Oracle® Cloud

Using Integrations in Oracle Integration 3

G36915-31
February 2026

Oracle Cloud Using Integrations in Oracle Integration 3,

G36915-31

Copyright © 2022, 2026, Oracle and/or its affiliates.

Primary Author: Oracle Corporation

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

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

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

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

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

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

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

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

Contents

About This Content

1 Get Started with Integrations

Find More Resources 2

2 Learn About Integrations

10 Things to Know Before You Start 1
Watch a Video 4
- What Is Oracle Integration 4
- Integration Elements 4
- Integration Patterns 5
- Integration Adapters 5
- Design an Integration 5

Workflows for Integrations 5
- Prerequisites for Developing Integrations 5
- Workflow for Developing an Integration 7
- Workflow for Testing Integrations 9
- Workflow for Deploying Automation Solutions 12
- Workflow for Monitoring Integrations 13
  - Initial Monitoring Tasks 13
  - Daily Monitoring Tasks 14
  - Occasional Monitoring Tasks 14

Tours and Tutorials 15
- Explore the Navigation Pane 17
- Tour: Explore an Integration That Sends a Hello World Message 21
- Tour: Explore a Multiple Verbs and Resources Invoke Integration 23
- Tutorial: Build Your First Integration From Scratch 27
- Tutorial: Build Your Second Integration From Scratch 53

Get Started with Projects 80
- About Projects 80
- Project Guidance If You've Used Packages 82
- Tour of the Projects Workspace 83

Copyright © 2022, 2026, Oracle and/or its affiliates.

Navigation to the Projects Workspace 83
Design Tab 84
Deploy Tab 87
Observe Tab 88
10 Tips for Working with Projects 90
4 Ways to Organize Projects 92
About Project Deployment 93
Restrictions for Projects 94
Understand Integration Styles 94
About Application Integrations 96
How to Choose Between Synchronous and Asynchronous Integrations 96
How to Specify if an Integration is Synchronous or Asynchronous 99
Differences Between Synchronous and Asynchronous Integrations 100
About Schedule Integrations 100
About Event Integrations 101
Understand Hybrid Integrations 101
About Connecting to On-Premises Applications 102
About Creating Hybrid Integrations 102
About the Connectivity Agent 103
Connection Patterns for Hybrid Integrations 107
About Connecting to Private Resources 111
Outbound Private Connection Pattern 111
Adapters that Support Connecting to Private Endpoints 112
Requirements for Connecting to a Private Endpoint 113
Integration Concepts 114
About Integrations 114
About Connections 115
Available Adapters for Connections 117
Understand Trigger and Invoke Connections 122
Trigger Connections 122
Invoke Connections 125
About Mappings 129
About Lookups 132
About Packages 134
About Business Identifiers for Tracking Fields in Messages 134
About Error Management 135
AI Innovation and Oracle Integration 136
Fast Track Automation with AI Assistant 136
Create Projects Using Natural Language 137
Create and Edit Integrations Using Natural Language 137
Schedule Integrations Using Natural Language 139
Automatically Create Descriptions for Projects, Integrations, and More 140

Copyright © 2022, 2026, Oracle and/or its affiliates.

Automatically Generate Documentation for Projects and Integrations 142
Query Data Using Natural Language 144
Troubleshoot Errors Faster with AI 146
Enable or Disable AI Features 147
Bring Intelligence to Automations 148
Augment Integrations with OCI AI Services 148
Expand Integrations with OpenAI LLM Models 149
Expand Integrations with Anthropic LLM Models 150
Execute Queries Against an OpenSearch Cluster 150
Automate with Agentic AI 150
Get Started with Observability 152
About the Observability Pages 153
Develop an Observability Strategy 156
About Building Your Own Observability Tools 158
About Oracle Integration 158
Start Oracle Integration 159
Keyboard Shortcuts in the Integration Canvas 159

3 Best Practices

Avoid Common Integration Pattern Pitfalls 1
Design Best Practices 5
Create Schedule Integrations Mindfully 5
Follow Guidelines for Synchronous Integrations 6
Design Long-Running Integrations as Asynchronous 7
Understand Timeouts in Synchronous Integrations 7
Send Data to Third-Party Systems Efficiently 7

4 Install and Manage the Connectivity Agent

Install the Connectivity Agent 1
Meet the Requirements for the Connectivity Agent 1
System Requirements 2
Network Requirements 2
Plan the Connectivity Agent Setup 3
Number of Agent Groups 4
Number of Connectivity Agents in an Agent Group 4
Number of Active Connectivity Agents 5
Workflows for the Connectivity Agent 7
Associate Multiple Connectivity Agents with an Agent Group 8
Associate One Connectivity Agent with an Agent Group 9
Use the Connectivity Agent in an Integration 10

Copyright © 2022, 2026, Oracle and/or its affiliates.

Create an Agent Group 10
Download and Install the Connectivity Agent 12
Installation Requirements 12
Download and Install the Agent 12
After Installation 14
Manage the Connectivity Agent 15
Updates Are Automatic for the Connectivity Agent 15
Run the Connectivity Agent as a Service 16
Run the Agent as a Service on Linux 16
Run the Agent as a Service on Windows 17
Install a Certificate on the Agent Host 17
Restart the Connectivity Agent 18
Configure the Connectivity Agent to Use mTLS to Access On-Premises Endpoints 19
Recover the Connectivity Agent in Case of Host Failure 20
Delete an Agent Group 21
Configure the Connectivity Agent to Use mTLS to Access On-Premises Endpoints 21
Upgrade the JDK 11 Agent to the JDK 17 Agent 21

5 Design and Monitor Integrations in Projects

Design a Project 1
Control Who Can Edit, View, and Monitor in a Project 2
Create or Import a Project 14
Create a Project By Selecting an Accelerator or Recipe that the AI Suggests 20
Copy an Integration into a Project 23
Use AI to Create an Integration 24
Use AI to Edit an Integration 33
Use AI to Create Integrations Based on Integrations from the Integration Store 40
Add and Share a Connection Across a Project 42
Add and Share a Lookup Across a Project 49
Create, Publish, and Subscribe to Events in a Project 52
Create an AI-Powered Event Description 53
Create an AI-Powered Project Description 54
Create Robots in a Project 56
Create Healthcare Integrations in a Project 56
Create B2B Integrations in a Project 57
Create Decision Models in a Project 58
Create Human in the Loop Workflows in a Project 58
Use AI to Generate Project Documentation 59
Use AI to Generate Integration Documentation 60
Manage a Project 61
Create and Manage a Project Deployment 62

Copyright © 2022, 2026, Oracle and/or its affiliates.

Understand the Integration Versioning Life Cycle in a Project Deployment 62
Create a Deployment 64
Manage Project Deployments 66
Activate or Deactivate a Project 66
Edit and Replace Dependent Resources in a Project 67
Export a Project 70
Clone a Project 73
Integrate Projects and Project Deployments with a GitHub Repository 74
Invoke Child Integrations Inside or Outside of Projects 76
Manage Accelerator Projects 80
Extend an Integration in an Accelerator Project 80
Upgrade an Accelerator Project and Merge Extensions 99
Behavior When You Activate an Updated Accelerator Project Deployment with Newer Integration Versions 106
Deploy Integration Endpoints to Oracle Cloud Infrastructure API Gateway 110
Clone an Integration in a Project 121
Update the Tracing Level of Integrations in a Project 122
Test REST Adapter Trigger Connection-Based Integrations in Projects 122
View the Dependent Relationships Between Project Resources 122
View Run Details About Integrations in Projects 130
View Project Status 130
Edit Project Description Details 131
Delete a Project 131
Add Retry Logic to Invoke Connections in Projects 131
Monitor Integrations in a Project 134
Monitor the Message Processing Status of Integrations in Projects 134
Track Instances in a Project 136
View Subscribed Instances in a Project 137
Manage Errors in a Project 137
View the Schedule Integration Runs Calendar in a Project 138
View the Design-Time Audit Log of a Project 138
Monitor Robots in a Project 139
View the Relationship Between Parent and Child Integration Tracking Instances in a Project 140
Monitor MLLP Adapter Connections and Queues 141

6 Create Connections

Inbound and Outbound Endpoints and Transport Layer Security Server Version Support 1
Create a Connection 1
Create a Connection with an Agent Group 3
Configure Connection Properties and Security Properties 4
Configure Connection Properties 4

Copyright © 2022, 2026, Oracle and/or its affiliates.

Configure Connection Security 8
Use OAuth 2.0 in Oracle Integration 13
Where Can You Use OAuth in Oracle Integration? 13
Use OAuth with Inbound Calls to Oracle Integration 15
Protect an Integration's REST Endpoint with OAuth 15
Protect an Integration's SOAP Endpoint with OAuth 16
Protect Application-Specific Endpoints with OAuth 18
Use OAuth with Outbound Calls from Oracle Integration 18
Invoke an Application's REST Endpoint with OAuth 18
Invoke an Application's SOAP Endpoint with OAuth 20
Invoke Oracle Fusion Application Endpoints with OAuth 21
Invoke Application-Specific Endpoints with OAuth 23
You Must Use OAuth with the Oracle Integration Developer APIs 23
What are the Oracle Integration Developer APIs 24
Do You Use the Runtime or Design-time URL? 24
Invoke Oracle Integration Developer APIs from an External Client 26
Invoke the Oracle Integration Developer APIs from within an Integration 28
Call the Developer APIs with Client Credentials 29
Call the Developer APIs with Authorization Code 33
Call the Developer APIs with JWT User Assertion 39
Call the Developer APIs with Resource Owner Password Credentials 42
FAQs for the Confidential Client Application 47
Troubleshoot OAuth 48
Token Refresh Error: How do I Check the Configuration? 48
Check Scopes Match Primary Audience in Your Oracle Integration Instance 52
Unauthorized Errors: Check Roles Assigned to Your Confidential Application 53
Test the Connection 54
Edit a Connection 54
Clone a Connection 55
Delete a Connection 55
Unlock a Connection 56
Refresh Integration Metadata 57
Manage Security Certificates 57
Upload a Certificate to Connect with External Services 58
Update or Delete a Certificate 60
Browse and Install Third-Party Adapters 61
View Preinstalled Adapters 61
Allow JSON Numbers with High Precision and Scale 62

7 Create Application Integrations

Create an Integration 1

Copyright © 2022, 2026, Oracle and/or its affiliates.

Design an Application Integration 4
Define Inbound Triggers, Outbound Invokes, and Actions 6
Add Suggestions, Invokes, and Actions Through an Inline Menu 10
Cut and Paste Elements in the Integration Canvas 12
Expand and Collapse Actions in the Integration Canvas 17
Create Global Variables 18
Define Ad-Hoc Maps 19
Import a Map File into an Integration 21
Add Actions to an Integration 21
Actions Category 21
Assign Values to Scalar Variables in an Assign Action 22
Translate an EDI Document with the B2B Action 25
Build Complex Assignment Statements with a Data Stitch Action 27
Convert HL7 Messages with a Healthcare Action 31
Log Messages with a Logger Action 35
Add Placeholder Notes with a Note Action 37
Send Notification Emails During Stages of the Integration with a Notification Action 37
Process Files in Schedule Integrations with a Stage File Action 41
Delay Integration Processing for a Specified Time Period with a Wait Action 56
Error Handling Category 60
Catch Faults with a Re-throw Fault Action 60
Throw Faults with a Throw New Fault Action 61
OCI AI Services Category 63
Prerequisites 63
Extract Document Information with a Document Understanding Action 66
Analyze and Extract Information from Images with a Vision Action 80
Generate, Embed, or Rerank Text with a Generative AI Action 86
Perform Text Analysis and Translation with a Language Action 90
Ask Questions and Receive Answers Based on Current Enterprise Data with a Generative AI Agents RAG Action 99
Transcribe Speech to Text and Synthesize Text to Speech with a Speech Action 101
Logic Category 106
Loop over Repeating Elements with a For-Each Action 107
Route Expressions with Switch Action Branches 110
Process Tasks in Parallel with a Parallel Action 113
Manage a Group of Actions and Fault Handlers with a Scope Action 120
Loop Over Actions or Invoke Connections While a Condition is Satisfied with a While Action 123
Call Category 125
Add a Decision Model to an Integration with a Decision Action 125
Interact with Files in File Server 126
Invoke A Child Integration with an Integration Action 132

Copyright © 2022, 2026, Oracle and/or its affiliates.

Add JavaScript Functions to an Integration with a JavaScript Action 133
Invoke Oracle Cloud Infrastructure Functions Directly from an Integration with an OCI Function Action 135
Invoke Oracle Cloud Infrastructure Object Storage from an Integration with an OCI Object Storage Action 145
Invoke a Process from an Integration with a Process Action 151
Publish Events in an Integration with a Publish Event Action 153
Call a Robot from an Integration 154
Receive Requests for Multiple Resources in a Single REST Adapter Trigger with a Pick Action 154
Add Global Fault Handling to Integrations 157
Use XPath Axis and Wildcard Expressions in the Expression Builder 159
Use Lookups in Variable Assignments 160
Define Fault Aggregation in Parallel Processing Scenarios 163
Assign Business Identifiers for Tracking Fields in Messages 165
Assign Business Identifiers 165
Delete Business Identifiers 166
Maintain Business Identifier Definitions During Trigger Connection/Schedule Integration Parameter Changes 166
Display Errors and Warnings in an Integration 167

8 Create Schedule Integrations

Create a Schedule Integration 1
Define the Integration Schedule 2
Decide When to Create the Integration Schedule 2
Define the Type of Integration Schedule to Run 3
Select When to Run the Integration Schedule 6
Create and Update Parameters in Schedule Integrations 8
Override Schedule Parameter Values 11
Run a Schedule Integration Immediately 12
Start and Pause an Integration Schedule 13
Change the User Submitting an Integration Schedule 16
Globally Change the Submitter of Integration Schedules 17
View Past and Present Schedule Integration Runs 17
View Future Schedule Integration Runs 17
Edit an Integration Schedule 18
Delete an Integration Schedule 18
Unlock an Integration Schedule 19
Convert a Schedule Integration to a REST Adapter-Triggered, Application Integration 19
Delete Scheduled or Ad-Hoc Runs 20
Monitor Schedule Integration Runs 21

Copyright © 2022, 2026, Oracle and/or its affiliates.

Understand Schedule Integration Behavior When Using the Local Time Zone 21

9 Create Event Integrations

About Publishing and Subscribing to Events 1
Available System Events 1
Life Cycle for Published Events 2
Restrictions 3
Create Events to Publish and Subscribe to in Integrations 3
Publish Events in an Integration 5
Subscribe to Events in an Integration 7
Retain an Event Subscription During Integration Deactivation 12
Define Custom Header Filters to Trigger Integrations 13
JQ Expression Syntax for Creating Filters 33
Share Events Across Projects 36

10 Map Data and Create Lookups

Map Data 1
Access the Mapper 1
Encode and Decode File Attachment Content 1
Import Map Files 3
Import a Map File into Oracle JDeveloper 3
Import a Map File into Oracle Integration 5
Manage Lookups 5
Create a Lookup 6
Add Adapters or Domain Names to a Lookup 7
Clone a Lookup 8
Delete a Lookup 8
Edit a Lookup 9
Update Lookups in Active Integrations 9
Export a Lookup 9
Import a Lookup 10
Create an AI-Powered Lookup Description 10
Create the lookupValue Function 12
Access the Build Lookup Function Wizard 12
Select the Lookup 15
Select the Source and Target Columns 15
Specify the Default Value 16
Review Your Lookup Table Selections 16

Copyright © 2022, 2026, Oracle and/or its affiliates.

11 Use JavaScript Libraries in Integrations

Use Libraries to Manage Integration Functions 1
View Library Details 2
Edit Library Function Values 2
Determine Which Integrations Use a Library Function 2
Delete a Library 3
Import a Library File 3
    Write JavaScript 4
Create an AI-Powered Library Description 10
Export a Library File 12
Update JavaScript Files 12
Use Library Functions in Integrations 15

12 Manage Integrations

Activate and Deactivate Integrations 1
    Reactivate Integrations after a Connection Update 2
    Deactivate an Integration 3
        Deactivate an Integration with Business Events 4
    Manage Tracing Levels on Integrations 5
    Activate an Integration 7
        Behavior of Running Instances When a Minor Integration Version is Activated 9
Test Integrations in Oracle Integration 12
    Test Integrations from Inside the Integration Canvas 13
    Test Integrations from Outside the Integration Canvas 17
        Operation Section 17
        Request and Response Sections 19
Generate an AI-Powered Integration Description 21
Edit and Replace Dependent Resources 22
    Edit and Replace Dependent Resources in a Package 23
    Edit and Replace Dependent Resources in an Integration 25
    Unlock Dependent Resources 28
Invoke a Child Integration from a Parent Integration 29
    Create a Co-located Integration with Header Support 32
        Backward Compatibility for Header Support 36
    Dynamically Invoke a Child Integration 36
Manage Packages 39
    View the Integrations in a Package 39
    Import a Package 40
    Export a Package 41
    Update the Package for an Integration 41

Copyright © 2022, 2026, Oracle and/or its affiliates.

Delete a Package 41
Convert a Package to a Project 42
Import and Export Integrations 42
Import an Integration 43
Import a Prebuilt Integration 44
Export an Integration 44
Clone the Design-Time Metadata of an Entire Service Instance 44
Step 1: Create an Object Storage Bucket and Construct the Storage URL 45
Step 2: Configure the Instance Object Storage Bucket 48
Step 3: Export and Import Design-Time Metadata Between Instances 49
Create an Export Job 50
Create an Import Job 52
Step 4: Perform Post-Processing Steps 54
Refresh Endpoints for Integrations 54
Set the Inactive Session Timeout Limit and Audit Logging Levels 57
Modify an Integration 59
Impact on the Mapper of Editing the Endpoint Information in an Integration 59
View the Actions and Connections in an Integration 60
Clone an Integration 61
Create a Draft Version of an Integration 62
Delete an Integration 63
Recover Unsaved Integration Changes 63
Override Design-Time Properties in an Integration 64
Use Metadata in Integrations 67
Cancel a Running Integration Instance 69
View the Contents of an Incomplete Integration 69
Edit an Integration with Incomplete Connections 70
Change the Time Zone 70

13 Monitor Integrations During Runtime

Get Started with Observability 1
View the Dashboard 3
Monitor the Number of Emails Sent from Oracle Integration 6
Check the Design Time Audit History for an Integration or Other Component 7
Check the Runtime Audit History for File Server 10
See the Current Status of All Integration Components 12
Check an Integration's Tracing Level 13
View Notification Alert Announcements and the Status of Asynchronous Operations 14
View Notification Alert Announcements 14
View the Progress of Asynchronous, Design-Time Operations 15
Monitor the Message Processing Status of Integrations 16

Copyright © 2022, 2026, Oracle and/or its affiliates.

Monitor Agents 20
Learn About the Payloads in the Activity Stream 21
Measure an Integration's Performance 23
Determine an Integration's Typical Processing Times 24
Determine an Integration's Performance on a Given Day 25
Determine Whether an Integration's Performance Is Consistent Over Time 25
Find Integration Instances with Increased Processing Times 26
Measure the Running Times of an Integration's Actions 27
Find Integrations That Experienced Errors 27
Find Integrations with Errors 28
Determine the Source of an Integration's Error 28
Find the Integration Error Rate for the Last 24 Hours 30
Measure Your Integration Errors Over Time 30
Find the Connections with the Most Errors 31
Find Integrations That Timed Out or Were Aborted 33
Manage Errors 33
View Error Messages and the Message Payload in the Activity Stream of Failed Instances 34
View an AI-Powered Error Message 42
Resubmit Failed Messages 43
Discard Errors 46
View the Status of Message Recovery 47
View Errors By Integration and Connection 48
View the Business Identifiers Used in Failed Integration Instances 49
Track Integration Instances 50
Filter the Display of Integration Instances 50
Search for Business Identifier Values 53
Track the Status of Integration Instances 58
Track the Integration Flow on the Instance Details Page 72
Determine Why an Integration Instance Stopped Running 76
Replay Integration Instances 78
Monitor Integrations That Subscribe to Events 80
View Minimal Details About Debug Tracing Level Instances for 32 Days 81
View the Calendar of Schedule Integration Runs 82
Diagnose and Manage Event-Based Oracle Fusion Applications Integrations 83
Monitor Business Event Delivery Status 85
Send System Status Reports with Notification Emails 87
Understand DKIM Authentication and SPF Record Details 87
Configure Notification Emails 88
Remove Email Addresses from the Suppression List 93
Change the Delivery Time of Notification Emails 93

Copyright © 2022, 2026, Oracle and/or its affiliates.

14 Troubleshoot Oracle Integration

Troubleshoot Notification Email Configuration 1
Troubleshoot the Connectivity Agent 2
Troubleshoot Private Endpoints 13
Troubleshoot Integration Runtime 13
- An Integration Runs Slowly 13
- Fewer Integration Instances Ran Than Usual 15
- Using Basic Authentication to Call Services When the Identity Domain is Configured With Multi-Factor Authentication Causes a 401-Unauthorized Error 16
- Perform Remedial Actions When the Payload Exceeds the Size Limit 16
- Remove Unprintable Control Characters from the Incoming Payload 17
- Notification Action Returns a <from> value is invalid Error 18
- Synchronous Integration Timeout Failure 19
- Activity Stream Details 19
- Message Dequeuing Takes Three to Four Seconds in Asynchronous Integrations 20
- 403 Error When Accessing an API Protected Using OAuth 20
- Troubleshoot Certificate Import Failures 20

Troubleshoot Integration Design Time 21
- Missing a Primary Business Identifier in an Integration 21
- Unable to Call a replaceAll JavaScript Function 21
- Conflict with Using Variables of String Data Type and Applying Mathematical Comparator Expressions 22
- Reduce Network Calls that Slow Down Processes Caused By Writing Record By Record 22
- Invoke Subprocess Integrations Using a Single REST API Connection 22
- Trigger Connection Cannot Receive an Empty Array in JSON if the Corresponding JSON Received Has an Empty JSON 23
- Object Storage Bucket Error Creation 23
- Manage Your Registered AI Engines 24

Troubleshoot Schedule Integrations 24
- Schedule Integrations Are Not Running on Time 24
- When a Schedule Integration Instance Gets Terminated 24

Troubleshoot Integration Activations 25
- Troubleshoot Integration Activations 25
- Delay in Message Processing of First Instance after Activation 26

A Oracle Integration Use Cases

Extract Content from an Invoice PDF Document with a Document Understanding Action A-1
Extract Information from a Custom Document with a Document Understanding Action A-5
Manage Oracle Cloud Infrastructure Buckets and Objects from an Integration A-9
Read Large XML Files Containing Multiple Namespaces A-16
Use a Stage File Write Operation with an Opaque Schema A-19

Copyright © 2022, 2026, Oracle and/or its affiliates.

Test REST Adapter Trigger-Based Integrations with Multipart Attachments on the Configure and run page A-20
Calculate the Sum of a Column or the Count of Rows While Processing Large Files A-21

Copyright © 2022, 2026, Oracle and/or its affiliates.

About This Content

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

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 at `http://cloud.oracle.com`
- *Using the Oracle Mapper with Oracle Integration 3*

Conventions

The following text conventions are used in this document.

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

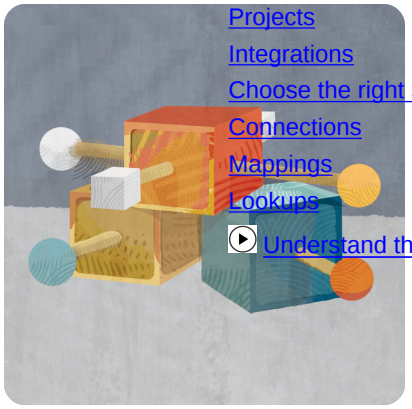
Copyright © 2022, 2026, Oracle and/or its affiliates.

1

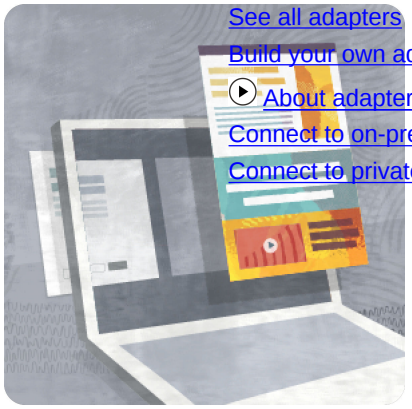
Get Started with Integrations

An integration completes a task or synchronizes data by connecting to applications' APIs. Browse the following popular topics to learn more.

Explore Integrations

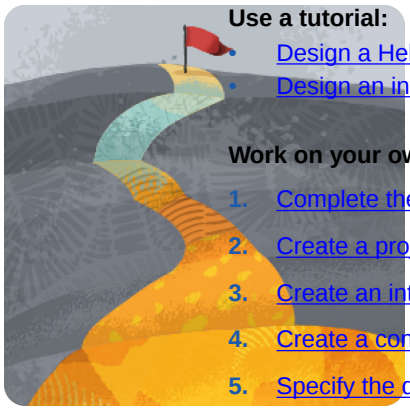
Image	Links
	Projects Integrations Choose the right style Connections Mappings Lookups Understand the elements of an integration

Connect Everything, Everywhere

Image	Links
	See all adapters Build your own adapter About adapters Connect to on-premises apps Connect to private resources

Copyright © 2022, 2026, Oracle and/or its affiliates.

Design an Integration

Image	Links
	**Use a tutorial:** • Design a Hello World integration • Design an integration that gets a weather report **Work on your own:** 1. Complete the prerequisites 2. Create a project Project tips Ways to organize projects 3. Create an integration 4. Create a connection Applications you can connect 5. Specify the data to move 6. Map terms that describe the same item 7. Upload JavaScript 8. Design the business logic with actions 9. Add global fault handling 10. Track payload fields in runtime 11. (Scheduled integrations only) Specify when an integration runs

Track Status and Runtime Success

Image	Links
	Get Started with Observability

Find More Resources

Oracle Integration offers many ways to stay connected and grow your expertise.

Resource	Description	Link
Documentation	Find step-by-step instructions, best practices, and more.	Get Started page on the Oracle Help Center

Copyright © 2022, 2026, Oracle and/or its affiliates.

Resource	Description	Link
Tutorials	Take a product tour, build an integration from start to finish, or complete a longer self-directed workshop with a LiveLab.	Tours and Tutorials in *Using Integrations in Oracle Integration 3*
Videos	Learn more about Oracle Integration by watching a short video.	Videos page on the Oracle Help Center
Live demos	Take a deep dive into new features, use cases, and more. Subscribe to the channel to get notified when a new live demo is available.	Oracle Integration on YouTube
Oracle Integration blog	Learn what's new from the product management team.	Oracle Integration blog
A-Team Blog	Learn from technical solution architects and software engineers.	A-Team Chronicles blog
Cloud Customer Connect	Connect with experts, ask or answer questions, and share your opinions.	Cloud Customer Connect
Newsletter	Receive a quarterly newsletter with product, event, and training information.	Subscribe to the Oracle Integration Newsletter
Webcasts	Join the following quarterly webcasts: • **Product Update Webcasts** Get product updates and watch live demos. • **Customer Success Webcasts** Get inspired by success stories and best practices, or present your own wins to the community.	Subscribe to the Oracle Integration Newsletter: Learn about upcoming webcasts by subscribing to the newsletter Integration Resources blog post: Watch past webcasts and learn about upcoming webcasts
Oracle Partners	Use the Oracle PartnerNetwork to find a partner to support your business goals.	Oracle Partner Finder To view this page, connect to the Oracle network.

Copyright © 2022, 2026, Oracle and/or its affiliates.

2

Learn About Integrations

Become an expert at integrations by learning about their components, completing product tours and tutorials, and reviewing tips.

Topics:

- 10 Things to Know Before You Start
- Watch a Video
- Workflows for Integrations
- Tours and Tutorials
- Get Started with Projects
- Understand Integration Styles
- Understand Hybrid Integrations
- Integration Concepts
- AI Innovation and Oracle Integration
- Get Started with Observability
- About Oracle Integration

10 Things to Know Before You Start

New to Oracle Integration? Whether you're a seasoned developer or have done a little work in low-code platforms, these pointers will help you get started with confidence and become an expert quickly.

Oracle Integration has an intuitive drag-and-drop interface that lets you design integrations quickly and easily. To ensure that you're building efficient integrations, spend some time developing your expertise before you start building.

1. Integrations Streamline Your Business

No matter your business or your goals, you can use Oracle Integration to automate your business processes. For example, you can:

- Send data between your applications.
- Connect applications in the cloud with on-premises software.
- Automate end-to-end solutions, such as procure to pay, inquiry to order, and hire to retire.
- Monitor your business processes.
- Recover from errors.

Several components help move data, including connections, mappings, and lookups. Understand how they work together so you can grow your skills and build integrations quickly and efficiently.

Learn more: About Integrations, About Connections, About Mappings, and About Lookups

Copyright © 2022, 2026, Oracle and/or its affiliates.

2. A Project Is Your Super-Powered Release Manager

Oracle Integration offers an intuitive user interface that places all the components that you need in a single solution, called a project. There, you design, manage, and observe related integrations. A project streamlines your day-to-day work and your integration deployment.

Learn more: About Projects

3. An Integration's Pattern Determines How it Starts

When you begin designing an integration, you choose the integration pattern to use. The pattern determines the event or activity that starts the integration.

The following integration patterns are available:

- **Application integrations**

  Create an application integration to move data after a specific business event occurs or after a change in an application occurs. For example, an application integration runs after a customer submits an order, or after an employee makes an update in an application.

  When the business event occurs, the application sends a request to Oracle Integration to start the integration. A trigger connection handles the request. You create the trigger connection using an adapter.

- **Event integrations**

  Create an event integration to move data when someone creates a specific event type in an application. This publish-subscribe pattern requires two integrations. The first integration detects that a specific event type has been created in an application and publishes the event. The second integration subscribes to the event.

- **Schedule integrations**

  Create a schedule integration to move data, either at a specific time and date or on demand.

Learn more: Understand Integration Styles

4. Application Integrations Can Be Asynchronous or Synchronous

You can design an application integration as either a synchronous or asynchronous integration. Synchronous integrations deliver immediate responses, whereas asynchronous integrations guarantee the delivery of requests, no matter how complex the integration is.

- Asynchronous integrations are guaranteed to run when they are accepted

  Use asynchronous integrations whenever you can. An asynchronous integration provides an immediate acknowledgment and, depending on the design, it can also provide a response upon completion.

- Synchronous integrations give immediate responses

  When your business process requires an immediate response, such as providing a customer with a confirmation number for an order, build a synchronous integration. A synchronous integration finishes quickly, provides a response back to the caller, and raises any errors back to the caller.

  Errors that occur for synchronous integrations are unrecoverable. For example, if an error occurs, you can't submit the failed integration instance.

See About Application Integrations.

Copyright © 2022, 2026, Oracle and/or its affiliates.

5. Schedule Integrations Should be Staggered and Simple

When your business response must run on a schedule, use a schedule integration. For example, use a schedule integration to run payroll at the end of the month or to synchronize your inventory every Friday at 5 PM.

Keep in mind the following guidance when creating schedule integrations:

- Stagger the start times.

  For example, if you need to run ten daily integrations at the end of the business day, start one at 5 PM, the next at 5:05 PM, and so on.
- Keep schedule integrations small and simple.

  Keep reading to learn tips for keeping schedule integrations small and simple.

See About Schedule Integrations.

6. You Can Combine Integrations with Different Patterns

If you need an immediate response or need an integration to run on a schedule, you can pair a synchronous or schedule integration with an asynchronous integration.

- **When to pair a synchronous and asynchronous integration**

  When a business process is complex and requires an immediate response, build a synchronous integration that completes two tasks:

  1. Returns a quick response.

     For example, a synchronous integration can return the confirmation number for a customer's order or complete a credit check for a customer before fulfilling an order.
  2. Kicks off one or more asynchronous integrations.

     The asynchronous integrations contain the bulk of the work, such as updating inventory, creating packing lists, and more.

  See How to Choose Between Synchronous and Asynchronous Integrations.
- **When to pair a schedule and asynchronous integration**

  If a business process is complex, build a schedule integration that completes two tasks:

  1. Starts the process at the right time.
  2. Kicks off one or more asynchronous integrations, which contain the bulk of the work.

> **Tip**
>
> No matter what type of integration you create, try to keep it as simple as possible. Simple integrations run more quickly than complex integrations. Of course, an integration can be only as simple as its corresponding business process, and some business processes are overwhelmingly complex. When designing for such complexity, remember that your integrations are competing for resources. Make sure that they all get a fair chance to complete as planned.

7. Service Limits Provide Guard Rails

Everyone must work within the service limits, which are the quotas and allowances for working with integrations. A common mistake that new users make is designing a complex integration that exceeds the service limits.

Copyright © 2022, 2026, Oracle and/or its affiliates.

Your focus right now is understandably on getting your business flow started, but spend some time thinking about the scenarios the integration will encounter in the real world. For example, before you start designing a synchronous integration that needs to process orders, consider the real-world size of the files that the integration must process. For example, if the order that the integration must process contains many lines, the order processing could take a long time, and the integration might time out.

You don't need to memorize the service limits. However, reference them often when planning and developing an integration. Know the limits up front so you can design effective and appropriate integrations from the beginning.

Learn more: Service Limits in *Provisioning and Administering Oracle Integration 3*

8. Best Practices Are Available

You don't have to be an expert at Oracle Integration to create best-in-class integrations. That's because we collected the best practices and antipatterns, also known as worst practices. Follow this guidance so you know the actions to take—and the mistakes to avoid. Review this guidance before and while you work so that you create integrations that go live without issues.

Knowing the best practices and antipatterns, along with the service limits, will make you the go-to resource for integration best practices at your organization.

Learn more: Design Best Practices

9. Fault Handlers Keep Things Moving When Mistakes Happen

Before you start designing your first integration, spend some time implementing a global fault handler. A fault handler translates cryptic error messages into actionable tasks. This small time investment keeps your business running smoothly and saves everyone time.

10. Test Now, or Rework After Go-Live

An integration that performs well in isolation might perform well under load...or it might not. You won't know how all your systems work together until you test them in a production-like environment. For instance, an on-premises application that's experiencing some latency could negatively impact your synchronous integration.

Before you move your integrations to a production environment, test them in a real-world scenario, under load. If you need to modify the integration, it's much faster and easier to rework it before you start using it in a production environment.

Watch a Video

Watch a video that introduces key concepts about integrations.

What Is Oracle Integration

Learn how Oracle Integration helps you connect applications and data, automate business processes, and innovate with AI.

Video

Integration Elements

Learn the basic vocabulary and concepts used in Oracle Integration.

Copyright © 2022, 2026, Oracle and/or its affiliates.

Video

Integration Patterns

Take a look at the various integration styles, or patterns, available in Oracle Integration, and how, why, and when you might use them.

Video

Integration Adapters

Get a high level look at adapters in Oracle Integration. Learn what they do, and what kinds are available for your integrations.

Video

Design an Integration

Learn how to design integrations quickly and efficiently using the large library of adapters in Oracle Integration.

Video

Workflows for Integrations

Complete several prerequisite tasks, and then you can start designing and developing an integration. Next, it's time to deploy to a higher environment and then test and monitor your integration.

Topics:

- Prerequisites for Developing Integrations
- Workflow for Developing an Integration
- Workflow for Deploying Automation Solutions
- Workflow for Testing Integrations
- Workflow for Monitoring Integrations

Prerequisites for Developing Integrations

Depending on your organization's requirements, you might need to perform some, all, or none of the prerequisites for designing integrations.

Complete the required prerequisites one time, before you start designing integrations.

Copyright © 2022, 2026, Oracle and/or its affiliates.

Step	Task	More information
1	Plan your work	Before you start building, spend some time planning your work. For example: 1. Understand the problem. An integration automates work that people would otherwise complete. Spend some time understanding the reason the work needs to be done, the current process, the problems with the current process, and the ideal workflow. Gathering requirements can be challenging, but this research becomes a worthwhile use of your time when you collect information that helps you deliver a superb solution. For example, you might need to: • Create an order in Oracle ERP Cloud when Oracle ERP Cloud receives an opportunity won event from a CRM. • Synchronize patient data every Friday at 5 PM. 2. Identify how to fulfill the requirements in an integration and plan your integration at a high level. 3. Identify the applications that you're connecting, and collect connectivity information. For example, collect the URLs and authentication details for the applications. 4. Determine whether a recipe or accelerator can help. Recipes and accelerators contain all the resources required for specific integration scenarios. See Recipes on the Oracle Help Center.
2	Complete the prerequisites for the applications you're connecting	**Required if:** The adapter you're using to connect to an application has prerequisites. Most adapters have prerequisites. **Get started:** Complete the prerequisites described in the guides for each adapter. Additionally, determine the security policies and connection login credentials that you'll use for the connections that are based on the adapter. See Configure Connection Properties.

Copyright © 2022, 2026, Oracle and/or its affiliates.

Step	Task	More information
3	Install the on-premises connectivity agent	**Required if:** An integration needs to connect to applications that reside in an on-premises network. **Not needed if:** You're connecting to cloud applications only. **Learn more:** About Connecting to On-Premises Applications . **Get started:** 1. Create a connectivity agent group. See Create an Agent Group. 2. Download and run the on-premises connectivity agent installer in your local environment. During installation setup, associate the on-premises connectivity agent with the agent group. See Install the Connectivity Agent or Associate Multiple Connectivity Agents with an Agent Group. After you install the connectivity agent, you can create a connection that uses the agent group and its associated connectivity agent. See Create a Connection with an Agent Group.
4	Configure a private endpoint	**Required if:** You need to connect to private resources and must route the traffic through a private channel. **Not needed if:** You don't need to connect to private resources, or you need to connect to private resources but don't need to route traffic through a private channel. **Learn more:** Connect to Private Resources in *Provisioning and Administering Oracle Integration 3*. **Get started:** 1. Complete the prerequisites for configuring a private endpoint. See Prerequisites for Configuring a Private Endpoint in *Provisioning and Administering Oracle Integration 3*. 2. Configure the private endpoint. See Configure a Private Endpoint for an Instance in *Provisioning and Administering Oracle Integration 3*. Later, when you create connections, you choose whether they use the private endpoint to connect to applications.

> **Note**
>
> A connection that doesn't use the connectivity agent or a private endpoint is routed through the public internet.

After completing the prerequisites, you can start designing integrations. See Workflow for Developing an Integration.

Workflow for Developing an Integration

To develop an integration, first create a project to work in. Then, create the integration and all its components, and specify the steps that the integration completes.

Copyright © 2022, 2026, Oracle and/or its affiliates.

Before completing these tasks, make sure you complete all prerequisites. See Prerequisites for Developing Integrations.

Step	Task	More information
1	Create a project	Create one or many integrations in each project. Most organizations group related integrations in a single project. **Learn more:** Get Started with Projects. **Get started:** Create or Import a Project.
2	Create an integration	Oracle recommends designing integrations in projects, but you can also design an integration outside of a project. When you create the integration, add trigger and invoke connections, assign business identifiers, and then map the data between the connections. **Learn more:** About Integrations. **Get started:** Copy an Integration into a Project or Create an Integration.
3	Choose the applications to connect	Connect to applications using connections. Create one connection for each application that you need to integrate. Typically, an administrator creates connections. You can use a connection in one or many integrations. **Learn more:** About Connections. **Get started:** Add and Share a Connection Across a Project.
4	Specify which data to move and how to move it	Specify the data to move and the way to move the data by adding trigger and invoke connections to an integration. At that time, you specify your requirements, including how to process the data. **Learn more:** About Connections. **Get started:** Understand Trigger and Invoke Connections.
5	Map different terms that describe the same item (required for only some integrations)	Use lookups to associate different values for the same item. For example, two applications might have different gender codes or country codes. Lookups ensure that the correct values appear in the target application. **Required if:** The applications you're connecting share and represent information differently. **Learn more:** About Lookups.
6	Upload JavaScript for use in your integration (optional)	When you need to call JavaScript from an integration, you can import the JavaScript file. **Learn more:** Use Libraries to Manage Integration Functions.

Copyright © 2022, 2026, Oracle and/or its affiliates.

Step	Task	More information
7	Design the business logic of your integration	Every integration performs one or more activities, such as looping over elements and adding routing expressions. Add actions to perform the activities that the integration must complete. **Get started:** • To review the wide range of available actions, see Add Actions to an Integration. • In addition to the actions that define the business logic of the integration, consider including the following actions as best practices: – Add email notifications to the integration You can use email notifications to send confirmations or inform you about faults that have occurred. Oracle Integration sends the email notifications at specific processing points for an integration, and the notifications help you quickly identify issues with important business processes. See Send Notification Emails During Stages of the Integration with a Notification Action. – Add global fault handling to the integration A fault handler translates cryptic error messages into actionable tasks so your business can keep running smoothly. If the APIs you're connecting to return error messages that are difficult to understand, consider adding global fault handling. See Add Global Fault Handling to Integrations. – Add fault handlers to a collection of actions and invokes When you move some business logic, including actions and invokes, into a scope, you can add fault handlers to the business logic. See Manage a Group of Actions and Fault Handlers with a Scope Action.
8	Select the payload fields to track in runtime	You track fields in messages during runtime by assigning business identifiers to the fields. **Get started:** Assign Business Identifiers for Tracking Fields in Messages.
9	Specify when your integration runs (schedule integrations only)	Schedule integrations run by the clock. Specify when to run the integration and more details when you schedule the integration. **Get started:** Define the Integration Schedule.

Next, deploy the integration to a higher environment, if needed. See Workflow for Deploying Automation Solutions. Or, you can test the integration in your development environment. See Workflow for Testing Integrations.

Workflow for Testing Integrations

You can test integrations in your development environment or in a higher environment, such as testing or production. When testing, check the integrations for errors and ensure that they run as expected. You should understand the goals and business logic of an integration so that you can identify issues quickly.

Copyright © 2022, 2026, Oracle and/or its affiliates.

If you need to deploy to a higher environment before testing, see Workflow for Deploying Automation Solutions.

Step	Task	More information
1	Activate the project or the integrations in the project	You have several options: • In any environment, you can activate the project, which activates all integrations in it. The Debug tracing level isn't available when you activate a project, but you can update the tracing level for individual integrations after activation. See Update the Tracing Level of Integrations in a Project. • If you're working in your development environment, you can also activate integrations individually. When you activate integrations individually, you can choose the tracing level for each integration. See Activate an Integration. **Note**: When you activate an integration, you choose the level of tracing information to log to the activity stream. Consider the following factors when choosing a tracing level: • To see a detailed log of every activity that the integration completes, such as when you're testing, select **Debug (Not recommended)** for the tracing level. After 24 hours, **Debug (Not recommended)** is automatically reset to **Production**. Debug tracing provides the most robust level of information in the activity stream and can help you find mistakes. However, Oracle recommends not using debug tracing in production for several reasons. Debug tracing captures a lot of data and can cause performance issues for an integration, and it might expose sensitive data. • If you're testing the invokes or the trigger of the integration, you don't need to use debug tracing. Instead, the **Audit** tracing level captures payloads for these activities.
2	Run the integration	To run an integration, invoke the endpoint URL for the integration. You have several options for invoking an endpoint URL: • **For any integration** You can trigger any integration from outside Oracle Integration. Copy the endpoint URL from within Oracle Integration, and use it in a client application to run the integration. For instance: – If the trigger for an integration is the creation of a specific event in Salesforce, create the event in Salesforce. – If an integration trigger uses the SOAP Adapter, use a SOAP user interface to trigger the endpoint URL. – If an integration trigger is based on the REST Adapter, use Postman to trigger it, or see the next bullet. • **For schedule integrations and integrations that use trigger-based REST Adapter connections** Run these integrations from within Oracle Integration. See Run a Schedule Integration Immediately and Test REST Adapter Trigger Connection-Based Integrations in Projects.

Copyright © 2022, 2026, Oracle and/or its affiliates.

Step	Task	More information
3	Check the activity stream for errors	Verify that the instance of the integration is finished. See Monitor Integrations in a Project. Next, verify that the integration finished running, and check the activity stream for errors. The activity stream depicts the message flow in the integration instance and indicates where errors occurred, if any. The activity stream contains all the information that you need to troubleshoot problems with your integrations. To learn more, see Learn About the Payloads in the Activity Stream. You can open the activity stream from several pages: • If you ran the integration from within Oracle Integration, open the activity stream from the Configure and run page, which appears after you run the integration. See Test REST Adapter Trigger Connection-Based Integrations in Projects. • If the integration is in a project, open the activity stream from the Instances tab, which is on the Observe tab. See Track Instances in a Project. • If the integration is outside a project, or if it's in a project that you have access to, open the activity stream from the Instances page, which is within Observability. See View Error Messages and the Message Payload in the Activity Stream of Failed Instances.
4	If an error occurred when the integration ran, address it and take the appropriate next steps	When the activity stream contains one or more errors, the integration isn't running as expected. 1. Identify the cause of the issue by reviewing the activity stream. Get more information about an error on the Instance Details page. This page provides a graphical view of the integration flow and also includes the activity stream. • If you're working in a project, see Track Instances in a Project. • If you're not working in a project, see View Error Messages and the Message Payload in the Activity Stream of Failed Instances. 2. Address the error as needed. For example: • Update the connection or the configuration of the lookup. • Fix the integration in your development environment and redeploy it. 3. Run the integration again, and repeat these steps as needed until the integration runs without errors.
5	Verify that the integration works as expected	For example: • Verify that the integration creates records as expected in the target application. • Verify that the integration handles expected failures, errors, and invalid data as expected.

Copyright © 2022, 2026, Oracle and/or its affiliates.

Workflow for Deploying Automation Solutions

You have several opportunities to deploy to higher environments. Many organizations deploy to a testing environment to simulate real-world usage, including testing under load. Additionally, everyone should deploy to a production environment when you're ready to go live. The steps you follow are the same for both environments.

> **Note**
>
> Complete these steps only if you're promoting to a higher environment and are working in a project:
>
> - If you use a single environment for development and testing, these steps apply only when you're promoting to your production environment. For guidance on testing, see Workflow for Testing Integrations.
> - If you're not using projects, follow your organization's standard procedures for deploying integrations to higher environments.

If you haven't developed the integration yet, see Workflow for Developing an Integration.

Step	Task	More information
1	Create a project deployment	Work in your development environment. Include the integrations and their versions in the project deployment. • To work in the user interface, see Create and Manage a Project Deployment. • To use the REST API, use the `deployments` endpoint. See Deployments REST Endpoints in *Developer API for Oracle Integration 3*.
2	Export the project deployment	Export the project deployment to the location of your choice, such as your hard drive or a location on your network. • To work in the user interface, you have the following options: – Export the entire project, including all of the integrations and components. See Export a Project. – Export a specific project deployment, including only the integrations and components that are part of the deployment. See Create and Manage a Project Deployment. • To use the REST API, use the `archive` endpoint. See Export a Project in *Developer API for Oracle Integration 3*.

Copyright © 2022, 2026, Oracle and/or its affiliates.

Step	Task	More information
3	Import the project deployment into the higher environment	Sign in to the environment to which you're deploying the integrations, such as the testing or production environment. Then, import the exported project into the environment. • To work in the user interface, see Create or Import a Project. • To use the REST API, use the `archive` endpoint. See Import(Add) a Project in *Developer API for Oracle Integration 3*. **Note:** To synchronize the project with a source control system, use the `archive` endpoint to export the entire project. See Export a Project in *Developer API for Oracle Integration 3*.
4	Configure the properties for the connections in the integrations	You must update your connection properties so that they reflect the higher environment that you imported into. See Configure Connection Properties and Security Properties.

Next, it's time to test the integration. See Workflow for Testing Integrations.

Workflow for Monitoring Integrations

Monitoring tasks typically fall into one of three categories: initial, daily, and occasional.

To learn more about monitoring in Oracle Integration, see Get Started with Observability.

Initial Monitoring Tasks

Before an automation solution goes live in production, optimize your monitoring workflow by completing a few tasks.

You might need to complete some tasks again when your organization prepares to release a major update to an automation solution.

Task	More information
Complete performance testing	See Measure an Integration's Performance.
Complete load testing	Check the scalability of a new automation solution by performing load testing. Load testing shows you how the automation solution will perform in a real-world scenario. For example, how will your order creation solution handle Black Friday? After completing load testing, measure performance. See Measure an Integration's Performance.
Prepare to go live	Document the automation solution, including its name, its use, the applications and connectivity agents that are involved, and contact names with backups for who gets involved and informed in the event of a failure. See View the Dashboard.
Update custom monitoring tools	If you maintain custom monitoring tools, determine whether to update them for the new automation solution. See About Building Your Own Observability Tools.

Copyright © 2022, 2026, Oracle and/or its affiliates.

Task	More information
Choose start times intentionally	For schedule integrations, make sure the start time doesn't conflict with other integrations. For example, don't start twenty schedule integrations at 5 PM. Instead, stagger the start times. See View the Calendar of Schedule Integration Runs.
Set up notifications	Notifications provide daily or hourly emails about failures or the overall health of all your integrations. See Send System Status Reports with Notification Emails.
Plan the hand-off	If you'll monitor your own automation solution, skip this row. However, if you'll monitor another person's automation solution, determine a hand-off schedule. For example, when the automation solution goes live, or after the automation solution runs successfully in production for a period of time. Setting expectations from the beginning helps ensure the successful launch of a new business process automation.
Verify that updates don't impact performance	After you version and update an automation solution, compare the average running times of the previous and new solutions to ensure that you haen't introduced performance regressions. See Determine an Integration's Typical Processing Times.

Daily Monitoring Tasks

Organizations typically perform these tasks at least daily, often at predetermined times.

Task	More information
Check whether integrations experienced errors	Find Integrations That Experienced Errors
Check whether integrations timed out or were canceled	Find Integrations That Timed Out or Were Aborted
Check for performance issues	Troubleshoot slow-running integrations and verify that integrations meet your organization's performance requirements. See Measure an Integration's Performance.
Identify recent updates	When an automation solution stops working as expected, an inadvertent update could be to blame. Check the audit history to find out what happened. See Check the Design Time Audit History for an Integration or Other Component.
Determine why fewer integration instances are running	Fewer Integration Instances Ran Than Usual
Check whether connectivity agents are available	If a connectivity agent isn't available, any integrations that use the agent fail. Inform the administrator who is responsible for the connectivity agent if any agents go down. See Monitor Agents.
Identify connections with the most errors	Find the Connections with the Most Errors

Occasional Monitoring Tasks

Routine analysis work supports operational excellence. Use your organization's operating procedures to determine how often you complete these tasks. For example, you might complete some tasks weekly, monthly, or quarterly.

Copyright © 2022, 2026, Oracle and/or its affiliates.

Task	More information
Check performance	Measure an Integration's Performance
Check tracing levels	Check an Integration's Tracing Level
Stay within the limits	Make sure that your organization's active integrations stay below the limit. On the Dashboards page, see the numbers in the **Active integrations** box. To explore all service limits, see Service Limits in *Provisioning and Administering Oracle Integration 3*.
Confirm the staggered start times of schedule integrations	If too many schedule integrations start at the same time, they might experience contention. Ease the contention by staggering the start times. See View the Calendar of Schedule Integration Runs.
Check the status of all components	See the Current Status of All Integration Components
Update documentation	Periodically review the documentation for your automation solution and make updates for organizational changes and employee turnover.

Tours and Tutorials

The following tours and tutorials introduce you to integrations and show the ease of designing, activating, invoking, and monitoring an integration.

Tours

Tours walk you through the design-time and runtime aspects of integrations and require no hands-on activities.

> **Note**
>
> Sample recipes and accelerators are available for download from the Integration Store. See Get Familiar with the Home Page in *Getting Started with Oracle Integration 3* and Recipes and Accelerators on the Oracle Help Center.

Tour	Complexity	Description
Explore the Navigation Pane	Low	Learn how the navigation menu helps you find your way around.
Tour: Explore an Integration That Sends a Hello World Message	None	Oracle Integration includes a recipe for a Hello World integration, which uses a simple log notification action with a REST Adapter in an application integration. In this tour, you review the design-time and runtime aspects of a Hello World integration.
Tour: Explore a Multiple Verbs and Resources Invoke Integration	Medium	Oracle Integration includes a recipe that creates and invokes a REST web service with multiple HTTP verbs and resources. In this tour, you review the design-time and runtime aspects of this integration.

Tutorials

In a tutorial, you build everything you need for an integration from start to finish.

Copyright © 2022, 2026, Oracle and/or its affiliates.

Tutorial	Description
Build Your First Integration from Scratch	Design, run, and monitor a Hello World integration from start to finish.
Build Your Second Integration from Scratch	Design, run, and monitor an integration that receives a weather report for a city. This integration invokes a weather application with the REST Adapter.
Tutorial: Create an Integration to Import and Process Bulk Files	Create a schedule integration to load and transform files from a secure File Transfer Protocol (FTP) location.
Tutorial: Build and Run Your First AI Agent	Create a very simple AI agent with one integration as an agentic AI tool.
Tutorial: Add Human Approval to Your AI Agent	Add human approval tool to an existing AI agent.
Tutorial: Read Files and Write the Files to File Server	Use the File server action to read one or more files that are on an FTP server and then write the files to File Server in Oracle Integration.
Get Started with the Rapid Adapter Builder	Learn the basics of the Rapid Adapter Builder and build a sample adapter for an application.

LiveLab Workshops

In a hands-on, self-directed workshop, gain expertise in a couple hours.

- **LiveLabs for new users**
  - Get Started with Oracle Integration 3
  - Get Started with Agentic AI
  - Get Started with Oracle Integration B2B
  - Get Started with Healthcare
  - Get Started with Process Automation
  - Get Started with Rapid Adapter Builder
  - Get Started with Robotic Process Automation (RPA)
- **LiveLabs for Oracle Fusion Cloud Applications**
  - ERP Cloud and B2B for Oracle Integration
  - ERP Cloud BIP Report
  - ERP Cloud Bulk Extract
  - ERP Cloud Business Events
  - ERP Cloud Business Services
  - ERP Cloud FBDI Import
  - ERP Cloud Real-Time Synchronization
  - HCM Cloud Bulk Data Import
  - HCM Cloud Directory Synchronization
- **Additional LiveLabs**
  - Automate Expenses with Agentic AI

Copyright © 2022, 2026, Oracle and/or its affiliates.

- Automate Invoice Processing with OCI Document Understanding
- Build Intelligent Invoice Automation with OCI Document Understanding and Decisions
- Decouple Producers and Subscribers Using Events
- File Server Events
- Understand the REST Adapter
- Use Built-In Actions to Harness OCI Services

Explore the Navigation Pane

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

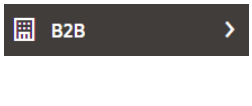
When you start Oracle Integration, the navigation pane is hidden. To view the navigation pane, click **Show/Hide navigation menu** ☰ in the top corner of the Home page.

The menu displayed in the navigation pane depends on your assigned role, your selection, and your location in Oracle Integration. The navigation pane may display the main menu or a submenu.

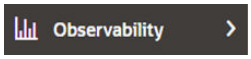
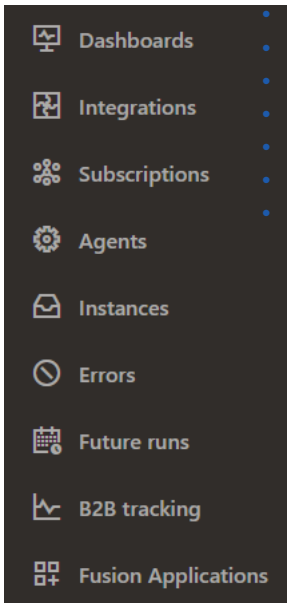
Browse the menu and submenu selections, which give you access to the Oracle Integration runtime environment, design-time environment, development tools, and administration settings.

Menu Selections	More Information
Home	Access the launch pad and high-level dashboard for Oracle Integration, search for recipes and accelerators, and open recently worked on projects.
Projects	Launch the page to create integration projects. An integration project consists of self-contained assets (integrations, connections, lookups, and JavaScript libraries) bundled into a solution that can be developed, managed, and monitored as a single unit from a single workspace. For more information about working with projects, see the following topics: • About Integration Projects • Designing, Managing, and Monitoring Integrations in Projects

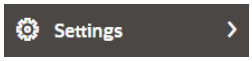
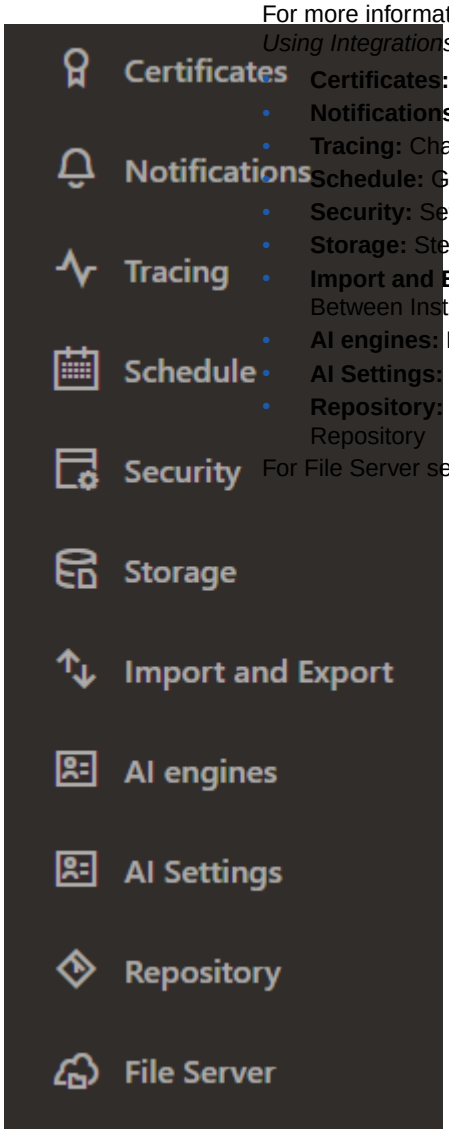
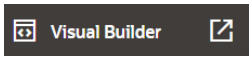
Copyright © 2022, 2026, Oracle and/or its affiliates.

Menu Selections	More Information
	Launch the design-time environment for integrations. Configure connections, and create and activate integrations. Work with lookups, libraries, packages, agents, and adapters. For more information about the submenu selections, see the following topics: • **Integrations:** Create an Integration • **Connections:** Create Connections • **Events:** Create Integrations to Publish and Subscribe to Events • **Lookups:** Map Data and Create Lookups • **Libraries:** Use JavaScript Libraries in Integrations • **Packages:** Manage Packages • **Agents:** Download and Run the Connectivity Agent Installer • **Adapters:** View Preinstalled Adapters
	Exchange business documents between Oracle Integration and a trading partner securely and reliably. For more information about the submenu selections, see the following topics: • **Documents:** Work with B2B Documents • **Schemas:** Work with B2B Schemas • **Trading partners:** Use B2B for Oracle Integration in Trading Partner Mode • **Host profile:** Define the Host Profile
	Exchange messages between Oracle Integration and healthcare organizations that use HL7 and FHIR. For more information about the submenu selections, see the following topics: • **HL7 messages**: Create a Custom HL7 Message • **Schemas**: Create a New Healthcare Schema • **FHIR profiles**: Import FHIR Profile Packages

Copyright © 2022, 2026, Oracle and/or its affiliates.

Menu Selections	More Information
 	Monitor integrations in the runtime environment. For more information about the submenu selections, see the following topics: • **Dashboards:** View the Dashboard • **Integrations:** Monitor the Message Processing Status of Integrations • **Subscriptions:** Monitor Event Subscription Integration Status • **Agents:** Monitor Agents • **Instances:** Track Integration Instances • **Errors:** Manage Errors • **Future runs:** View the Scheduled Integration Runs Calendar • **B2B tracking:** Track B2B Messages • **Fusion Applications:** Diagnose and Manage Event-Based Oracle Fusion Applications Integrations

Copyright © 2022, 2026, Oracle and/or its affiliates.

Menu Selections	More Information
	Configure settings for security certificates, notifications, tracing, schedules, time limits for inactive sessions, storage buckets, import and export of design-time metadata, and artificial intelligence (AI) engines. Access File Server settings for preferences, users, groups, and folders. For more information about the submenu selections, see the following topics in *Using Integrations in Oracle Integration 3*: • **Certificates:** Manage Security Certificates • **Notifications:** Configure Notification Emails • **Tracing:** Change the Tracing Level on an Active Integration • **Schedule:** Globally Change the Submitter of Integration Schedules • **Security:** Set the Time Limit for Inactive Sessions • **Storage:** Step 2: Configure the Instance Object Storage Bucket • **Import and Export:** Step 3: Export and Import Design-Time Metadata Between Instances • **AI engines:** Manage Your Registered AI Engines • **AI Settings:** Enable or Disable AI Features • **Repository:** Integrate Projects and Project Deployments with a GitHub Repository For File Server settings, see Administer File Server.
	
	Launch Visual Builder to create and publish web and mobile applications. For more information about working with Visual Builder, see Get Started with Visual Builder in *Developing Applications with Oracle Visual Builder in Oracle Integration 3*.
	Launch Oracle Cloud Infrastructure Process Automation to rapidly design, automate, and manage business processes in the cloud. For more information about working with Oracle Cloud Infrastructure Process Automation, see Use Process Automation with Oracle Integration 3 in *Using Oracle Cloud Infrastructure Process Automation*.

Copyright © 2022, 2026, Oracle and/or its affiliates.

Tour: Explore an Integration That Sends a Hello World Message

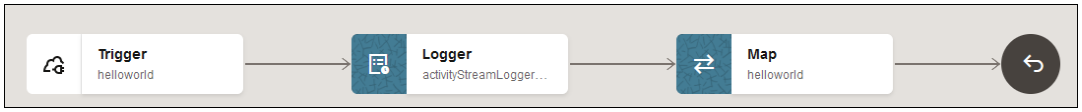
This tour provides a walkthrough of a Hello World integration, including steps to monitor the integration.

To see a similar Hello World integration in your instance, you can install a Hello World recipe. Or, simply review the screenshots and steps below to understand the integration and your workflow. You can also build a Hello World integration from scratch. See Tutorial: Build Your First Integration From Scratch.

Step 1. View the Integration Design

This integration is based on an application integration. It triggers the REST Adapter when you specify the integration name. The integration creates a logging message and sends it to the activity stream for logging.

1. In the navigation pane, click **Design**, then **Integrations**.
   This integration consists of the following:



   - A REST Adapter is configured as a trigger (inbound) connection in the integration. The REST Adapter is configured with a resource endpoint of `/input/{Helloworld}` and a GET operation. The REST Adapter is configured to receive a message and respond with a message. The REST Adapter is triggered when you specify an integration name on the Configure and run page.
   - A logger action is configured with a `concat` function that accepts the integration name you specify on the Configure and run page and concatenates it to the message.

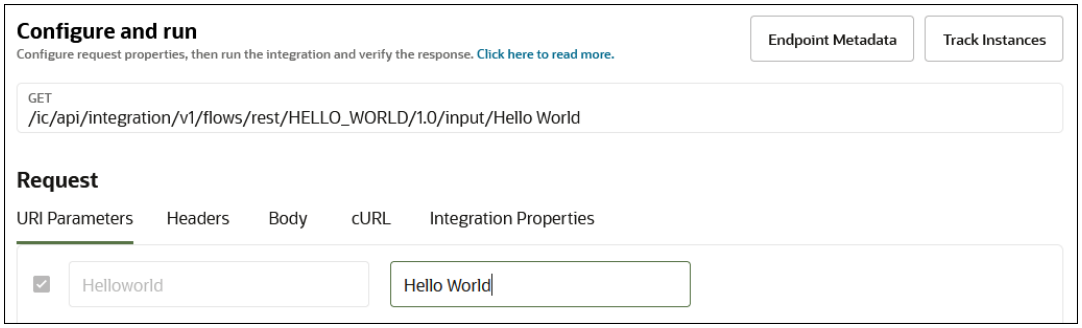
     ```
     concat('Integration name is :', name)
     ```

   - A mapper maps the source value to the target message element.
   - A response is returned.

Step 2. View How to Run the Integration

1. Hover your cursor over **Hello World**, then click **Activate** .
2. Click **Activate** when prompted.
3. Wait for the icon to turn green and the word **Active** to appear in the **Status** column, indicating that the integration is activated.
4. Hover your cursor over **Hello World**, then click and select **Run**. The Configure and run page is displayed. This page is available only for REST Adapter-trigger based integrations and scheduled integrations.
5. Enter `Hello World`, and click **Run**.

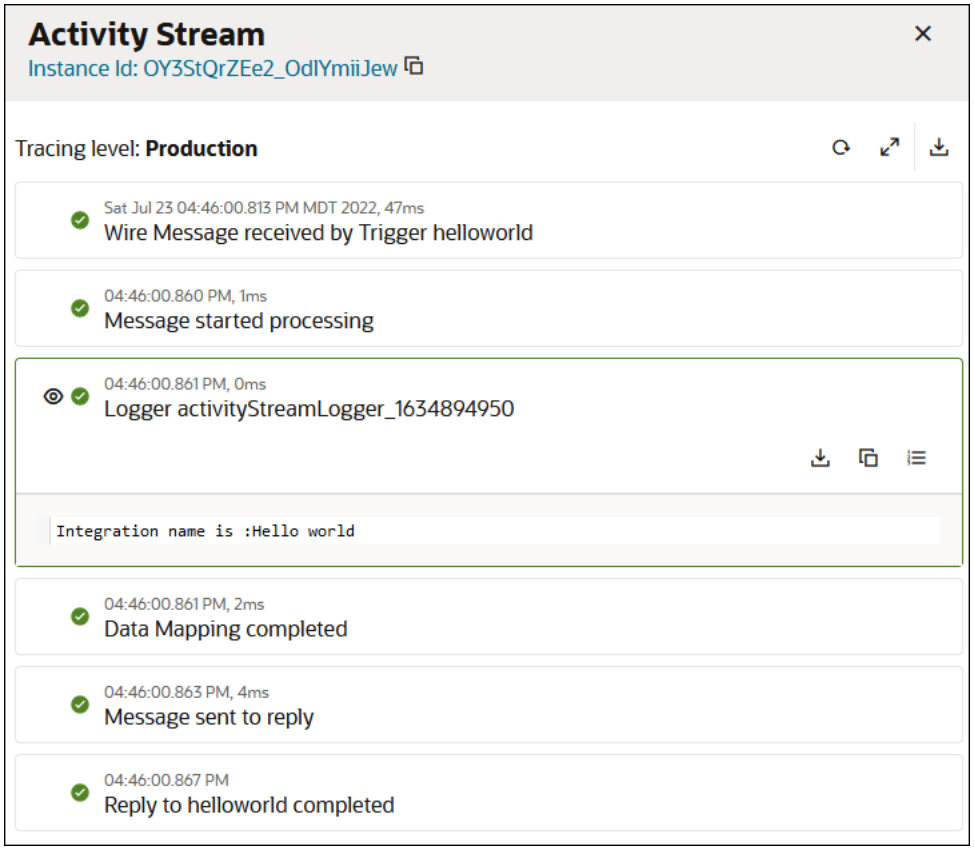
Copyright © 2022, 2026, Oracle and/or its affiliates.



Step 3. View the Movement of Messages Through the Integration

The Activity Stream panel slides open on the Configure and run page. The activity stream shows the status of message processing through the major milestones of the integration.

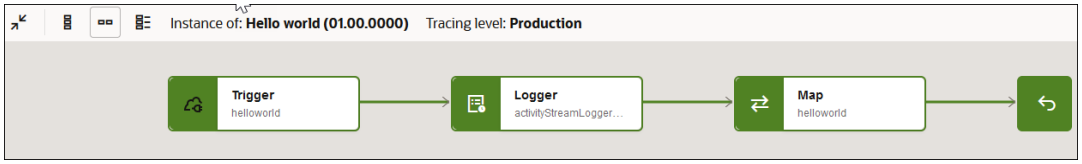
1. Expand the logger action to see the message.



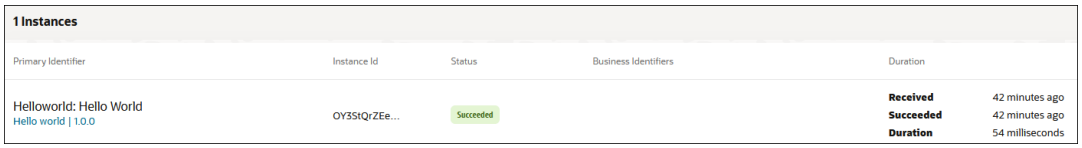
Step 4. View How to Monitor the Integration

1. Click the instance ID at the top of the Activity Stream panel. The Instance Details page provides a graphical view of the integration flow. The same Activity Stream panel is also visible on the right.

Copyright © 2022, 2026, Oracle and/or its affiliates.



2. Click **<** to return to the Configure and run page.
3. Click **Track Instances** and note that the **Hello World** integration instance is listed as **Succeeded** in the **Status** column.



4. In the navigation pane, click **Observability**, then **Integrations**.

   The message is listed as successfully received and processed without any errors.



Learn More About The Features in this Tour

- Create Application Integrations
- REST Adapter Capabilities
- Log Messages with a Logger Action
- Test Integrations from Outside the Integration Canvas
- Getting Started with the Mapper

Tour: Explore a Multiple Verbs and Resources Invoke Integration

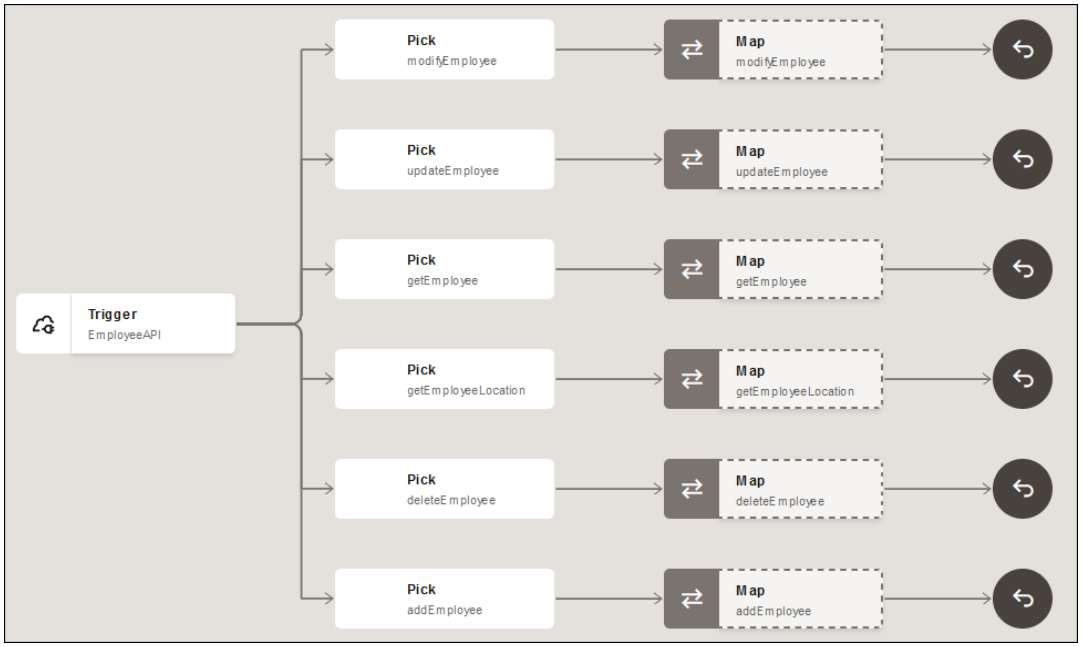
This example shows the Use Multiple Verbs and Resources Invoke recipe, which demonstrates how to use multiple HTTP verbs and resources configured in the REST Adapter trigger connection of an integration. This recipe is available from the Integration Store.

You can configure a single REST Adapter to include multiple HTTP verbs such as GET, PUT, POST, PATCH, and DELETE. You can also configure the REST Adapter with multiple resources. For this example, a single `/emp` resource and an `/emp/loc` subresource are used.

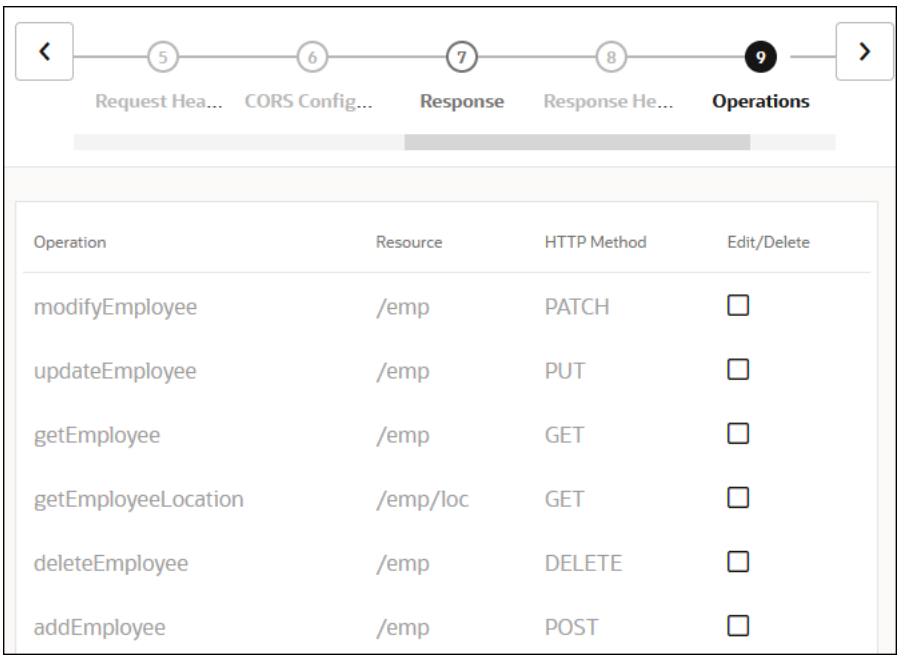
Step 1. View the Integration Design

1. In the navigation pane, click **Design**, then **Integrations**.
   This integration consists of the following:

Copyright © 2022, 2026, Oracle and/or its affiliates.



- A REST Adapter connection (**EmployeeAPI**) is configured as a trigger (inbound) connection in the integration. The REST Adapter connection is configured with multiple operations: **modifyEmployee** (**PATCH**), **updateEmployee** (**PUT**), **getEmployee** (**GET**), **getEmployeeLocation** (**GET**), **deleteEmployee** (**DELETE**), and **addEmployee** (**POST**).



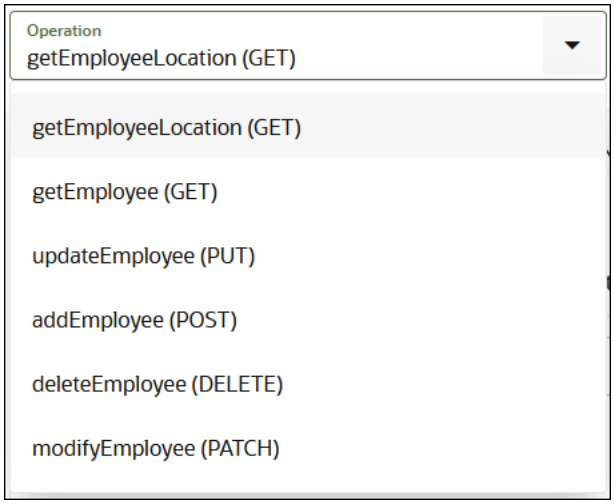
- The REST Adapter is triggered when you specify the URL.
- Each operation is shown as a different path that branches out from the trigger connection.
- Each branch has a map and a reply action configured.

Copyright © 2022, 2026, Oracle and/or its affiliates.

- The tracking configuration for each branch can be viewed by clicking **Actions** ⋮, then **tracking** on each pick action.

Step 2. View How to Run the Integration

1. In the navigation pane, click **Design**, then **Integrations**.
2. Hover your cursor over **Sample Multiple Verbs and Resources Invoke**, then click **Activate** ⏻.
3. Click **Activate** when prompted.
4. Wait for the icon to turn green and the word **Active** to appear in the **Status** column, indicating that the integration is activated.
5. Hover your cursor over **Sample Multiple Verbs and Resources Invoke**, then click **Actions** ⋮ and select **Run**. The Configure and run page is displayed with details about running, tracking, and testing the integration.
6. Click the **Operation** drop-down list to view all configured options and HTTP verbs.
7. From the drop-down list, select the operation to perform. For this example, **addEmployee (POST)** is selected.



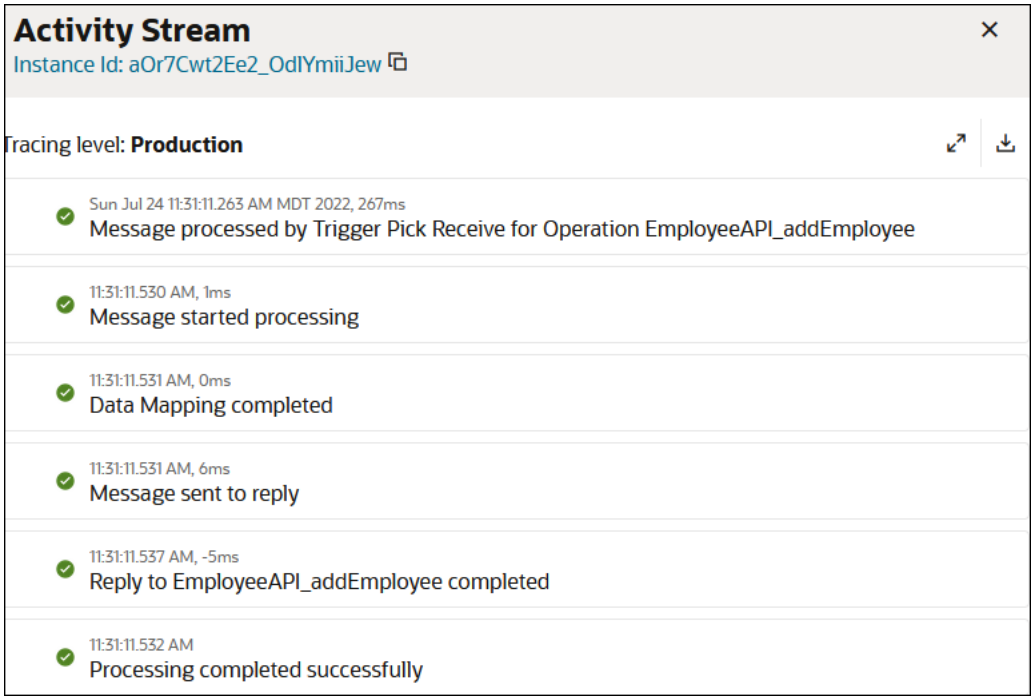
The section following the drop-down list shows all the options available for the selected operation such as **URI Parameters** and **Body**.

8. Click **Run** to run the selected operation.

Step 3. View the Movement of Messages Through the Integration

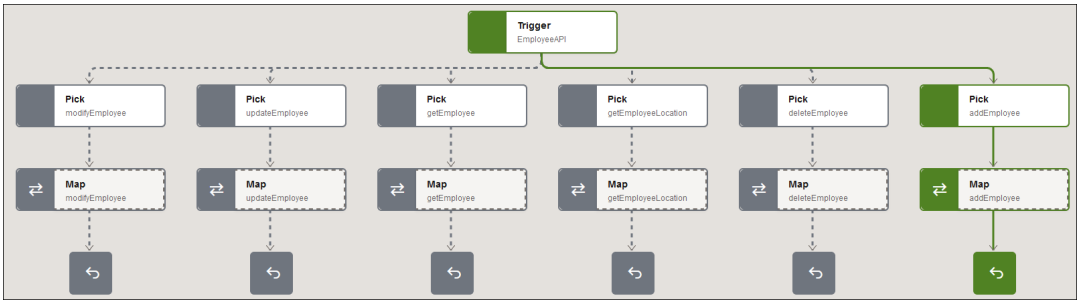
The Activity Stream panel slides open on the Configure and run page. The activity stream shows the status of message processing through the major milestones of the integration.

Copyright © 2022, 2026, Oracle and/or its affiliates.

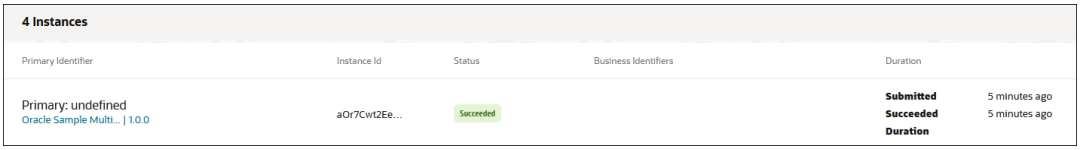


Step 4. View How to Monitor the Integration

1. Click the instance ID at the top of the Activity Stream panel. The Instance Details page provides a graphical view of the integration flow. The path taken by this instance is highlighted in green, indicating it was successful. The same Activity Stream panel is also visible on the right.

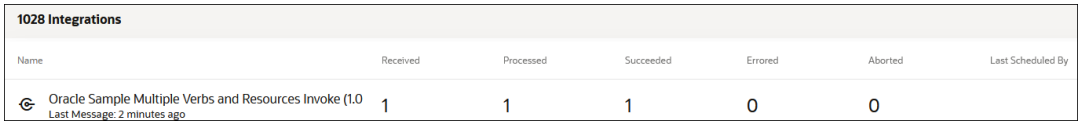


2. Click **<** to return to the Configure and run page.
3. Click **Track Instances** and note that the **Sample Multiple Verbs and Resources Invoke** integration instance is listed as **Succeeded** in the **Status** column.



4. In the navigation pane, click **Observability**, then **Integrations**.
The message is listed as successfully received and processed without any errors.

Copyright © 2022, 2026, Oracle and/or its affiliates.



Learn More About the Features in This Tour

- Create Application Integrations
- Receive Requests for Multiple Resources in a Single REST Adapter Trigger with a Pick Action
- Test Integrations from Outside the Integration Canvas
- Getting Started with the Mapper

Tutorial: Build Your First Integration From Scratch

Let's walk through how to create, design, activate, run, and monitor your first integration completely from scratch. You work inside a project. A project contains related resources, including integrations, connections, and more. You can design, manage, and monitor everything in one place.

- What Do I Learn from this Tutorial?
- Step 1. Create a Project in Which to Design, Manage, and Monitor Your Integration
- Step 2. Create an Integration
- Step 3. Create a Trigger Connection to Allow Your Integration to Receive Messages
- Step 4. Specify How the Trigger Connection Receives Messages in the Integration
- Step 5. Design How the Integration Receives Messages
- Step 6. Define a Business Identifier to Track Message Delivery at Runtime
- Step 7. Activate and Run the Integration
- Step 8. Monitor Message Delivery Status at Runtime
- Learn More About The Features in this Tutorial

What Do I Learn from this Tutorial?

This tutorial walks you through performing the following tasks:

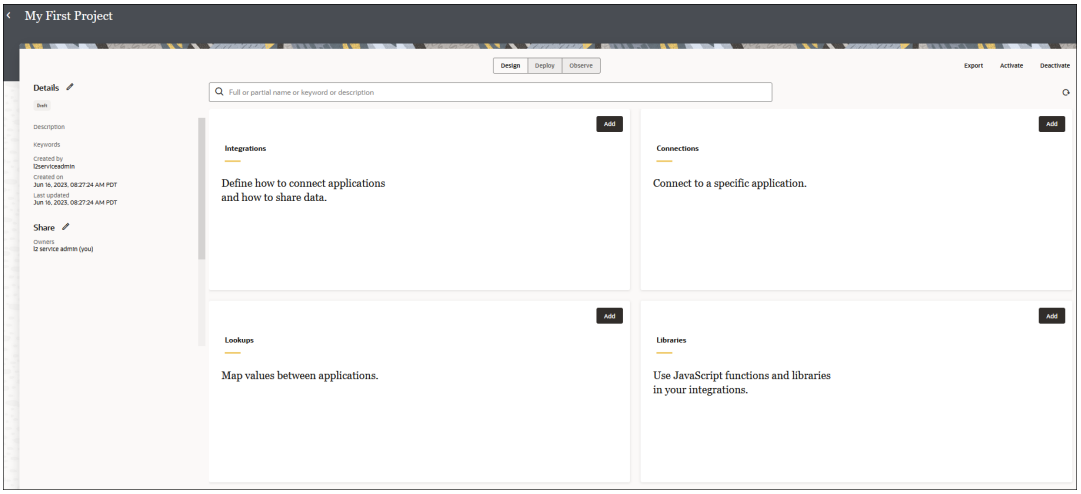
- Creating a project in which you perform all design, management, and monitoring tasks.
- Creating and configuring an adapter connection that lets your integration receive inbound messages.
- Creating and configuring an integration that takes either of two paths to process the inbound message:
  - If an email address is specified
  - If an email address is not specified
- Running (triggering) the integration by specifying a simple message consisting of a name, an optional flow ID, and an email address.
- Monitoring the results of the integration run.

Copyright © 2022, 2026, Oracle and/or its affiliates.

Step 1. Create a Project in Which to Design, Manage, and Monitor Your Integration

You manage the entire life cycle of an integration from within a project.

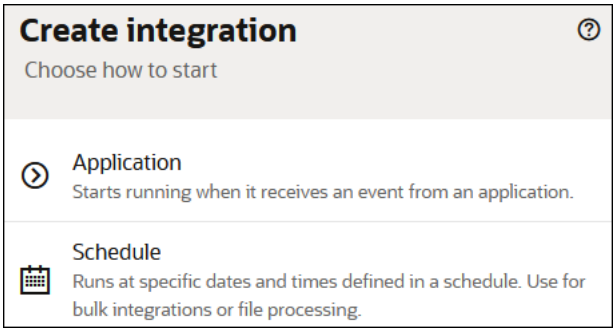
1. In the navigation pane, click **Projects**.
2. Click **Add**, then select **Create**.
3. In the **Name** field, enter `My First Project`.
4. Leave all other fields as they are, then click **Create**.
   The page for your first project is displayed. Your project contains sections for creating, designing, deploying, and monitoring your integration.



Step 2. Create an Integration

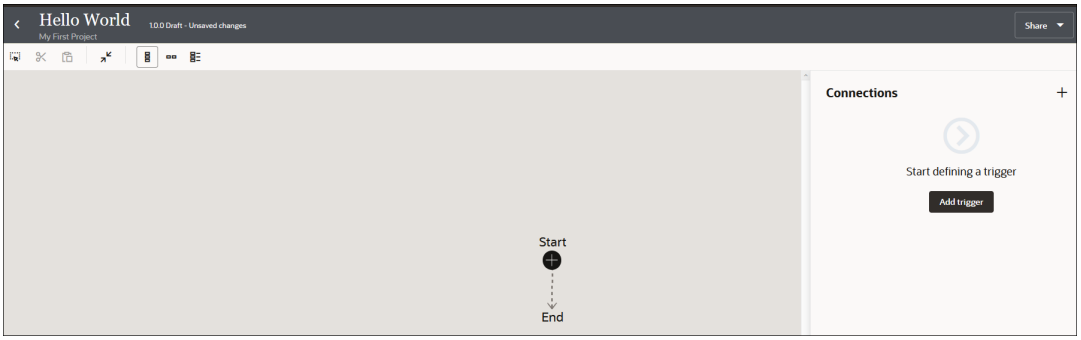
Let's create your first integration inside the project.

1. In the **Integrations** section, click **Add**.
   The Add integration panel opens.
2. Click **Create**, then select **Application**.



3. In the **Name** field, enter `Hello World`.
4. Leave all other fields as they are, then click **Create**.
   The integration canvas opens with an option for creating a trigger connection. The trigger connection lets your integration receive inbound messages from an application.

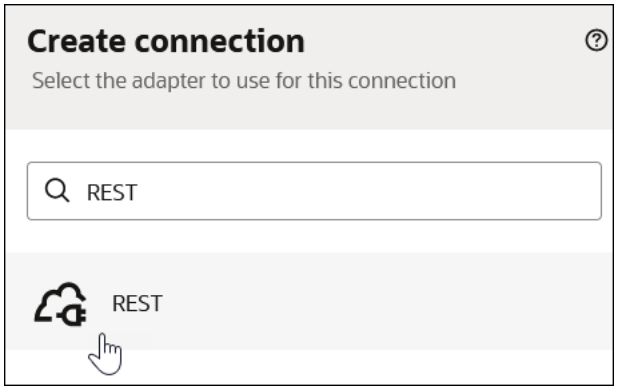
Copyright © 2022, 2026, Oracle and/or its affiliates.



Step 3. Create a Trigger Connection to Allow Your Integration to Receive Messages

Now let's create a trigger connection inside the integration canvas. The trigger (source) connection allows your integration to receive inbound messages from an application.

1. In the Connections panel, click **Add trigger**.
   The Create connection panel opens.
2. Enter REST in the search field.
3. Select **REST** as the adapter connection to use. For this tutorial, the REST Adapter processes inbound messages sent to Oracle Integration.



The page for configuring the REST Adapter opens.

a. Enter the following information.

Field	Description
Name	Enter `Trigger1`.
Role	Select **Trigger**. Roles perform the following tasks. • **Trigger**: Enables an inbound application (endpoint) to trigger (run) the integration. • **Invoke**: Enables your integration to invoke an outbound application (endpoint).

b. Ignore all other fields, and click **Create**.
   The page for configuring security and delivery methods for the REST Adapter appears.

c. Enter the following information:

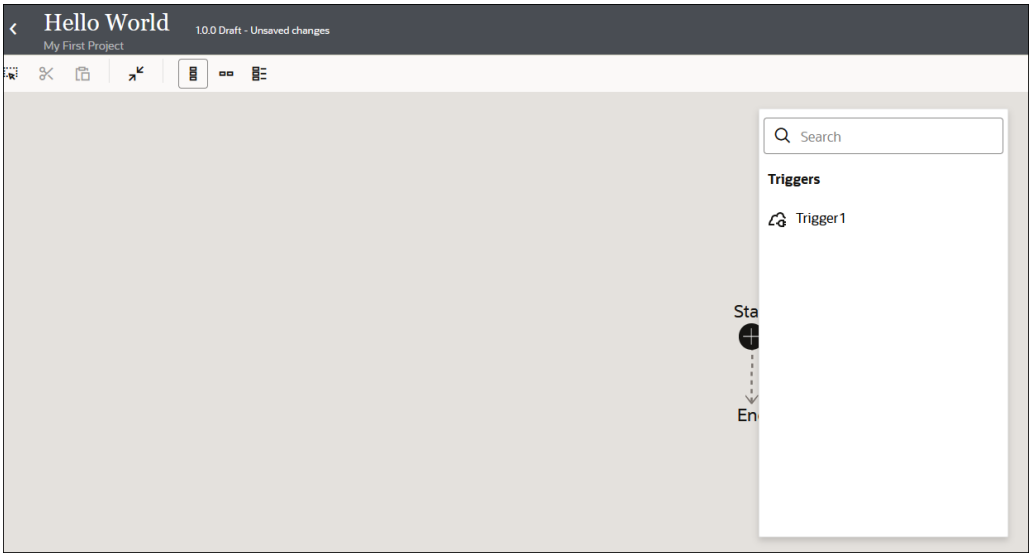
Copyright © 2022, 2026, Oracle and/or its affiliates.

Field	Description
Security	From the list, select **OAuth 2.0 Or Basic Authentication** to secure incoming messages.
Access type	Ensure **Public gateway** is selected. This access type uses the public internet to receive messages.

d. Click **Test** to confirm that you configured the connection successfully. When testing completes, a success message is displayed.

e. Click **Save**.

f. Click **Back** to return to the integration canvas. A menu shows that the newly-created **Trigger1** connection is available for selection.



Step 4. Specify How the Trigger Connection Receives Messages in the Integration

Now let's configure the REST Adapter connection in the integration. During configuration, you specify details such as the operation, the relative resource URI, the query parameters, and the message payload content.

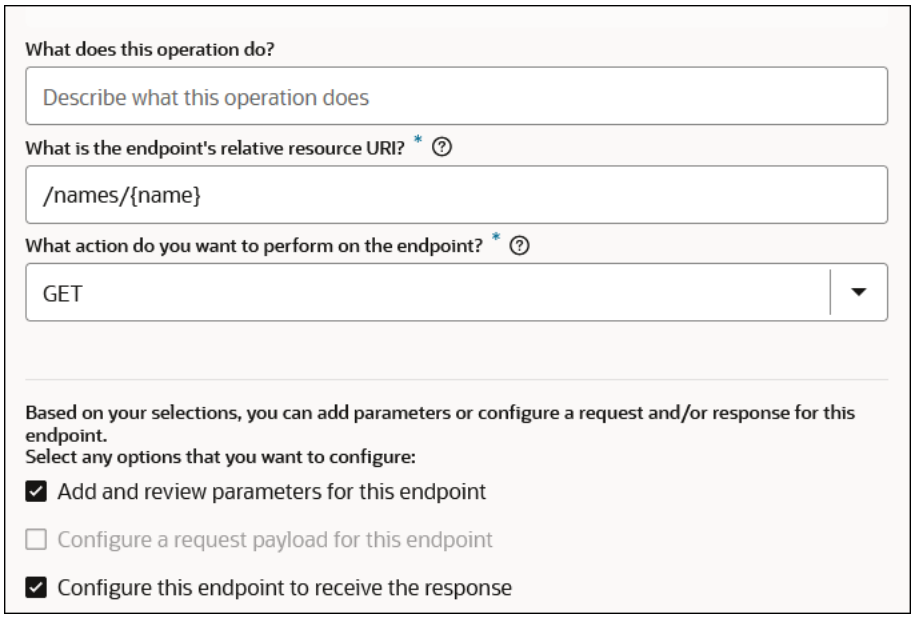
1. Select **Trigger1**.
   The Basic Info page of the Adapter Endpoint Configuration Wizard opens. This wizard enables you to configure the **Trigger1** connection for use in the integration.
2. In the **What do you want to call your endpoint** field, enter `helloWorld`, then click **Continue** to access the Resource Configuration page.
3. Enter the following information.

Field	Description
What is the endpoint's relative resource URI	Enter the following: `/names/{name}`
What action do you want to perform on the endpoint	Select **GET**.

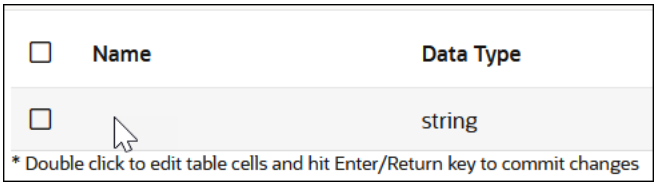
Copyright © 2022, 2026, Oracle and/or its affiliates.

Field	Description
Select any options that you want to configure	Scroll to the bottom of the page and ensure that the following options are selected. • **Add and review parameters for this endpoint** • **Configure this endpoint to receive the response**

The completed Resource Configuration page looks as follows.

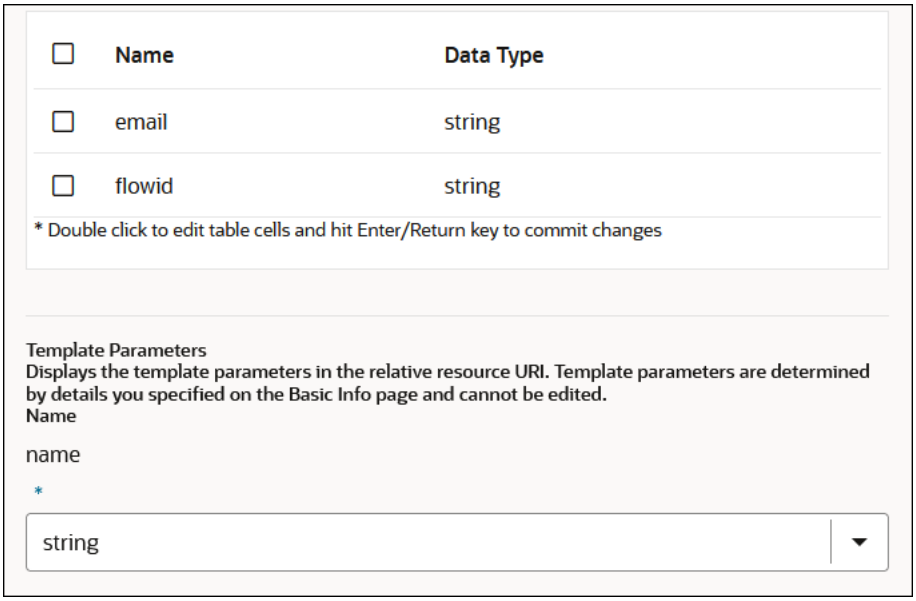


4. Click **Continue** to access the Request Parameters page.
   a. In the **Specify Query Parameters** section, click **Add**.
   b. Double-click the row in the **Name** column to enter a value.



   c. Enter `email`, then select **string** from the **Data Type** list.
   d. Click **Add** again.
   e. Enter `flowid`, then select **string** from the **Data Type** list.
   Both query parameters are displayed. Note also that the **name** template parameter from the relative resource URI you specified in Step 3 is also included. You select these parameters later when designing the business logic for your integration.

Copyright © 2022, 2026, Oracle and/or its affiliates.



f. Click outside the **flowid** row.

g. Click **Continue** to access the Response page.

5. Enter the following information.

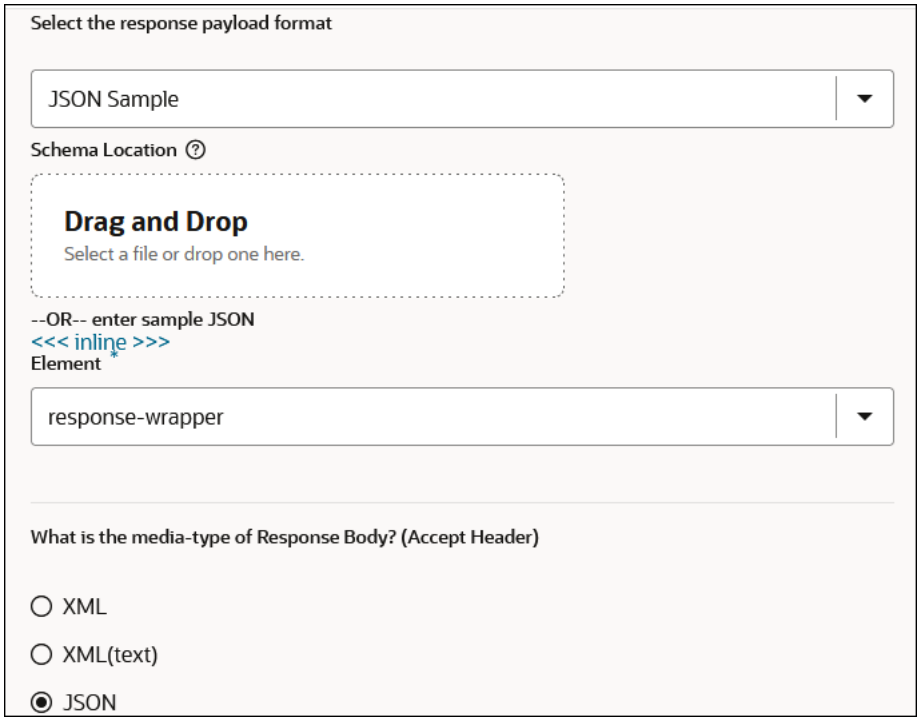
Field	Description
Select the response payload format	Select **JSON Sample**.

<<< inline >>>	Click this field and enter the following content, then click **OK**.
	
	`{`
	`  "Hello" : "Name",`
	`  "Message" : "Email Sent",`
	`  "Email" : "EmailAddress"`
	`}`

Element	Ensure **response-wrapper** is selected.
What is the media-type of Response Body	Ensure **JSON** is selected.

The completed Response page looks as follows.

Copyright © 2022, 2026, Oracle and/or its affiliates.



6. Click **Continue** to access the Summary page.
7. Review your selections, then click **Finish**.
   The REST Adapter trigger connection is now fully configured for use in the integration.

Step 5. Design How the Integration Receives Messages

Now let's fully design the business logic for your integration. You first create and design a switch action with two paths. The path taken at runtime is based on whether you specify an email address:

- Route 1: This path is taken if you specify an email address.
- Otherwise: This path is taken if you don't specify an email address.

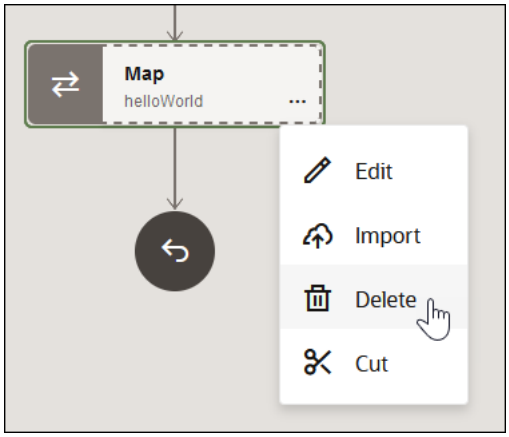
> **Note**
>
> You use the Expression Builder to design much of the business logic for your integration. Be careful while entering your expressions. Simple mistakes such as forgetting a comma or entering an extra blank space can cause validation failure.

Design the Route 1 path of the integration

Let's first delete the map action that was automatically created when you configured the REST Adapter trigger connection in the integration. Instead, you add the necessary map actions for mapping source data to target data inside both switch action paths.

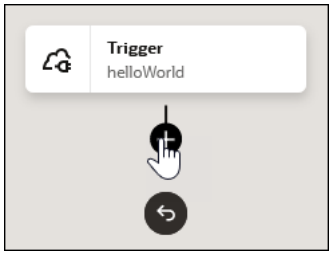
1. In the map action, click **Actions** ⋮, then select **Delete**.

Copyright © 2022, 2026, Oracle and/or its affiliates.

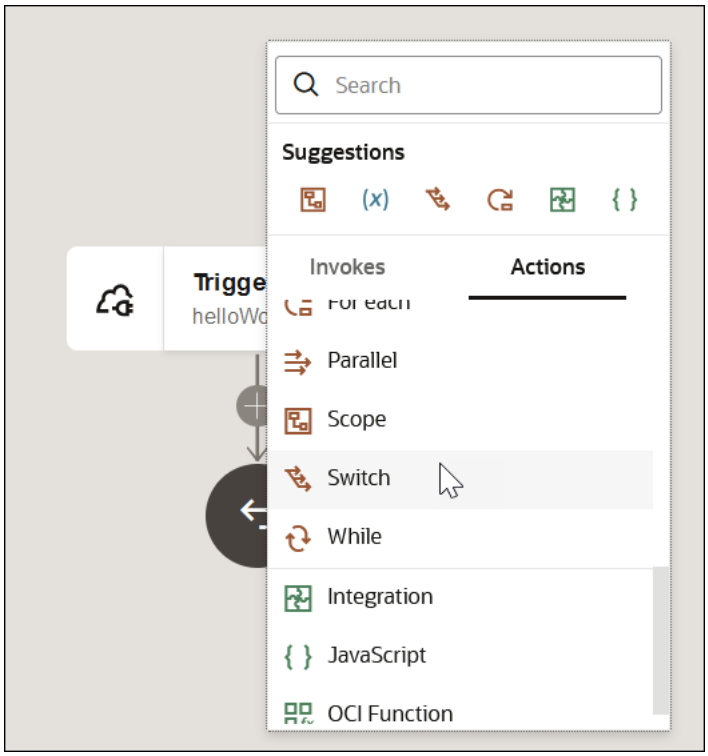


2. Click **Delete** when prompted.
   Now let's add a switch action to define routing expressions in your integration. The switch action takes the first path that evaluates to true. The other paths are ignored.

3. Hover over the line below the trigger connection and click **Add** .

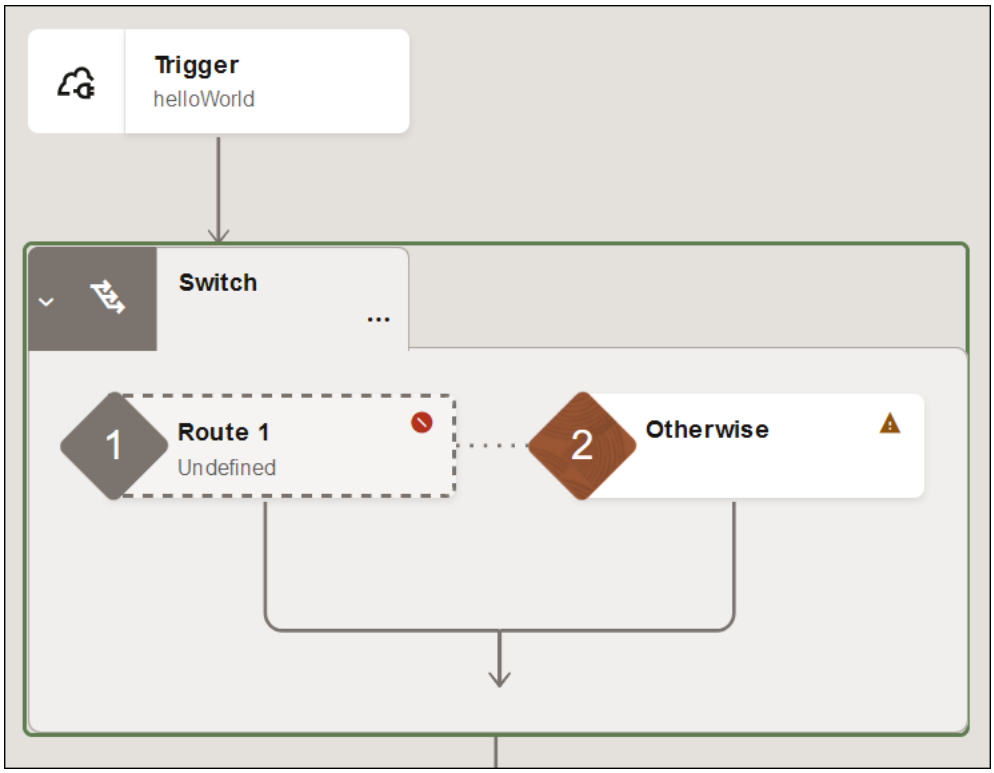


4. Click **Actions**, then **Switch**.



Copyright © 2022, 2026, Oracle and/or its affiliates.

5. In the switch action, click **Actions** ⋯, then select **Add**, and then **Otherwise**.
A switch action with two branches is displayed.



a. Hover your cursor inside **Route 1**, and select **Actions** ⋯, then **Edit**.
The Configure route panel opens.

b. Click **Edit**.



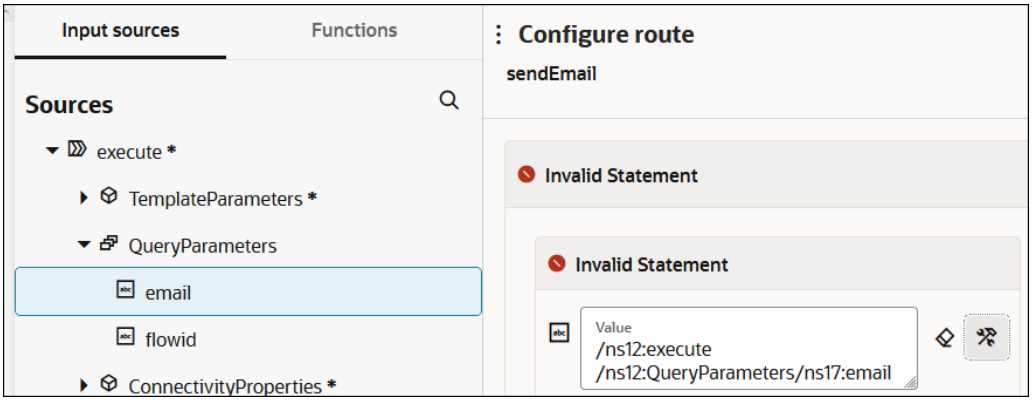
c. Enter `sendEmail` as the name for the switch action, then click **Apply**.

d. To the right of the **Value** field, click **Switch to Developer View**.

e. In the **Sources** tree, expand **QueryParameters**. These are the query parameters you defined when configuring the REST Adapter in the integration.

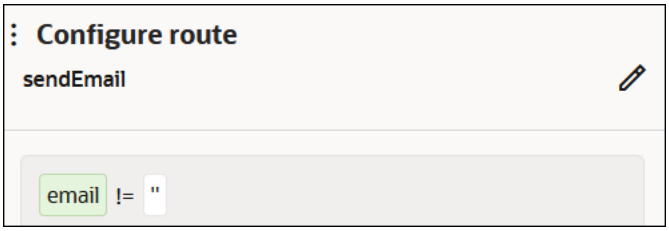
f. Drag **email** to the **Value** field. This parameter lets you specify the email address that gets logged to the activity stream when you test and run the integration.

Copyright © 2022, 2026, Oracle and/or its affiliates.



g. In the **Operator** list, select **!=**.

h. In the **Value** field, enter two single quotes ' ' with *no* blank space in between.
The completed routing expression looks as follows. When the email address that you specify at runtime is not an empty value, this route of the switch action is taken.



> **Note**
>
> As you design this integration, note that an error icon can appear on the right side of the page. Click this icon for details. Errors may occur due to mistakes you make in the Expression Builder. Follow the instructions in the Error panel to resolve the errors. Other errors that appear are resolved as you complete the design of all parts of this integration. Click outside the error panel to close it.
>
> 

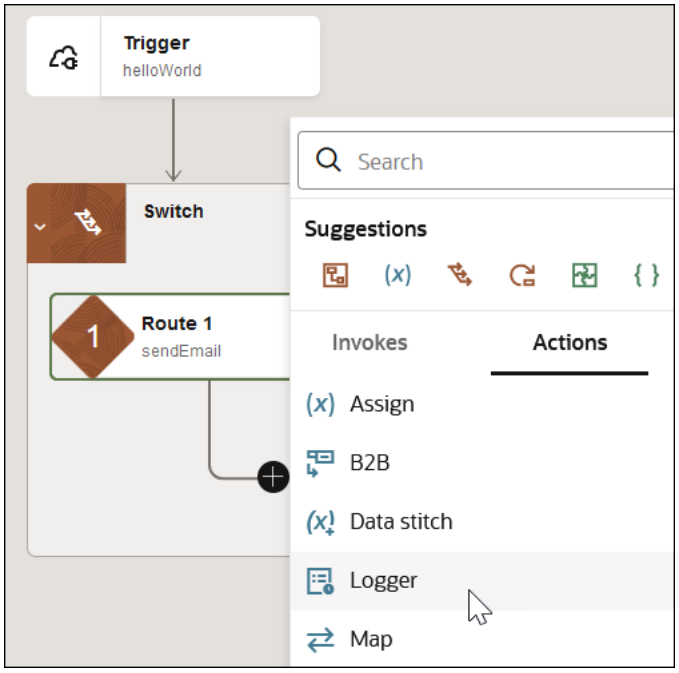
i. Click **Save**, then click outside the Configure route panel to close it.

Now let's add a logger action. This action enables you to log messages to the activity stream at runtime for diagnostic purposes.

6. Drag your cursor along the line below **Route 1** until the **Add** icon appears.

7. Click **Add**, then select **Actions**, then **Logger**.

Copyright © 2022, 2026, Oracle and/or its affiliates.



8. The Configure logger panel appears. The **Input Sources** and **Functions** tabs appear in the upper left part of the panel. You use the elements available under these tabs to build a logger expression.

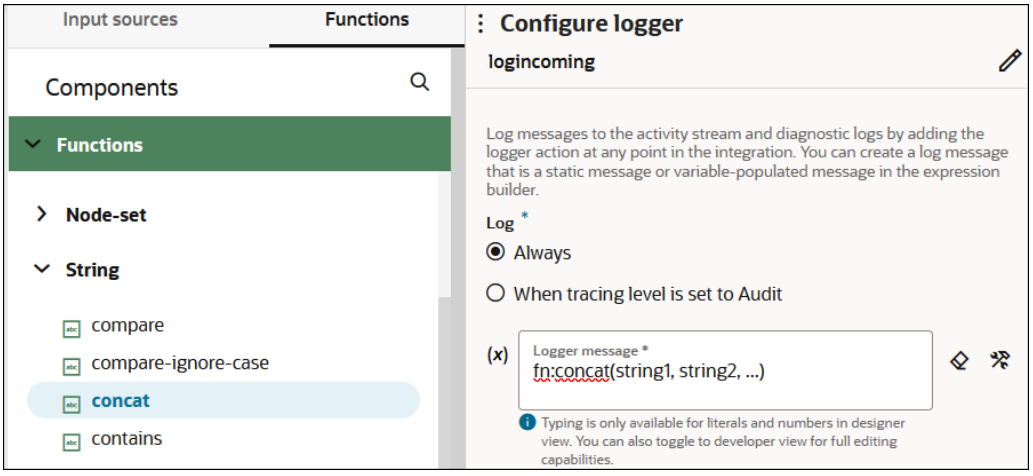
   a. Click **Edit** .

   b. Enter `logincoming` as the name for the logger action, then click **Apply** .

   c. To the right of the **Logger message** field, click **Switch to Developer Mode** .

   d. Click the **Functions** tab, then expand **String**.

   e. Drag a **concat** function into the **Logger message** field.



   f. Remove the content between the parentheses.

Copyright © 2022, 2026, Oracle and/or its affiliates.

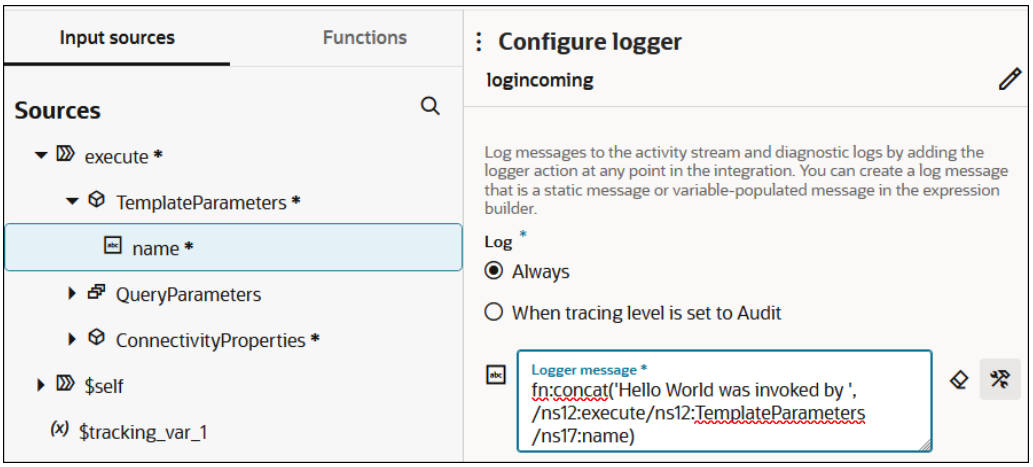
g. After the opening parenthesis, enter the following:

```
'Hello World was invoked by ',
```

Ensure that you enter a blank space before the single quote and after the comma.

h. Click the **Input sources** tab, and expand **TemplateParameters**.

i. Drag **name** before the closing parenthesis.



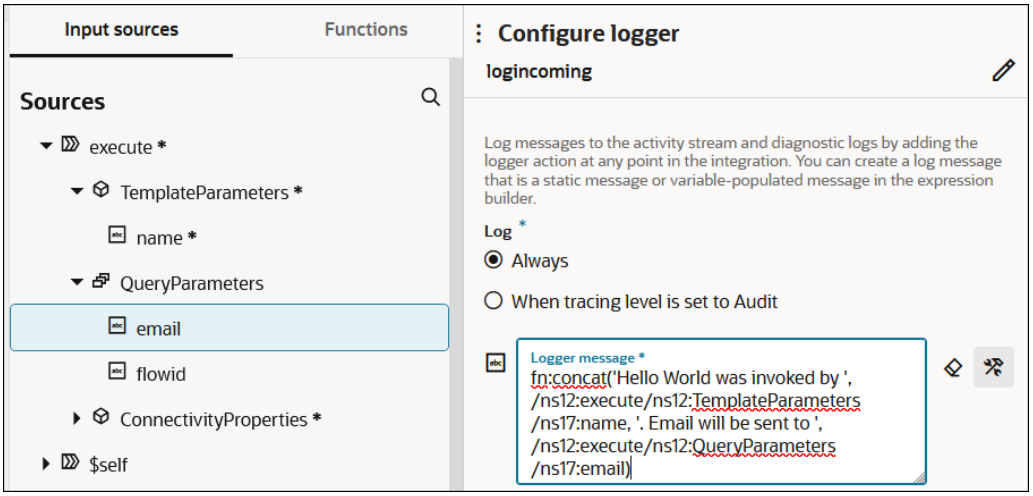
j. Before the closing parenthesis, enter the following:

```
, '. Email will be sent to ',
```

Note the single blank space after the last comma.

k. Expand **QueryParameters**.

l. Before the closing parenthesis, drag **email**.

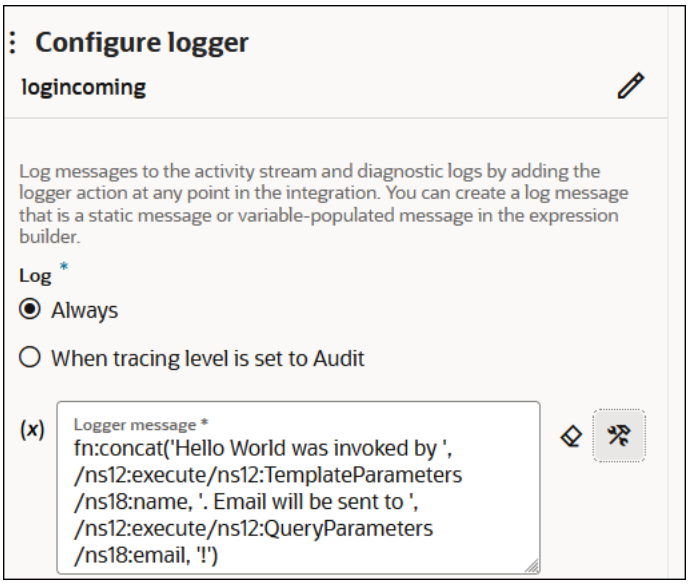


m. Before the closing parenthesis, enter the following:

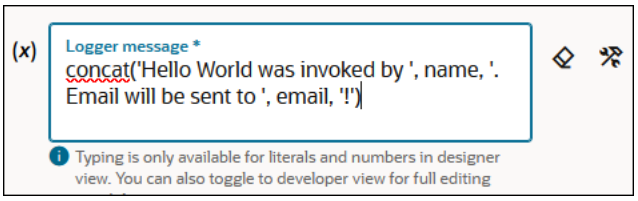
```
, '!'
```


Copyright © 2022, 2026, Oracle and/or its affiliates.

The completed logger action looks as follows.



n. Click **Switch to Designer View** to toggle to designer view, which shows a more user-friendly version of the expression.



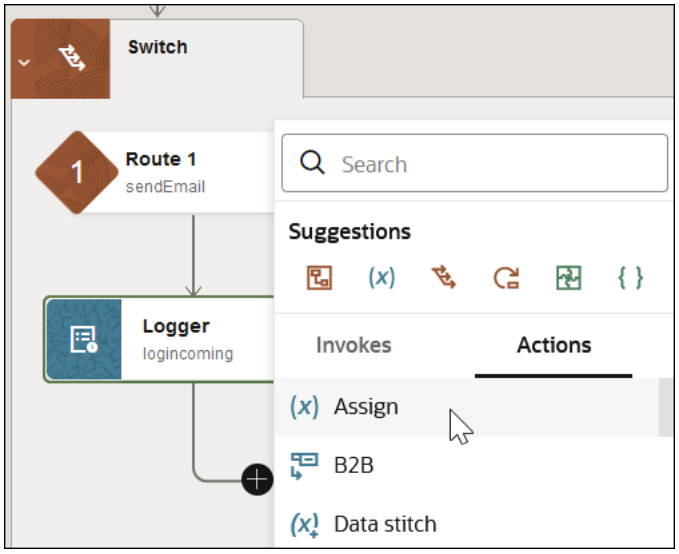
o. Click **Save**, then click outside the Configure logger panel to close it.

Now let's assign an optional secondary business identifier to track payload fields in messages during runtime. Each integration requires at least one business identifier (known as the primary business identifier) to track the status of your running instance (for example, was the run successful or did the run fail). Later in this tutorial, you assign the primary business identifier from a separate location in the integration canvas.

9. Drag your cursor along the line below **Logger** until the **Add** icon appears.

10. Click **Add**, then select **Actions**, then **Assign**.

Copyright © 2022, 2026, Oracle and/or its affiliates.

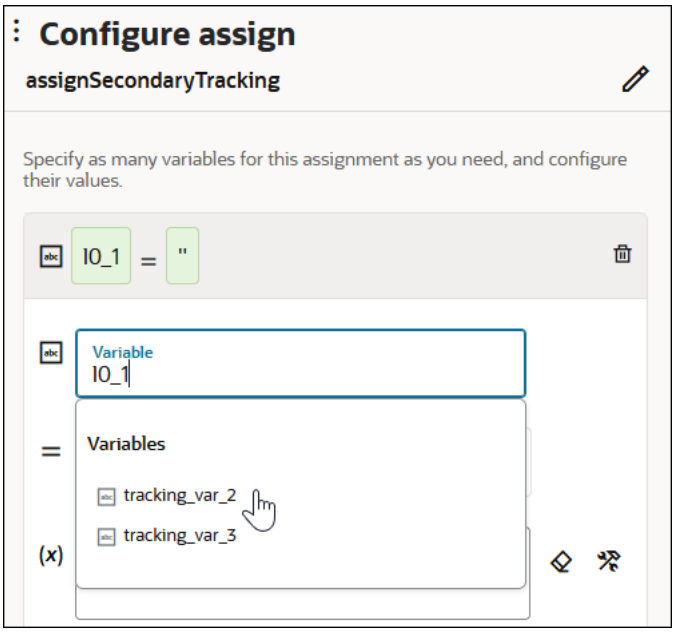


a. Click **Edit** .

b. Enter `assignSecondaryTracking` as the name for the assign action, then click **Apply** .

c. Click **Add** .

d. Click inside the **Variable** field, and select **tracking_var_2** from the list.

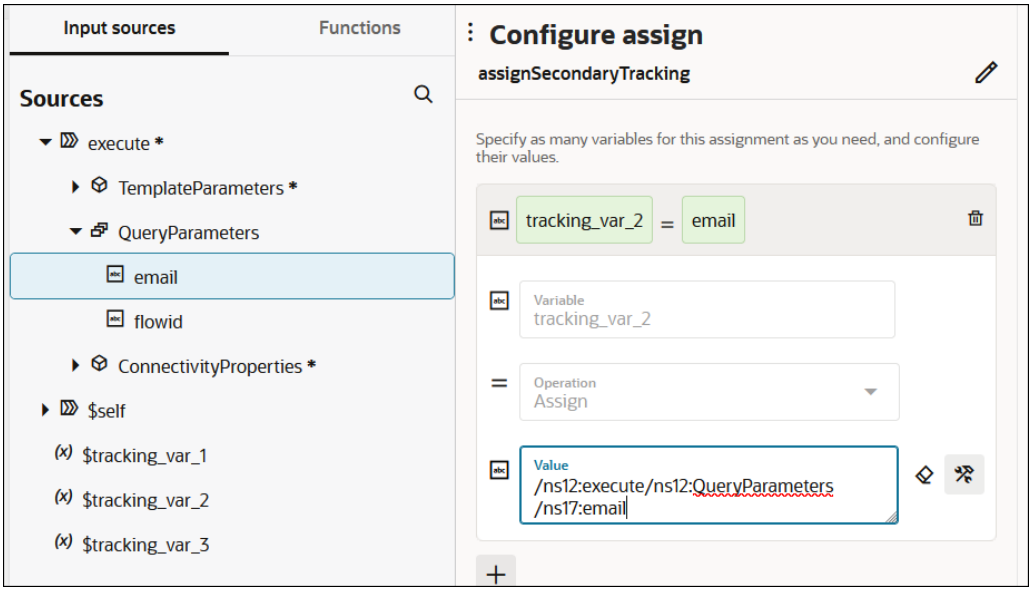


e. To the right of the **Value** field, click **Switch to Developer View** .

f. Clear the content from the **Value** field.

g. Expand **QueryParameters**, and drag **email** into the **Value** field.

Copyright © 2022, 2026, Oracle and/or its affiliates.

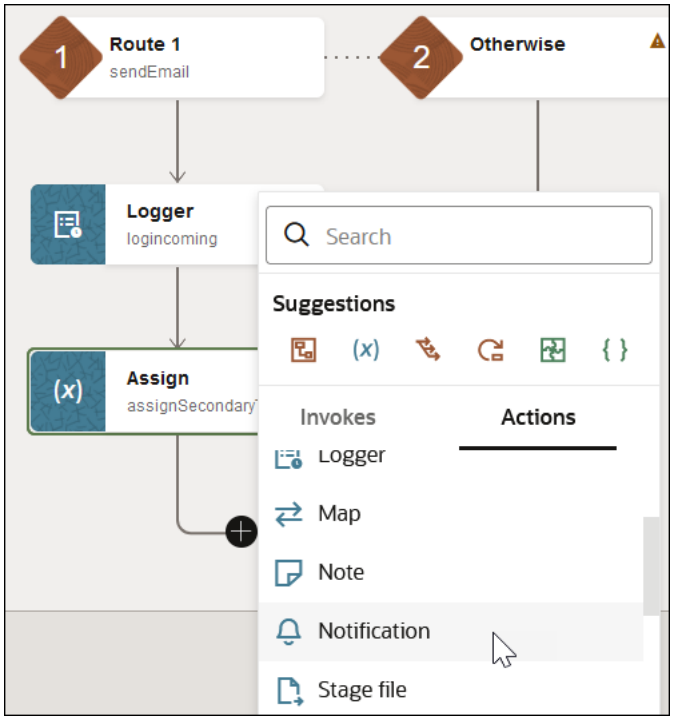


h. Click **Save**, then click outside the Configure assign panel to close it.

Now let's design the contents of the notification email you receive after running the integration.

11. Drag your cursor along the line below **Assign** until the **Add** icon appears.

12. Click **Add**, then select **Actions**, then **Notification**.



The Configure notification panel opens.

a. Click **Edit**.

Copyright © 2022, 2026, Oracle and/or its affiliates.

b. Enter `sendEmail` as the name for the notification action, then click **Apply** ✓.

c. Click the **To** field.
The **Sources** panel opens.

d. Click **Switch to Developer View** .

e. Expand **QueryParameters**, and drag **email** into the **To** field.

f. Click the **From** field.

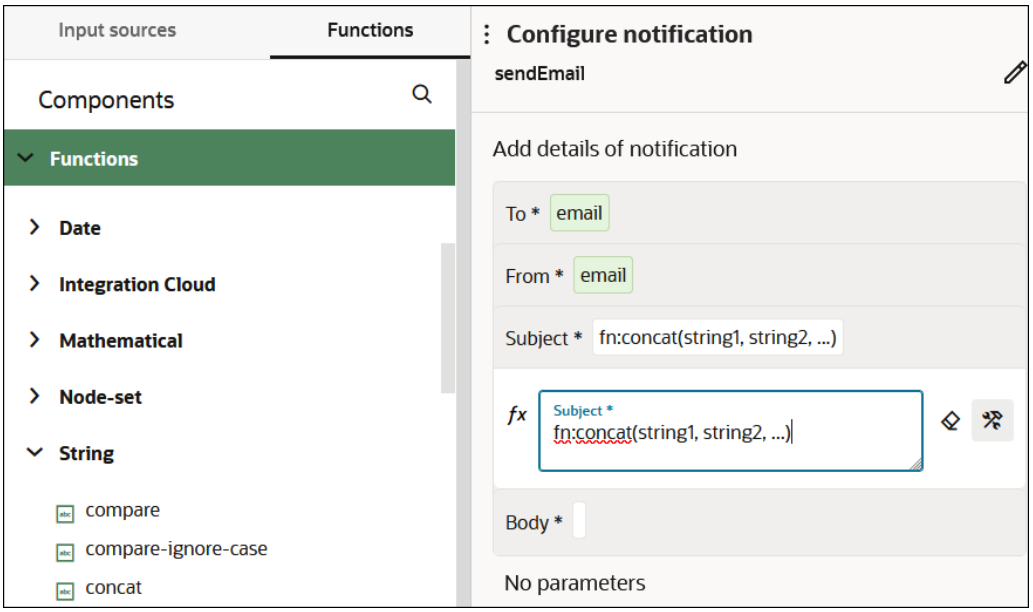
g. Expand **QueryParameters**, and drag **email** into the **From** field.

h. Click **Switch to Developer View** again.

i. Click the **Subject** field.

j. Click the **Functions** tab, then expand **String**.

k. Drag **concat** into the **Subject** field.



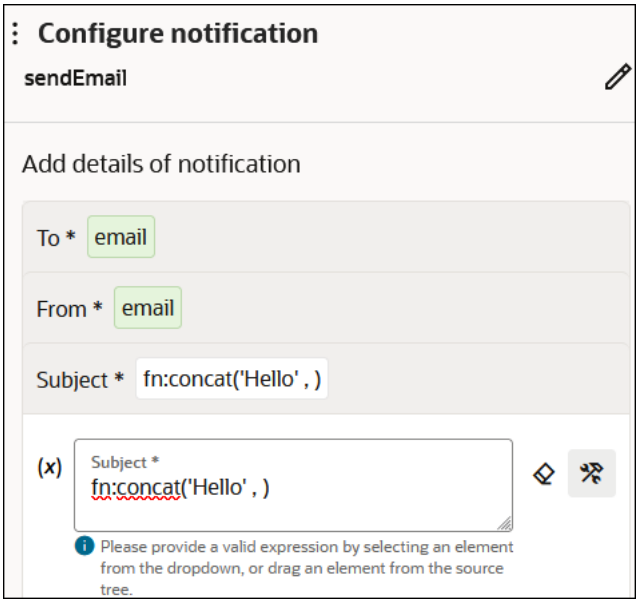
l. Remove the content between the parentheses.

m. After the opening parenthesis, enter the following:

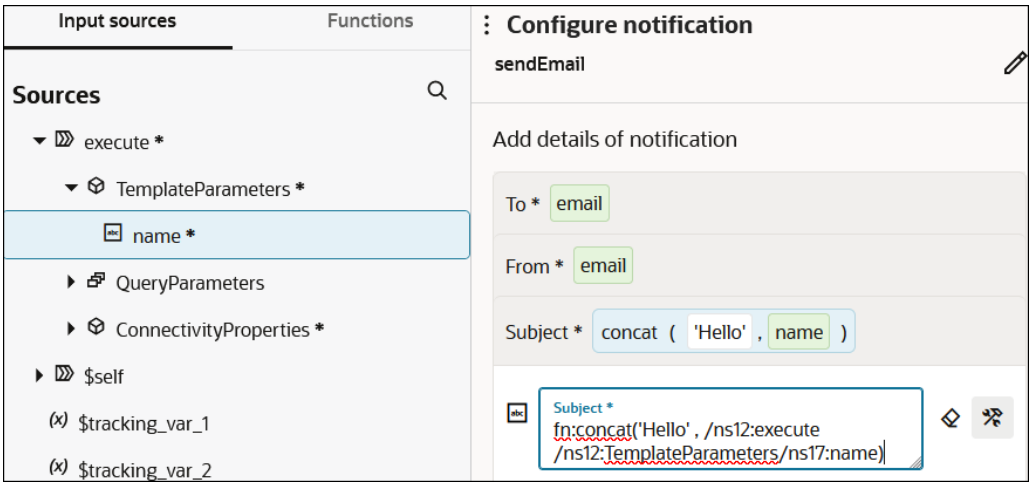
```
'Hello' ,
```

Note the blank space after the comma.

Copyright © 2022, 2026, Oracle and/or its affiliates.



n. Click the **Input sources** tab, expand **TemplateParameters**, and drag **name** into the **concat** function before the closing parenthesis.



o. Click the **Body** field and enter the following text. This is the email message you receive when the integration runs successfully.

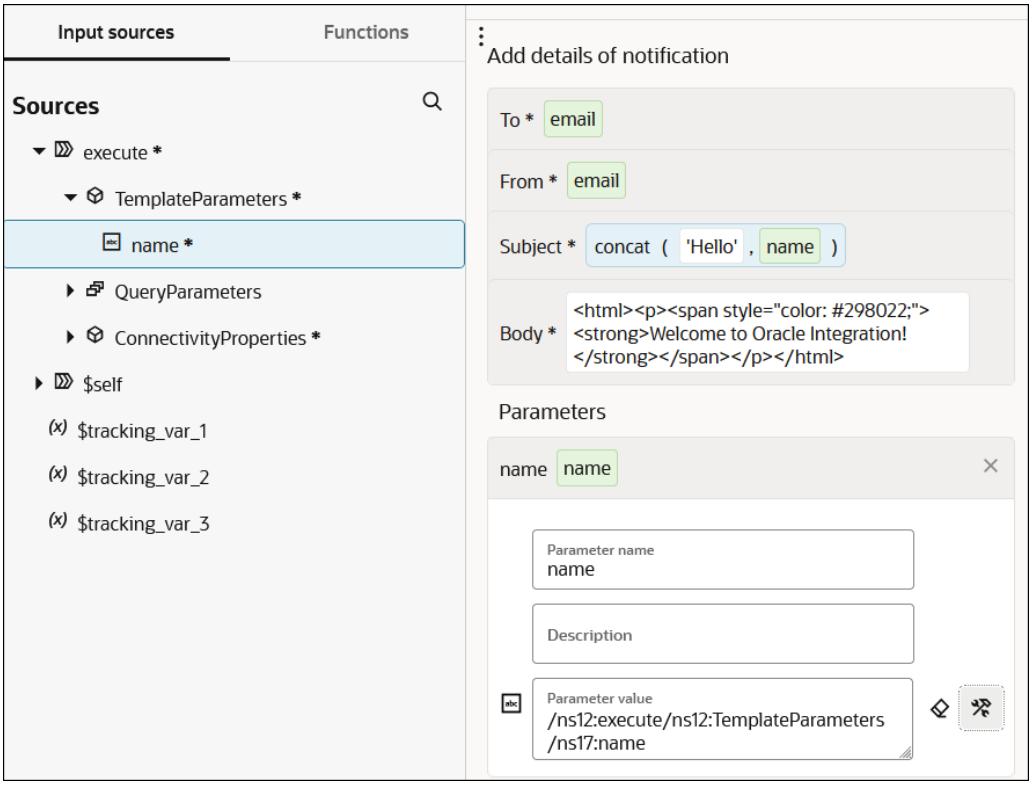
```
<html><p><span style="color: #298022;"><strong>Welcome to Oracle
Integration!</strong></span></p></html>
```

p. Click + in the **No parameters** section at the bottom.

q. In the **Parameter** field, enter `name`.

r. Click the **Input sources** tab, expand **TemplateParameters**, and drag **name** into the **Parameter value** field.

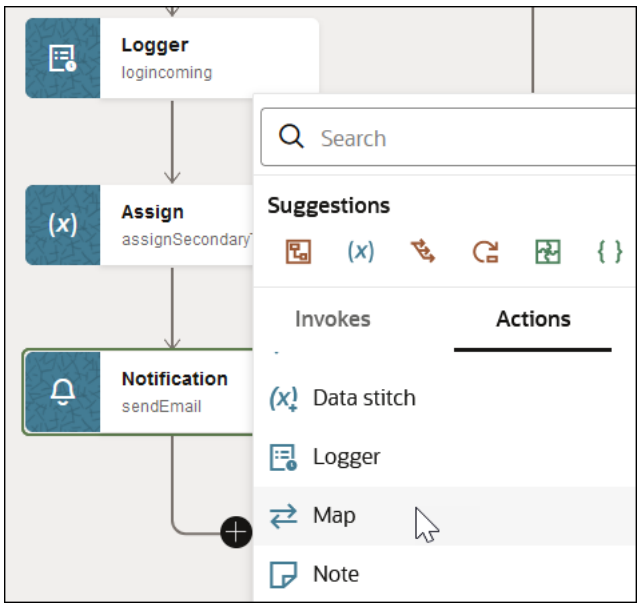
Copyright © 2022, 2026, Oracle and/or its affiliates.



s. Click **Save**, then click outside the Configure notification panel to close it.

13. Drag your cursor along the line below **Notification** until the **Add** icon appears.

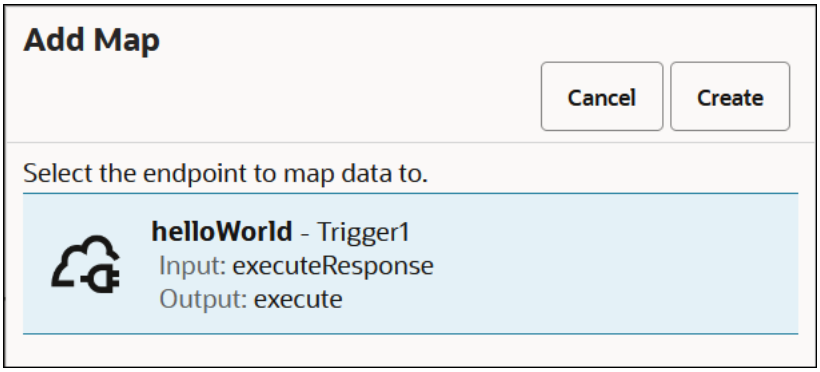
14. Click **Add**, then select **Actions**, then **Map**.



The Add Map panel opens.

a. Select the **helloWorld** endpoint to which to map data, and click **Create**.

Copyright © 2022, 2026, Oracle and/or its affiliates.

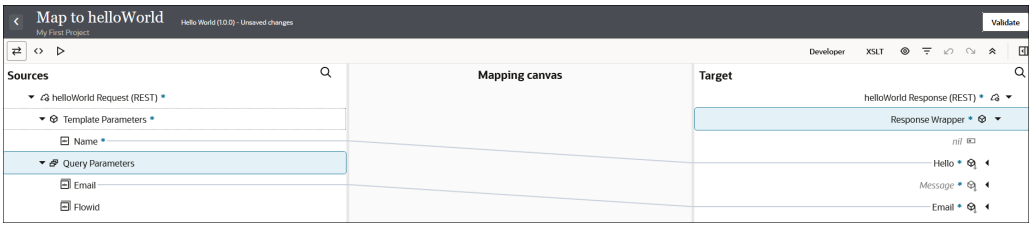


The mapper opens.

b. Expand **Template Parameters** and **Query Parameters** in the **Sources** section and **Response Wrapper** in the **Target** section.

c. Perform the following mappings:

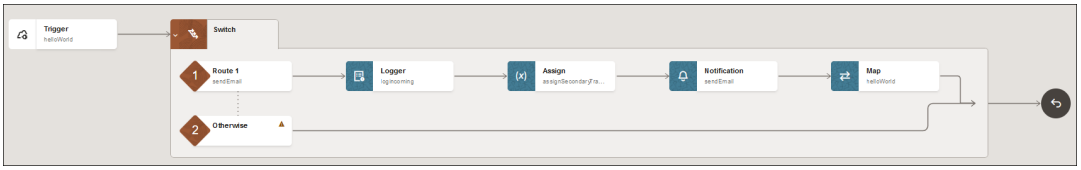
- Drag the source **Name** element to the target **Hello** element.
- Drag the source **Email** element to the target **Email** element.



d. Click **Validate**.

e. Click **Back** to exit the mapper and return to the integration canvas. Your mappings are automatically saved.

15. Above the integration, click **Horizontal** to see a horizontal view of the integration. You have now completed configuration of the first route of the switch action.



Design the Otherwise path of the integration

Now let's design the second (Otherwise) path of the integration.

1. Drag your cursor along the line to the right of **Otherwise** until the **Add** icon appears.
2. Click **Add**, then select **Actions**, then **Logger**.

Copyright © 2022, 2026, Oracle and/or its affiliates.



a. Click **Edit** .

b. Enter `logincomingWithoutEmail` as the name for the logger action, then click **Apply** .

c. In the **Logger message** field, click **Switch to Developer Mode** .

d. Click the **Functions** tab, then expand **String**.

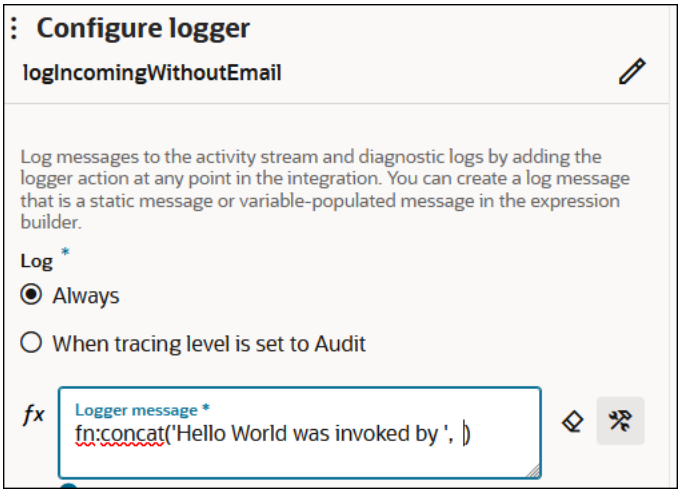
e. Drag a **concat** function into the **Logger message** field.

f. Remove the content between the parentheses.

g. After the opening parenthesis, enter the following:

```
'Hello World was invoked by ',
```

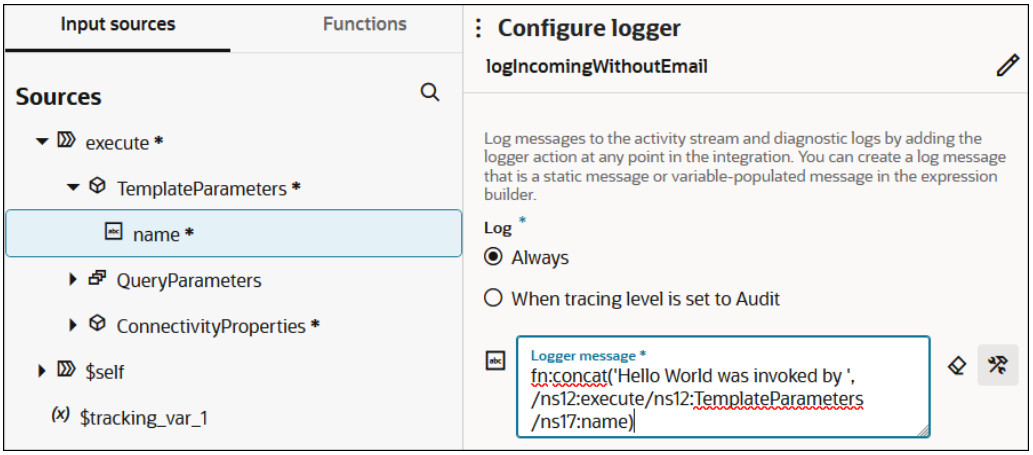
Note the single blank space after the comma.



h. Click the **Input sources** tab, and expand **TemplateParameters**.

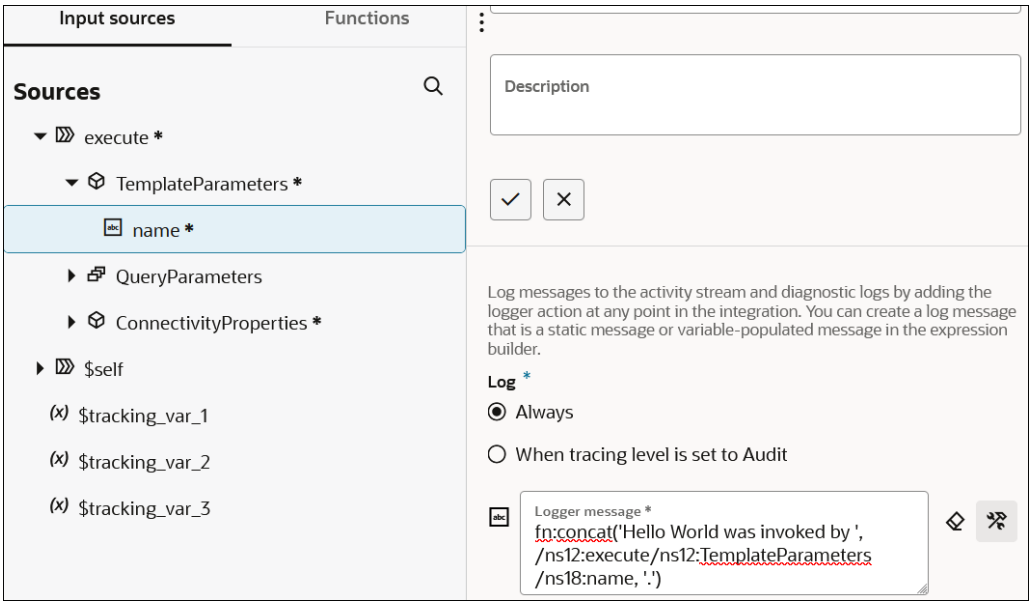
i. Drag **name** before the closing parenthesis.

Copyright © 2022, 2026, Oracle and/or its affiliates.



j. Before the closing parenthesis, enter the following:

```
, '.'
```

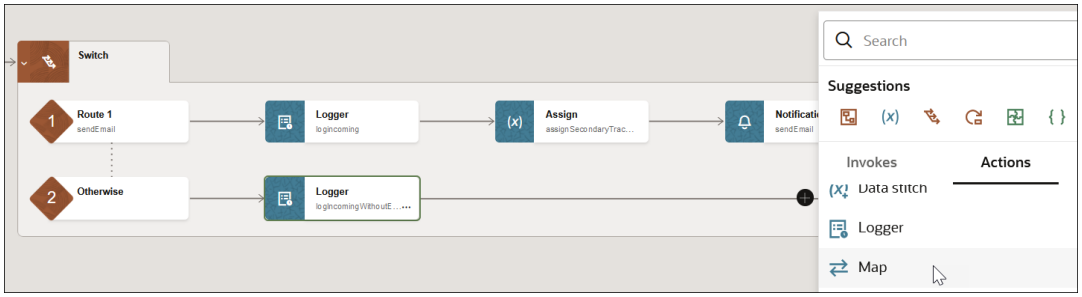


k. Click **Save**, then click outside the Configure logger panel to close it.

3. Drag your cursor along the line to the right of the **Logger** until the **Add** icon appears.

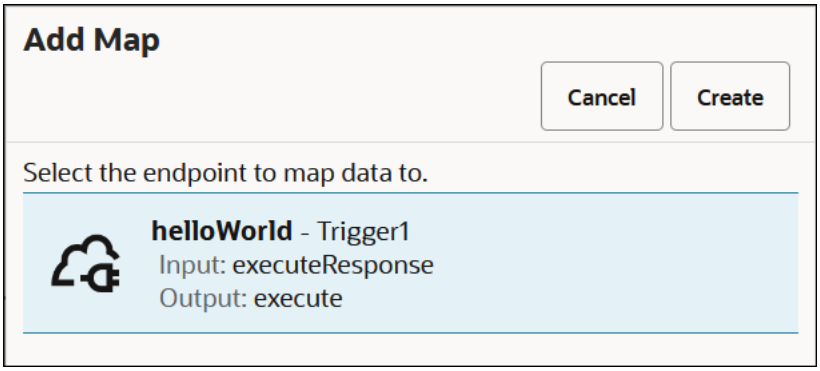
4. Click **Add**, then select **Actions**, then **Map**.

Copyright © 2022, 2026, Oracle and/or its affiliates.



The Add Map panel opens.

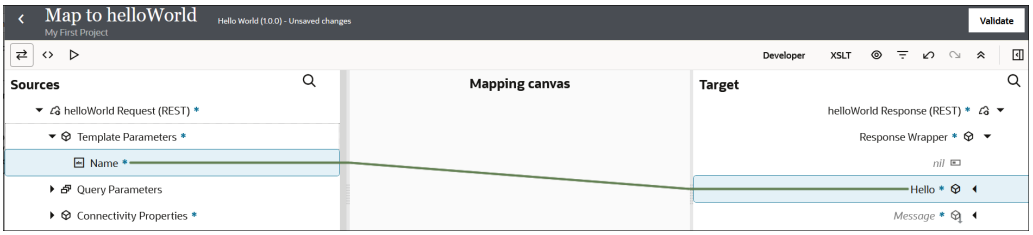
a. Select the **helloWorld** endpoint to which to map data and click **Create**.



The mapper opens.

b. Expand **Template Parameters** in the **Sources** section and **Response Wrapper** in the **Target** section.

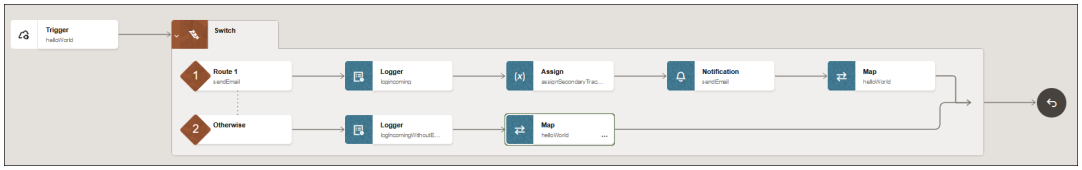
c. Drag the source **Name** element to the target **Hello** element.



d. Click **Validate**.

e. Click **Back** to exit the mapper and return to the integration canvas. Your mappings are automatically saved.

Both routes are now configured.

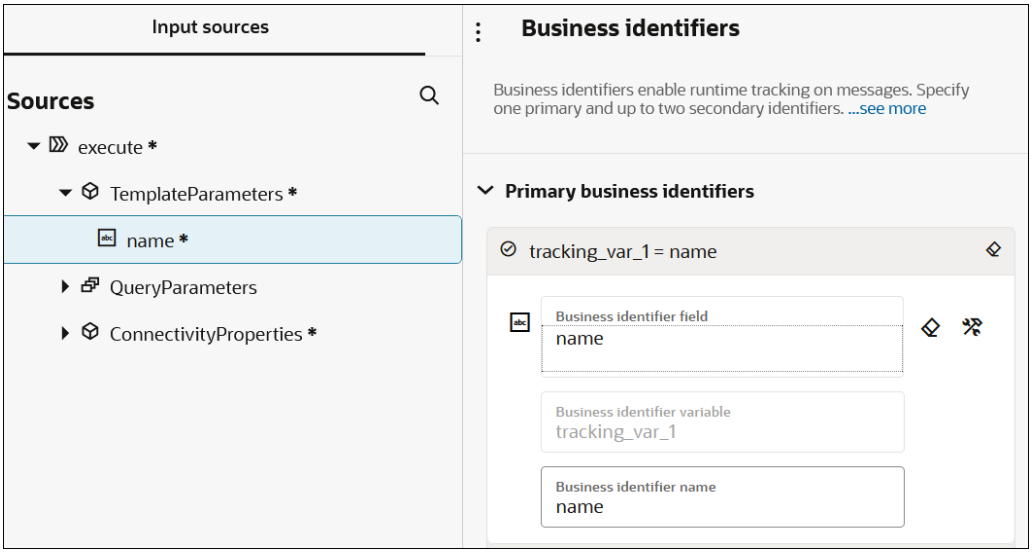


Copyright © 2022, 2026, Oracle and/or its affiliates.

Step 6. Define a Business Identifier to Track Message Delivery at Runtime

Business identifiers enable you to track payload fields in messages during runtime. Each integration requires a primary business identifier. Without specifying a business identifier, you cannot complete design integration or run your integration.

1. Above the integration, click **Business Identifier** (I).
   a. Expand **TemplateParameters**, and drag **name** to the **Business identifier** field.

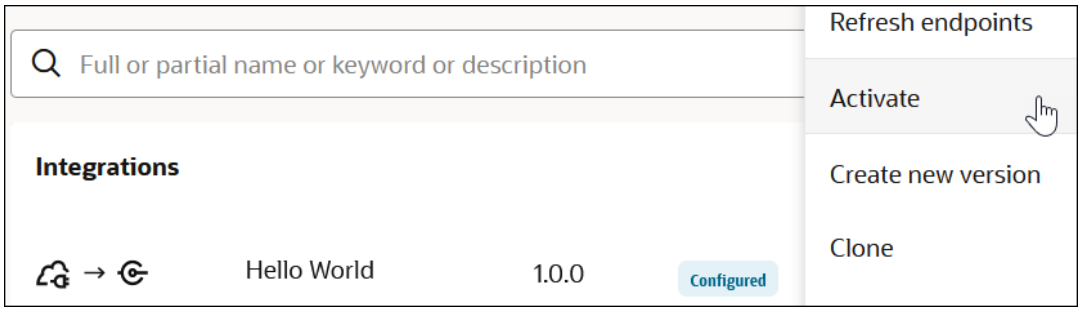


   b. Click **Save**, then click outside the Business identifiers panel to close it.
2. Check for any red error icons. If you see any, click the icon for details and resolve those errors.
3. Click **Back** < to exit the integration canvas and return to the project page.

Step 7. Activate and Run the Integration

You are now ready to activate the integration to the runtime environment. Once activated, you can run the integration.

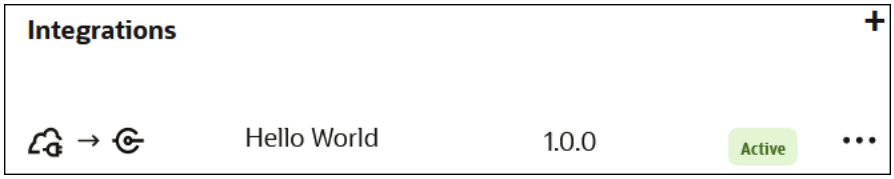
1. In the **Integrations** section, click **Actions** ⋮, then select **Activate**.



2. Keep the tracing level set to **Production**.
3. Click **Activate**.

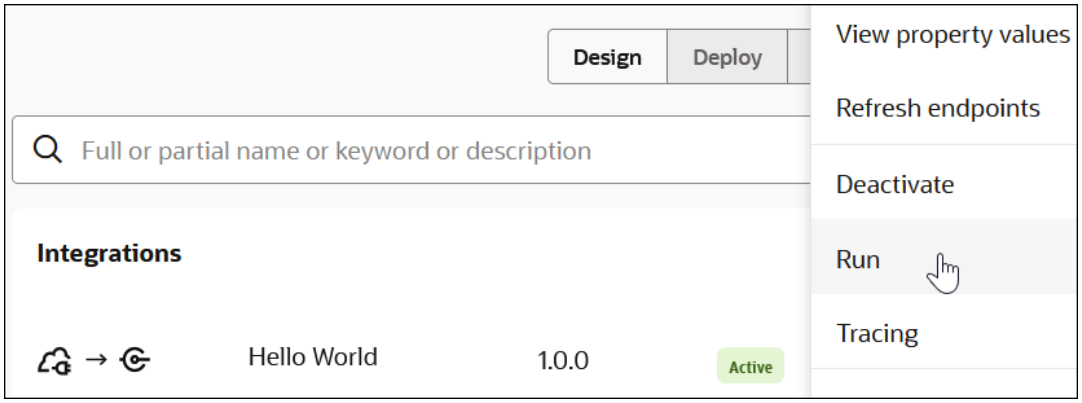
Copyright © 2022, 2026, Oracle and/or its affiliates.

4. Click **Refresh** periodically.
The integration status changes to **Active**.



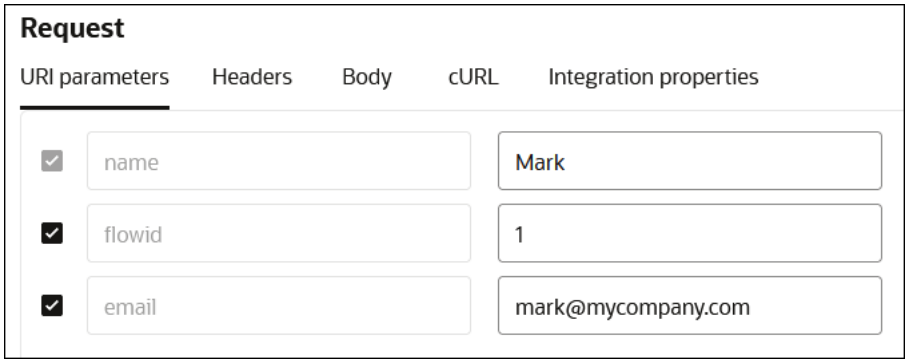
You are now ready to run the integration and send a message.

5. In the **Integrations** section, click **Actions**, then select **Run**.



The Configure and run page enables you to run and test REST Adapter trigger-based integrations. This page provides a way of testing the sending of an inbound message from an actual application.

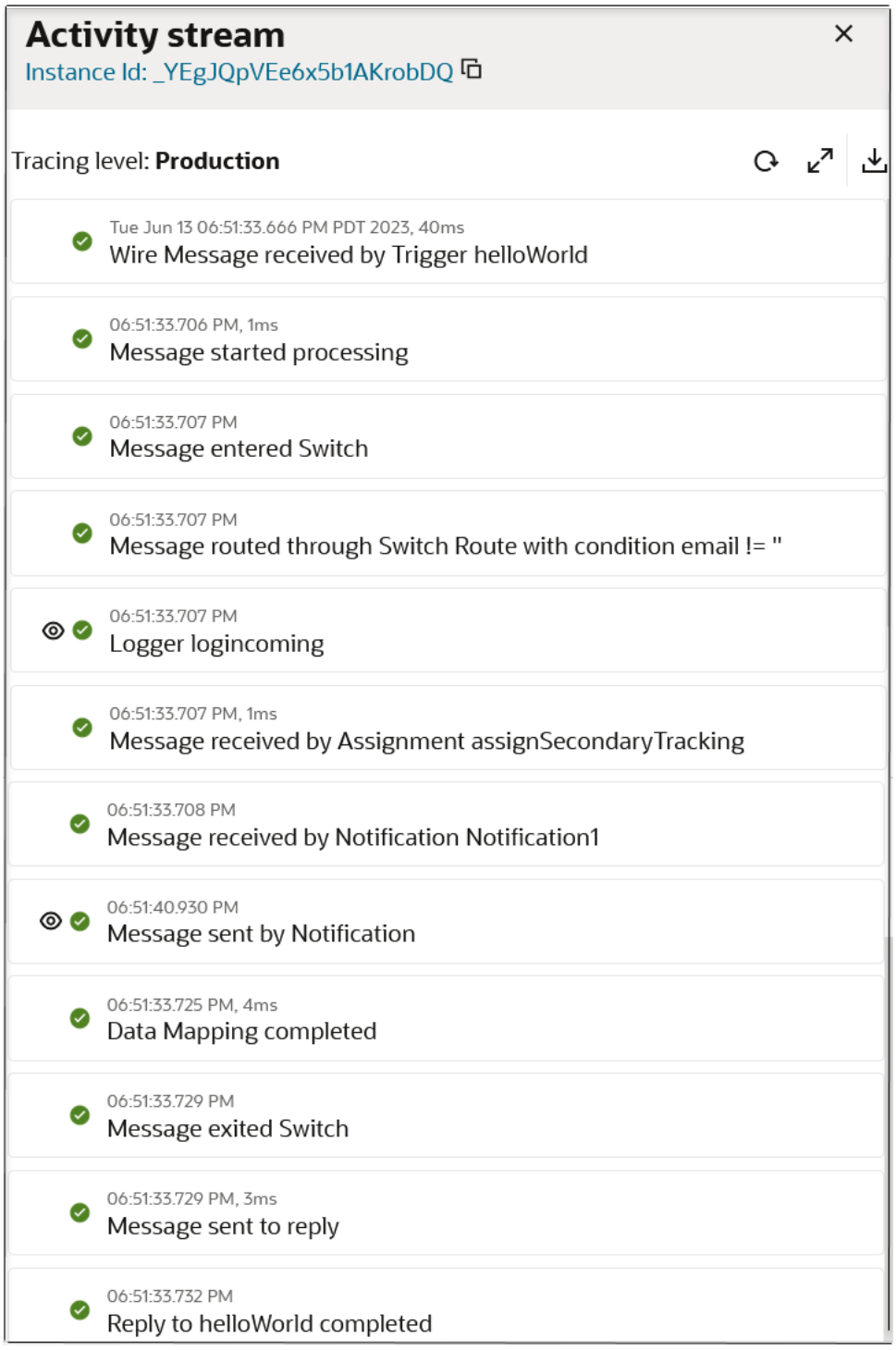
6. In the **Request** section, specify your name, a flow ID number, and your email address.



7. Click **Run** to send your inbound message to the integration.

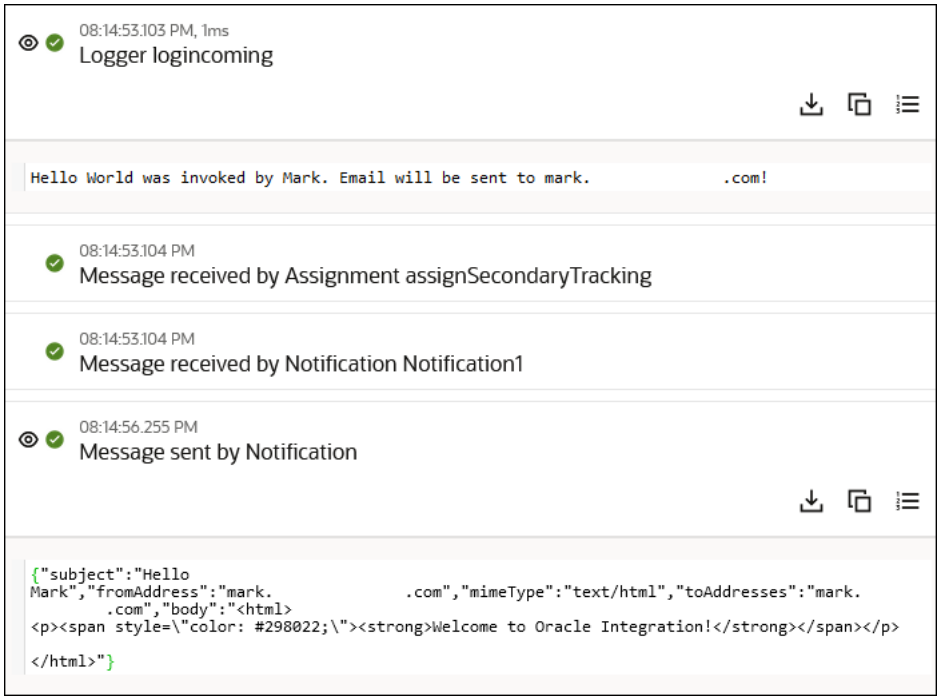
8. Click **Refresh** periodically.
The Activity stream panel opens and shows the movement of the message through the actions that you designed (switch, logger, mapper, and others). The unique instance ID that was created for this integration run is displayed at the top. All milestones in this activity stream are green, indicating the successful delivery of this message.

Copyright © 2022, 2026, Oracle and/or its affiliates.



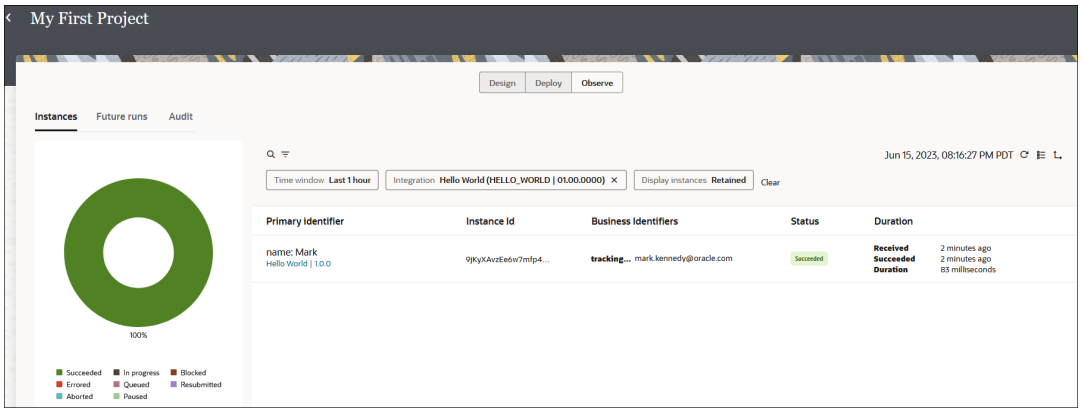
9. Click **View** next to the logger and notification actions to view the message payload that passed through the integration.

Copyright © 2022, 2026, Oracle and/or its affiliates.

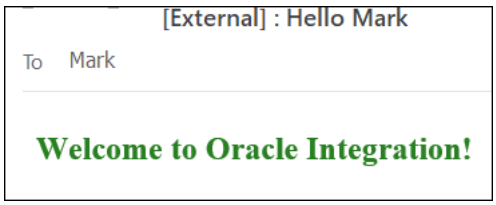


Step 8. Monitor Message Delivery Status at Runtime

1. Close the Activity steam panel and click **Track instances**.
   The **Observe** tab in your project opens. This tab enables you to monitor the status of your integration instance. The status of the instance is displayed as **Succeeded**.



2. Check your email inbox for a new message.
3. Open the email and note the following message:



Copyright © 2022, 2026, Oracle and/or its affiliates.

Congratulations! You have successfully created, designed, activated, run, and monitored your first integration. See the sections below for more information about the capabilities used in this tutorial.

Learn More About The Features in this Tutorial

- About Projects
- About Integrations
- About Connections
- Add Actions to an Integration
- Assign Business Identifiers
- Activate an Integration
- Test Integrations from Outside the Integration Canvas
- Monitor Integrations in a Project

Tutorial: Build Your Second Integration From Scratch

Let's walk through how to create, design, activate, run, and monitor a second integration completely from scratch. For this tutorial, you create an integration that invokes a weather application to return a weather report for a city that you specify.

- What Do I Learn with this Tutorial
- Step 1. Complete Prerequisites
- (Optional) Step 2. Test the API Key Outside of Oracle Integration
- Step 3. Create a Project
- Step 4. Create an Integration
- Step 5. Create a Trigger Connection to Define a City Parameter and the Content of the Weather Report Response
- Step 6. Create an Invoke Connection to Call the OpenWeather Application
- Step 7. Specify How the Invoke Connection Retrieves the Latitude and Longitude Coordinates for the City
- Step 8. Specify How the Invoke Connection Retrieves the Weather Report for the City
- Step 9. Map the Weather Report Response Returned from the OpenWeather Application
- Step 10. Define a Business Identifier
- Step 11. Activate and Run the Integration
- Step 12. Monitor the Integration
- Learn More About The Features in this Tutorial

What Do I Learn with this Tutorial

This tutorial walks you through performing the following tasks:

- Creating a project in which to perform all design, management, and monitoring tasks.
- Creating an integration in which to design your business requirements.

Copyright © 2022, 2026, Oracle and/or its affiliates.

- Creating and configuring a trigger connection in which you define a city parameter and the contents of the weather report to receive back from the OpenWeather application (city, latitude, longitude, and other content).
- Creating and configuring an invoke connection to obtain the latitude and longitude coordinates for the specified city from the OpenWeather application.
- Creating and configuring a second invoke connection to obtain the weather report from the OpenWeather application.
- Specifying a city to start the integration and receiving the weather report back for that city from the OpenWeather application.
- Monitoring the results of the integration run.

Step 1. Complete Prerequisites

When you configure the REST Adapter as an invoke connection to use in this integration, you select the API Key Based Authentication security policy. This policy requires you to specify an API key. Create an account at the OpenWeather application to obtain an API key.

1. Go to the OpenWeather Application.
2. Create an account to use the application.
3. Sign in by specifying your email address and password.
4. Click **API keys** or click your profile link in the upper right corner.
5. Copy and save your API key value.

(Optional) Step 2. Test the API Key Outside of Oracle Integration

When you test REST Adapter configuration later in this tutorial, a message appears indicating that the API key is not validated as part of the test. You must validate the API key separately. While there may be nothing to indicate that your API key value is invalid, it is typically a best practice to test your connection outside of Oracle Integration to completely validate the API key.

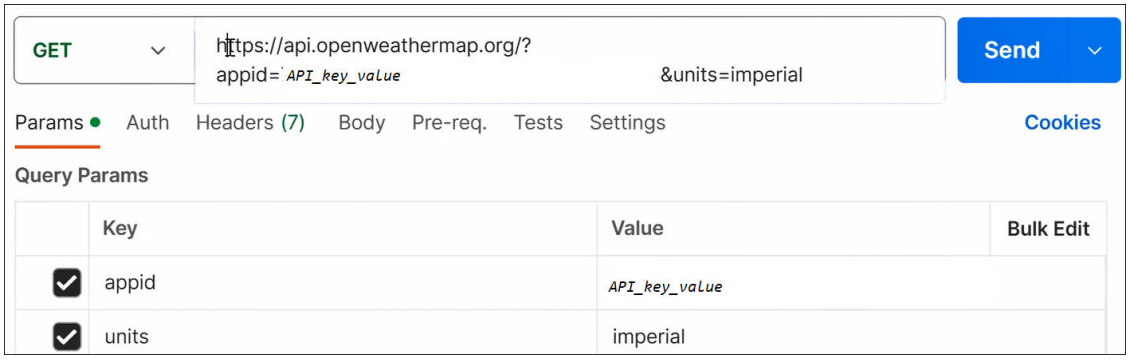
You can use a third-party tool such as Postman to test your API key connection outside of Oracle Integration. This test provides a way to completely validate the API key. For example:

```
https://api.openweathermap.org/?appid=API_key_value&units=imperial
```

Where *API_key_value* is the key you obtained from the OpenWeather application.

For example, from a third-party tool such as Postman, you specify and test the URL and query parameter values for `appid` and `units`. Because this tutorial returns the temperature in degrees Fahrenheit, `imperial` is specified. You can also specify `metric` to return the temperature in degrees Celsius.

Copyright © 2022, 2026, Oracle and/or its affiliates.



Step 3. Create a Project

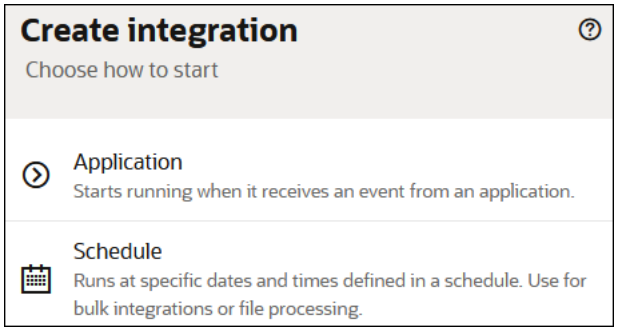
You manage the entire life cycle of an integration from within a project.

1. In the navigation pane, click **Projects**.
2. Click **Add**, then select **Create**.
3. In the **Name** field, enter `My Weather Project`.
4. Leave all remaining fields as they are, then click **Create**.
   The page for your project is displayed. Your project contains sections for creating, designing, deploying, and monitoring your integration resources (integrations, connections, lookups, and JavaScript libraries).

Step 4. Create an Integration

Let's now create an integration inside the project.

1. In the **Integrations** section, click **Add**.
   The Add integration panel opens.
2. Click **Create**, then select **Application**.



3. In the **Name** field, enter `Weather By City`.
4. Leave all remaining fields as they are, then click **Create**.
   The integration canvas opens with an option for creating a trigger connection. The trigger connection enables your integration to receive inbound messages from an application.

Copyright © 2022, 2026, Oracle and/or its affiliates.

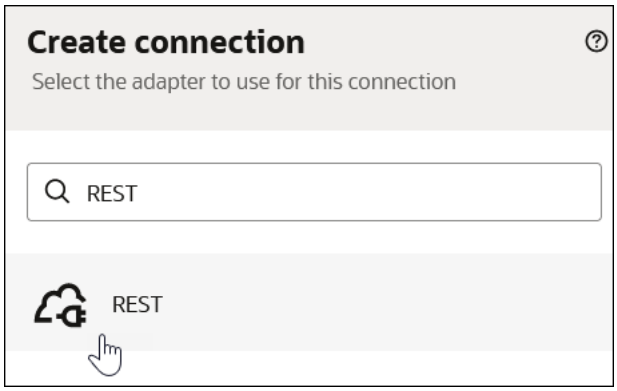


> **Note**
>
> As you design this integration, an error icon can appear on the right side of the page. Errors may occur due to mistakes you make in the mapper or Adapter Endpoint Configuration Wizard. Other errors that appear are resolved as you complete the design of all parts of this integration. Click the error icon to view details. To resolve the error, you can either follow the instructions in the Error panel or wait to see if your completed design resolves the issues. Click outside the Error panel to close it.
>
> 

Step 5. Create a Trigger Connection to Define a City Parameter and the Content of the Weather Report Response

Let's now create a trigger connection inside the integration canvas. The trigger (source) connection allows you to define a city parameter and the content of the weather report to receive back from the OpenWeather application (city, latitude, longitude, and other content).

1. In the Connections panel of the integration canvas, click **Add trigger**.
   The Create connection panel opens.
2. Enter REST in the search field.
3. Select **REST** as the adapter connection to use. For this tutorial, the REST Adapter processes inbound messages sent to Oracle Integration.



4. The page for configuring the REST Adapter opens.

Copyright © 2022, 2026, Oracle and/or its affiliates.

a. Enter the following information.

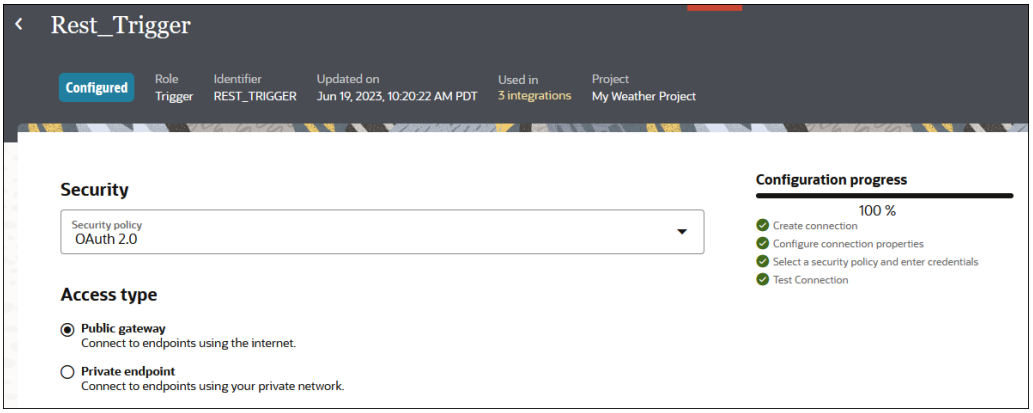
Field	Description
Name	Enter `Rest_Trigger`.
Role	Select **Trigger**. Roles perform the following tasks. • **Trigger**: Enables an inbound application (endpoint) to trigger (run) the integration. • **Invoke**: Enables your integration to invoke an outbound application (endpoint). • **Trigger and Invoke**: Enables you to create a connection that can be used as both a trigger and invoke. For example, you use the connection as a trigger in one integration and as an invoke in another integration.

b. Leave all remaining fields as they are, then click **Create**.
The page for configuring security and delivery methods for the REST Adapter trigger connection appears.

c. Enter the following information:

Field	Description
Security	Select **OAuth 2.0** to secure incoming messages.
Access type	Ensure **Public gateway** is selected. This access type uses the public internet to receive messages.

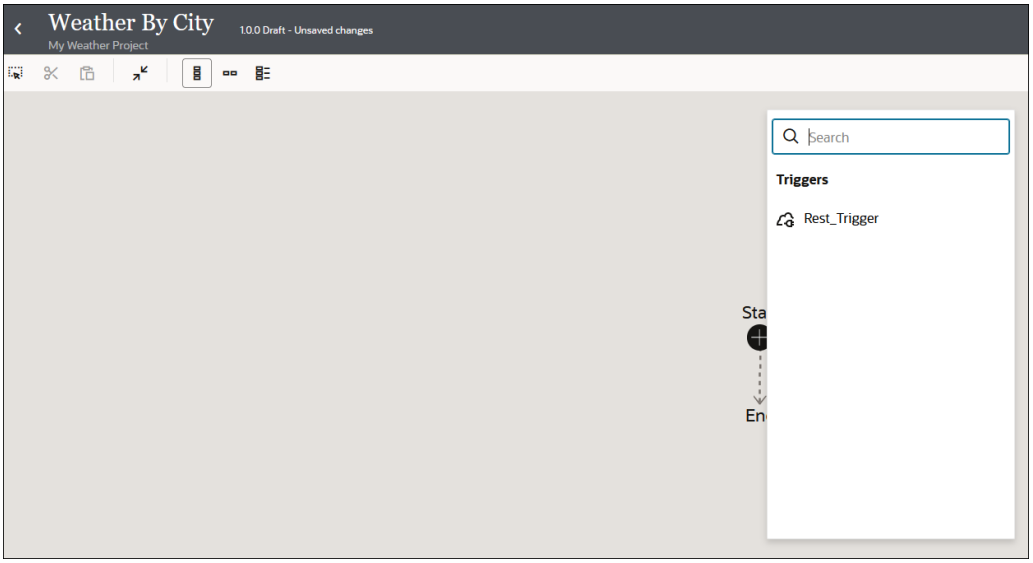
d. Click **Test** to confirm that you configured the connection successfully. When testing completes, a success message is displayed. Configuration progress is now 100% complete.



e. Click **Save**.

f. Click **Back** to return to the integration canvas. A menu shows that the newly-created **Rest_Trigger** connection is available for selection.

Copyright © 2022, 2026, Oracle and/or its affiliates.



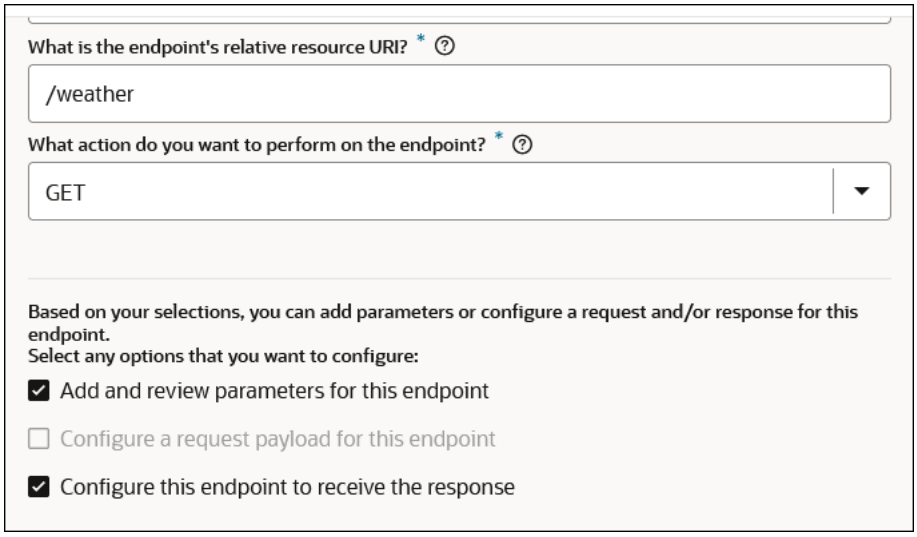
Let's now configure the REST Adapter trigger connection in the integration. During configuration, you specify details such as the operation, the relative resource URI, the query parameters, and the message payload content.

5. In the integration canvas, click the **Rest_Trigger** connection shown in the menu.
The Basic Info page of the Adapter Endpoint Configuration Wizard opens. This wizard enables you to configure the **Rest_Trigger** connection for use in the integration.
6. In the **What do you want to call your endpoint** field, enter `GetWeatherByCity`, then click **Continue** to access the Resource Configuration page.
7. Enter the following information.

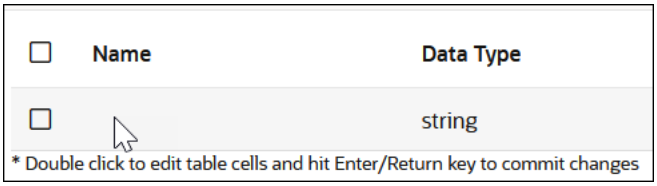
Field	Description
What is the endpoint's relative resource URI	Enter the following: `/weather`
What action do you want to perform on the endpoint	Select **GET**.
Select any options that you want to configure	Scroll to the bottom of the page and select the following options: • **Add and review parameters for this endpoint** • **Configure this endpoint to receive the response**

8. Leave all remaining fields as they are.
The completed Resource Configuration page looks as follows.

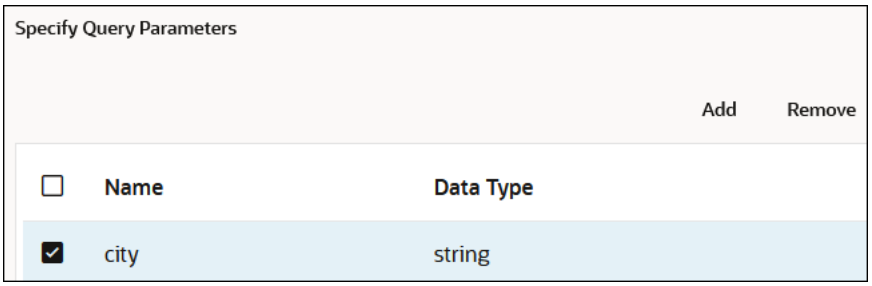
Copyright © 2022, 2026, Oracle and/or its affiliates.



9. Click **Continue** to access the Request Parameters page.
   a. In the **Specify Query Parameters** section, click **Add**.
   b. Double-click the row in the **Name** column to enter a value.



   c. Enter `city`, then select **string** from the **Data Type** list.



   At runtime, you specify the city on the Configure and run page to receive a weather report.
   d. Click outside the **city** row.
   e. Click **Continue** to access the Response page.
10. Enter the following information.

Field	Description
Select the response payload format	Select **JSON Sample**.

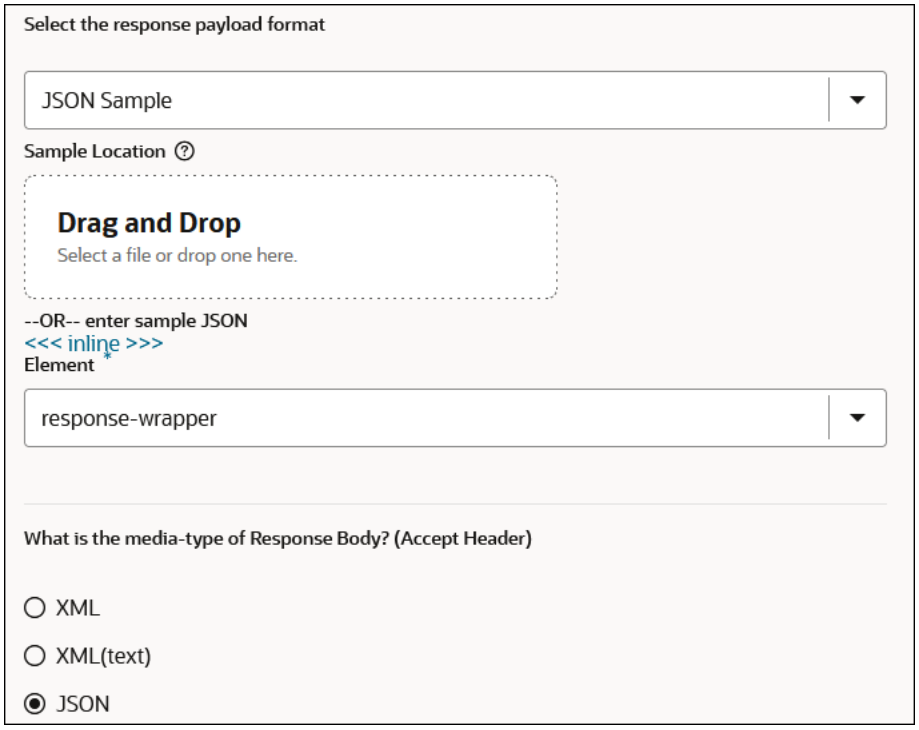
Copyright © 2022, 2026, Oracle and/or its affiliates.

Field	Description

<<< inline >>>	Click this field and enter the following content, then click **OK**.
	
	`{`
	`  "city" : "Timbuktu",`
	`  "longitude" : 1.1,`
	`  "latitude" : 1.2,`
	`  "country" : "AF",`
	`  "description" : "clear sky",`
	`  "temp" : 20.6`
	`}`

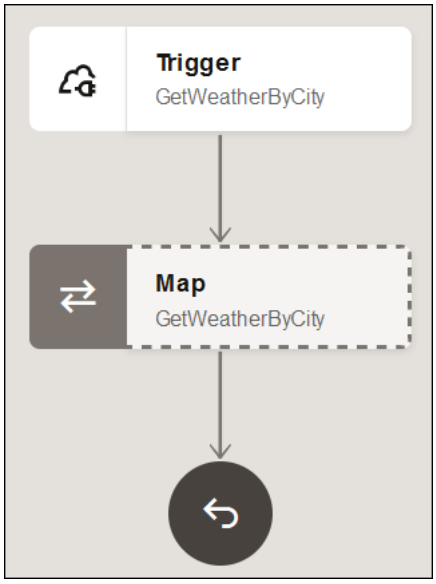
Element	Ensure **response-wrapper** is selected.
What is the media-type of Response Body	Ensure **JSON** is selected.

The completed Response page looks as follows.



11. Click **Continue** to access the Summary page.
12. Review your selections, then click **Finish**.
    The REST Adapter trigger connection is now fully configured for use in the integration. A map action is also automatically created for mapping data between the trigger connection and the invoke connection that you create in the next section.

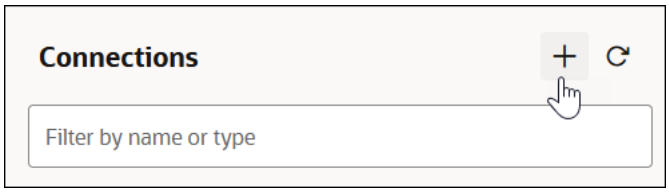
Copyright © 2022, 2026, Oracle and/or its affiliates.



Step 6. Create an Invoke Connection to Call the OpenWeather Application

Let's now create an invoke connection inside the integration canvas to call the OpenWeather application and receive weather information.

1. In the right pane, click **Invokes** .
   The Create connection panel opens.
2. Click **Create Connection** to create a new connection.



3. Select **REST** as the adapter connection to use. For this tutorial, the REST Adapter invokes the OpenWeather application.
   a. Enter the following information.

Field	Description
Name	Enter `Weather`.
Role	Select **Invoke**. This role enables your integration to invoke the OpenWeather application.

   b. Leave all remaining fields as they are, then click **Create**.
   The page for configuring security and delivery methods for the REST Adapter invoke connection appears.
   c. Enter the following information:

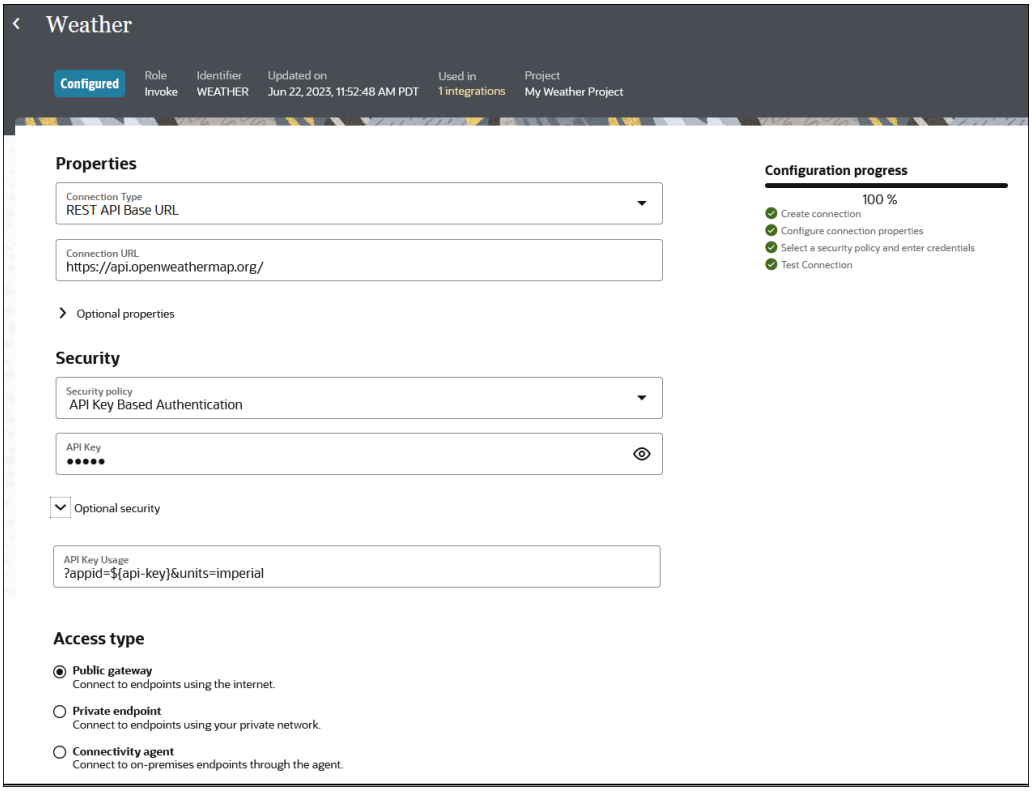
Copyright © 2022, 2026, Oracle and/or its affiliates.

Field	Description
Connection Type	Select **REST API Base URL** to secure the message.
Connection URL	Enter the following. `https://api.openweathermap.org/`
Security policy	Select **API Key Based Authentication**. This is the authentication security used by the OpenWeather application.
API Key	Enter the API key you obtained in Step 1. Complete Prerequisites.

API Key Usage	Expand **Optional security**, then enter one of the following key usages. This is how the key is passed to the OpenWeather application. The value of `imperial` indicates that we want to return the temperature in degrees Fahrenheit.
	
	`?appid=${api-key}&units=imperial`
	
	This tutorial describes how to return the temperature in degrees Fahrenheit. If you want, you can return the temperature in degrees Celsius by specifying the following value:
	
	`?appid=${api-key}&units=metric`

Access type	Ensure **Public gateway** is selected. This access type uses the public internet to send messages.

Copyright © 2022, 2026, Oracle and/or its affiliates.



d. Click **Test** to confirm that you configured the connection successfully. When testing completes, the following message is displayed.

```
Connection Weather was tested successfully.
   The API key is not validated as part of this test. You must validate
   the API key separately before using them with this security policy.
```

While the API key may be completely valid, it cannot be completely validated from within Oracle Integration. You have the option of completely validating the API key in (Optional) Step 2. Test the API Key Outside of Oracle Integration.

e. Click **Save**.

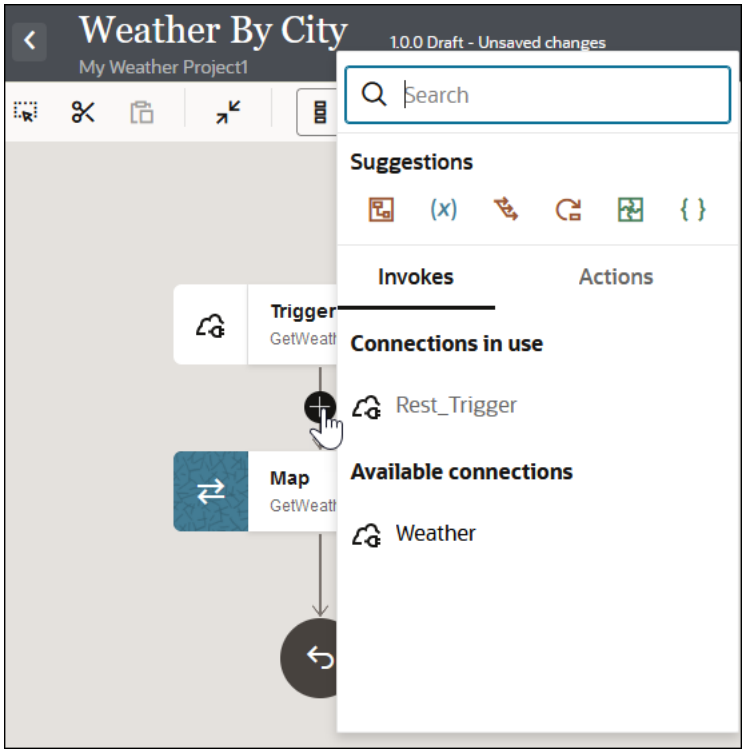
f. Click **Back** to return to the integration canvas.

Step 7. Specify How the Invoke Connection Retrieves the Latitude and Longitude Coordinates for the City

Let's now configure the REST Adapter invoke connection in the integration to retrieve the latitude and longitude coordinates for the city you specify when running the integration. During configuration, you specify details such as the operation, the relative resource URI, the query parameters, and the message payload content.

1. Drag your cursor along the line below the **GetWeatherByCity** trigger until the **Add** icon appears.
2. Select **Invokes**, then select the **Weather** connection.

Copyright © 2022, 2026, Oracle and/or its affiliates.



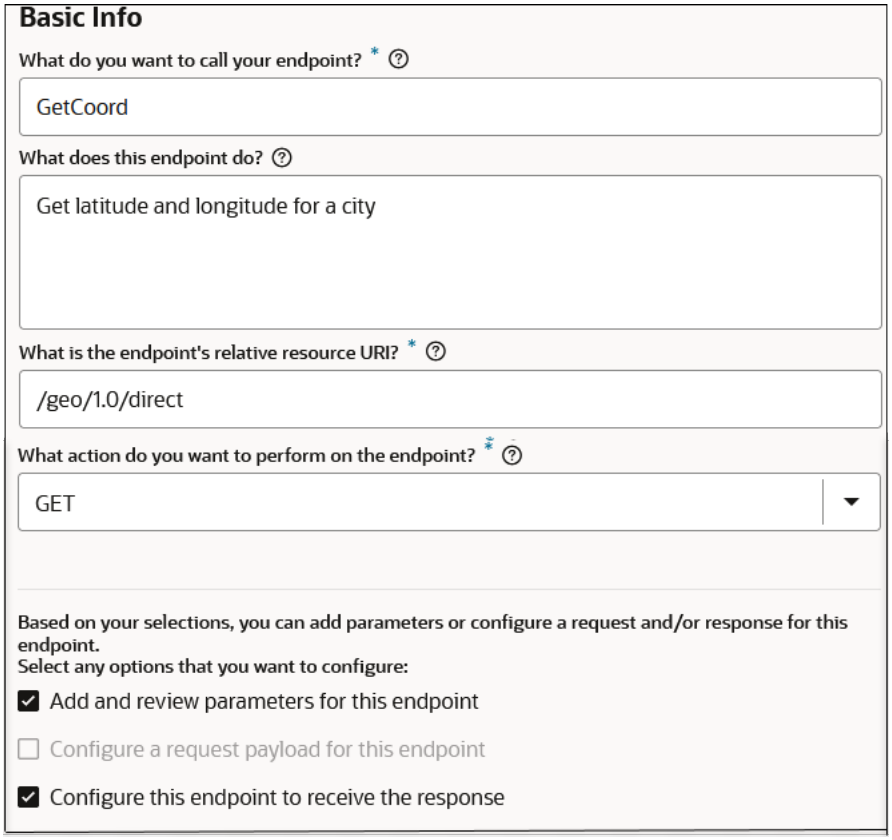
The Adapter Endpoint Configuration Wizard opens.

3. Enter the following information.

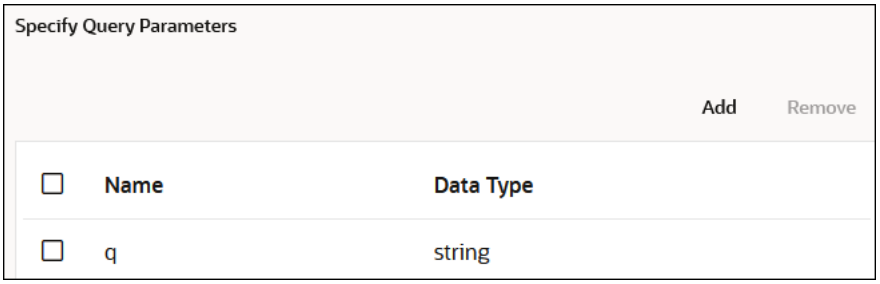
Field	Description
What do you want to call your endpoint	Enter the following: `GetCoord`
What does this endpoint do	Enter the following: `Get latitude and longitude for a city`
What is the endpoint's relative resource URI	Enter the following: `/geo/1.0/direct`
What action do you want to perform on the endpoint	Select **GET**.
Select any options that you want to configure	Scroll to the bottom of the page and select the following options: • **Add and review parameters for this endpoint** • **Configure this endpoint to receive the response**

4. Leave all remaining fields as they are.
The completed Basic Info page looks as follows.

Copyright © 2022, 2026, Oracle and/or its affiliates.



5. Click **Continue** to access the Request Parameters page.
   a. In the **Specify Query Parameters** section, click **Add**.
   b. Double-click the row in the **Name** column to enter a value.
   c. Enter q, then select **string** from the **Data Type** list.



   d. Click outside the **q** row.
   e. Click **Continue** to access the Response page.
6. Enter the following information.

Field	Description
Select the response payload format	Select **JSON Sample**.

Copyright © 2022, 2026, Oracle and/or its affiliates.

Field	Description

<<< inline >>>	Click this field and enter the following content, then click **OK**.
	
	`[ {`
	`  "name" : "London",`
	`  "local_names" : {`
	`  "af" : "Londen",`
	`  "ascii" : "London",`
	`  "az" : "London",`
	`  "ca" : "Londres",`
	`  "da" : "London",`
	`  "de" : "London",`
	`  "en" : "London",`
	`  "eu" : "Londres",`
	`  "feature_name" : "London",`
	`  "fi" : "Lontoo",`
	`  "fr" : "Londres",`
	`  "gl" : "Londres",`
	`  "hr" : "London",`
	`  "hu" : "London",`
	`  "id" : "London",`
	`  "it" : "Londra",`
	`  "la" : "Londinium",`
	`  "lt" : "Londonas",`
	`  "nl" : "Londen",`
	`  "no" : "London",`
	`  "pl" : "Londyn",`
	`  "pt" : "Londres",`
	`  "ro" : "Londra",`
	`  "sl" : "London",`
	`  "tr" : "Londra",`
	`  "zu" : "ILondon"`
	`  },`
	`  "lat" : 51.5085,`
	`  "lon" : -0.1257,`
	`  "country" : "GB"`
	`}, {`
	`  "name" : "London",`
	`  "local_names" : {`
	`  "ascii" : "London",`
	`  "de" : "London",`
	`  "en" : "London",`
	`  "feature_name" : "London",`
	`  "fi" : "London",`
	`  "fr" : "London",`
	`  "lt" : "Londonas",`
	`  "nl" : "London",`
	`  "pl" : "London",`
	`  "pt" : "London"`
	`  },`
	`  "lat" : 42.9834,`
	`  "lon" : -81.233,`
	`  "country" : "CA"`

Copyright © 2022, 2026, Oracle and/or its affiliates.

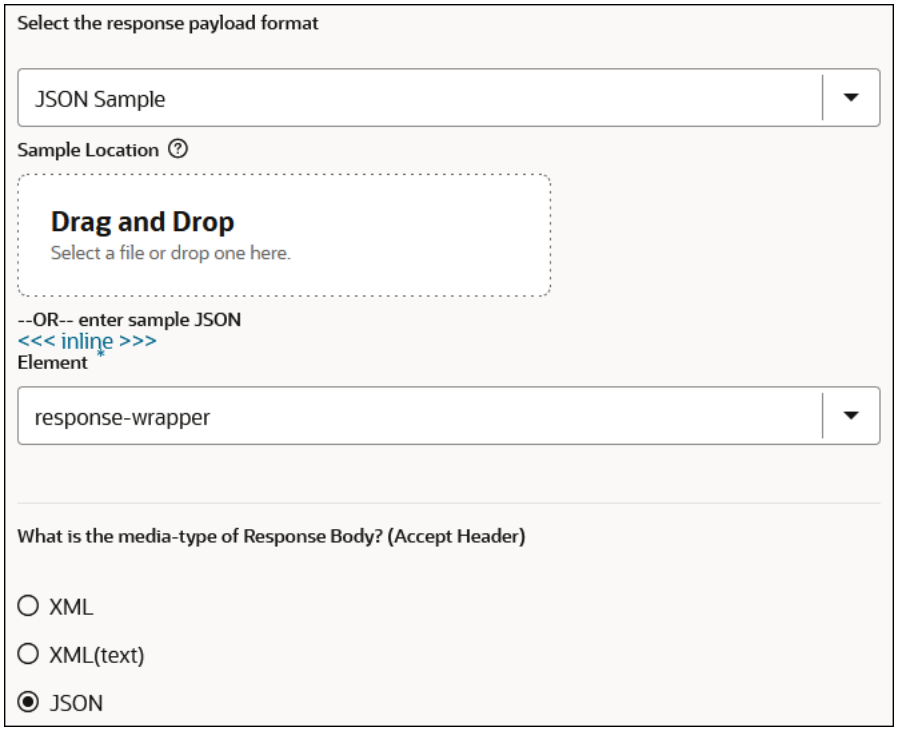
Field	Description

	<pre>}, {
	  "name" : "London",
	  "local_names" : {
	  "ascii" : "London",
	  "en" : "London",
	  "feature_name" : "London"
	  },
	  "lat" : 39.8865,
	  "lon" : -83.4483,
	  "country" : "US",
	  "state" : "OH"
	}, {
	  "name" : "London",
	  "local_names" : {
	  "ascii" : "London",
	  "en" : "London",
	  "feature_name" : "London"
	  },
	  "lat" : 37.129,
	  "lon" : -84.0833,
	  "country" : "US",
	  "state" : "KY"
	}, {
	  "name" : "London",
	  "local_names" : {
	  "ascii" : "London",
	  "ca" : "Londres",
	  "en" : "London",
	  "feature_name" : "London"
	  },
	  "lat" : 36.4761,
	  "lon" : -119.4432,
	  "country" : "US",
	  "state" : "CA"
	}]</pre>

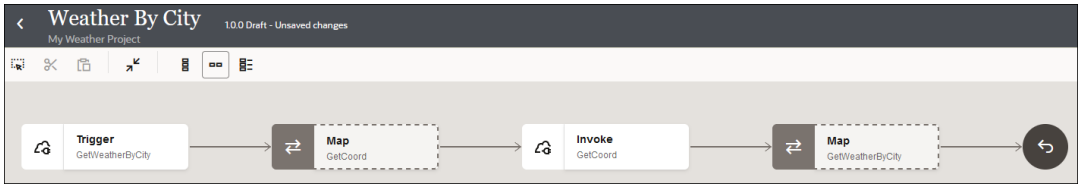
Element	Ensure **response-wrapper** is selected.
What is the media-type of Response Body	Ensure **JSON** is selected.

The completed Response page looks as follows.

Copyright © 2022, 2026, Oracle and/or its affiliates.



7. Click **Continue** to access the Summary page.
8. Review your selections, then click **Finish**.
9. Above the integration, click **Horizontal** to see a horizontal view of the integration. Note that after you created and configured the **GetCoord** invoke connection in the integration, a map was automatically added after it.



When you configured the **GetWeatherByCity** trigger connection in the integration canvas, a map was automatically created. Let's now configure the first map.

1. Double-click the map between the **GetWeatherByCity** trigger connection and **GetCoord** invoke connection.

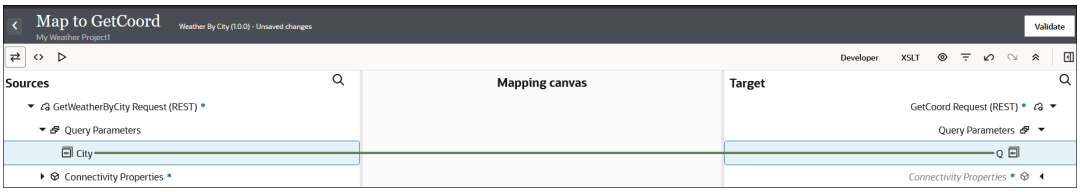


The mapper opens.

2. In the **Sources** section, expand **Query Parameters**.
3. In the **Target** section, expand **Query Parameters**.

Copyright © 2022, 2026, Oracle and/or its affiliates.

4. Drag the source **City** element to the target **Q** element. This maps the city to its latitude and longitude coordinates. You created these query parameters earlier when you configured the trigger and invoke connections in the Adapter Endpoint Configuration Wizard.



5. Click **Validate** and wait for the following message to appear.

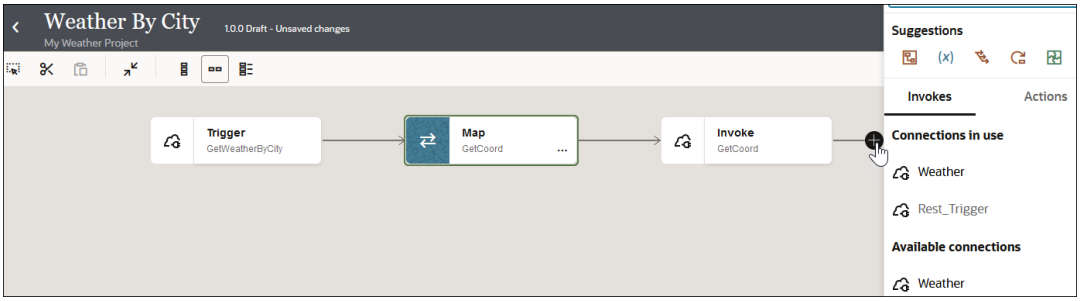
```
Map to GetCoord successfully validated.
```

6. Click **Back** to exit the mapper and return to the integration canvas. Your mappings are automatically saved.

Step 8. Specify How the Invoke Connection Retrieves the Weather Report for the City

Let's now configure a second REST Adapter invoke connection in the integration to obtain the weather report from the OpenWeather application.

1. Drag your cursor along the line after the **GetCoord** invoke connection until the **Add** icon appears.



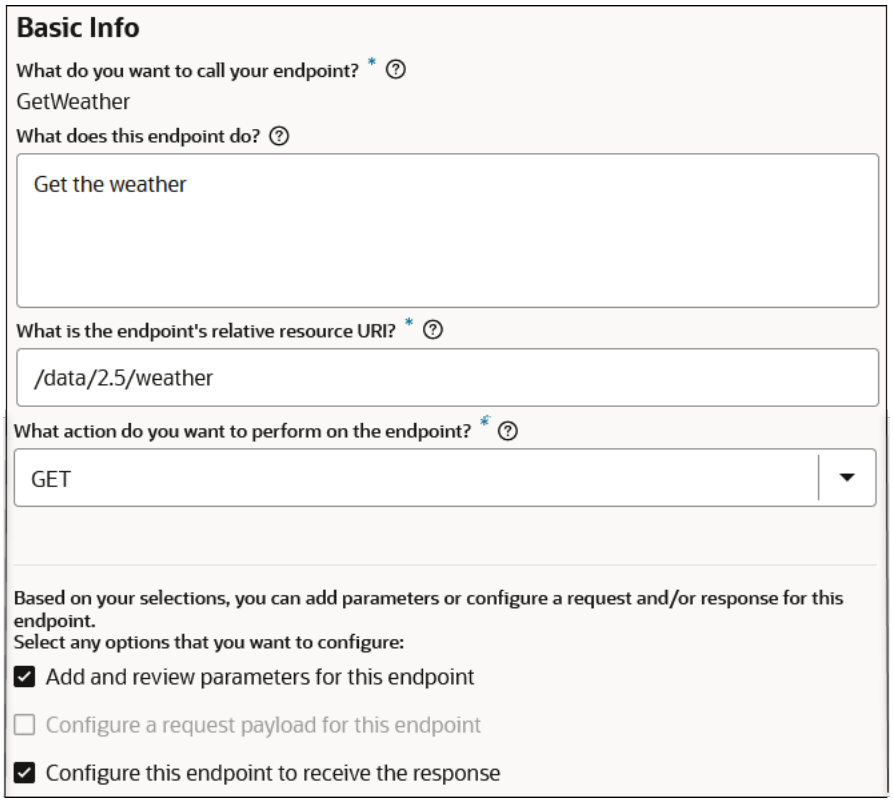
2. Select **Invokes**, then select the **Weather** connection again.
The Adapter Endpoint Configuration Wizard opens.
3. Enter the following information.

Field	Description
What do you want to call your endpoint	Enter the following: `GetWeather`
What does this endpoint do	Enter the following: `Get the weather`

Copyright © 2022, 2026, Oracle and/or its affiliates.

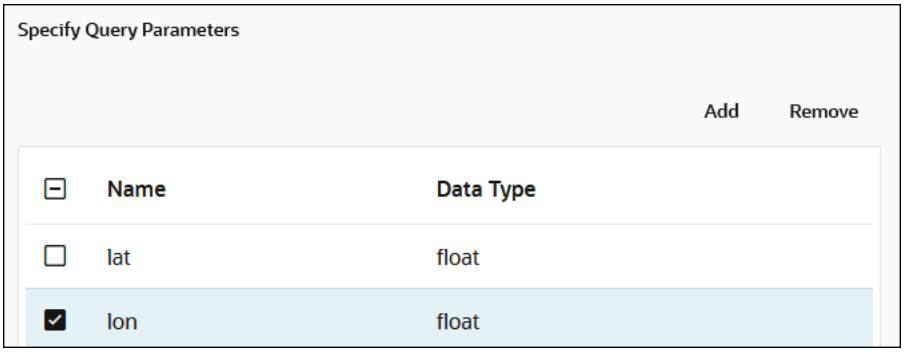
Field	Description
What is the endpoint's relative resource URI	Enter the following: `/data/2.5/weather`
What action do you want to perform on the endpoint	Select **GET**.
Select any options that you want to configure	Scroll to the bottom of the page and select the following options: • **Add and review parameters for this endpoint** • **Configure this endpoint to receive the response**

4. Leave all remaining fields as they are.
The completed Basic Info page looks as follows.



5. Click **Continue** to access the Request Parameters page.
   a. In the **Specify Query Parameters** section, click **Add**.
   b. Double-click the row in the **Name** column to enter a value.
   c. Enter `lat`, then select **float** from the **Data Type** list.
   d. Click **Add** again.
   e. Double-click the row in the **Name** column to enter a value.
   f. Enter `lon`, then select **float** from the **Data Type** list.

Copyright © 2022, 2026, Oracle and/or its affiliates.



g. Click outside the **lon** row.

h. Click **Continue** to access the Response page.

6. Enter the following information.

Field	Description
Select the response payload format	Select **JSON Sample**.

Copyright © 2022, 2026, Oracle and/or its affiliates.

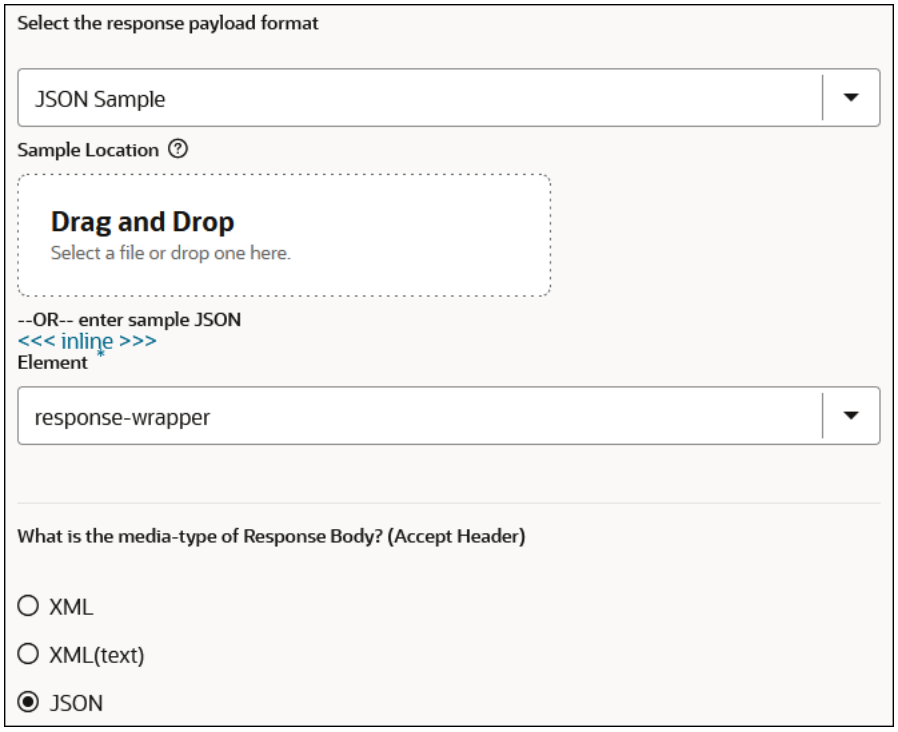
Field	Description

<<< inline >>>	Click this field and enter the following content, then click **OK**.
	
	`{`
	`  "coord" : {`
	`  "lon" : -122.08,`
	`  "lat" : 37.39`
	`  },`
	`  "weather" : [ {`
	`  "id" : 800,`
	`  "main" : "Clear",`
	`  "description" : "clear sky",`
	`  "icon" : "01d"`
	`  } ],`
	`  "base" : "stations",`
	`  "main" : {`
	`  "temp" : 282.55,`
	`  "feels_like" : 281.86,`
	`  "temp_min" : 280.37,`
	`  "temp_max" : 284.26,`
	`  "pressure" : 1023,`
	`  "humidity" : 100`
	`  },`
	`  "visibility" : 10000,`
	`  "wind" : {`
	`  "speed" : 1.5,`
	`  "deg" : 350`
	`  },`
	`  "clouds" : {`
	`  "all" : 1`
	`  },`
	`  "dt" : 1560350645,`
	`  "sys" : {`
	`  "type" : 1,`
	`  "id" : 5122,`
	`  "message" : 0.0139,`
	`  "country" : "US",`
	`  "sunrise" : 1560343627,`
	`  "sunset" : 1560396563`
	`  },`
	`  "timezone" : -25200,`
	`  "id" : 420006353,`
	`  "name" : "Mountain View",`
	`  "cod" : 200`
	`}`

Element	Ensure **response-wrapper** is selected.
What is the media-type of Response Body	Ensure **JSON** is selected.

The completed Response page looks as follows.

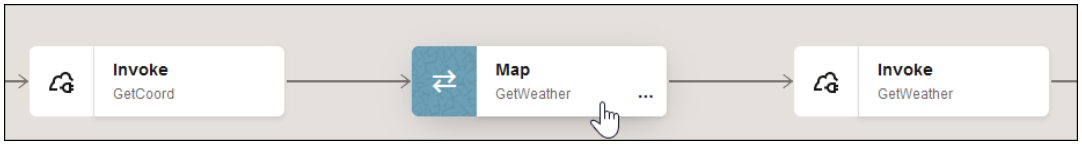
Copyright © 2022, 2026, Oracle and/or its affiliates.



7. Click **Continue** to access the Summary page.
8. Review your selections, then click **Finish**.

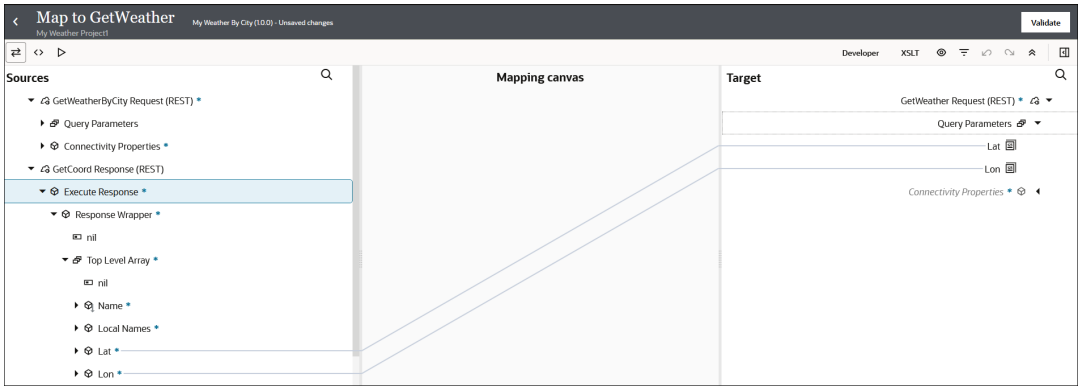
When you configured the **GetCoord** invoke connection in the integration canvas, a map action was automatically created. Let's now map the latitude and the longitude coordinates to the call to get the weather from the OpenWeather application.

1. Double-click the map between the **GetCoord** invoke connection and **GetWeather** invoke connection.



   The mapper opens.
2. In the **Sources** section, expand **GetCoord Response (REST)**, then **Execute Response**, then **Response Wrapper**, and then **Top Level Array**.
3. In the **Target** section, expand **GetWeather Request (REST)**, and then **Query Parameters**.
4. Perform the following mapping:
   - Drag the source **Lat** element to the target **Lat** element.
   - Drag the source **Lon** element to the target **Lon** element.

Copyright © 2022, 2026, Oracle and/or its affiliates.



5. Click **Validate** and wait for the following message to appear.

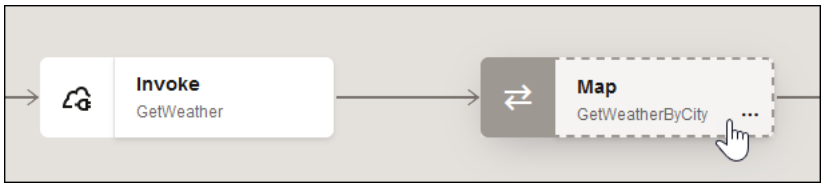
   ```
   Map to GetWeather successfully validated.
   ```

6. Click **Back** to exit the mapper and return to the integration canvas. Your mappings are automatically saved.

Step 9. Map the Weather Report Response Returned from the OpenWeather Application

When you configured the **GetWeather** invoke connection in the integration canvas, a map action was automatically created. Let's now map the response information you receive back from the OpenWeather application.

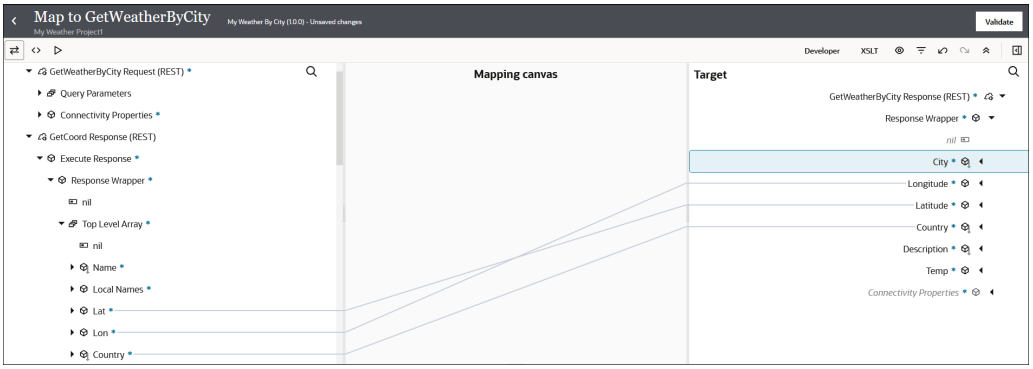
1. Double-click the map after the **GetWeather** invoke connection.



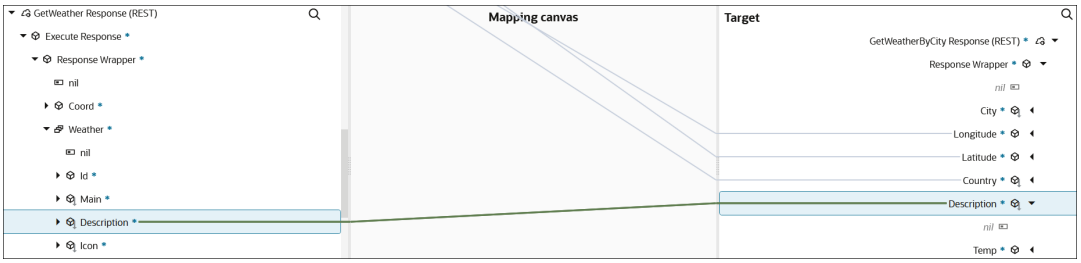
   The mapper opens.

2. In the **Sources** section, expand **GetCoord Response (REST)**, then **Execute Response**, then **Response Wrapper**, and then **Top Level Array**.
3. In the **Target** section, expand **GetWeatherByCity Response (REST)**, and then **Response Wrapper**.
4. Perform the following mapping:
   - Drag the source **Lat** element to the target **Latitude** element.
   - Drag the source **Lon** element to the target **Longitude** element.
   - Drag the source **Country** element to the target **Country** element.

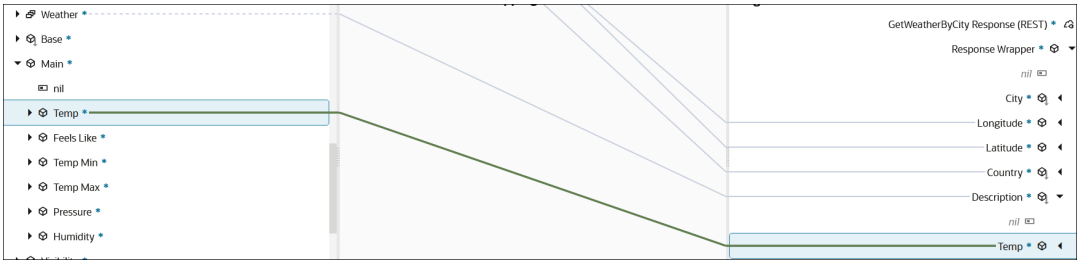
Copyright © 2022, 2026, Oracle and/or its affiliates.



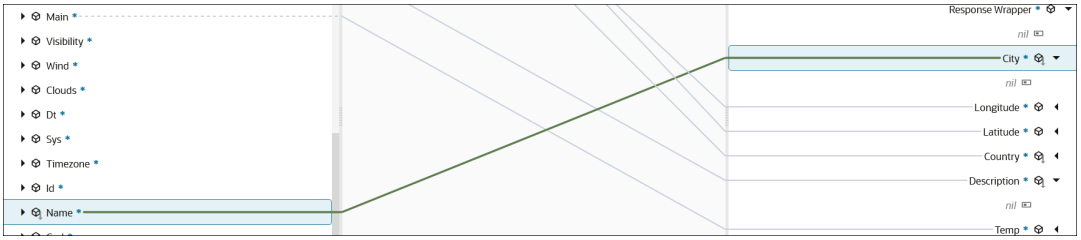
5. Scroll down in the **Sources** section and expand **GetWeather Response (REST)**, then **Execute Response**, then **Response Wrapper**, and then **Weather**.
6. Drag the source **Description** element to the target **Description** element.



7. Collapse the source **Weather** element in the **Sources** section.
8. Scroll down in the **Sources** section and expand the **Main** element.
9. Drag the source **Temp** element to the target **Temp** element.



10. Collapse the source **Main** element in the **Sources** section.
11. Scroll down in the **Sources** section and drag the source **Name** element to the target **City** element.



Copyright © 2022, 2026, Oracle and/or its affiliates.

12. Click **Validate** and wait for the following message.

    ```
    Map to GetWeatherByCity successfully validated.
    ```

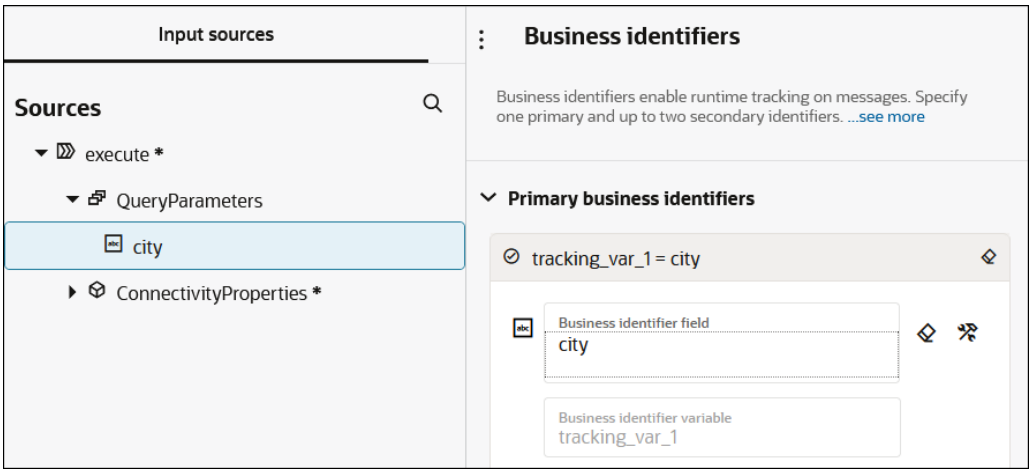
13. Click **Back** to exit the mapper and return to the integration canvas. Your mappings are automatically saved.

Step 10. Define a Business Identifier

Let's now define the primary business identifier for the integration. Business identifiers enable you to track payload fields in messages during runtime. Each integration requires a primary business identifier.

1. Above the integration, click **Business Identifier** .

   a. Expand **QueryParameters**, and drag **city** to the **Business identifier** field.



   b. Click **Save**, then click outside the Business identifiers panel to close it. The business identifier you specify here is visible when you monitor your completed integration instance in the **Observe** tab in Step 12. Monitor the Integration.

2. Check for any red error icons. If you see any, click the icon for details and resolve those errors.
   The complete integration design looks as follows.



3. Click **Back** to exit the integration canvas and return to the project page.

Step 11. Activate and Run the Integration

You are now ready to activate the integration to the runtime environment. Once activated, you can run the integration.

1. In the **Integrations** section, click **Actions** , then select **Activate**.

Copyright © 2022, 2026, Oracle and/or its affiliates.



2. Click **Activate**. Keep the tracing level set to **Production**. If you receive runtime errors, the **Debug (Not recommended)** selection provides detailed information in the activity stream and is useful for testing and troubleshooting any runtime errors.

3. Click **Refresh** periodically.
The integration status changes to **Active**.

   You are now ready to run the integration and send a message.

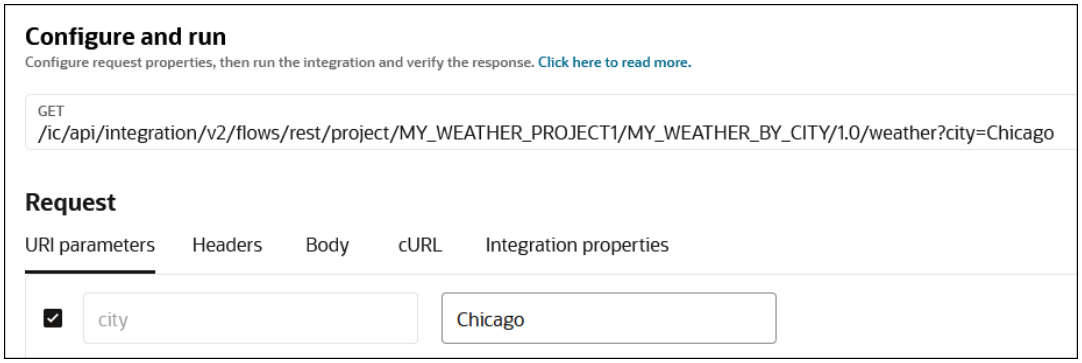
4. In the **Integrations** section, click **Actions** , then select **Run**.



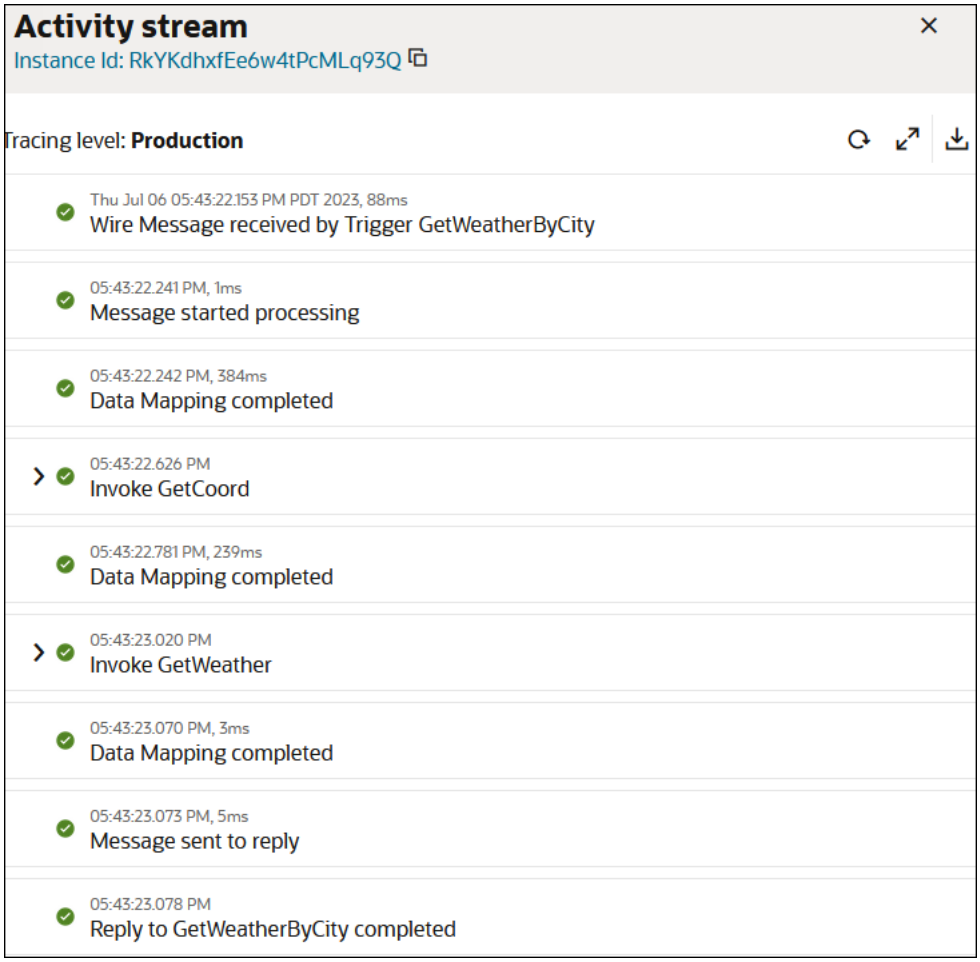
   The Configure and run page enables you to run and test REST Adapter trigger-based integrations. This page provides a way of testing the delivery of an inbound message to your integration.

5. In the **Request** section, specify a city, and click **Run** to send your inbound message to the integration.

Copyright © 2022, 2026, Oracle and/or its affiliates.



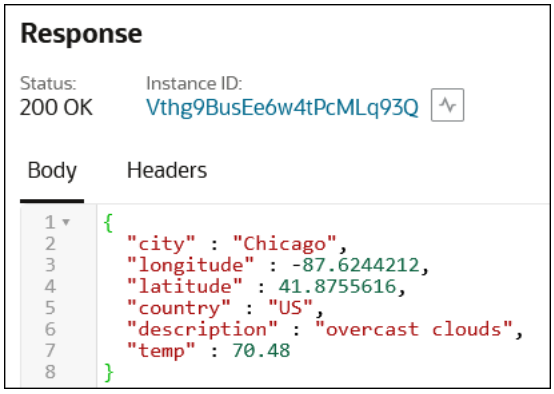
The Activity stream panel opens and shows the movement of the message through the actions that you designed (triggers, invokes, and maps). The unique instance ID that was created for this integration run is displayed at the top. All milestones in this activity stream are green, indicating a successful run of this integration instance.



.

6. View the response message at the bottom of the Configure and run page. The temperature of the city you specified is displayed (for this example, Chicago), along with its longitude and latitude coordinates, country, and current weather description.

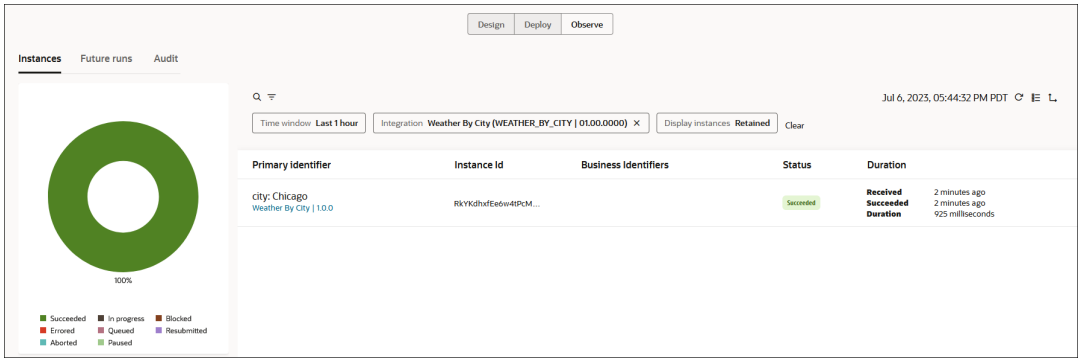
Copyright © 2022, 2026, Oracle and/or its affiliates.



The response information you received back from the OpenWeather application is based on the target-to-source element mappings you performed in the **GetWeatherByCity** map in Step 9. Map the Weather Report Response Returned from the OpenWeather Application.

Step 12. Monitor the Integration

1. Close the Activity steam panel and click **Track instances**.
   The **Observe** tab in your project opens. This tab enables you to monitor the status of your integration instance. The status of the instance is displayed as **Succeeded**. Note that the primary business identifier (**city**) that you defined in Step 10. Define a Business Identifier is visible in the **Primary Identifier** column.



Congratulation! You have successfully created, designed, activated, run, and monitored your second integration. See the section below for more information about the capabilities used in this tutorial.

Learn More About The Features in this Tutorial

- About Projects
- About Integrations
- About Connections
- Understand Trigger and Invoke Connections
- Map Data
- Assign Business Identifiers
- Activate an Integration

Copyright © 2022, 2026, Oracle and/or its affiliates.

- Test Integrations from Outside the Integration Canvas
- Monitor Integrations in a Project

Get Started with Projects

Oracle recommends designing your integrations in projects. Projects provide a single workspace for designing, managing, and monitoring integrations, agentic AI, B2B, decisions, healthcare, human in the loop, and robots. Use projects to perform release management, manage access to automation solutions, and keep everything organized.

Topics:

- About Projects
- Project Guidance If You've Used Packages
- Tour of the Projects Workspace
- 10 Tips for Working with Projects
- 4 Ways to Organize Projects
- About Project Deployment
- Restrictions for Projects

About Projects

A project provides a single unified workspace for all stakeholders to design, manage, and monitor automation solutions, including integrations, agentic AI, B2B, decisions, healthcare, human in the loop, and robots. Additionally, projects provide robust life-cycle management and streamlined updates to prebuilt integrations.

How Projects Help You

With everything at your fingertips, you'll build faster. And, with easy updates to integrations based on accelerators, you'll future-proof your prebuilt integrations. Finally, deploy quickly and with confidence using project deployments. Keep reading to learn more.

1. **Manage the releases of your automation solutions**

   With project deployments, you can move components to higher environments, such as testing or production, quickly and efficiently. A project deployment represents the current snapshot of all the integration and robot versions that must be in the environment you're promoting to. Enjoy improved stability in your higher environments and easier rollbacks when things go wrong.

   See About Project Deployment.

2. **Control access to automation solutions and their components using fine-grained access control**

   With projects, you control the users and groups who can edit, view, and monitor an automation solution, including any integrations, agentic AI, B2B, decisions, healthcare, human in the loop, and robots, using role-based access control (RBAC). Creating automation solutions within a project is the only way to control access at this fine-grained level.

   See Control Who Can Edit, View, and Monitor in a Project.

3. **Build, manage, and monitor everything in one place**

Copyright © 2022, 2026, Oracle and/or its affiliates.

A project helps you get started quickly and confidently because you build all the components that an automation solution needs within a project. You don't need to click all over the user interface to find the right page to create a component. It's all right there within your project, including the ability to monitor your automation solutions.

How many integrations should you include in a project? What guardrails are in place to govern your use of projects? Get answers to these questions and more. See:

- 10 Tips for Working with Projects
- 4 Ways to Organize Projects
- Restrictions for Projects

4. **Build faster by reusing components**

   After you've built an integration or two within a project, creating additional integrations is even faster and easier. The connections, JavaScript libraries, and lookups from existing integrations are all at your fingertips, offering easy one-stop shopping. The same is true for agentic AI, B2B, decisions, healthcare, human in the loop, and robots. Just grab what you need from the streamlined user interface and start building.

5. **Future-proof your prebuilt integrations with easy updates**

   Accelerators provide prebuilt integrations that you can easily customize, but what happens to your customized integration when a new version of the accelerator is released? When you install the accelerator in a project, you can automatically update to the new version without reworking your customizations. Use accelerators with confidence, knowing that you can always accept the newest version without having to set aside time to modify and troubleshoot your integration.

   See Extend an Integration in an Accelerator Project and Upgrade an Accelerator Project and Merge Extensions.

6. **Facilitate the exchange of business documents using projects**

   In a project, you can design, manage, and monitor B2B integrations. Use an AS2 Adapter connection as a trigger or an invoke connection, or use an FTP Adapter connection as an invoke connection. Track the processing of B2B messages as integration instances during runtime in the project.

   See Introduction to B2B for Oracle Integration in *Using B2B for Oracle Integration 3*.

Factors to Consider Before Importing Integrations into Projects

If you've created automation solutions outside of a project, consider the following factors when planning to import them into a project:

- The conversion process requires a small time investment.
- When you move an integration into a project, the integration endpoint changes.
- All components in a project are subject to the service limits.

  See Service Limits in *Provisioning and Administering Oracle Integration 3*.

Of course, projects provide many benefits, which typically justify the time costs of migration.

Ready to Convert Your Packages to Projects?

If you upgraded from Oracle Integration Generation 2, you can convert your packages to projects at any time. But first, familiarize yourself with the differences between packages and projects. See Project Guidance If You've Used Packages.

Copyright © 2022, 2026, Oracle and/or its affiliates.

Get Started with Projects

See Design and Monitor Integrations in Projects.

Project Guidance If You've Used Packages

If you upgraded from Oracle Integration Generation 2, some of your integrations might be in packages. Learn the key differences between packages and projects and how to get started using projects.

Key Differences Between Packages and Projects

Area	Packages	Projects
Overall usage and goals	Packages primarily offer help with organization. For example, you can collect related integrations in a single package. You can also import and export packages, including their contents.	Projects offer many benefits, including improved release management, organization of artifacts, and a single unified workspace for designing, managing, and monitoring integrations. See About Projects.
Access control	Everyone who works in Oracle Integration can access the integrations in a package. Similarly, everyone can access integrations that aren't part of a package.	Projects offer fine-grained access control. You choose the people who can edit, view, and monitor the integrations and other assets in each project. See Control Who Can Edit, View, and Monitor in a Project.
Deployment	Creating a continuous integration and continuous deployment (CI/CD) pipeline and delivering updated integrations to higher environments requires work outside Oracle Integration. For instance, you can export a package and use the solution of your choice, such as Oracle Visual Studio, APIs, or shell scripts, to move the integrations into a testing or production environment.	If you've deployed integrations from packages before, deployment is easier and faster from a project. Projects provide built-in deployment capabilities for improved release management, including controlled deployment. Deploy your integrations where you design them, from within a project, without needing to create a pipeline or run a build. See Workflow for Deploying Automation Solutions.
Observability	Packages don't offer monitoring capabilities. You monitor integrations individually on the Observability pages.	You have two options for monitoring the integrations in a project: The Observe pages in the project and the Observability pages, which are outside the project. Monitoring within the project provides helpful context for the integrations.

Workflow for Converting a Package to a Project

If you used packages in Oracle Integration Generation 2, you can convert them to projects in just a couple of minutes using the package conversion tool.

The package conversion tool moves everything in the package into a project, including integrations, connections, lookups, and JavaScript libraries. The existing package remains untouched.

Copyright © 2022, 2026, Oracle and/or its affiliates.

> **Note**
>
> You can't convert accelerator and recipe packages into projects.

Step	Task	More information
1	Create a project from a package	Convert a Package to a Project
2	Control access to the project by selecting the people who can edit, view, and monitor the project	If you don't assign access, everyone's access within Oracle Integration determines the work they can do in the project. Control Who Can Edit, View, and Monitor in a Project

After you've created the project, you can deploy the integrations within it. See Create and Manage a Project Deployment.

Workflow for Importing Standalone Integrations into Projects

Ready to start moving your standalone integrations into projects? All you need to do is spend a few minutes importing the integrations and their components into one or more projects.

Step	Task	More information
1	Create a strategy for organizing projects	4 Ways to Organize Projects
2	Create a project	Create or Import a Project
3	Export the integrations that you want to import into a project	Export an Integration
4	Add one or more integrations to the project	Copy an Integration into a Project
5	Configure the properties for the connections in the integrations	Configure Connection Properties and Security Properties

Tour of the Projects Workspace

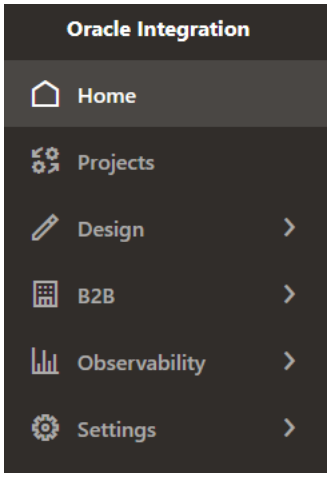
The Projects workspace contains several tabs for designing, managing, and monitoring automation solutions.

Navigation to the Projects Workspace

A **Projects** option is available in the navigation pane.

In the Projects workspace, you can complete all your design, management, and monitoring tasks. You don't need to navigate to the **Design** and **Observability** pages when working in a project.

Copyright © 2022, 2026, Oracle and/or its affiliates.

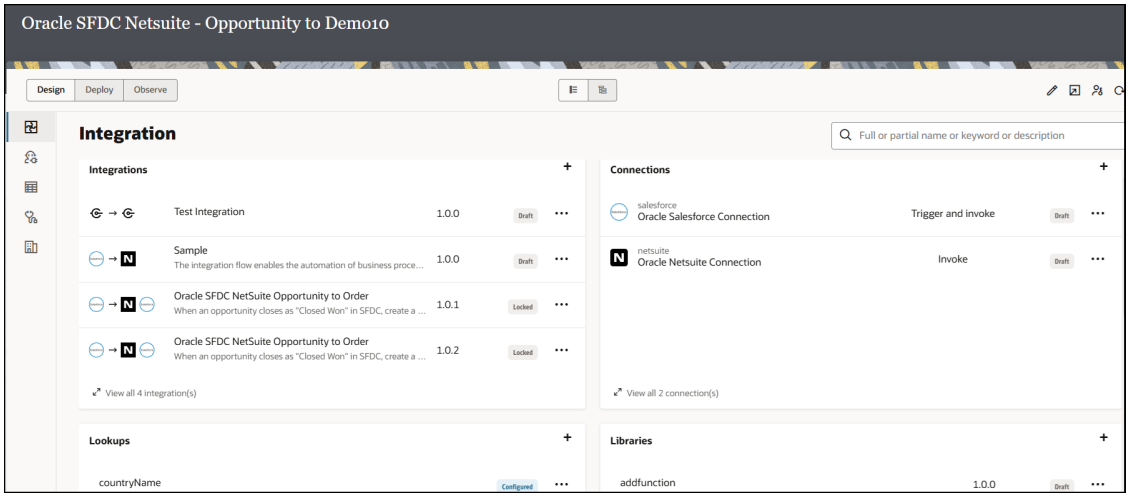


Tip

When you move around the project workspace performing tasks, a **Back** button is provided that enables you to easily return to your previous location under the **Projects** option.

Design Tab

Design and manage integrations on the Design tab, which appears by default when you open a project. The design tab includes a vertical navigation bar to navigate among the pages.



The Design tab contains the following pages.

Integration Page

The following components are available on the Integration page on the Design tab.

Copyright © 2022, 2026, Oracle and/or its affiliates.

Component	Description
Integrations section	Create integrations in this section. Additional tasks that you perform in this section include: • Viewing, editing, cloning, deleting, searching, activating, and deactivating integrations. Click **View all integration(s)** at the bottom of the list of integrations to show the status of any schedule integrations (for example, stopped, paused, no schedule defined, and others). • Enabling or disabling the ability for REST Adapter-triggered and AS2 Adapter-triggered integration instances to be replayable at runtime. • Creating a new version of an integration. • Setting tracing levels. • Refreshing endpoints. • Updating property values. • Running REST Adapter trigger-based integrations and schedule integrations. • Viewing run details.
Connections section	Create connections for your integrations in this section. Additional tasks that you perform in this section include: • Editing, deleting, and searching for connections. • Refreshing metadata.
Lookups section	Create lookups for your integrations in this section. Additional tasks that you perform in this section include editing, deleting, and searching for lookups.
Libraries section	Create JavaScript libraries for your integrations in this section. Additional tasks that you perform in this section include deleting, updating, and searching for libraries.
Events area	Create events to publish and subscribe to in integrations. Additional tasks that you perform in this section include cloning, editing, deleting, and viewing events.

Robot Page

The following components are available on the Robot page on the Design tab.

For information about working with robots and their components, see Learn About Robots and Build a Robot in *Using Robots in Oracle Integration 3*.

Component	Description
Robots section	Create and edit robots, create new versions of robots, associate and remove environment pools, activate and deactivate robots, delete robots, and more in this section.
Environment pools section	Create and edit environment pools, add and remove environments from the pools, and more in this section.
Robot connection types section	Create, edit, and delete robot connection types in this section.
Robot connections section	Create, edit, and delete robot connections in this section.

Copyright © 2022, 2026, Oracle and/or its affiliates.

Human in the Loop Page

The following components are available on the Human in the loop page on the Design tab.

Component	Description
Workflows section	Create and edit workflows, create new versions of workflows, manage data in workflows, activate and deactivate workflows, delete workflows, and more in this section.
Forms section	Create and edit forms, work with form controls, create new versions of forms, activate and deactivate forms, delete forms, and more in this section.

Healthcare Page

The following components are available on the Healthcare page on the Design tab.

Component	Description
HL7 messages and schemas	Create HL7 messages and schemas in this section. See Create Oracle Integration for Healthcare Schemas and Documents in *Using Oracle Integration for Healthcare in Oracle Integration 3*. Additional healthcare-related tasks that you can perform include: • Cloning, editing, viewing, and deleting schemas. • Cloning, editing, viewing, and deleting messages. • Exporting a project that includes healthcare resources (schemas, messages, and integrations using a healthcare action) and importing that project into another instance. • Cloning a project that includes healthcare resources (schemas, messages, and integrations using a healthcare action) You can select the HL7 message when configuring the healthcare action in an integration in the **Integrations** section of the project. See Convert HL7 Messages with a Healthcare Action in *Using Oracle Integration for Healthcare in Oracle Integration 3*.

B2B Page

The following components are available on the B2B page on the Design tab.

Copyright © 2022, 2026, Oracle and/or its affiliates.

Component	Description
Documents and schemas	Create or import documents and schemas in this section. See B2B Schemas and B2B Documents in *Using B2B for Oracle Integration 3*. Additional B2B for Oracle Integration-related tasks that you can perform include: • Editing, viewing, cloning, exporting, and deleting schemas, and generating an implementation guide. • Editing, viewing, cloning, exporting, and deleting documents. • Exporting a project that includes resources (schemas, documents, and integrations using a B2B action) and importing that project into another instance. • Cloning a project that includes B2B resources (schemas, documents, and integrations using a B2B action). You can select the document when configuring the B2B action in an integration in the **Integrations** section of the project. See Use the B2B Action in Standalone Mode in *Using B2B for Oracle Integration 3*.
Trading partners	Create or import trading partners in this section. You can also edit, view, export, and delete trading partners. See Use B2B for Oracle Integration in Trading Partner Mode in *Using B2B for Oracle Integration 3*.

Not all B2B features are currently supported in projects. See Design B2B Integrations in Projects or in Standalone Environments.

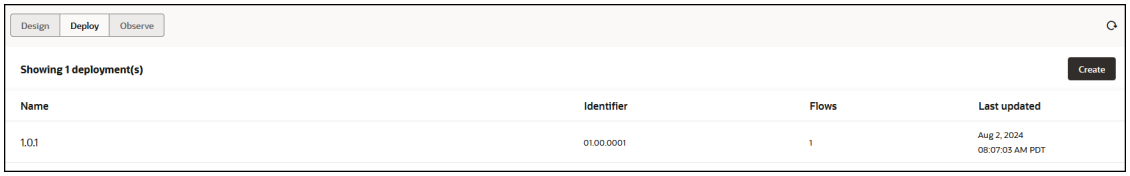
Components on All Pages

Some components are available on all pages in the Design tab.

Component	Description
View dependencies	View the dependent relationships between all integrations, connections, lookups, JavaScript libraries, and events in a project.
Edit details	View integration description details.
Export	Export a project.
Share	Assign a project owner and specify the people who can edit, view, and monitor resources in the project.
Refresh	Refresh the area you are in separately from other areas (for example, healthcare).

Deploy Tab

Deploy one or more automation solutions in a project on the Deploy tab.



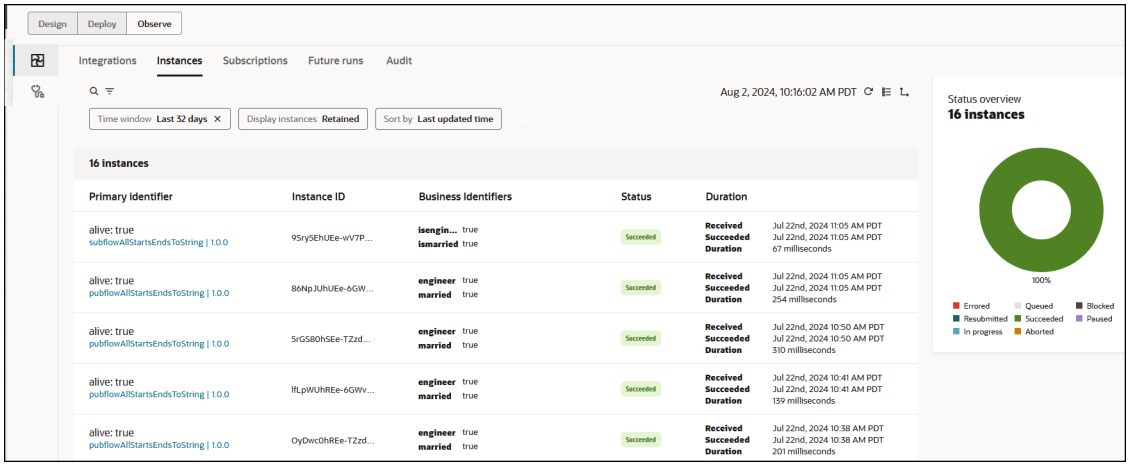
Copyright © 2022, 2026, Oracle and/or its affiliates.

The Deploy tab contains the following components:

Component	Description
Create button	Create a project deployment.
Options that appear when you point to a project deployment	Contains options for exporting, activating, and deactivating a project deployment.
Actions ⋮ menu, which appears when you point to a project deployment	Contains options for editing, viewing, cloning, and deleting a project deployment.

Observe Tab

Monitor your automation solutions on the Observe tab.



The Observe tab contains the following pages.

Integration Page

The following tabs are available on the Integration page on the Observe tab.

Tab	Description
Integrations tab	See high-level information about your integrations and their messages.

Copyright © 2022, 2026, Oracle and/or its affiliates.

Tab	Description
Instances tab	This tab contains the following components: • A table for tracking the status of integration instances through business identifiers and viewing the activity stream and message payloads of an instance. Use a clickable graph to filter the display of the state of instances. You can also use the **Filter** icon to perform the same task. • A section for viewing instance errors. You can resubmit failed instances, discard failed instances, view message recovery status, and view basic and detailed error messages. • An **Actions** menu for each instance to perform other tasks such as viewing the integration schedule and child instances and replaying REST Adapter-triggered and AS2 Adapter-triggered integration instances that you configured as replayable during integration activation.
Subscriptions tab	Shows the currently subscribed integrations, the event, retained events, and subscriber status.
Future runs tab	View a calendar of scheduled integration runs.
Audit tab	View and download the design-time audit log of actions performed in a project.

Robot Page

The following tabs are available on the Robot page on the Observe tab.

Tab	Description
Robot instances tab	This tab contains a table for tracking the status of robot instances through business identifiers and viewing the activity stream and message payloads of an instance. Use a clickable graph to filter the display of the state of instances. You can also use the **Filter** icon to perform the same task.

B2B Page

The following tabs are available on the B2B page on the Observe tab.

Copyright © 2022, 2026, Oracle and/or its affiliates.

Tab	Description
Business messages	This tab contains the following components: • A filter for sorting on last message created, date range, trading partner name, message direction, message type, status, resubmit status, document status, document standard. message ID, wire message ID, transport message ID, integration instance ID, error code, document version, document type, business identifier, parent message ID, resubmitted message ID, collaboration ID, group control number, transaction control number, and interchange control number. • A table for viewing the document, trading partner, document status, business identifier, and creation date. • A **View details** link for accessing business message details, such as properties, message tags, payloads, and correlations. • An **Actions** menu for viewing wire messages and functional acknowledgments.
Wire messages	This tab contains the following components: • A filter for sorting on last message created, date range, trading partner name, message direction, message type, status, resubmit status, transport, message ID, transport message ID, integration instance ID, error code, payload name, parent message ID, and resubmitted message ID. • A table for viewing the transport, trading partner, transport status, transport message ID, creation date, payload name, and payload size. • A **View details** for accessing wire message details, such as properties, message tags, payloads, and correlations. • An **Actions** menu for viewing business messages and the message disposition notification (MDN).

10 Tips for Working with Projects

Ready to be an expert at projects? These tips will help you get started on the right path.

1. Allow Your Business Goals to Inform Your Strategy for Projects

With a project, you can create a custom solution that supports your business goals. See 4 Ways to Organize Projects.

2. Aim for No More Than Twenty Integrations per Project

The optimal number of integrations for each project is the number that makes sense for your business goals. Oracle recommends limiting the integrations in a project to about twenty, at least when you're starting out, so that the assets within a project are still easy to find and navigate at a glance. Whatever number you choose, make sure you stay within the service limits. See Service Limits in *Provisioning and Administering Oracle Integration 3*.

3. Remember That Organization Isn't Always About Separation

Managing related integrations in a project is useful, but what if you want to call a child integration from integrations that are in multiple projects? Not to worry, projects can accommodate this requirement, too. When you create an integration in a project, choose whether to make the integration available in other projects, in addition to its own project.

See Invoke Child Integrations Inside or Outside of Projects and Create an Integration.

Copyright © 2022, 2026, Oracle and/or its affiliates.

4. Keep Everything Secure by Giving the Minimum Access Required

The tenancy administrator assigns application roles that grant people access to work in Oracle Integration. The tenancy administrator assigns these roles in the Oracle Cloud Console.

Additionally, the project owner, who is typically either a developer or an administrator, manages access to projects using role-based access control (RBAC). The project owner assigns these roles in a project in Oracle Integration.

To work in a project, a person must have both the required application roles and the required project roles. Project roles never override application roles. As a best practice, administrators and project owners should give people only the minimum access they need to do their jobs.

Reminder: You can control access to projects, but the control extends only to design time, not to runtime. Those with the endpoint, the credentials, and an OAuth token can run the integration at runtime.

5. Plan Your Working Times if You're Working with a Team

Your entire team can work in a project at the same time, but the same is not true of integrations. When a person opens an integration for editing, the integration is locked, and no one else can edit it. Make sure you plan your work accordingly so that everyone can complete their work in a timely way.

6. Be Descriptive When Versioning and Deploying

When you version an integration and when you create a project deployment, make sure you include information about the changes in the Description field. This information helps you select the right integration version when you create the project deployment and select the right project deployment to export.

7. Adopt a Versioning Strategy for Integrations

Version numbers for integrations use the following format: *xx.yy.zzzz*, where each *x*, *y*, and *z* is a number that you specify.

Typically in versioning, the *xx* numbers correspond to major versions, the *yy* numbers correspond to minor versions, and the *zzzz* numbers correspond to patch versions. You can increment version numbers however you want, but Oracle recommends establishing standard operating procedures for when you increment each number. For instance:

- Increment the *xx* number by one when you add an integration to the project deployment.

  The endpoint URLs for the integrations in the project deployment don't change when you increment the version number.
- Increment the *yy* number by one for significant changes to integrations.
- Increment the *zzzz* number by one for minor changes.

8. Choose Carefully When Creating Project Deployments

Deploying multiple integrations at the same time is easy, thanks to fine-grained deployment creations in a project. All you need to do is select the integrations to include and their versions. Just make sure you select the correct integrations and versions to include in the project deployment.

See About Project Deployment.

Copyright © 2022, 2026, Oracle and/or its affiliates.

9. Update Your Workflows

Projects offer streamlined workflows for all your integration work, from designing to monitoring. Spend less time navigating from place to place in the user interface and more time meeting your organization's integration needs.

See Workflows for Integrations .

10. Monitor Integrations on the Observe Page

When you want to see what's happening with all the integrations in a project, go to the Observe page within a project. Here, you can see at-a-glance status and error information for all of the project's integrations.

Additionally, as long as you have access to a project, you can monitor the integrations on the Observability pages.

4 Ways to Organize Projects

Ready to get started with projects, but want some organization inspiration? Keep reading for ways you can use projects to organize your integrations.

The number of projects that you create and the integrations that you include in each project is up to you, allowing you to create a custom solution that supports your business goals.

1. Organize by Product or Line of Business

When you organize by product, you can organize the work of each line of business that works in an instance. For example, create:

- One or more projects for ERP integrations

  For example, one project for accounts receivable integrations in ERP, another for order capture in ERP, another for year end in ERP, and so on.
- One or more projects for HCM integrations

  For example, one project for onboarding employees in HCM, and so on.
- Additional projects as needed.

2. Organize by Solution

When you organize by solution, all components related to a solution are in a single project.

For example, create one project for all components related to synchronizing Salesforce opportunities with Oracle Fusion Cloud Applications, and create additional projects for other solutions.

3. Keep Common Integrations Together

Projects optimize your day by keeping everything you need at your fingertips. If you have a number of logging assets or error handling assets, you can group them in one or two projects so that you can find them more easily.

For example, you might create one or more integrations that handle error handling and call the integrations from your other integrations. Keeping all the integrations related to error handling in a single project makes the integrations easy to find.

Copyright © 2022, 2026, Oracle and/or its affiliates.

4. Mix and Match

Why limit yourself to one method of organization? Projects are highly customizable so that you can create as many as you need. Mix and match these options, or come up with your own ideas.

About Project Deployment

Use a project deployment to move automation solutions with integrations and robots to a higher environment, such as from development to testing. Project deployments offer convenience, easier activation of integrations, and improved stability of your testing and production environments.

If your process for deploying automation solutions to a new environment has focused on moving only updated integrations, a project deployment is going to offer new capabilities, including a more stable production environment. To take advantage of all the benefits of production deployments, you'll need to approach your promotion differently. Keep reading to learn more.

Think Beyond New and Updated Integrations and Robots

Before working in projects, when you needed to deploy a new or updated integration to a testing or production environment, you probably moved only the new or updated integration.

To take advantage of the convenience of project deployments, including easier activation of integrations, you can continue this practice and include only new and updated integrations in each project deployment. However, to gain improved stability in your testing and production environments, you should deploy differently when working with project deployments.

Project Deployments Are the Current Snapshot

Think of a project deployment as a snapshot of all the integrations that must be in the environment that you're promoting to. For instance, consider a project with fifteen integrations. You included ten of the integrations in an earlier deployment and promoted them to your testing environment. You're now ready to promote the remaining five integrations to the testing environment.

Before working in projects, you probably would have moved only the remaining five integrations to testing. However, the best practice for project deployments is to include the five integrations as well as the ten other integrations in the project deployment.

Enjoy Improved Stability and Easier Rollbacks

If the ten integrations in testing haven't changed, why should you include them in the project deployment? The reason is stability.

If you include all fifteen integrations in the production deployment and experience any issues in the testing environment, you can roll back to the previous stable version of your product deployment.

Issues in a testing environment are expected and likely aren't going to derail your business. However, imagine the power of a project deployment in your production environment. Your business will stay on track, and you can troubleshoot and resolve any issues in your lower environments. For example, if a new or updated integration introduces any issues, you can:

1. Perform a rollback by deactivating the faulty individual integration to avoid impacting unaffected integrations included in the project deployment.
2. Reactive the prior stable project deployment.

Copyright © 2022, 2026, Oracle and/or its affiliates.

Once the faulty integration is fixed, you can create and activate a new project deployment that includes the fixed version, along with all the unaffected integrations. Your business will stay on track, and you can troubleshoot and resolve any issues in your lower environments.

See Life Cycle of a Project Deployment that Requires a Rollback.

Workflow for Deploying Automation Solutions

When your automation solutions are finished, a few fast and easy steps promote the integrations, robots, and more to your testing or production environments. See Workflow for Deploying Automation Solutions.

Restrictions for Projects

Be aware of the restrictions for projects.

Contents of a project

- There are service limits on the number of integrations, project deployments, connections, lookups, and JavaScript libraries you can create in a project. See Service Limits in *Provisioning and Administering Oracle Integration 3*.

  To stay within the project deployment limit, delete old versions periodically. If you need to keep older versions for auditing purposes, you can export the project deployments and archive them using your organization's archiving tools, such as a source control system.
- Lookups and JavaScript libraries are visible and usable only in a single project.

  Your uncluttered view shows you only the assets in which you are interested.

  Similarly, assets created in standalone integrations or integrations in packages are not visible or usable in a project.
- You can use certificates, which are defined globally, from any project.

Security and other restrictions

- You cannot invoke one project from another project.
- You cannot migrate a package of integrations to a project.
- You cannot clone individual JavaScript libraries within a project.

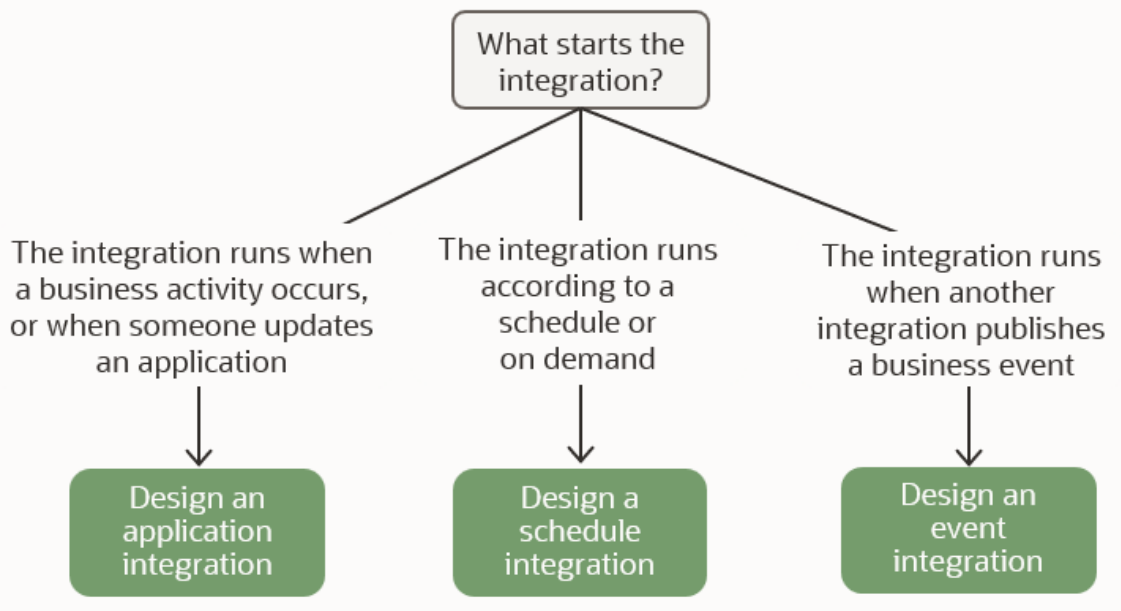
Understand Integration Styles

You can select from several styles when creating an integration in the Create integration panel.

What Kind of Integration Should You Design?

When you create an integration, the first decision to make is the integration model to use. The main difference among the models is the way that they start running.

Copyright © 2022, 2026, Oracle and/or its affiliates.



Available Integration Styles

Style	Description
Application	Create an application integration (previously known as app-driven orchestration) that uses an event or a business object to trigger the integration. For example, you create an integration with an Oracle Service Cloud (RightNow) Adapter as a trigger and an Oracle CX Sales and B2B Service Adapter as an invoke. The Oracle Service Cloud (RightNow) Adapter subscribes to an event from the Oracle CX Sales and B2B Service application to trigger the integration. Application integrations include many actions for performing specific tasks, such as the following: • Switch actions to create multiple routing expressions. • For-each actions for looping over repeating elements. • Assign actions for assigning values to scalar variables. • Ad-hoc mappings on switch actions branches. • Callback actions (to end a process and respond back to the sender) and end actions (to end a process without responding back to the sender) in asynchronous integrations. See About Application Integrations and Create Application Integrations.
Schedule	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 About Schedule Integrations and Create Schedule Integrations.
Event	Create and select events for publishing and subscribing to in integrations. You define the events in JSON-formatted or XML schema files. **Note**: If you created integrations using the **Publish to OIC** and **Subscribe to OIC** integration styles (often called pub/sub) in Oracle Integration Generation 2, these capabilities and more are available in Oracle Integration 3. To create an integration that publishes an event, create an application integration. To create an integration that subscribes to published events, create an event integration. See About Event Integrations Create Event Integrations.

Copyright © 2022, 2026, Oracle and/or its affiliates.

About Application Integrations

An application integration runs when a business activity occurs, such as a customer submitting an order; or when someone adds or changes something in an integration. Application integrations can be synchronous or asynchronous, depending on whether they provide an immediate response.

Quick Introduction to Sync and Async

- **If you require real-time information, create a synchronous integration.**

  For the sake of efficiency, you might want to pair it with an asynchronous integration.

- **To perform work in batch operations, create an asynchronous integration.**

  An asynchronous integration is sometimes called fire and forget.

Real-World Analogy: Phone versus Email

Consider a situation that you've probably encountered: You need a coworker who is in a different location to complete a task for you.

You have several options for contacting your coworker:

- **If the task is urgent and time sensitive, you call your coworker**

  You need to know that they received your request and that they'll start and finish the task quickly.

  In the integration world, this scenario is **equivalent to a synchronous integration**. Synchronous integrations send immediate responses and run quickly, usually in seconds or less.

- **If the task isn't urgent or time sensitive, you email your coworker**

  You won't get a confirmation that they have read the message, but you know that your coworker is very responsive to requests. They'll fit the task into their schedule as soon as they can.

  In the integration world, this scenario is **equivalent to an asynchronous integration**. Oracle Integration adds the requests for asynchronous integrations to a queue and processes the requests as soon as it can.

Learn More About Synchronous and Asynchronous Integrations

- How to Choose Between Synchronous and Asynchronous Integrations
- How to Specify if an Integration is Synchronous or Asynchronous
- Differences Between Synchronous and Asynchronous Integrations

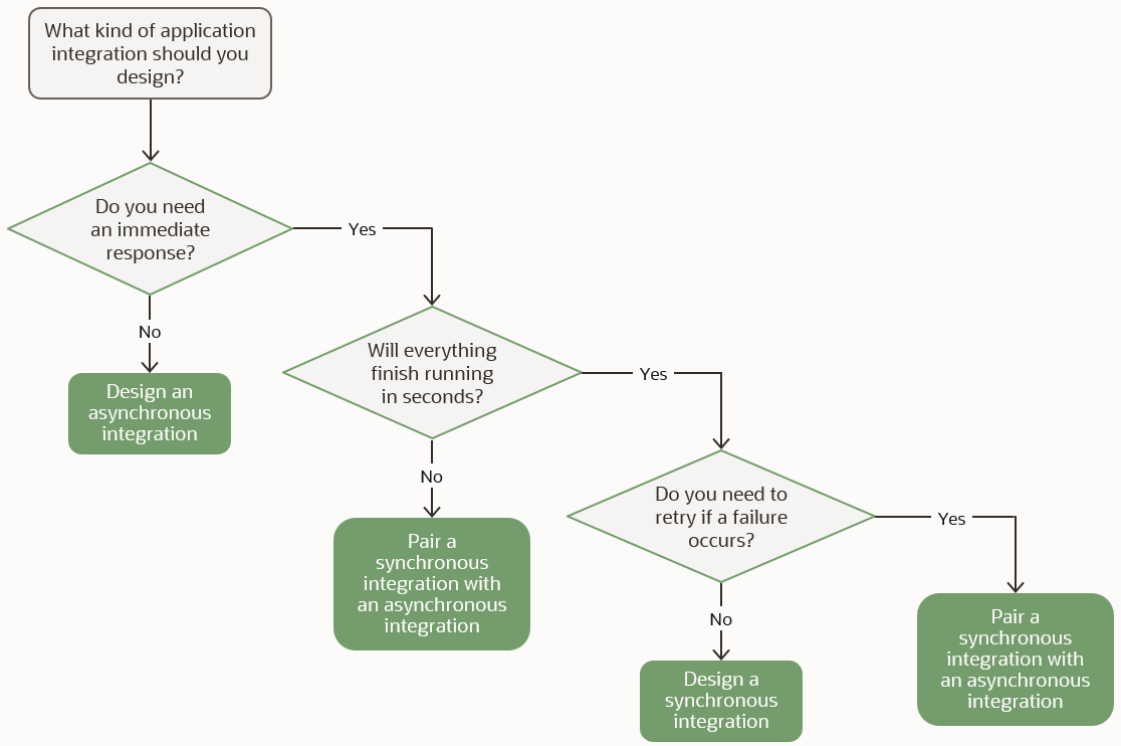
How to Choose Between Synchronous and Asynchronous Integrations

An application integration can be synchronous or asynchronous. Learn when to create an asynchronous integration, a synchronous integration, or a combination of both.

What Kind of Application Integration Should You Design?

You have several options for application integrations: They can be synchronous, asynchronous, or a combination. When you know you need an application integration, the following decision tree helps you choose the type of application integration.

Copyright © 2022, 2026, Oracle and/or its affiliates.



Detailed Explanation

The following table walks you through the questions in the previous decision tree and provides additional context for the questions and answers.

Copyright © 2022, 2026, Oracle and/or its affiliates.

Question and details	Examples	Next steps
Do you need an immediate response? Sometimes, the client application that starts the integration requires an immediate response.	**No, I don't need an immediate response** If you're updating a record in an application, you typically don't need an immediate response. For example: • You need to synchronize inventory between two applications. • You need to create a packing list for an order. • You need to create a new employee's accounts in various business systems after you add a new record for them in your human capital management (HCM) software. **Yes, I need an immediate response** • A customer places an order on your website and wants to see an order confirmation page right away. • You must complete a credit check for a customer before fulfilling an order.	**No, I don't need an immediate response** Create an asynchronous integration. You can stop reading here. **Yes, I need an immediate response** Go to the next question.
Will everything finish running in seconds? Synchronous integrations should finish in milliseconds or seconds. Asynchronous integrations are better suited to more complex or longer-running integrations, such as integrations that must process hundreds of records. To make this decision, draw upon your and your coworkers' experience with building integrations. If everyone is new to building integrations, consider building the integration, testing it under real-world circumstances, and checking the integration's performance in the activity stream. If the integration's running time is too long, you can always move some of the logic into a separate synchronous integration.	**No, the integration can't run in seconds** Consider an online retailer that collects all orders in a single file. One time per day, they run an integration that loops through the orders and processes them. The processing work includes validation, enrichment, and posting to an order-processing application. This integration takes several minutes to run. **Yes, the integration can run in seconds** Consider a bank that lets customers check their bank balances in a mobile app. The customers want to know their balances immediately, not in a minute or two.	**No, the integration can't run in seconds** Create a synchronous integration that provides an immediate response, and use it to run an asynchronous integration that performs the longer or more complex task. For example, when a synchronous integration provides a customer with an order confirmation number, the synchronous integration can kick off an asynchronous integration that completes the remaining order-related tasks, including creating an account for the customer, registering the customer for marketing emails, updating inventory, and generating a packing list, and sending the packing list to the warehouse. **Yes, the integration can run in seconds** Go to the next question.

Copyright © 2022, 2026, Oracle and/or its affiliates.

Question and details	Examples	Next steps
Do you need to retry if a failure occurs? Integration failures typically occur when the client application that the integration connects to goes down. Synchronous and asynchronous integrations handle failures differently: • When a synchronous integration fails, Oracle Integration returns a failure response to the client. You cannot rerun the integration. Sometimes, you don't want to rerun an integration, typically because the information that the integration obtains is time sensitive and is relevant only to an immediate need. • Oracle Integration saves all requests for asynchronous integrations. Therefore, if the client application is down or otherwise unavailable when Oracle Integration sends a request for an asynchronous integration, you can rerun the integration instance in the future.	**No, I don't need to rerun the integration** Rerunning an integration doesn't make sense for the following situations: • Obtaining the current price for a stock ticker. • Getting the current inventory numbers for an item. • Determining how many seats are currently available for a show. **Yes, I need to rerun the integration** Reunning an integration makes sense for the following situations: • Updating inventory in an application after the application went down for maintenance. After the application is available again, you need to rerun the integration so that your inventory can be synchronized for the start of business tomorrow. • Attaching a document to a new employee's record after someone creates the employee's record in the HCM software. After the employee record is created, you need to be able to attach the document.	**No, I don't need to rerun the integration** Create a synchronous integration. **Yes, I need to rerun the integration** Create a synchronous integration that provides an immediate response, and use it to run an asynchronous integration that contains the actions that you need to rerun.

How to Specify if an Integration is Synchronous or Asynchronous

When you define the trigger for an integration, you must determine whether to configure the endpoint to receive a response. This selection determines whether the integration is synchronous or asynchronous.

- When you configure the endpoint to receive a response, the integration becomes synchronous.
- When you don't configure the endpoint to receive a response, the integration becomes asynchronous.

Copyright © 2022, 2026, Oracle and/or its affiliates.

Differences Between Synchronous and Asynchronous Integrations

An application integration can be synchronous or asynchronous. Synchronous and asynchronous integrations have some key differences, including whether they provide an immediate response.

Question	Synchronous integration	Asynchronous integration
Optimal use	Best for delivering real-time information	Best for performing work in batch operations
Response	Oracle Integration sends a response after the integration finishes running, so you know the integration succeeded	Oracle Integration sends a response immediately after receiving the request, so you know Oracle Integration received the request The response is an HTTP 202 Accepted message You don't receive a response when the integration finishes running unless you configure an invoke to provide a notification
Run timing	The integration runs immediately	Oracle Integration places the integration in a queue
Payload	 The response includes a payload	 The response doesn't include a payload
Guaranteed delivery of requests	 The integration doesn't guarantee the delivery of its requests	 The integration guarantees delivery of its requests
Long-running tasks	 The integration cannot perform long-running tasks	 The integration can perform long-running tasks
Subscribe to events	 The integration cannot subscribe to events	 The integration can subscribe to events
Rerun after failure	 You cannot rerun the integration if it fails	 You can rerun the integration if it fails

About Schedule Integrations

A schedule integration starts moving data when you tell it to, either on the day and time of your choice or on demand. You can specify a regular cadence for the integration, such as a daily, twice daily, weekly, or any schedule that your business requires.

When to Use Schedule Integrations

A schedule integration solves a number of business problems. For example:

- Every day at 5 PM, synchronize your inventory in all of your applications.

Copyright © 2022, 2026, Oracle and/or its affiliates.

- Four business days before every pay date, run payroll.

When Not to Use Schedule Integrations

Don't use a schedule integration to get real-time data. If you need real-time data, create an application integration that is synchronous.

About Event Integrations

An event integration subscribes to events that you create and publish in a separate integration. You create the events you want to publish and subscribe to in JSON-formatted or XML schema files.

Advantages of Event Integrations

- Integrations are decoupled to make them easier to maintain. Event messages move between different integrations without the integrations being aware of each other.
- Integrations are easily scalable. You can add additional subscribers to an existing event without concern for interrupting the existing event flow between publishers and subscribers.
- Events are made available to multiple integrations for consumption, allowing applications across your organization to simultaneously receive relevant payload.
- Custom headers can be defined in events, meaning that if the filter condition defined for the header is satisfied at runtime, the subscribing integration is triggered. If the filter condition is not satisfied, the integration is not triggered.

Event Design Process

You follow a simple design process for events in Oracle Integration:

- Create an event in JSON or XML schema format. You can optionally define custom headers in events, meaning that if the filter condition defined is satisfied at runtime, the subscribing integration is triggered.
- Create a publishing integration (application or schedule) in which you add a **Publish event** action for selecting the event.
- Create an event integration to subscribe to the event.

Learn More About Event Integrations

See Create Event Integrations.

Understand Hybrid Integrations

You can configure access to resources residing in an on-premises network or private resources that are available in your virtual cloud network (VCN) in Oracle Integration.

Topics:

- About Connecting to On-Premises Applications
- About Connecting to Private Resources

Copyright © 2022, 2026, Oracle and/or its affiliates.

About Connecting to On-Premises Applications

Use the connectivity agent to configure access to resources residing in an on-premises network in Oracle Integration.

Topics:

- About Creating Hybrid Integrations
- About the Connectivity Agent
- Connection Patterns for Hybrid Integrations

About Creating Hybrid Integrations

Many business use cases require integration between applications hosted on *public cloud* and resources residing in an *on-premises network* or *private cloud*. To facilitate such hybrid integrations, Oracle Integration provides the necessary infrastructure and architecture patterns.

For example, consider a business case where a quote or sales order configured through an Oracle Configure Price Quote (CPQ) application must be sent to an Oracle E-Business Suite application, hosted in an on-premises network, for creation and fulfillment of the sales order.

Capabilities of Hybrid Integrations

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

- Access SOAP/REST endpoints exposed by applications such as Oracle E-Business Suite, Siebel, and JD Edwards, and any on-premises home grown SOAP/REST APIs
- Access non-HTTP-based endpoints such as databases, JMS, AQ, local file systems, SAP, and others

Use Cases

These 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
- Synchronize customers that are added/updated in an on-premises SAP application with SaaS applications such as Oracle CX Sales and B2B Service Adapter, Oracle CPQ, Oracle Service Cloud, and Salesforce.com

Use the Connectivity Agent in Hybrid Integrations

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

Connection Patterns

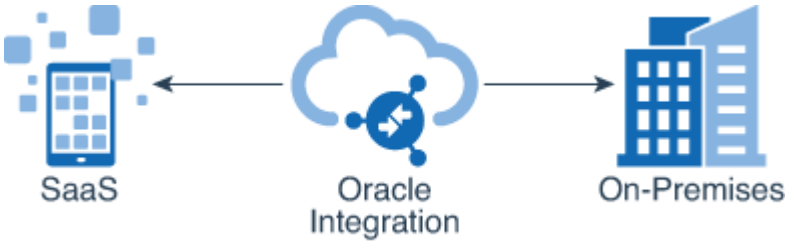
For different connection patterns you can use to create hybrid integrations, see Connection Patterns for Hybrid Integrations.

Copyright © 2022, 2026, Oracle and/or its affiliates.

About the Connectivity Agent

Using the connectivity agent, you can create hybrid integrations and exchange messages between applications in private or on-premises networks and Oracle Integration. The connectivity agent framework extends reach of Oracle Integration to non-internet accessible systems. The connectivity agent also provides multithreading support, which allows for multiple executors to perform downstream message processing.

For supported connectivity agent message payload sizes, see Service Limits in *Provisioning and Administering Oracle Integration 3*.



Connectivity Agent Components

The connectivity agent consists of the following components.

Component	Description
Cloud agent	This agent is installed and runs in Oracle Integration and supports communication with on-premises applications.
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. A connectivity agent can support multiple target systems. An Oracle Integration instance can support multiple agent groups, each with up to two agents, in a cloud/on premises topology. The on-premises connectivity agent establishes connections with the cloud agent for message processing.

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.

Copyright © 2022, 2026, Oracle and/or its affiliates.

Feature	More information
Network	• The connectivity agent opens no ports on the on-premises system for communication. • 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.
Communications security	• All communication is secured using TLS. • The on-premises connectivity agent registers with Oracle Integration over TLS using OAuth with the connectivity agent-specific authentication generated by Oracle Integration. • The on-premises connectivity agent processes requests by pulling messages from Oracle Integration across TLS. • The on-premises connectivity agent posts responses by pushing messages to Oracle Integration across TLS.
Access to the connectivity agent	Depending on your setup, either the Service Gateway or a NAT Gateway provides access to Oracle Integration and other resources. This setup is different from Oracle Integration Generation 2 and is a requirement of OAuth 2.0. • The **Service Gateway** on the Oracle VCN provides access to identity domains **in the same region** as the connectivity agent. See Access to Oracle Services: Service Gateway. • A **NAT Gateway** provides access to identity domains **in a different region** than the connectivity agent. See NAT Gateway.
Requests	• The on-premises connectivity agent checks for work by making outbound requests through the firewall. • All communication is initiated by the on-premises connectivity agent. Every interaction that the connectivity agent has with Oracle Integration follows a prescribed paradigm in which the connectivity agent pulls information from your Oracle Integration instance.
Connection details	The on-premises connectivity agent connections are configured by the agent retrieving the configuration details from Oracle Integration.
Adapter support	Adapters such as the Oracle E-Business Suite Adapter, Oracle JD Edwards EnterpriseOne Adapter, Oracle Utilities Adapter, and Oracle Siebel Adapter even when enabled with the connectivity agent invoke Oracle Integration for direct message processing when invoking integrations.
Data security	No data is persisted in the on-premises agent.
Additional details	• The on-premises connectivity agent does not support trigger/polling for REST/SOAP endpoints. • No existing J2EE container is required to deploy the on-premises connectivity agent.

Adapter Connections that Work with the Connectivity Agent

Any callbacks using REST/SOAP do not go through agent, but go directly to Oracle Integration. The on-premises connectivity 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:

  - Amazon Simple Notification Service (SNS) Adapter
  - Amazon Simple Queue Service (SQS) Adapter

Copyright © 2022, 2026, Oracle and/or its affiliates.

- Amazon Simple Storage Service (S3) Adapter
- Apache Kafka Adapter
- ArcGIS (ESRI) Adapter
- Microsoft Entra ID Adapter
- Azure Event Grid Adapter
- Azure Service Bus Adapter
- Azure Storage Adapter
- Confluent Adapter
- File Adapter
- FTP Adapter
- GCP Pub Sub Adapter
- GCP Storage Adapter
- GraphQL Adapter
- IBM DB2 Adapter
- IBM MQ Series JMS Adapter
- MLLP Adapter
- MySQL Adapter
- Netezza Adapter
- OData Adapter
- Oracle Advanced Queuing (AQ) Adapter
- Oracle Advanced Queuing (AQ) JMS Adapter
- Oracle Autonomous Data Warehouse Adapter
- Oracle Autonomous Transaction Processing Adapter
- Oracle Database Adapter
- Oracle Database Cloud Service Adapter
- Oracle E-Business Suite Adapter
- Oracle JD Edwards EnterpriseOne Adapter
- Oracle Primavera P6 EPPM Adapter
- Oracle Siebel Adapter
- Oracle SOA Suite Adapter
- Oracle Utilities Adapter
- Oracle WebLogic JMS Adapter
- PostgreSQL Adapter
- REST Adapter
- SAP Adapter
- SAP ASE (Sybase) Adapter
- Snowflake Adapter
- SOAP Adapter

Copyright © 2022, 2026, Oracle and/or its affiliates.

- **Inbound (trigger) adapters**

  The following adapters can be configured as trigger connections in an integration. Trigger adapter use with the connectivity agent is accomplished with polling or direct message processing:

  - Polling:
    - Apache Kafka Adapter
    - Microsoft Entra ID Adapter
    - Azure Service Bus Adapter
    - Confluent Adapter
    - File Adapter
    - IBM DB2 Adapter
    - IBM MQ Series JMS Adapter
    - Microsoft SQL Server Adapter
    - MySQL Adapter
    - Netezza Adapter
    - Oracle Advanced Queuing (AQ) Adapter
    - Oracle Autonomous Data Warehouse Adapter
    - Oracle Autonomous Transaction Processing Adapter
    - Oracle Database Adapter
    - Oracle Database Cloud Service Adapter
    - Oracle WebLogic JMS Adapter
    - PostgreSQL Adapter
    - SAP Adapter
    - SAP ASE (Sybase) Adapter
  - Direct message processing:
    - Oracle E-Business Suite Adapter
    - Oracle JD Edwards EnterpriseOne Adapter
    - Oracle Siebel Adapter
    - Oracle Utilities Adapter

Differences from Oracle Integration Generation 2

If you upgraded from Oracle Integration Generation 2 to Oracle Integration 3, be aware of how the connectivity agent has changed.

Copyright © 2022, 2026, Oracle and/or its affiliates.

Area	More information
OAuth 2.0 token based authentication, not basic authentication	Oracle Integration Generation 2 supported basic authentication for the connectivity agent. Oracle Integration 3 does not support basic authentication for the connectivity agent. Instead, Oracle Integration 3 supports only OAuth 2.0 token-based authentication, which is more secure than basic authentication. OAuth 2.0 requires a service gateway or NAT gateway to provide access to certain resources. See Network Requirements.
Simpler configuration	The configuration of the connectivity agent is simpler in Oracle Integration 3 than in Oracle Integration Generation 2.

Connection Patterns for Hybrid Integrations

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

> **Note**
>
> You are responsible for configuring your own network routing rules. If you don't know how to configure these rules, contact your network team for assistance.

You can set up a connection over the public internet or select to configure an exclusive connection using FastConnect, which provides a faster, more reliable networking experience compared to the internet. You'll use the connectivity agent to communicate with Oracle Integration irrespective of the connection pattern you select; employing FastConnect only ensures that the traffic between your private (on-premises) network and Oracle Integration doesn't go over the public internet and remains private.

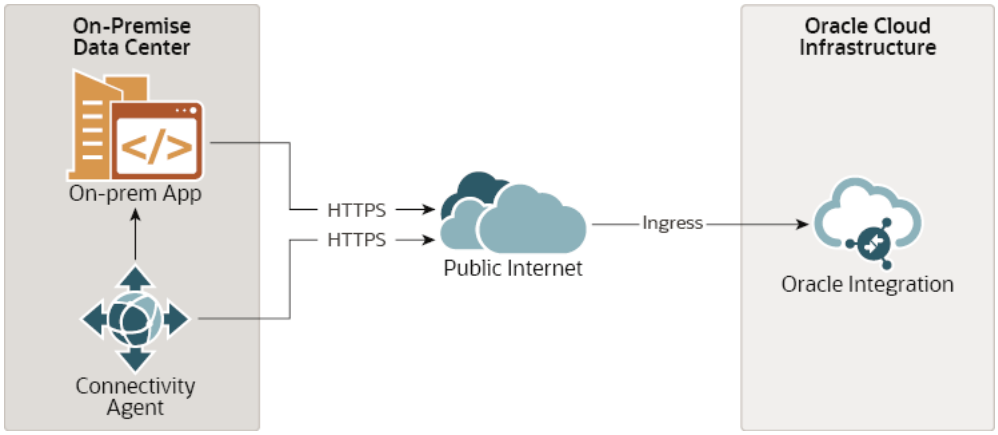
Public Internet Pattern

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

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

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

Copyright © 2022, 2026, Oracle and/or its affiliates.



Oracle Cloud Infrastructure Only - Virtual Cloud Network Pattern

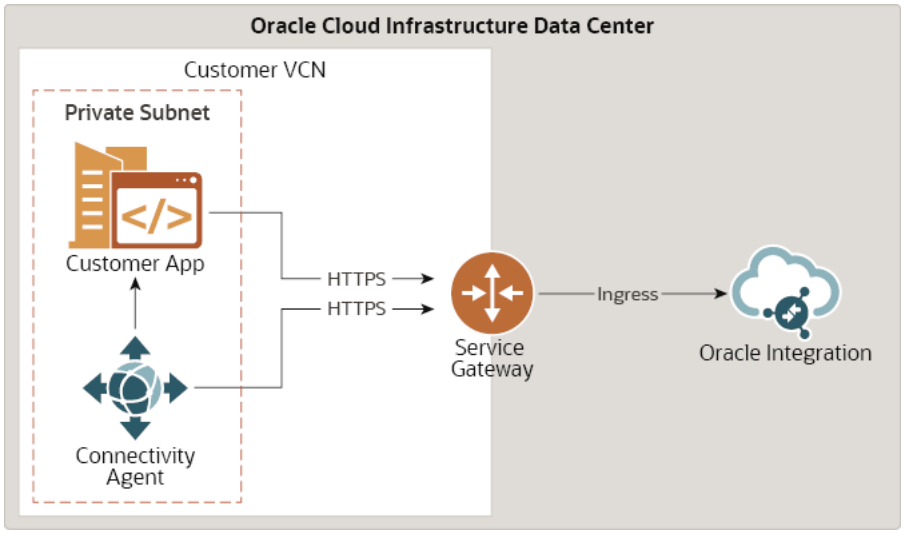
Install the connectivity agent in your Virtual Cloud Network (VCN) within Oracle Cloud Infrastructure, and configure a service gateway to route the traffic from the VCN to Oracle Integration.

Use this pattern if you have applications, such as Oracle E-Business Suite, running in a private subnet within Oracle Cloud Infrastructure. In this case, all traffic is routed locally and the public internet is not involved.

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

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

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



Copyright © 2022, 2026, Oracle and/or its affiliates.

FastConnect Public Peering Pattern

Install the connectivity agent on your private (on-premises) network, and set up an exclusive connection between your network and Oracle Integration using a FastConnect public peering link. The inbound and outbound traffic to Oracle Integration goes through the FastConnect link.

This connection pattern provides a faster and more reliable networking experience compared to the public internet pattern.

Follow the steps listed here to configure this pattern:

1. Subscribe to FastConnect with the public peering option. Currently, Oracle Integration directly supports only public peering with FastConnect. If you want to use the private peering option, you'll need to additionally use a VCN and a service gateway. See the FastConnect Private Peering Patterns.

   For detailed information on requirements and best practices for setting up an Oracle Cloud Infrastructure FastConnect, see FastConnect.

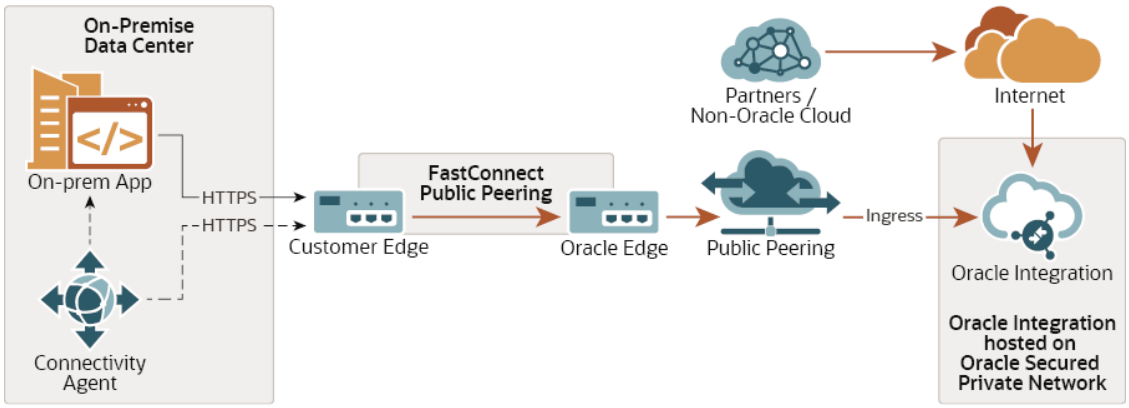
2. Configure your private (on-premises) network to route traffic through FastConnect.

   The FastConnect link contains the public IP addresses of Oracle Integration.

3. Finally, configure the connectivity agent to handle the outbound traffic from Oracle Integration to the on-premises application.

   The connectivity agent also acts as a client to FastConnect and uses public peering.

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



FastConnect Private Peering Patterns

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

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

Copyright © 2022, 2026, Oracle and/or its affiliates.

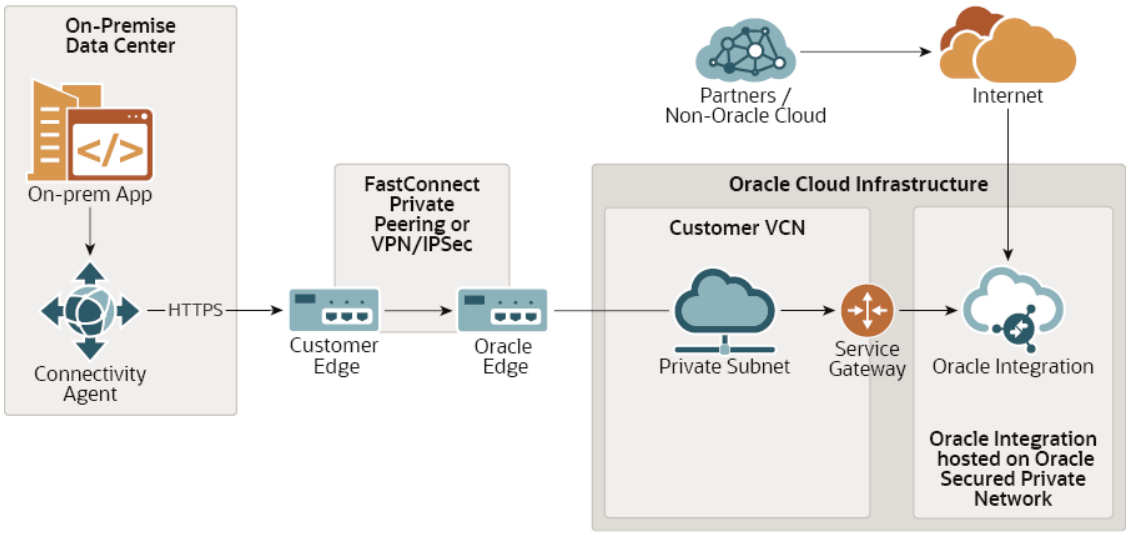
Connectivity Agent Deployed in Private or On-Premises Network

Install the connectivity agent in your private (on-premises) network, and set up a private connection between your network and VCN using FastConnect private peering or VPN. In addition, configure a service gateway to route the traffic from the VCN to Oracle Integration.

To use FastConnect, you must first subscribe to FastConnect with the private peering option. See FastConnect. The FastConnect link contains the private IP addresses of the VCN.

If you want to use VPN, see VPN Connect.

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



Connectivity Agent Deployed in a VCN

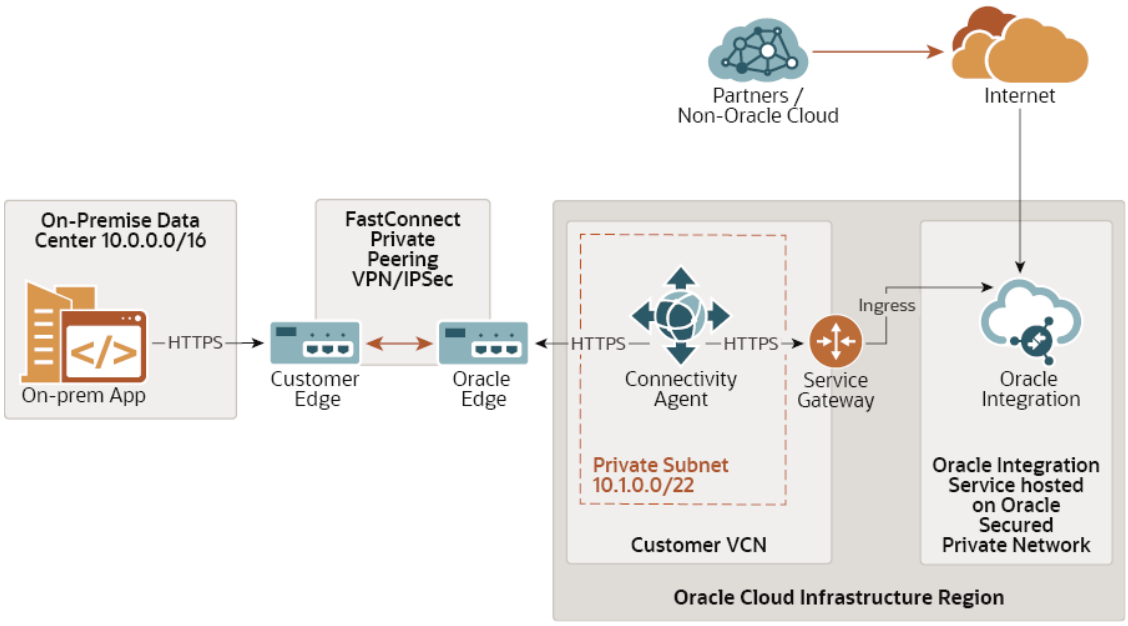
Install the connectivity agent in your VCN within Oracle Cloud Infrastructure, and set up a private connection between your network and the connectivity agent using FastConnect private peering or VPN. In addition, configure a service gateway to route the traffic from the VCN to Oracle Integration.

You can use this pattern if you have limited capacity or resource constraints on your data center.

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

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

Copyright © 2022, 2026, Oracle and/or its affiliates.



About Connecting to Private Resources

You can configure access to private resources available in your virtual cloud network (VCN) in Oracle Integration.

Outbound Private Connection Pattern

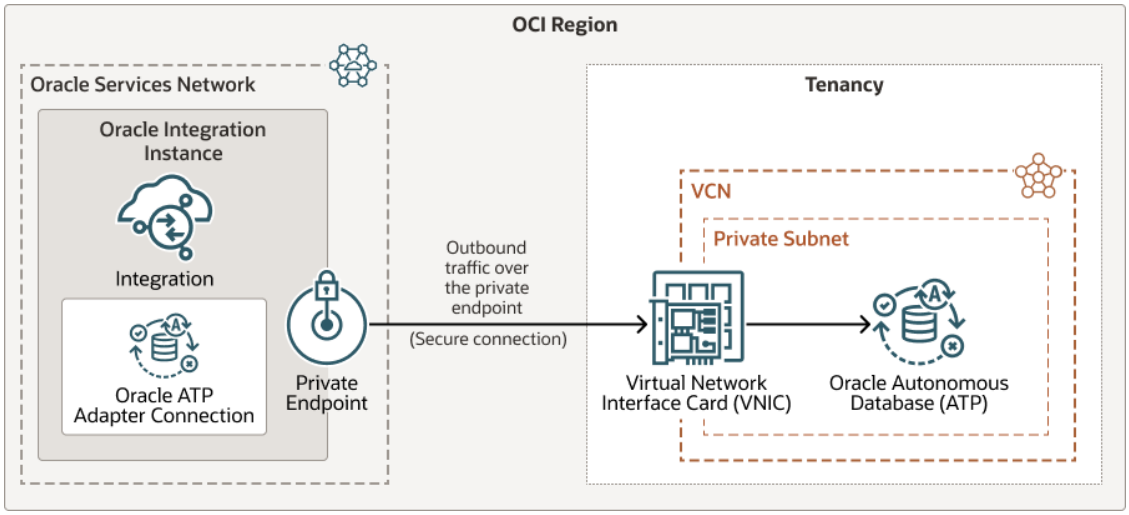
An outbound private connection pattern uses a private endpoint to connect to private resources that are in your virtual cloud network (VCN).

Create a virtual cloud network (VCN) and subnet, and add private resources to the subnet. Next, configure a private endpoint.

Outbound traffic from Oracle Integration goes through a private channel that is set up within the Oracle Cloud and connects to the private resources in the subnet. The traffic never goes through the public internet.

To get started, see Connect to Private Resources in *Provisioning and Administering Oracle Integration 3*.

Copyright © 2022, 2026, Oracle and/or its affiliates.



Adapters that Support Connecting to Private Endpoints

You can configure access to private endpoints in Oracle Integration with the following adapters. You select the **Private endpoint** option in the **Access type** section of the Connections page.

- Apache Kafka Adapter
- AS2 Adapter
- eBay Marketplace Adapter
- FTP Adapter
- GraphQL Adapter
- HubSpot Adapter
- Jira Adapter
- Magento Commerce Cloud Adapter
- Marketo Adapter
- Microsoft Dynamics 365 CRM Adapter
- OData Adapter
- OpenSearch Adapter
- Oracle Advanced Queuing (AQ) Adapter
- Oracle Advanced Queuing (AQ) JMS Adapter
- Oracle Autonomous Data Warehouse Adapter
- Oracle Autonomous Transaction Processing Adapter
- Oracle Cloud Infrastructure Streaming Service Adapter
- Oracle CPQ Adapter
- Oracle CX Sales and B2B Service Adapter
- Oracle Database Cloud Service Adapter
- Oracle E-Business Suite Adapter
- Oracle ERP Cloud Adapter
- Oracle Fusion Field Service Adapter

Copyright © 2022, 2026, Oracle and/or its affiliates.

- Oracle HCM Cloud Adapter
- Oracle Intelligent Advisor Adapter
- Oracle Primavera Cloud Adapter
- Oracle Primavera P6 EPPM Adapter
- Oracle Service Cloud (RightNow) Adapter
- Oracle Utilities Adapter
- PayPal Adapter
- QuickBooks Adapter
- Oracle Responsys Adapter
- RabbitMQ Adapter
- REST Adapter
- Salesforce Adapter
- Salesforce Commerce Cloud Adapter
- Salesforce REST Adapter (inbound and outbound directions)
- SAP Ariba Adapter
- SAP Commerce Cloud (Hybris) Adapter
- SAP Concur Adapter
- SAP SuccessFactors Adapter
- SAP S/4HANA Cloud Adapter
- ServiceNow Adapter
- Shopify Adapter
- Shopify GraphQL Adapter
- SOAP Adapter
- SugarCRM Adapter
- Workday Adapter
- Zendesk Adapter

Requirements for Connecting to a Private Endpoint

When configuring access to a private endpoints in Oracle Integration, keep in mind several requirements.

- To connect to private endpoints, you must complete prerequisite tasks in the Oracle Cloud Console. Failure to do so results in errors when testing the connection. See Connect to Private Resources in *Provisioning and Administering Oracle Integration 3*.
- When configuring an adapter on the Connections page to connect to endpoints using a private network, specify the fully-qualified domain name (FQDN) and *not* the IP address. If you enter an IP address, validation fails when you click **Test**.

Copyright © 2022, 2026, Oracle and/or its affiliates.

Integration Concepts

In every integration, you define information about the applications you are integrating and specify the source and destination of the data that the integration moves. Additionally, you define the logic that controls the flow of data along with the mapping of data between different applications.

Topics:

- About Integrations
- About Connections
- Available Adapters for Connections
- Understand Trigger and Invoke Connections
- About Mappings
- About Lookups
- About Packages
- About Business Identifiers for Tracking Fields in Messages
- About Error Management

About Integrations

Integrations improve the efficiency and effectiveness of your business by sending data to applications and getting data from applications, seamlessly and without human intervention. Consider all the manual interventions that keep your business processes flowing as expected. An integration removes that manual work, thereby simplifying and streamlining your operations.

How Integrations Work

An integration moves data from one application to another. For the integration to pull data from one application and push the data to another application, the two applications' REST APIs must communicate with each other. An integration allows this communication to occur.

Moving Data, One Step at a Time

To understand an integration, think of an assembly line where an unfinished product moves from station to station in a specific order. Each station has special tools that build the product.

Of course, an integration is different from an assembly line. While an assembly line builds a product, an integration moves data between applications. However, both an assembly line and an integration follow a series of steps in a specific order. An integration follows these steps so that it can move data. You specify the steps and their order when you build the integration.

Related Components

In every integration, you specify how to communicate to applications using connections. Additionally, you specify how data is moved between applications by creating one or more mappings. When an integration needs to relate different terms used by different applications for the same item, such as a numeric country code used in one application and an alphabetic country code is used in another, you can use a lookup to relate the values.

See About Connections, About Mappings, and About Lookups.

Copyright © 2022, 2026, Oracle and/or its affiliates.

Save Time with Recipes and Accelerators

Oracle offers several time-saving features for integrations, including recipes and accelerators. Recipes and accelerators are custom built for specific business needs and include everything you need to build an integration, including mappings and lookups. Before you start designing an integration, review the library of recipes and accelerators and see if they can help you jump start your integration work. See Recipes and Accelerators on the Oracle Help Center.

About Connections

A connection tells your integration how to connect to an application and what to do after connecting. Most integrations contain at least one connection.

Key Points for Connections

1. Every connection is based on an adapter.
2. Two types of connections are available:
   - Trigger connections run the integration and pull data from a source application.
   - Invoke connections send data to a target application.
3. To connect to an application from an integration, you must perform the following steps:
   a. Create a connection to an application.
   b. Add the connection to an integration.

Keep reading for more details.

Connections Are Based on Adapters

When you create a connection, you choose the adapter to base the connection on. An adapter provides an interface so that an integration can connect with an application.

Oracle Integration provides an always growing library of application-specific adapters, such as Oracle CX Sales and B2B Service Adapter and the Oracle Service Cloud (RightNow) Adapter.

Additionally, some adapters let you connect to just about any product's REST APIs, even if an adapter isn't available for the product. These adapters are the most popular and include:

- REST Adapter
- SOAP Adapter
- FTP Adapter

For a list of all available adapters, see Adapters on the Oracle Help Center.

Each Adapter Requires Different Information

After choosing an adapter, you specify the information that the adapter requires. For example, for a connection based on the Oracle Service Cloud (RightNow) Adapter, you must specify the WSDL URL, security policy, and security credentials to connect to it.

Copyright © 2022, 2026, Oracle and/or its affiliates.

> **Tip**
>
> If the connection is based on the REST Adapter, you can test the integration without having to deploy it. See Test REST Adapter Trigger Connection-Based Integrations in Projects.

Two-Step Process for Connections

An integration sends or receives data for one or more applications using connections. For example, a connection based on the Oracle Service Cloud (RightNow) Adapter can send and receive data from Oracle Service Cloud (RightNow). Adding a connection to an integration is a multi-step process:

1. Create the connection.

   Specify the adapter and the role of the connection, such as trigger connection, invoke connection, or trigger and invoke connection. Depending on the adapter and role you choose, you must provide additional information. For example, if you choose a database adapter, enter the IP address for the database, the user name and password for connecting to the database, and the security policy for securing the data. The role that you choose determines the security policies that are available for the connection.

   After creating the connection, test it to ensure that it connects to the application.

   See Add and Share a Connection Across a Project and Understand Trigger and Invoke Connections.

2. Configure the connection for use in an integration.

   You can use each connection in as many integrations as you want, and you can use the connection to move different data.

   For example, consider ten integrations that send data to Oracle ERP Cloud. All the integrations use the same invoke connection that is based on the Oracle ERP Cloud Adapter. However, each connection sends different data. When you add a connection to an integration, you choose the data to move in the source application and specify how it moves into the target application. If you need to process the data, you specify that information, too. Examples of processing tasks include the following:

   - Define the type of operations to perform on the data.
   - Specify the business objects and fields against which to perform the operations.
   - Define the required schemas.

   See Configure Connection Properties and Security Properties.

Typical Components of Integrations

Most integrations contain at least one of each of the following components:

- **Trigger connection**

  A trigger connection can be a scheduled trigger that is based on a schedule, or it can be a trigger that is based on a connection you define. A trigger connection does two things: It starts running the integration, and it pulls data from a source application.

  For example, consider an integration that automates the creation of an order when a customer approves an opportunity, such as a quote. The integration connects to Salesforce using the Salesforce Adapter. The connection is a trigger connection, and it pulls opportunity data from Salesforce.

Copyright © 2022, 2026, Oracle and/or its affiliates.

Every integration must contain a trigger connection.

- **Invoke connection**

  Create an invoke connection, also called a target connection, when a target application must receive data from the source application using the integration.

  Most integrations contain at least one invoke connection. For example, if your integration obtains the lowest price for an item from suppliers, it uses one invoke connection for each supplier to receive this data.

  In the Salesforce opportunity-to-order scenario, Salesforce is both the source application and the target application. The integration collects opportunity information from Salesforce and uses it to automatically generate an order in Salesforce.

- **Mapping between the trigger connection and invoke connection**

  The mapping between the connections specifies how the integration transfers data from the trigger (source) application to the invoke (target) application.

  The integration uses the connections to sends request messages to applications and to receive response messages from applications. You map the request and response messages as required by your use case. For example, you can map a request message to a response message, or map two request messages.

  Most integrations contain at least one mapping.

Available Adapters for Connections

You can create connections using application-specific adapters. The list of adapters is always growing. Additionally, you can use the REST, SOAP, and FTP adapters to connect to just about any product, even if it doesn't yet have an adapter. You can also build your own adapter with the Rapid Adapter Builder.

- List of Available Adapters
- Build Your Own Adapter with the Rapid Adapter Builder

List of Available Adapters

Adapter	For Information
Adobe Sign Adapter	*Using the Adobe Sign Adapter with Oracle Integration 3*
Anthropic Adapter	*Using the Anthropic Adapter with Oracle Integration 3*
Apache Kafka Adapter	*Using the Apache Kafka Adapter with Oracle Integration 3*
Asana Adapter	*Using the Asana Adapter with Oracle Integration 3*
AS2 Adapter	*Using the AS2 Adapter with Oracle Integration 3*
AS4 Adapter	*Using the AS4 Adapter with Oracle Integration 3*
Amazon Simple Notification Service (SNS) Adapter	*Using the Amazon Simple Notification Service (SNS) Adapter with Oracle Integration 3*
Amazon Simple Queue Service (SQS) Adapter	*Using the Amazon Simple Queue Service (SQS) Adapter with Oracle Integration 3*
Amazon Simple Storage Service (S3) Adapter	*Using the Amazon Simple Storage Service (S3) Adapter with Oracle Integration 3*
ArcGIS (ESRI) Adapter	*Using the ArcGIS (ESRI) Adapter with Oracle Integration 3*

Copyright © 2022, 2026, Oracle and/or its affiliates.

Adapter	For Information
Microsoft Entra ID Adapter	*Using the Microsoft Entra ID Adapter with Oracle Integration 3*
Azure Event Grid Adapter	*Using the Azure Event Grid Adapter with Oracle Integration 3*
Azure Service Bus Adapter	*Using the Azure Service Bus Adapter with Oracle Integration 3*
Azure Storage Adapter	*Using the Azure Storage Adapter with Oracle Integration 3*
Box Adapter	*Using the Box Adapter with Oracle Integration 3*
Confluent Adapter	*Using the Confluent Adapter with Oracle Integration 3*
Coupa Procurement Adapter	*Using the Oracle Procurement Adapter for Oracle Integration 3*
Coupa Sourcing Adapter	*Using the Coupa Sourcing Adapter with Oracle Integration 3*
DocuSign Adapter	*Using the DocuSign Adapter with Oracle Integration 3*
eBay Marketplace Adapter	*Using the eBay Marketplace Adapter with Oracle Integration 3*
Eventbrite Adapter	*Using the Eventbrite Adapter with Oracle Integration 3*
Facebook Adapter	*Using the Facebook Adapter with Oracle Integration 3*
FHIR Adapter	*Using the FHIR Adapter with Oracle Integration 3*
File Adapter	*Using the File Adapter with Oracle Integration 3*
FTP Adapter	*Using the FTP Adapter with Oracle Integration 3*
GCP Pub Sub Adapter	*Using the GCP Pub Sub Adapter with Oracle Integration 3*
GCP Storage Adapter	*Using the GCP Storage Adapter with Oracle Integration 3*
GitHub Adapter	*Using the GitHub Adapter with Oracle Integration 3*
GraphQL Adapter	*Using the GraphQL Adapter with Oracle Integration 3*
Google Gmail Adapter	*Using the Google Gmail Adapter with Oracle Integration 3*
Google Calendar Adapter	*Using the Google Calendar Adapter with Oracle Integration 3*
Google Tasks Adapter	*Using the Google Tasks Adapter with Oracle Integration 3*
HubSpot Adapter	*Using the HubSpot Adapter with Oracle Integration 3*
IBM DB2 Adapter	*Using the IBM DB2 Adapter with Oracle Integration 3*
IBM MQ Series JMS Adapter	*Using the IBM MQ Series JMS Adapter with Oracle Integration 3*
Jira Adapter	*Using the Jira Adapter with Oracle Integration 3*
LinkedIn Adapter	*Using the LinkedIn Adapter with Oracle Integration 3*
Magento Commerce Cloud Adapter	*Using the Magento Commerce Cloud Adapter with Oracle Integration 3*

Copyright © 2022, 2026, Oracle and/or its affiliates.

Adapter	For Information
Mailchimp Adapter	*Using the Mailchimp Adapter with Oracle Integration 3*
Marketo Adapter	*Using the Marketo Adapter with Oracle Integration 3*
Microsoft Office 365 Calendar Adapter	*Using the Microsoft Office 365 Calendar Adapter with Oracle Integration 3*
Microsoft Office 365 People Adapter	*Using the Microsoft Office 365 People Adapter with Oracle Integration 3*
Microsoft Office 365 Outlook Adapter	*Using the Microsoft Office 365 Outlook Adapter with Oracle Integration 3*
Microsoft SQL Server Adapter	*Using the Microsoft SQL Server Adapter with Oracle Integration 3*
MLLP Adapter	*Using the MLLP Adapter with Oracle Integration 3*
Microsoft Dynamics 365 CRM Adapter	*Using the Microsoft Dynamics 365 CRM Adapter with Oracle Integration 3*
Microsoft SharePoint Adapter	*Using the Microsoft SharePoint Adapter with Oracle Integration 3*
MySQL Adapter	*Using the MySQL Adapter with Oracle Integration 3*
Netezza Adapter	*Using the Netezza Adapter with Oracle Integration 3*
OData Adapter	*Using the OData Adapter with Oracle Integration 3*
OpenAI Adapter	*Using the OpenAI Adapter with Oracle Integration 3*
OpenSearch Adapter	*Using the OpenSearch Adapter with Oracle Integration 3*
Oracle Aconex Cloud Adapter	*Using the Oracle Aconex Cloud Adapter with Oracle Integration 3*
Oracle Advanced Queuing (AQ) Adapter	*Using the Oracle Advanced Queuing (AQ) Adapter with Oracle Integration 3*
Oracle Advanced Queuing (AQ) JMS Adapter	*Using the Oracle Advanced Queuing (AQ) JMS Adapter with Oracle Integration 3*
Oracle Autonomous Data Warehouse Adapter	*Using the Oracle Autonomous Data Warehouse Adapter with Oracle Integration 3*
Oracle Autonomous Transaction Processing Adapter	*Using the Oracle Autonomous Transaction Processing Adapter with Oracle Integration 3*
Oracle Cloud Infrastructure Streaming Service Adapter	*Using the Oracle Cloud Infrastructure Streaming Service Adapter with Oracle Integration 3*
Oracle Commerce Cloud Adapter	*Using the Oracle Commerce Cloud Adapter with Oracle Integration 3*
Oracle CPQ Adapter	*Using the Oracle CPQ Adapter with Oracle Integration 3*
Oracle Database Adapter	*Using the Oracle Database Adapter with Oracle Integration 3*
Oracle Database Cloud Service Adapter	*Using the Oracle Database Cloud Service Adapter with Oracle Integration 3*
Oracle E-Business Suite Adapter	*Using the Oracle E-Business Suite Adapter with Oracle Integration 3*
Oracle Eloqua Cloud Adapter	*Using the Oracle Eloqua Cloud Adapter with Oracle Integration 3*
Oracle CX Sales and B2B Service Adapter	*Using the Oracle CX Sales and B2B Service Adapter with Oracle Integration 3*

Copyright © 2022, 2026, Oracle and/or its affiliates.

Adapter	For Information
Oracle Enterprise Performance Management Cloud Adapter	*Using the Oracle Enterprise Performance Management Cloud Adapter with Oracle Integration 3*
Oracle ERP Cloud Adapter	*Using the Oracle ERP Cloud Adapter with Oracle Integration 3*
Oracle Fusion Field Service Adapter	*Using the Oracle Fusion Field Service Adapter with Oracle Integration 3*
Oracle HCM Cloud Adapter	*Using the Oracle HCM Cloud Adapter with Oracle Integration 3*
Oracle Health Inventory Management Adapter	*Using the Oracle Health Inventory Management Adapter with Oracle Integration 3*
Oracle Hospitality Adapter	*Using the Oracle Hospitality Adapter with Oracle Integration 3*
Oracle Intelligent Advisor Adapter	*Using the Oracle Intelligent Advisor Adapter with Oracle Integration 3*
Oracle Intelligent Track and Trace Adapter	*Using the Oracle Intelligent Track and Trace Adapter with Oracle Integration 3*
Oracle JD Edwards EnterpriseOne Adapter	*Using the Oracle JD Edwards EnterpriseOne Adapter with Oracle Integration 3*
Oracle Logistics Adapter	*Using the Oracle Logistics Adapter with Oracle Integration 3*
Oracle NetSuite Adapter	*Using the Oracle NetSuite Adapter with Oracle Integration 3*
Oracle NetSuite RST Adapter	*Using the Oracle NetSuite RST Adapter with Oracle Integration 3*
Oracle Primavera Cloud Adapter	*Using the Oracle Primavera Cloud Adapter with Oracle Integration 3*
Oracle Primavera Cloud Data Service Adapter	*Using the Oracle Primavera Cloud Data Service Adapter with Oracle Integration 3*
Oracle Primavera P6 EPPM Adapter	*Using the Oracle Primavera P6 EPPM Adapter with Oracle Integration 3*
Oracle Responsys Adapter	*Using the Oracle Responsys Adapter with Oracle Integration 3*
Oracle Service Cloud (RightNow) Adapter	*Using the Oracle Service Cloud (RightNow) Adapter with Oracle Integration 3*
Oracle Siebel Adapter	*Using the Oracle Siebel Adapter with Oracle Integration 3*
Oracle SOA Suite Adapter	*Using the Oracle SOA Suite Adapter with Oracle Integration 3*
Oracle Taleo Enterprise Edition Adapter	*Using the Oracle Taleo Enterprise Edition Adapter with Oracle Integration 3*
Oracle Unity Adapter	*Using the Oracle Unity Adapter with Oracle Integration*
Oracle Utilities Adapter	*Using the Oracle Utilities Adapter with Oracle Integration 3*
Oracle WebLogic JMS Adapter	*Using the Oracle WebLogic JMS Adapter with Oracle Integration 3*
PayPal Adapter	*Using the PayPal Adapter with Oracle Integration 3*
PostgreSQL Adapter	*Using the PostgreSQL Adapter with Oracle Integration 3*
Primavera Unifier Adapter	*Using the Oracle Primavera Unifier Adapter with Oracle Integration 3*

Copyright © 2022, 2026, Oracle and/or its affiliates.

Adapter	For Information
RabbitMQ Adapter	*Using the RabbitMQ Adapter with Oracle Integration 3*
REST Adapter	*Using the REST Adapter with Oracle Integration 3*
RosettaNet Adapter	*Using the RosettaNet Adapter with Oracle Integration 3*
Salesforce Adapter	*Using the Salesforce Adapter with Oracle Integration 3*
Salesforce Commerce Cloud Adapter	*Using the Salesforce Commerce Cloud Adapter with Oracle Integration 3*
Salesforce Marketing Cloud Adapter	*Using the Salesforce Marketing Cloud Adapter with Oracle Integration 3*
Salesforce REST Adapter	*Using the Salesforce REST Adapter with Oracle Integration 3*
SAP Adapter	*Using the SAP Adapter with Oracle Integration 3*
SAP Ariba Adapter	*Using the SAP Ariba Adapter with Oracle Integration 3*
SAP ASE (Sybase) Adapter	*Using the SAP ASE (Sybase) Adapter with Oracle Integration 3*
SAP Commerce Cloud (Hybris) Adapter	*Using the SAP Commerce Cloud (Hybris) Adapter with Oracle Integration 3*
SAP Concur Adapter	*Using the SAP Concur Adapter with Oracle Integration 3*
SAP SuccessFactors Adapter	*Using the SAP SuccessFactors Adapter with Oracle Integration 3*
SAP S/4HANA Cloud Adapter	*Using the SAP S/4HANA Cloud Adapter with Oracle Integration 3*
ServiceNow Adapter	*Using the ServiceNow Adapter with Oracle Integration 3*
Shopify Adapter	*Using the Shopify Adapter with Oracle Integration 3*
Shopify GraphQL Adapter	*Using the Shopify GraphQL Adapter with Oracle Integration 3*
Slack Adapter	*Using the Slack Adapter with Oracle Integration 3*
Snowflake Adapter	*Using the Snowflake Adapter with Oracle Integration 3*
SOAP Adapter	*Using the SOAP Adapter with Oracle Integration 3*
SugarCRM Adapter	*Using the SugarCRM Adapter with Oracle Integration 3*
SurveyMonkey Adapter	*Using the SurveyMonkey Adapter with Oracle Integration 3*
Trello Adapter	*Using the Trello Adapter with Oracle Integration 3*
Twilio Adapter	*Using the Twilio Adapter with Oracle Integration 3*
Twitter Adapter	*Using the Twitter Adapter with Oracle Integration 3*
Workday Adapter	*Using the Workday Adapter with Oracle Integration 3*
Zendesk Adapter	*Using the Zendesk Adapter with Oracle Integration 3*
Zuora Adapter	*Using the Zuora Adapter with Oracle Integration 3*

Copyright © 2022, 2026, Oracle and/or its affiliates.

Build Your Own Adapter with the Rapid Adapter Builder

Not finding the adapter you need in the list? You can build your own adapter with the Rapid Adapter Builder.

The Rapid Adapter Builder transforms the adapter experience in Oracle Integration by allowing you to build an adapter for any application that exposes REST APIs.

Oracle provides a Visual Studio Code extension for the Rapid Adapter Builder. Use the extension to develop, validate, and publish an adapter to an Oracle Integration instance.

See Learn About the Rapid Adapter Builder in Oracle Integration in *Using the Rapid Adapter Builder with Oracle Integration 3*.

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. You invoke the Adapter Endpoint Configuration Wizard to guide you through configuration of this adapter connection.

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

> **Note**
>
> A trigger adapter connection is not required in schedule integrations. The schedule acts as the trigger in schedule integrations.

Trigger Connections

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

Adapter	For Information
Amazon Simple Notification Service (SNS) Adapter	*Using the Amazon Simple Notification Service (SNS) Adapter with Oracle Integration 3*
Amazon Simple Queue Service (SQS) Adapter	*Using the Amazon Simple Queue Service (SQS) Adapter with Oracle Integration 3*
Apache Kafka Adapter	*Using the Apache Kafka Adapter with Oracle Integration 3*
ArcGIS (ESRI) Adapter	*Using the ArcGIS (ESRI) Adapter with Oracle Integration 3*

Copyright © 2022, 2026, Oracle and/or its affiliates.

Adapter	For Information
AS2 Adapter	*Using the AS2 Adapter with Oracle Integration 3*
Microsoft Entra ID Adapter	*Using the Microsoft Entra ID Adapter with Oracle Integration 3*
Azure Event Grid Adapter	*Using the Azure Event Grid Adapter with Oracle Integration 3*
Azure Service Bus Adapter	*Using the Azure Service Bus Adapter with Oracle Integration 3*
Box Adapter	*Using the Box Adapter with Oracle Integration 3*
Confluent Adapter	*Using the Confluent Adapter with Oracle Integration 3*
File Adapter	*Using the File Adapter with Oracle Integration 3*
GCP Pub Sub Adapter	*Using the GCP Pub Sub Adapter with Oracle Integration 3*
GitHub Adapter	*Using the GitHub Adapter with Oracle Integration 3*
HubSpot Adapter	*Using the HubSpot Adapter with Oracle Integration 3*
IBM DB2 Adapter	*Using the IBM DB2 Adapter with Oracle Integration 3*
IBM MQ Series JMS Adapter	*Using the IBM MQ Series JMS Adapter with Oracle Integration 3*
Jira Adapter	*Using the Jira Adapter with Oracle Integration 3*
Microsoft SQL Server Adapter	*Using the Microsoft SQL Server Adapter with Oracle Integration 3*
MLLP Adapter	*Using the MLLP Adapter with Oracle Integration 3*
Using the Microsoft Dynamics 365 CRM Adapter with Oracle Integration 3	*Using the Microsoft Dynamics 365 CRM Adapter with Oracle Integration 3*
Microsoft SharePoint Adapter	*Using the Microsoft SharePoint Adapter with Oracle Integration 3*
MySQL Adapter	*Using the MySQL Adapter with Oracle Integration 3*
Netezza Adapter	*Using the Netezza Adapter with Oracle Integration 3*
Oracle Advanced Queuing (AQ) Adapter	*Using the Oracle Advanced Queuing (AQ) Adapter with Oracle Integration 3*
Oracle Advanced Queuing (AQ) JMS Adapter	*Using the Oracle Advanced Queuing (AQ) JMS Adapter with Oracle Integration 3*
Oracle Autonomous Data Warehouse Adapter	*Using the Oracle Autonomous Data Warehouse Adapter with Oracle Integration 3*
Oracle Autonomous Transaction Processing Adapter	*Using the Oracle Autonomous Transaction Processing Adapter with Oracle Integration 3*
Oracle Cloud Infrastructure Streaming Service Adapter	*Using the Oracle Cloud Infrastructure Streaming Service Adapter with Oracle Integration 3*
Oracle Commerce Cloud Adapter	*Using the Oracle Commerce Cloud Adapter with Oracle Integration 3*
Oracle CPQ Adapter	*Using the Oracle CPQ Adapter with Oracle Integration* 3
Oracle Database Adapter	*Using the Oracle Database Adapter with Oracle Integration 3*
Oracle Database Cloud Service Adapter	*Using the Oracle Database Cloud Service Adapter with Oracle Integration 3*
Oracle E-Business Suite Adapter	*Using Oracle E-Business Suite Adapter with Oracle Integration 3*

Copyright © 2022, 2026, Oracle and/or its affiliates.

Adapter	For Information
Oracle Eloqua Cloud Adapter	*Using the Oracle Eloqua Cloud Adapter with Oracle Integration 3*
Oracle CX Sales and B2B Service Adapter	*Using the Oracle CX Sales and B2B Service Adapter with Oracle Integration 3*
Oracle ERP Cloud Adapter	*Using the Oracle ERP Cloud Adapter with Oracle Integration 3*
Oracle Fusion Field Service Adapter	*Using the Oracle Fusion Field Service Adapter with Oracle Integration 3*
Oracle HCM Cloud Adapter	*Using the Oracle HCM Cloud Adapter with Oracle Integration 3*
Oracle Hospitality Adapter	*Using the Oracle Hospitality Adapter with Oracle Integration 3*
Oracle JD Edwards EnterpriseOne Adapter	*Using the JD Edwards EnterpriseOne Adapter with Oracle Integration 3*
Oracle Intelligent Advisor Adapter	*Using the Oracle Intelligent Advisor Adapter with Oracle Integration 3*
Oracle Logistics Adapter	*Oracle Logistics Adapter with Oracle Integration 3*
Oracle Service Cloud (RightNow) Adapter	*Using the Oracle Service Cloud (RightNow) Adapter with Oracle Integration 3*
Oracle Siebel Adapter	*Using the Oracle Siebel Adapter with Oracle Integration 3*
Oracle WebLogic JMS Adapter	*Using the Oracle WebLogic JMS Adapter with Oracle Integration 3*
Oracle Utilities Adapter	*Using the Oracle Utilities Adapter with Oracle Integration 3*
PayPal Adapter	*Using the PayPal Adapter with Oracle Integration 3*
PostgreSQL Adapter	*Using the PostgreSQL Adapter with Oracle Integration 3*
Primavera Unifier Adapter	*Using the Primavera Unifier Adapter with Oracle Integration 3*
RabbitMQ Adapter	*Using the RabbitMQ Adapter with Oracle Integration 3*
REST Adapter	*Using the REST Adapter with Oracle Integration 3*
RosettaNet Adapter	*Using the RosettaNet Adapter with Oracle Integration 3*
Salesforce Adapter	*Using the Salesforce Adapter with Oracle Integration 3*
SAP Adapter	*Using the SAP Adapter with Oracle Integration 3*
Salesforce Marketing Cloud Adapter	*Using the Salesforce Marketing Cloud Adapter with Oracle Integration 3*
Salesforce REST Adapter	*Using the Salesforce REST Adapter with Oracle Integration 3*
SAP Adapter	*Using the SAP Adapter with Oracle Integration 3*
SAP Ariba Adapter	*Using the SAP Ariba Adapter with Oracle Integration 3*
SAP ASE (Sybase) Adapter	*Using the SAP ASE (Sybase) Adapter for Oracle Integration 3*
SAP SuccessFactors Adapter	*Using the SAP SuccessFactors Adapter with Oracle Integration 3*
ServiceNow Adapter	*Using the ServiceNow Adapter with Oracle Integration 3*

Copyright © 2022, 2026, Oracle and/or its affiliates.

Adapter	For Information
Shopify Adapter	*Using the Shopify Adapter with Oracle Integration 3*
Shopify GraphQL Adapter	*Using the Shopify GraphQL Adapter with Oracle Integration 3*
Slack Adapter	*Using the Slack Adapter with Oracle Integration 3*
SOAP Adapter	*Using the SOAP Adapter with Oracle Integration 3*
Zendesk Adapter	*Using the Zendesk Adapter with Oracle Integration 3*
Zuora Adapter	*Using the Zuora Adapter with Oracle Integration 3*

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 for each adapter connection.

Adapter	For Information
Adobe Sign Adapter	*Using the Adobe Sign Adapter with Oracle Integration 3*
Anthropic Adapter	*Using the Anthropic Adapter with Oracle Integration 3*
Apache Kafka Adapter	*Using the Apache Kafka Adapter with Oracle Integration 3*
ArcGIS (ESRI) Adapter	*Using the ArcGIS (ESRI) Adapter with Oracle Integration 3*
Asana Adapter	*Using the Asana Adapter with Oracle Integration 3*
AS2 Adapter	*Using the AS2 Adapter with Oracle Integration 3*
AS4 Adapter	*Using the AS4 Adapter with Oracle Integration 3*
Amazon Simple Notification Service (SNS) Adapter	*Using the Amazon Simple Notification Service (SNS) Adapter with Oracle Integration 3*
Amazon Simple Queue Service (SQS) Adapter	*Using the Amazon Simple Queue Service (SQS) Adapter with Oracle Integration 3*
Amazon Simple Storage Service (S3) Adapter	*Using the Amazon Simple Storage Service (S3) Adapter with Oracle Integration 3*
Azure Event Grid Adapter	*Using the Azure Event Grid Adapter with Oracle Integration 3*
Azure Service Bus Adapter	*Using the Azure Service Bus Adapter with Oracle Integration 3*
Microsoft Entra ID Adapter	*Using the Microsoft Entra ID Adapter with Oracle Integration 3*
Azure Storage Adapter	*Using the Azure Storage Adapter with Oracle Integration 3*
Box Adapter	*Using the Box Adapter with Oracle Integration 3*
Confluent Adapter	*Using the Confluent Adapter with Oracle Integration 3*
Coupa Procurement Adapter	*Using the Coupa Procurement Adapter with Oracle Integration 3*
Coupa Sourcing Adapter	*Using the Coupa Sourcing Adapter with Oracle Integration 3*

Copyright © 2022, 2026, Oracle and/or its affiliates.

Adapter	For Information
DocuSign Adapter	*Using the DocuSign Adapter with Oracle Integration 3*
eBay Marketplace Adapter	*Using the eBay Marketplace Adapter with Oracle Integration 3*
Eventbrite Adapter	*Using the Eventbrite Adapter with Oracle Integration 3*
Facebook Adapter	*Using the Facebook Adapter with Oracle Integration 3*
FHIR Adapter	*Using the FHIR Adapter with Oracle Integration 3*
File Adapter	*Using the File Adapter with Oracle Integration 3*
FTP Adapter	*Using the FTP Adapter with Oracle Integration 3*
GCP Pub Sub Adapter	*Using the GCP Pub Sub Adapter with Oracle Integration 3*
GCP Storage Adapter	*Using the GCP Storage Adapter with Oracle Integration 3*
GitHub Adapter	*Using the GitHub Adapter with Oracle Integration 3*
GraphQL Adapter	*Using the GraphQL Adapter with Oracle Integration 3*
Google Gmail Adapter	*Using the Goggle Gmail Adapter with Oracle Integration 3*
Google Calendar Adapter	*Using the Google Calendar Adapter with Oracle Integration 3*
Google Tasks Adapter	*Using the Google Tasks Adapter with Oracle Integration 3*
HubSpot Adapter	*Using the HubSpot Adapter with Oracle Integration 3*
IBM DB2 Adapter	*Using the IBM DB2 Adapter with Oracle Integration 3*
IBM MQ Series JMS Adapter	*Using the IBM MQ Series JMS Adapter with Oracle Integration 3*
Jira Adapter	*Using the Jira Adapter with Oracle Integration 3*
LinkedIn Adapter	*Using the LinkedIn Adapter with Oracle Integration 3*
Magento Commerce Cloud Adapter	*Using the Magento Commerce Cloud Adapter with Oracle Integration 3*
Mailchimp Adapter	*Using the Mailchimp Adapter with Oracle Integration 3*
Marketo Adapter	*Using the Marketo Adapter with Oracle Integration 3*
Microsoft Office 365 Calendar Adapter	*Using the Microsoft Office 365 Calendar Adapter with Oracle Integration 3*
Microsoft Office 365 People Adapter	*Using the Microsoft Office 365 People Adapter with Oracle Integration 3*
Microsoft Office 365 Outlook Adapter	*Using the Microsoft Office 365 Outlook Adapter with Oracle Integration 3*
Microsoft SQL Server Adapter	*Using the Microsoft SQL Server Adapter with Oracle Integration 3*
MLLP Adapter	*Using the MLLP Adapter with Oracle Integration 3*
Microsoft Dynamics 365 CRM Adapter	*Using the Microsoft Dynamics 365 CRM Adapter with Oracle Integration 3*

Copyright © 2022, 2026, Oracle and/or its affiliates.

Adapter	For Information
Microsoft SharePoint Adapter	*Using the Microsoft SharePoint Adapter with Oracle Integration 3*
MySQL Adapter	*Using the MySQL Adapter with Oracle Integration 3*
Netezza Adapter	*Using the Netezza Adapter with Oracle Integration 3*
OData Adapter	*Using the OData Adapter with Oracle Integration 3*
OpenAI Adapter	*Using the OpenAI Adapter with Oracle Integration 3*
OpenSearch Adapter	*Using the OpenSearch Adapter with Oracle Integration 3*
Oracle Aconex Cloud Adapter	*Using the Oracle Aconex Cloud Adapter with Oracle Integration 3*
Oracle Advanced Queuing (AQ) Adapter	*Using the Oracle Advanced Queuing (AQ) Adapter with Oracle Integration 3*
Oracle Advanced Queuing (AQ) JMS Adapter	*Using the Oracle Advanced Queuing (AQ) JMS Adapter with Oracle Integration 3*
Oracle Autonomous Data Warehouse Adapter	*Using the Oracle Autonomous Data Warehouse Adapter with Oracle Integration 3*
Oracle Autonomous Transaction Processing Adapter	*Using the Oracle Autonomous Transaction Processing Adapter with Oracle Integration 3*
Oracle Cloud Infrastructure Streaming Service Adapter	*Using the Oracle Cloud Infrastructure Streaming Service Adapter with Oracle Integration 3*
Oracle Commerce Cloud Adapter	*Using the Oracle Commerce Cloud Adapter with Oracle Integration 3*
Oracle CPQ Adapter	*Using the Oracle CPQ Adapter with Oracle Integration 3*
Oracle Database Adapter	*Using the Oracle Database Adapter with Oracle Integration 3*
Oracle Eloqua Cloud Adapter	*Using the Oracle Eloqua Cloud Adapter with Oracle Integration 3*
Oracle CX Sales and B2B Service Adapter	*Using the Oracle CX Sales and B2B Service Adapter with Oracle Integration 3*
Oracle Database Cloud Service Adapter	*Using the Oracle Database Cloud Service Adapter with Oracle Integration 3*
Oracle E-Business Suite Adapter	*Using Oracle E-Business Suite Adapter with Oracle Integration 3*
Oracle ERP Cloud Adapter	*Using the Oracle ERP Cloud Adapter with Oracle Integration 3*
Oracle Fusion Field Service Adapter	*Using the Oracle Fusion Field Service Adapter with Oracle Integration 3*
Oracle HCM Cloud Adapter	*Using the Oracle HCM Cloud Adapter with Oracle Integration 3*
Oracle Health Inventory Management Adapter	*Using the Oracle Health Inventory Management Adapter with Oracle Integration 3*
Oracle Hospitality Adapter	*Using the Oracle Hospitality Adapter with Oracle Integration 3*
Oracle Intelligent Advisor Adapter	*Using the Oracle Intelligent Advisor Adapter with Oracle Integration 3*
Oracle Intelligent Track and Trace Adapter	*Using the Oracle Intelligent Track and Trace Adapter with Oracle Integration 3*
Oracle JD Edwards EnterpriseOne Adapter	*Using the JD Edwards EnterpriseOne Adapter with Oracle Integration 3*

Copyright © 2022, 2026, Oracle and/or its affiliates.

Adapter	For Information
Oracle Logistics Adapter	*Using the Oracle Logistics Cloud Adapter with Oracle Integration 3*
Oracle NetSuite Adapter	*Using the Oracle NetSuite Adapter with Oracle Integration 3*
Oracle NetSuite RST Adapter	*Using the Oracle NetSuite RST Adapter with Oracle Integration 3*
Oracle Primavera Cloud Adapter	*Using the Oracle Primavera Cloud Adapter with Oracle Integration 3*
Oracle Primavera Cloud Data Service Adapter	*Using the Oracle Primavera Cloud Data Service Adapter with Oracle Integration 3*
Oracle Primavera P6 EPPM Adapter	*Using the Oracle Primavera P6 EPPM Adapter with Oracle Integration 3*
Oracle Responsys Adapter	*Using the Oracle Responsys Adapter with Oracle Integration 3*
Oracle Service Cloud (RightNow) Adapter	*Using the Oracle Service Cloud (RightNow) Adapter with Oracle Integration 3*
Oracle Siebel Adapter	*Using the Oracle Siebel Adapter with Oracle Integration 3*
Oracle SOA Suite Adapter	*Using the Oracle SOA Suite Adapter with Oracle Integration 3*
Oracle Taleo Enterprise Edition Adapter	*Using the Oracle Taleo Enterprise Edition Adapter with Oracle Integration 3*
Oracle Unity Adapter	*Using the Oracle Unity Adapter with Oracle Integration*
Oracle Utilities Adapter	*Using the Oracle Utilities Adapter with Oracle Integration 3*
Oracle WebLogic JMS Adapter	*Using the Oracle WebLogic JMS Adapter with Oracle Integration 3*
PayPal Adapter	*Using the PayPal Adapter with Oracle Integration 3*
PostgreSQL Adapter	*Using the PostgreSQL Adapter with Oracle Integration 3*
Primavera Unifier Adapter	*Using the Primavera Unifier Adapter with Oracle Integration 3*
QuickBooks Adapter	*Using the QuickBooks Adapter with Oracle Integration 3*
RabbitMQ Adapter	*Using the RabbitMQ Adapter with Oracle Integration 3*
REST Adapter	*Using the REST Adapter with Oracle Integration 3*
RosettaNet Adapter	*Using the RosettaNet Adapter with Oracle Integration 3*
Salesforce Adapter	*Using the Salesforce Adapter with Oracle Integration 3*
Salesforce Commerce Cloud Adapter	*Using the Salesforce Commerce Cloud Adapter with Oracle Integration 3*
Salesforce Marketing Cloud Adapter	*Using the Salesforce Marketing Cloud Adapter with Oracle Integration 3*
Salesforce REST Adapter	*Using the Salesforce REST Adapter with Oracle Integration 3*
SAP Adapter	*Using the SAP Adapter with Oracle Integration 3*
SAP Ariba Adapter	*Using the SAP Ariba Adapter with Oracle Integration 3*

Copyright © 2022, 2026, Oracle and/or its affiliates.

Adapter	For Information
SAP ASE (Sybase) Adapter	*Using the SAP ASE (Sybase) Adapter for Oracle Integration 3*
SAP Commerce Cloud (Hybris) Adapter	*Using the SAP Commerce Cloud (Hybris) Adapter with Oracle Integration 3*
SAP Concur Adapter	*Using the SAP Concur Adapter with Oracle Integration 3*
SAP SuccessFactors Adapter	*Using the SAP SuccessFactors Adapter with Oracle Integration 3*
SAP S/4HANA Cloud Adapter	*Using the S/4HANA Adapter with Oracle Integration 3*
ServiceNow Adapter	*Using the ServiceNow Adapter with Oracle Integration 3*
Shopify Adapter	*Using the Shopify Adapter with Oracle Integration 3*
Shopify GraphQL Adapter	*Using the Shopify GraphQL Adapter with Oracle Integration 3*
Slack Adapter	*Using the Slack Adapter with Oracle Integration 3*
SOAP Adapter	*Using the SOAP Adapter with Oracle Integration 3*
Snowflake Adapter	*Using the Snowflake Adapter with Oracle Integration 3*
SugarCRM Adapter	*Using the SugarCRM Adapter with Oracle Integration 3*
SurveyMonkey Adapter	*Using the SurveyMonkey Adapter with Oracle Integration 3*
Trello Adapter	*Using the Trello Adapter with Oracle Integration 3*
Twilio Adapter	*Using the Twilio Adapter with Oracle Integration 3*
Twitter Adapter	*Using the Twitter Adapter with Oracle Integration 3*
Workday Adapter	*Using the Workday Adapter with Oracle Integration 3*
Zendesk Adapter	*Using the Zendesk Adapter with Oracle Integration 3*
Zuora Adapter	*Using the Zuora Adapter with Oracle Integration 3*

About Mappings

Most integrations include one or more mappings, which specify the source and destination of the data that the integration moves.

A mapping ensures that the appropriate data is taken from a source application and then placed in the correct location in the target application.

Example

Imagine you need to create a Jira account for a new employee. The source of the employee's personal information is Oracle HCM Cloud.

To create the account, an integration must move the employee's personal information from Oracle HCM Cloud (the source application) to Jira (the target application). The mapping for the integration identifies the following information:

- The database fields that contain the source data in Oracle HCM Cloud.
- The database fields that receive the data in Jira.

Copyright © 2022, 2026, Oracle and/or its affiliates.

Mappings and connections work together in your integration. Connections identify the applications that an integration connects to, and a mapping identifies the data to move, including its source and destination fields.

When Terms or Values Differ, Use a Lookup

When you create a mapping, you choose the source and destination of the data that the integration moves. But what if the applications use different terms or values to describe the same thing? Use a lookup to translate the values so that the target application receives the expected value. See About Lookups.

Connections Determine the Fields in the Trees

The connections determine the data structures that you see. For instance, a connection that is based on the Jira Adapter shows only the Jira database fields that are available.

When you create mappings, expand each tree as needed to find the correct field.

Deeper Dive into Mappings

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

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

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

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

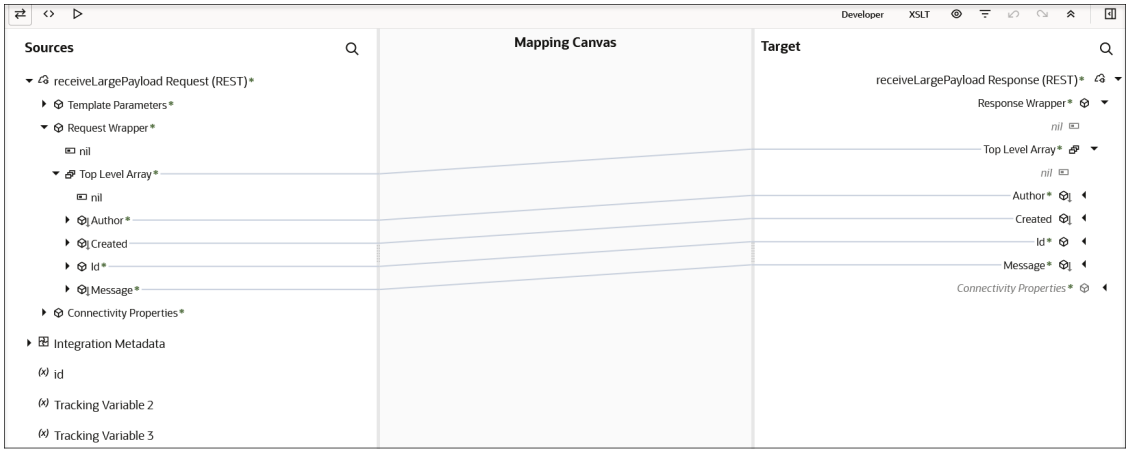
Extended data types are also supported.

Use the Mapper to Define Mappings

After you add a connection to an integration, an empty mapping is created in the integration canvas.

To define the mapping in the mapper, simply double-click the mapping.

Copyright © 2022, 2026, Oracle and/or its affiliates.



The mapper contains trees on the left and right and a canvas in the middle:

- The left tree is the data structure for the source application.
- The right tree is the data structure for the target application.
- On the canvas in the middle, view the lines that connect the entries in each tree.

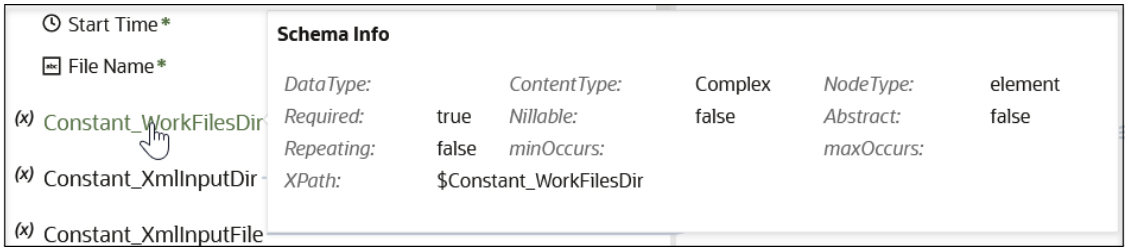
  An icon appears on a line when you update the mapping, such as by adding a function to it.

Connecting fields in the left and right trees is a simple drag-and-drop operation.

Tour of the Mapper

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 **All** from the **Fields** list and **Required** from the **Types** list, and click **Apply**.

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



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

Elements of the Mapper Toolbar

The mapper toolbar provides the following functionality.

Copyright © 2022, 2026, Oracle and/or its affiliates.

Element	Description
	Click to return to the mapping canvas when you are inside the Code or Test page.
	View the XSLT code being created when 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.
	Click to select sources for mapping recommendations.
Select all	This option is enabled by default when mappings are recommended. Click it to disable it, if you don't want to select all the mapping recommendations.
Show mappings	When mappings are recommended, this option appears in the mapper toolbar if you have existing mappings and it is enabled by default. When it is enabled, the mapper shows existing mappings and recommended mappings. Click it to disable it, if you want the mapper to show only recommended mappings and not show existing mappings.
	Click to accept recommended mappings when using mapping recommendations.
Developer	Click to disable user-friendly, source and target element names in the mapper. By default, user-friendly element names are shown.
XSLT	Click to show the XSLT functions.
	Select the following options: • Show the namespace prefixes on source and target element nodes. • Show the types (prefixes and data types) on source and target element nodes.
	Filter the display of element nodes, error messages, and warnings in the source or target data structures.
	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.
	Redo the action that was undone.
	Maximize the size of the mapper. This is useful when working with large schemas.
	Add functions, operators, and XSLT expressions to your mappings.

How to Create a Mapping

A separate guide walks you through creating mappings using the mapper. See Get Started with the Mapper in *Using the Oracle Mapper with Oracle Integration 3*.

About Lookups

When you create a mapping, you choose the source and destination of the data that the integration moves. But what if the applications use different terms or values to describe the

Copyright © 2022, 2026, Oracle and/or its affiliates.

same thing? Use a lookup to translate the values so that the target application receives the expected value.

When Terms or Values Differ, Create a Lookup

A lookup associates the value that an application uses for an item with the value used by other applications for the same item. A lookup is a table, and it can define one or more terms or values for two or more applications.

An integration might require you to include several fields in a lookup, depending on the applications that you're integrating and the number of fields that require mapping.

Examples

Consider an integration in which the source application uses numeric country codes and the target application uses alphabetical country codes. In such a case, add all the country code values to a lookup so that the mapping translates the numeric country codes to alphabetic country codes and displays the expected value in the target application.

Other examples of fields that might require lookups include currency codes and gender codes.

What Happens When You Don't Create a Lookup

If the applications that you're integrating use different terms or values and you don't create a lookup, the target application won't receive data the way that you expect. For example, you might attempt to copy a numeric country code into an alphabetic country code field.

Available Either Globally or Within a Project

A lookup is automatically available in multiple integrations, with some limitations:

- When you create lookups in a project

  The lookups are available only within the project. You can use the lookups in any mapping that you create in any integration in the project.

- When you create lookups outside a project

  The lookups are available only outside projects. You can use the lookups in any mapping that you create in any integration, as long as the integration is not in a project.

A Lookup Ensures That the Destination Application Receives the Correct Value

When an integration that calls a lookup runs, the integration checks the values in the lookup and matches the different codes. For example, if a source application provides a country code of 91 for India, the lookup table matches the value to alphabetic code of IN and sends the IN value to the destination application.

Lookup values are static. Runtime activities reference them but don't change them.

See Map Data and Create Lookups.

Copyright © 2022, 2026, Oracle and/or its affiliates.

About Packages

If you upgraded from Oracle Integration Generation 2, you might have some integrations in packages. In general, Oracle recommends designing integrations in projects rather than packages.

Packages Group Integrations

A package is a structure in which you can group one or more integrations. Packages let you easily import and export a group of integrations in Oracle Integration.

The Oracle Cloud Marketplace offers packages with prebuilt integrations that Oracle provides. You can import these packages to save time designing your integrations. You can also import and export packages that contain integrations that you or others have created.

Use Projects, Not Packages

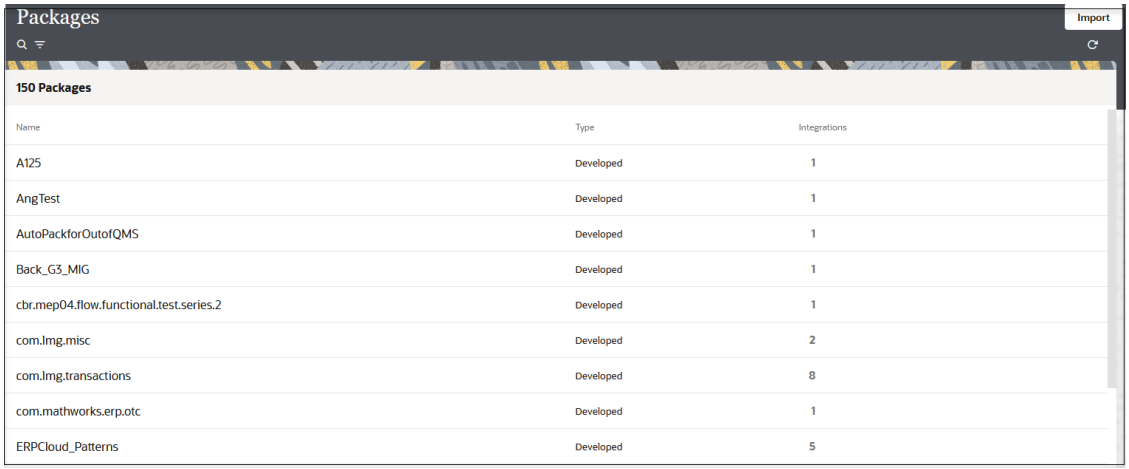
Packages are still available to use; however, Oracle recommends using projects instead. See Get Started with Projects.

If you decide to use packages to group your integrations, remember that every package must include at least one integration. You can't lock a package to exclude other users of your Oracle Integration instance.

Packages Page

You create a package in the Create integration dialog, which appears when you create an integration. You can also update an integration's package in the Primary Information dialog.

View, delete, import, and export packages on the Packages page.



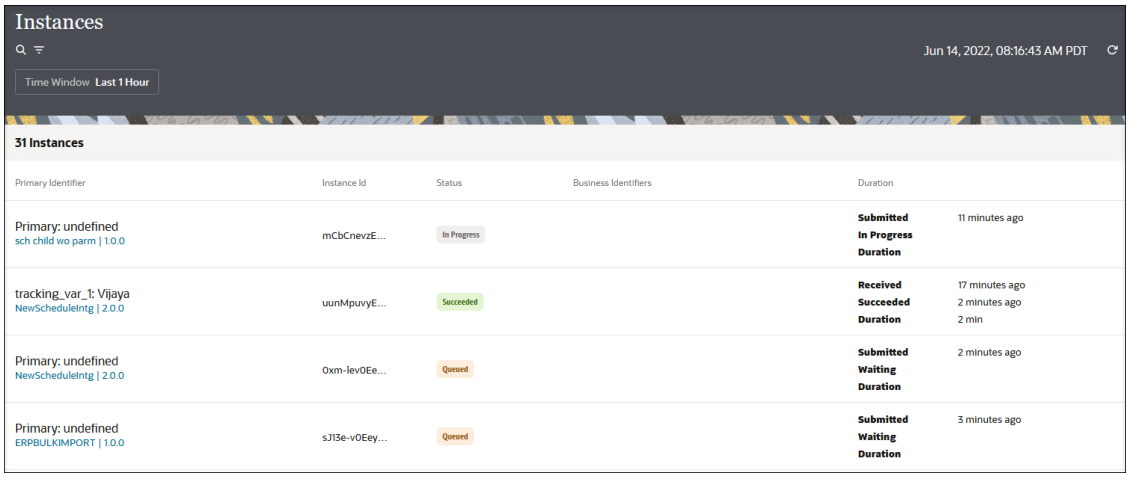
See Manage Packages and Create an Integration.

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.

Copyright © 2022, 2026, Oracle and/or its affiliates.

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



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

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

About Error Management

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

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

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

Integration Failure Scenarios

Integrations can fail for reasons such as the following:

Copyright © 2022, 2026, Oracle and/or its affiliates.

- 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.
- Any other Oracle Integration system failure.

See Manage Errors.

AI Innovation and Oracle Integration

Oracle Integration partners with you on your journey to innovate with AI.

Use AI Assistant to Get Things Done Faster

AI assistant in Oracle Integration brings you increased usability, collaboration, and productivity. Tap into AI to automatically create integrations, define schedules, describe integration components, generate FHIR and ATP queries, and resolve errors.

See Fast Track Automation with AI Assistant.

Add AI Power to Automations

Uncover endless possibilities with direct connection from integrations to Oracle Cloud Infrastructure(OCI) AI Services, and OpenAI and Anthropic models. Incorporate OCI AI Services, and OpenAI and Anthropic large language models into your integrations to add automated text and image processing, content generation, analysis capabilities, and more.

See Bring Intelligence to Automations.

Solve Complex Automation with AI Agents

Create agentic AI tools from integrations and call integrations from any Model Context Protocol (MCP) client. Design and build AI agents to work autonomously, and continuously adapt to your business.

See Automate with Agentic AI.

> **Note**
>
> AI feature support varies by region. See AI Feature Availability in *Provisioning and Administering Oracle Integration 3*.

Fast Track Automation with AI Assistant

Maximize development productivity in Oracle Integration with embedded AI. Use AI where you work to automatically create integrations and projects, define schedules, document project and

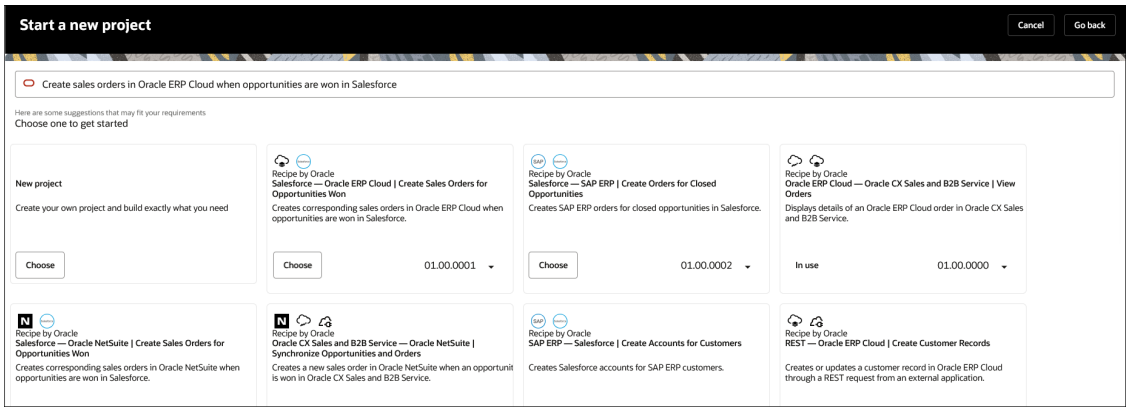
Copyright © 2022, 2026, Oracle and/or its affiliates.

integration components, generate ATP SQL queries, generate FHIR search queries, and resolve errors. Seamless AI, faster automation.

> **Note**
>
> - AI features are available in select regions. See AI Feature Availability in *Provisioning and Administering Oracle Integration 3*.
> - You can enable or disable the AI assistance features. See Enable or Disable AI Features.

Create Projects Using Natural Language

Tell Oracle Integration what you want in natural language. Select to create a project from an existing suggested accelerator or recipe, or create a project from scratch.



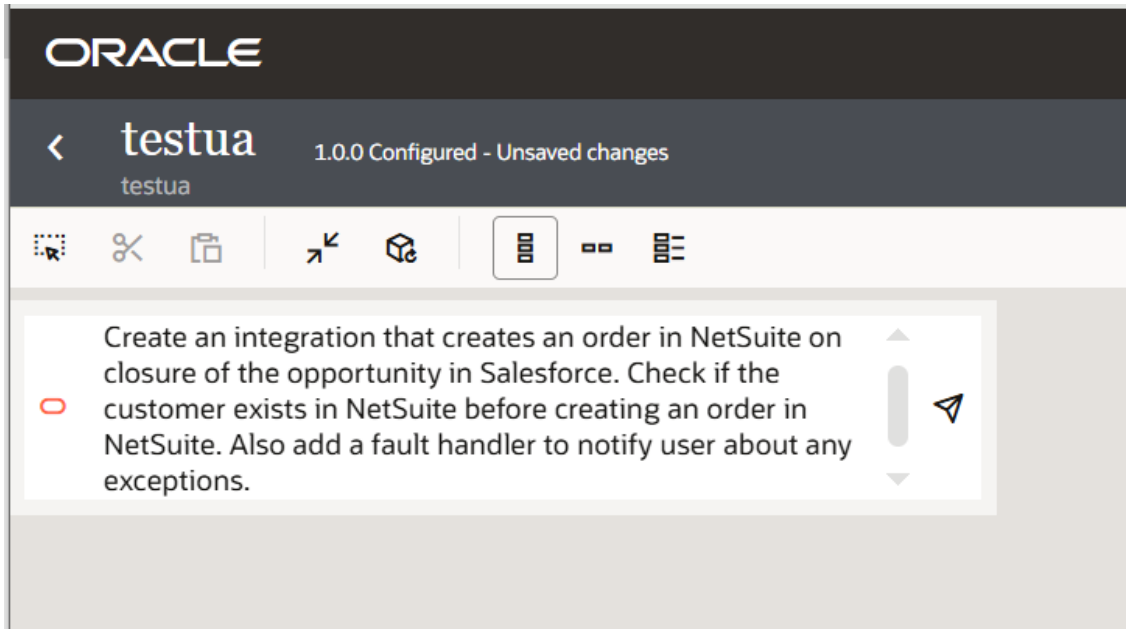
Try it out: Create a Project By Selecting an Accelerator or Recipe that the AI Suggests

Create and Edit Integrations Using Natural Language

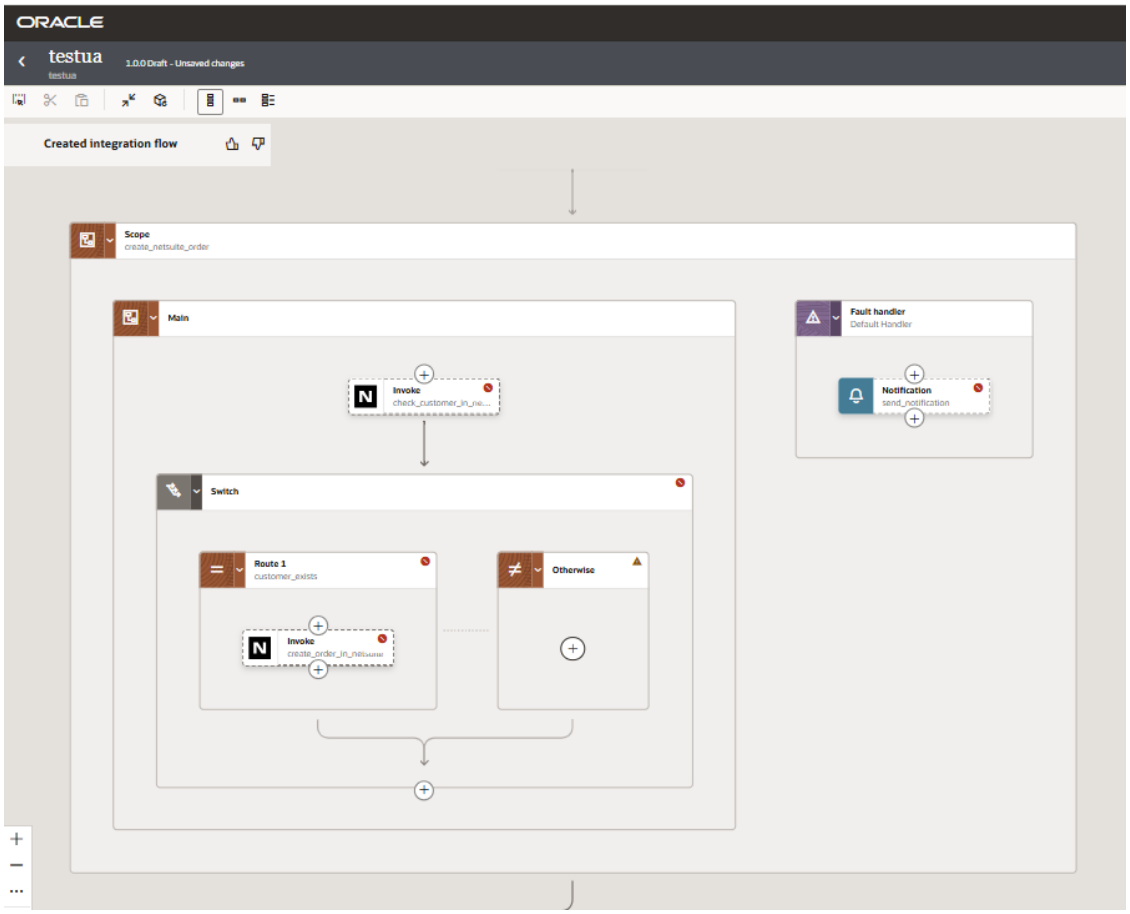
Just tell Oracle Integration what you want in natural language. The integration structure is created for you, and trigger and invoke connections are also configured within the integration. If you want to change something in the integration, you can also use natural language to edit the integration.

Tell Oracle Integration what you want in your integration:

Copyright © 2022, 2026, Oracle and/or its affiliates.



And an integration is generated for you.



Try it out:

- Use AI to Create an Integration

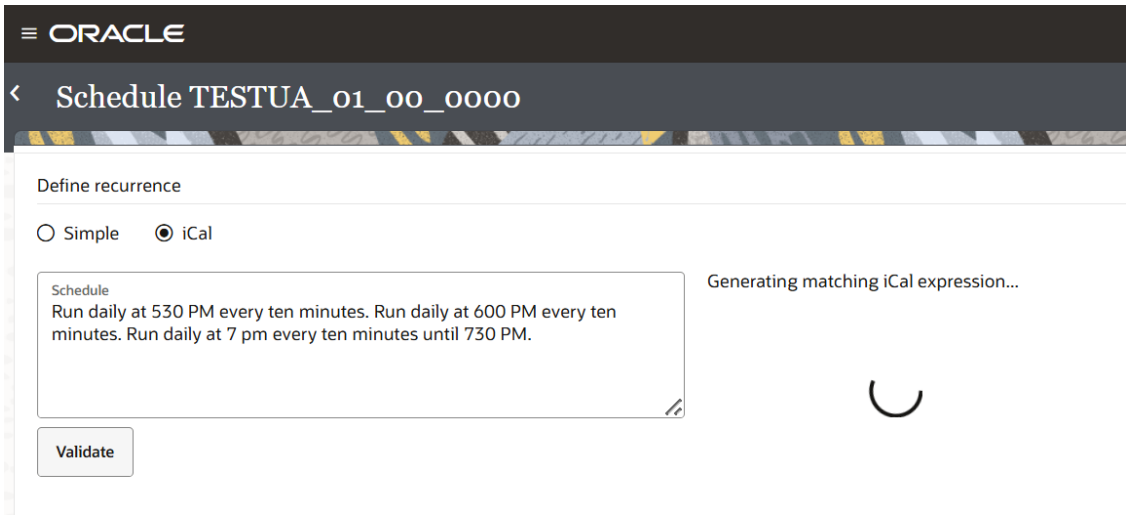
Copyright © 2022, 2026, Oracle and/or its affiliates.

- Use AI to Create Integrations Based on Integrations from the Integration Store
- Use AI to Edit an Integration

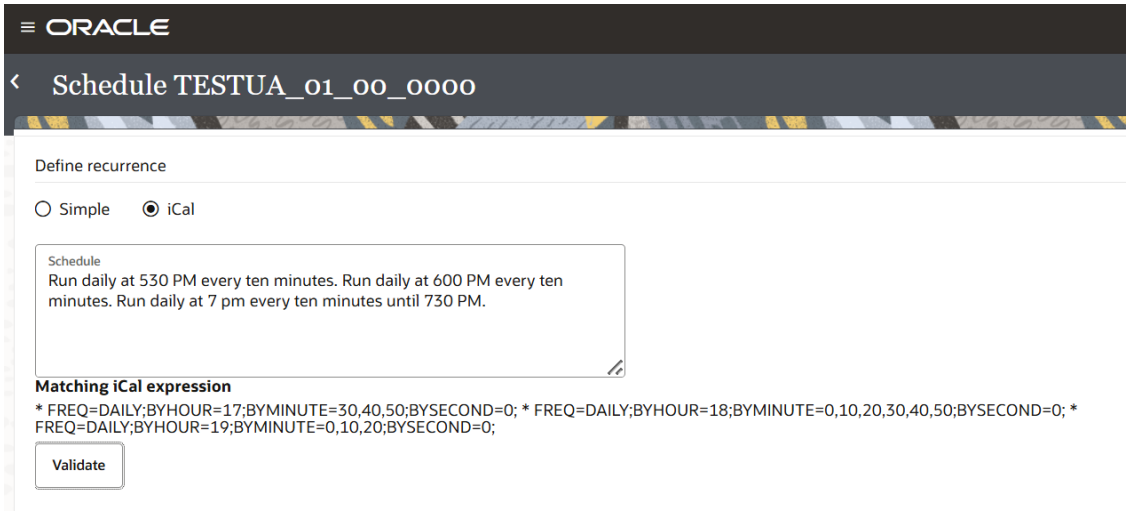
Schedule Integrations Using Natural Language

The integration needs scheduling? You don't have to write iCal expressions. Tell Oracle Integration what you want in natural language and the schedule is created for you, including the iCal expression.

Tell Oracle Integration the schedule you want to create:



And the matching iCal expression is automatically created.



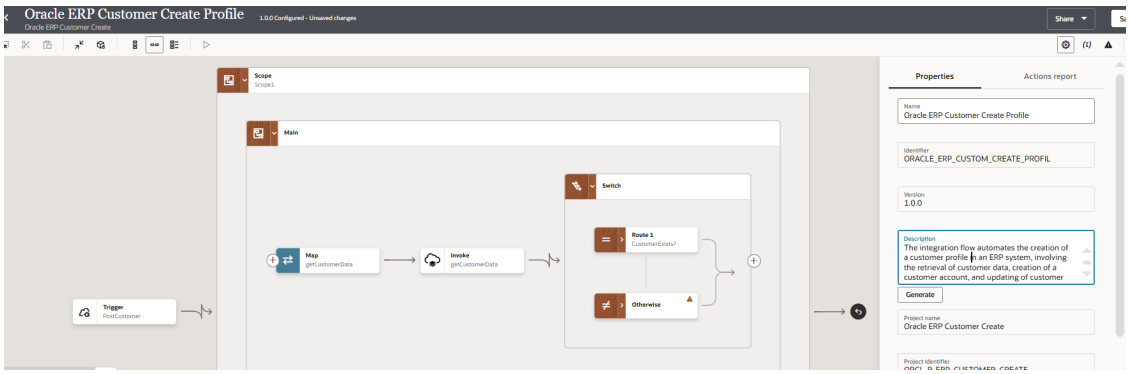
Try it out: Create an AI-Generated iCal Schedule by Entering Natural Language

Copyright © 2022, 2026, Oracle and/or its affiliates.

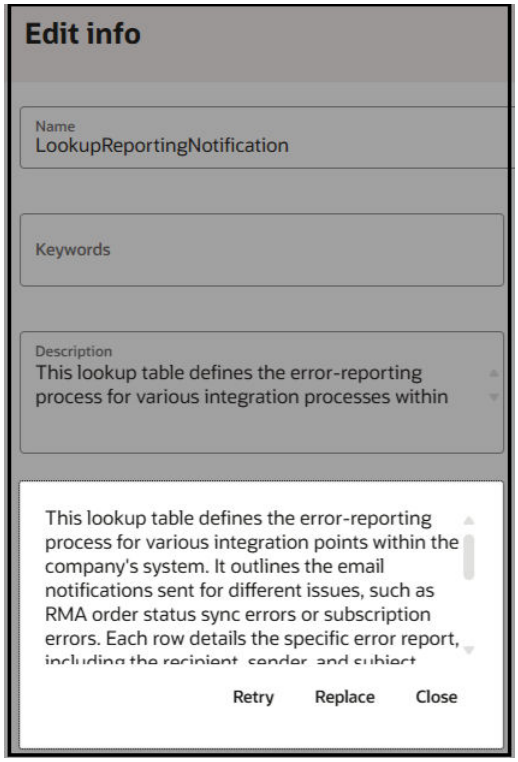
Automatically Create Descriptions for Projects, Integrations, and More

Enable collaboration at the touch of a button. Generate descriptions for projects, integrations, libraries, lookups, events, B2B trading partners, healthcare schemas, custom FHIR resources, and HL7 messages.

Generate Integration Descriptions

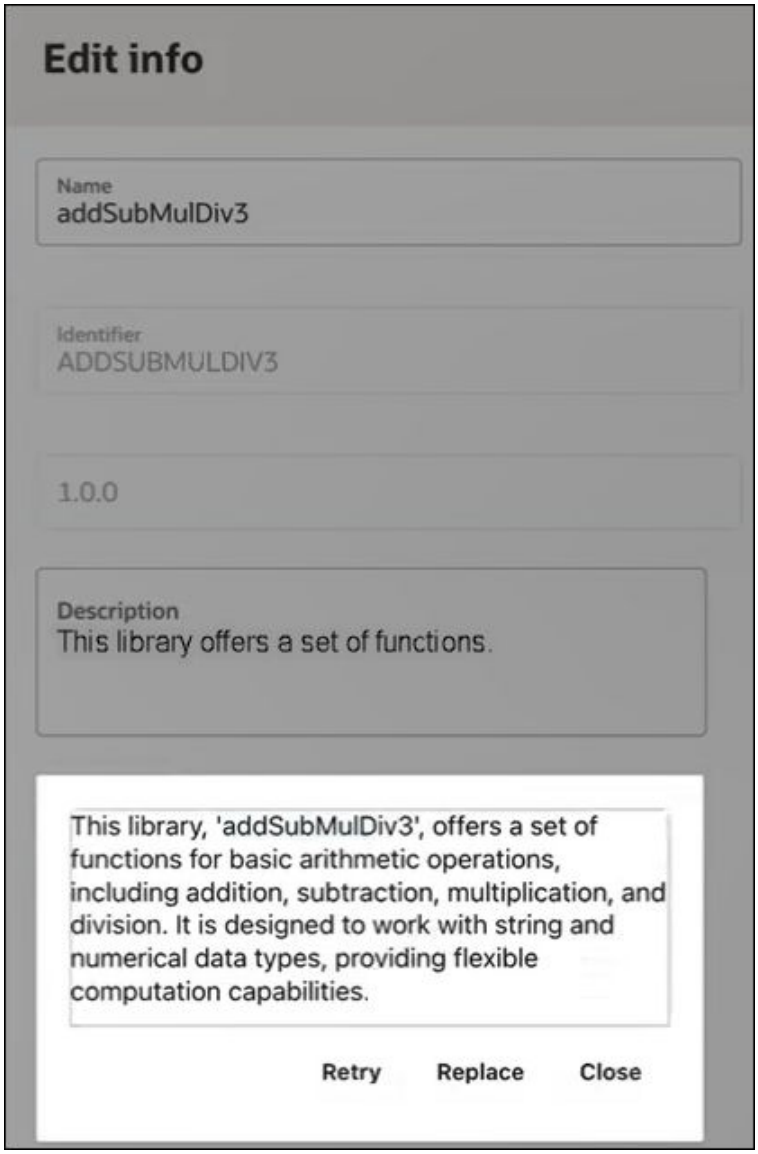


Generate Lookup Descriptions



Generate Library Descriptions

Copyright © 2022, 2026, Oracle and/or its affiliates.



Try it out:

- Healthcare:
  - Edit a Custom FHIR Resource
  - Edit a Healthcare Schema
  - Edit an HL7 Message
- Events: Create an AI-Powered Event Description
- Integrations: Create an AI-Powered Integration Description
- Libraries: Create an AI-Powered Library Description
- Lookups: Create an AI-Powered Lookup Description
- Projects: Create an AI-Powered Project Description
- B2B Trading Partners: Create an AI-Powered Trading Partner Description

Copyright © 2022, 2026, Oracle and/or its affiliates.

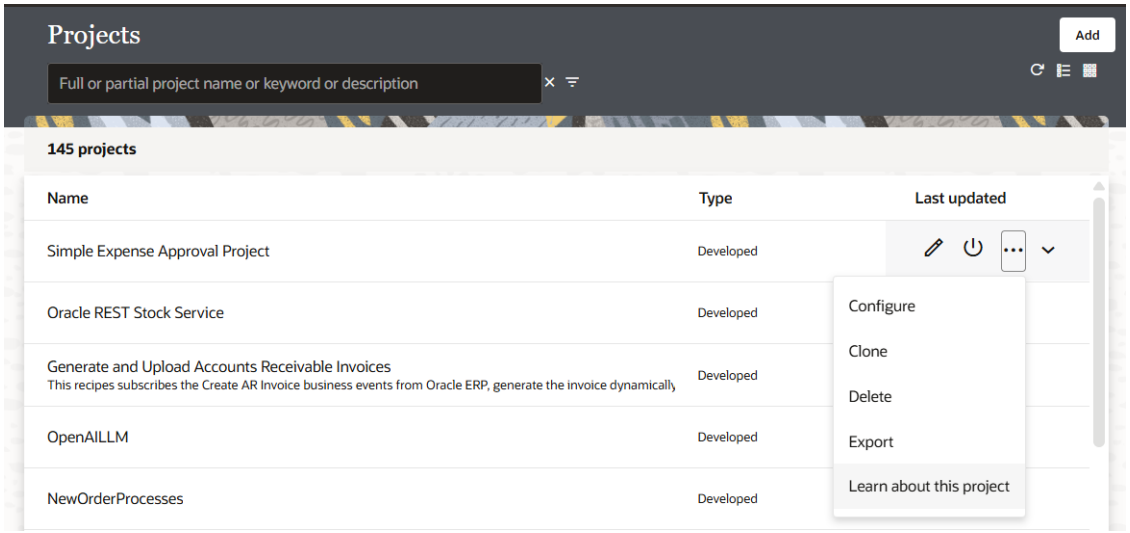
Automatically Generate Documentation for Projects and Integrations

Maximize collaboration by automatically documenting projects and integrations. You can even save as PDF. Click **Learn about** to generate documentation for a project or an integration. You can also use the **Actions** ... menu when viewing projects or integrations as a list.

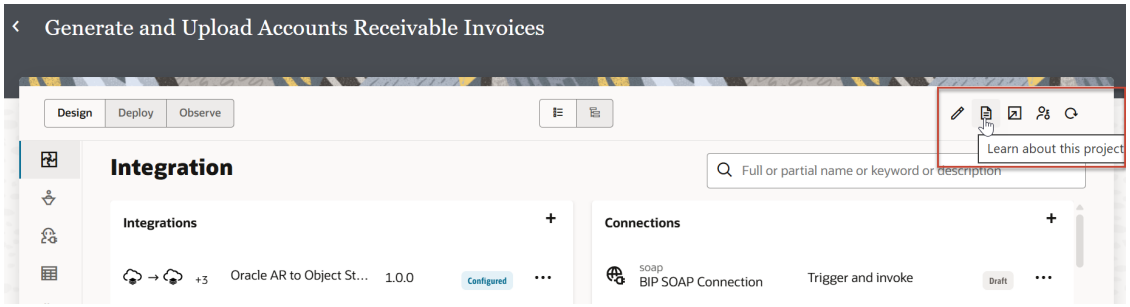
Generate Project Documentation

Generate project documentation and know all about the project in minutes. Find out about integrations in the project, technologies used, shared resources, deployments, and change history.

Generate project documentation from the project list:

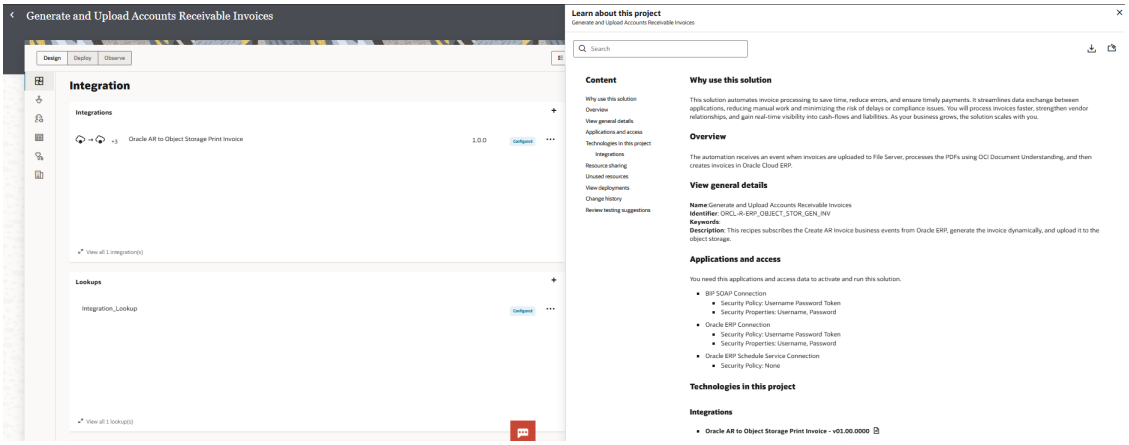


Or, generate project documentation from an individual project:



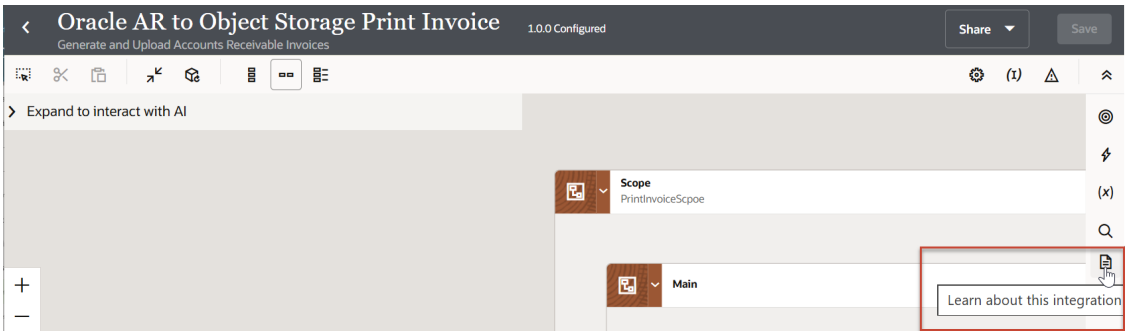
Get information about applications in the project, integrations, technologies, change history, and more.

Copyright © 2022, 2026, Oracle and/or its affiliates.

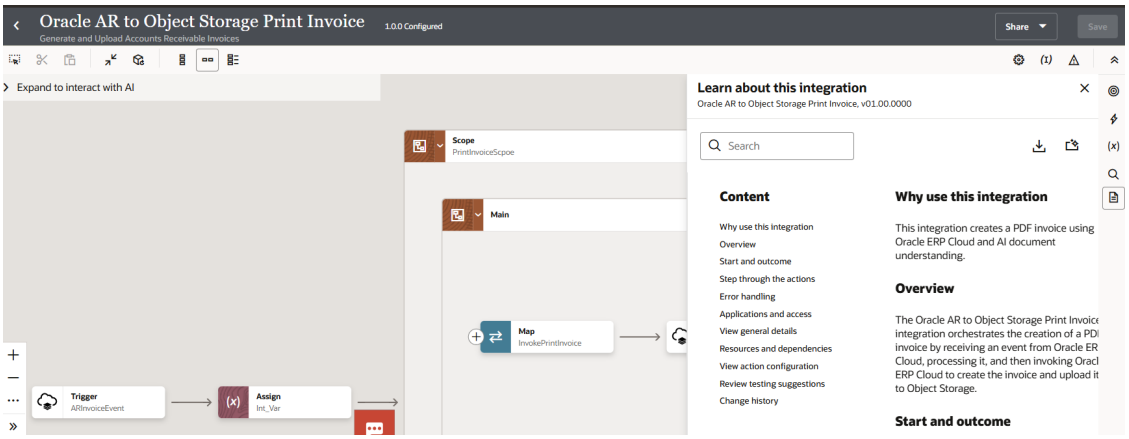


Generate Integration Documentation

Generate integration documentation at the click of a button.



And get information about the actions in the integration, connections, dependencies, and more.



Try it out:

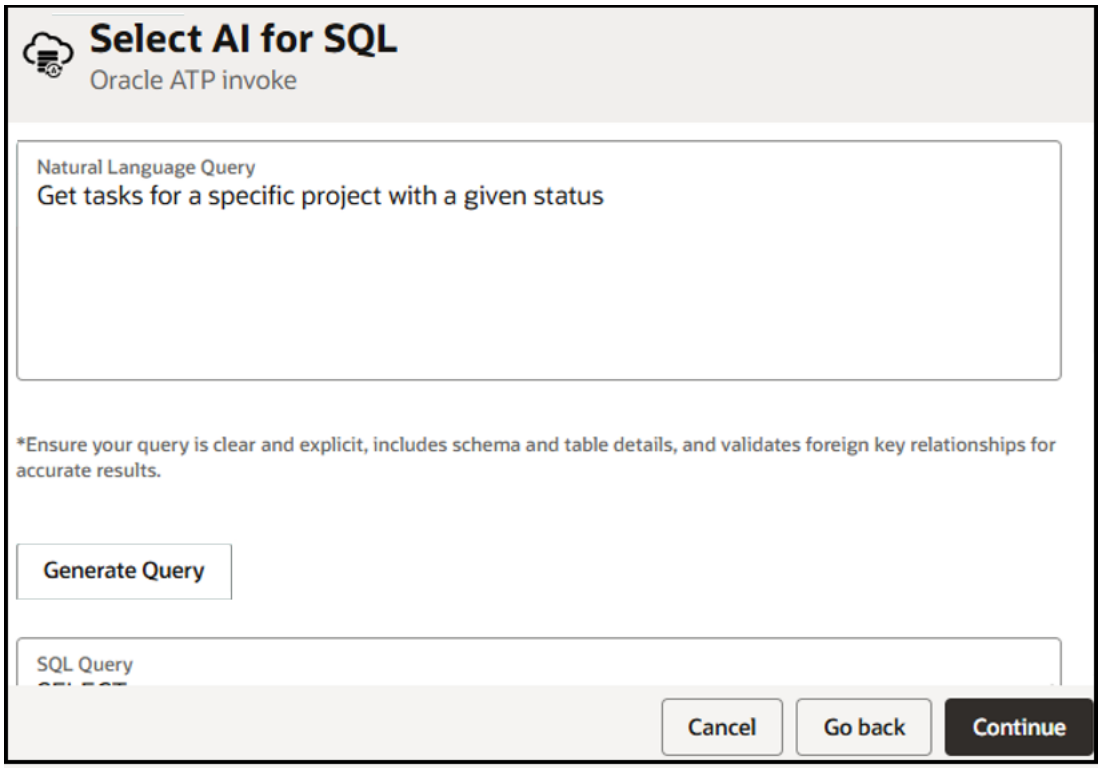
- Use AI to Generate Project Documentation
- Use AI to Generate Integration Documentation

Copyright © 2022, 2026, Oracle and/or its affiliates.

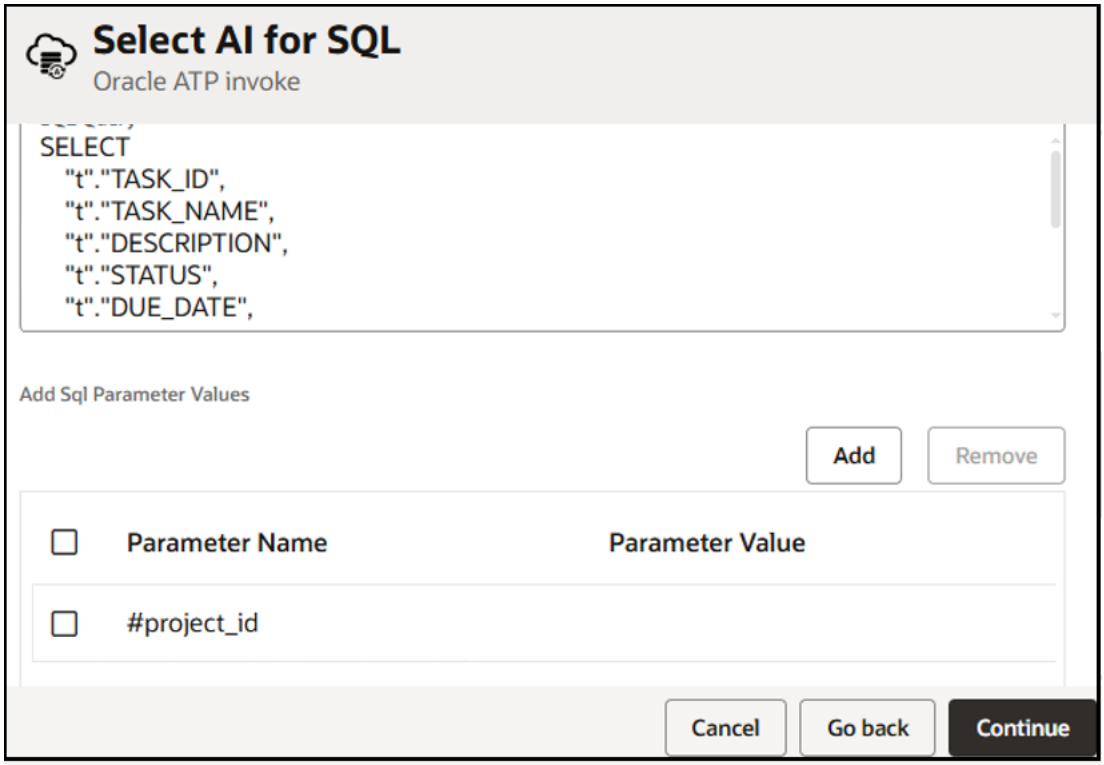
Query Data Using Natural Language

Tell Oracle Integration what you want to find and a query is automatically generated for you. Use it in the Oracle Autonomous Transaction Processing Adapter and FHIR Adapter.

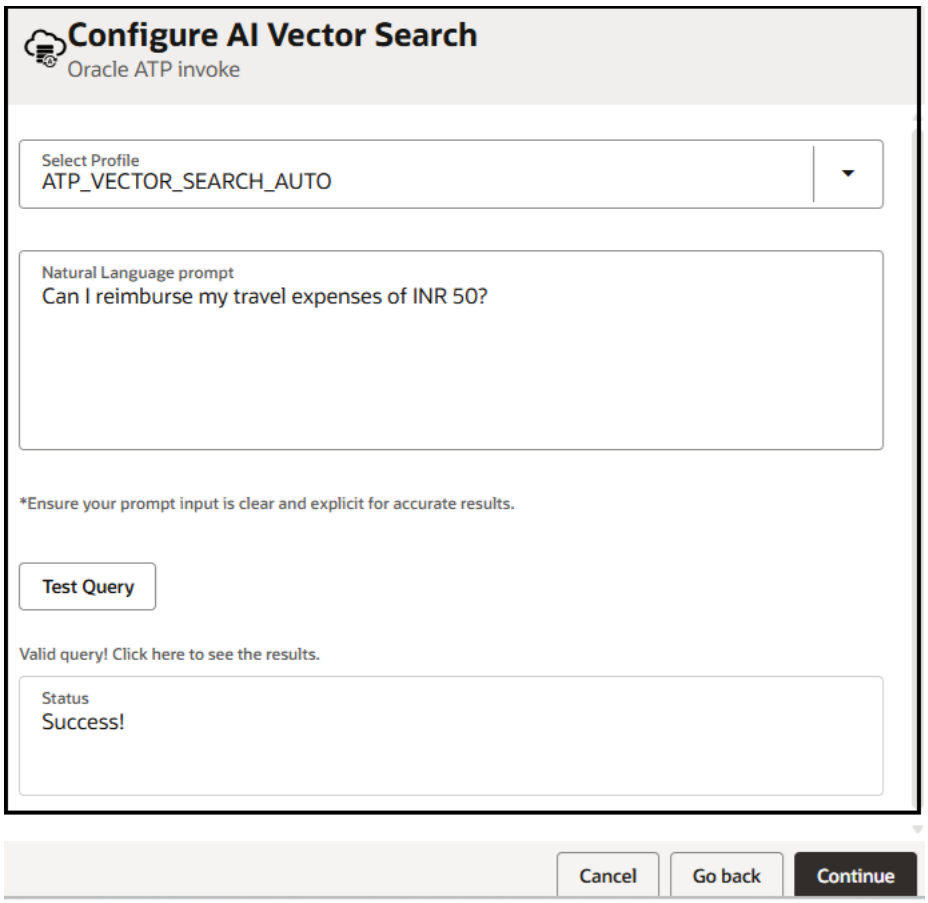
Generate ATP SQL Queries



Copyright © 2022, 2026, Oracle and/or its affiliates.

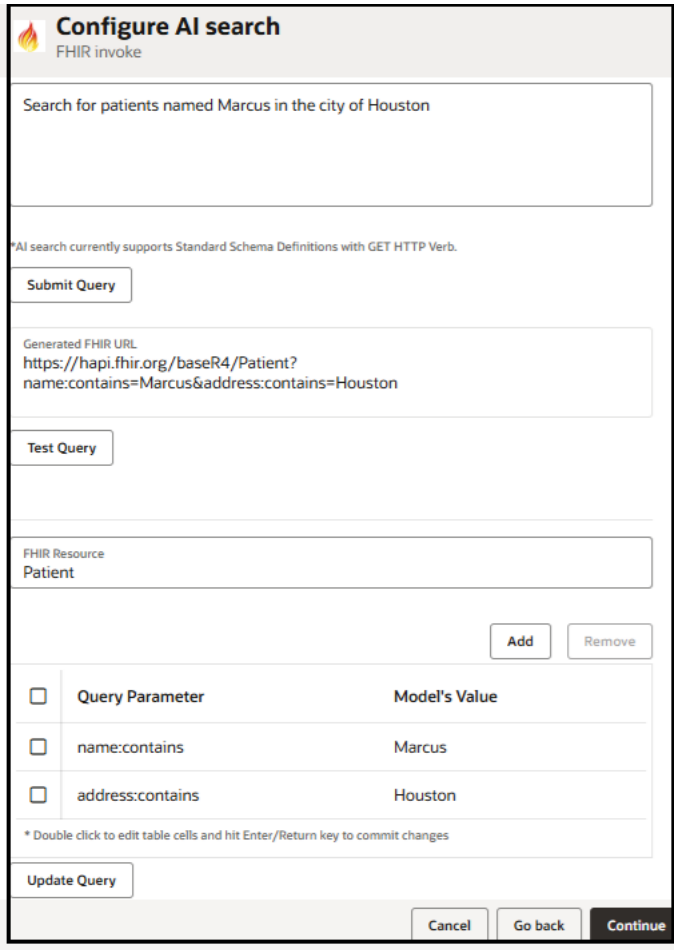


Search an ATP Vector Database



Copyright © 2022, 2026, Oracle and/or its affiliates.

Generate FHIR Search Queries



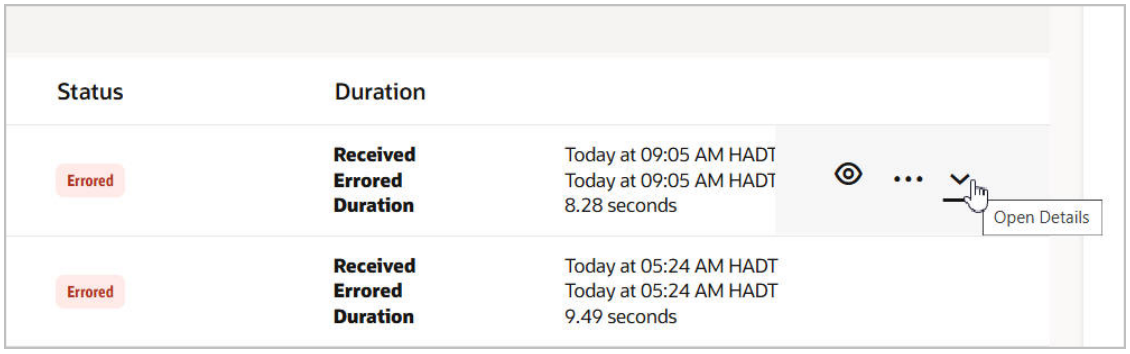
Try it out:

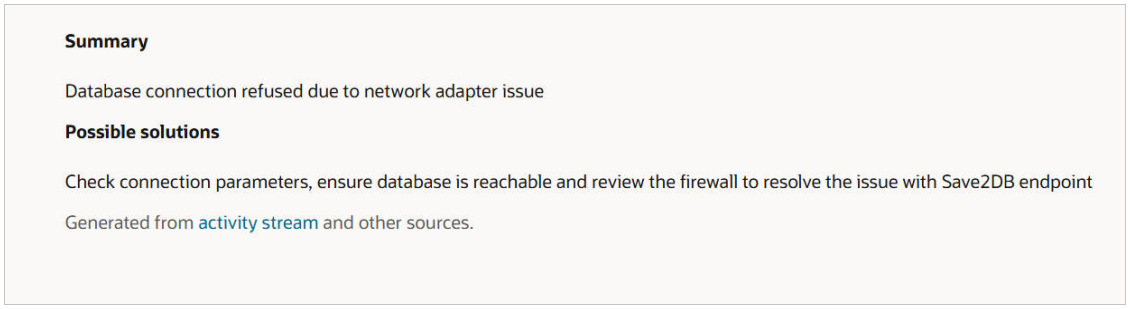
- Oracle Autonomous Transaction Processing Adapter: Create an AI-Generated SQL Query
- Oracle Autonomous Transaction Processing Adapter: Search an ATP Vector Database
- FHIR Adapter: Create an AI-Generated FHIR Search Query

Troubleshoot Errors Faster with AI

Harness AI to troubleshoot error messages for failed integration instances and errored B2B messages and determine the next course of action. View additional details to display the AI error summary with troubleshooting suggestions.

Copyright © 2022, 2026, Oracle and/or its affiliates.





Try it out:

- View an AI-Powered Error Message
- View an AI-Powered Summary for Errored B2B Messages

Enable or Disable AI Features

You can manage the AI assistance features for your Oracle Integration service instance. By default, the AI assistance features are enabled in your Oracle Integration instance.

> **Note**
>
> AI features are available in select regions. See AI Feature Availability in *Provisioning and Administering Oracle Integration 3*.

To enable or disable the AI assistance features for your Oracle Integration service instance, perform the following steps:

1. In the navigation pane, click **Settings**, then **AI Settings**.
   The AI Settings page appears.
2. On the AI Settings page, the **Enable AI Features** option is enabled by default. You can disable it if you want to disable the AI assistance features. You can enable or disable the option, as needed.
3. In the upper right corner of the page, click **Save** to save your settings.

Copyright © 2022, 2026, Oracle and/or its affiliates.

Bring Intelligence to Automations

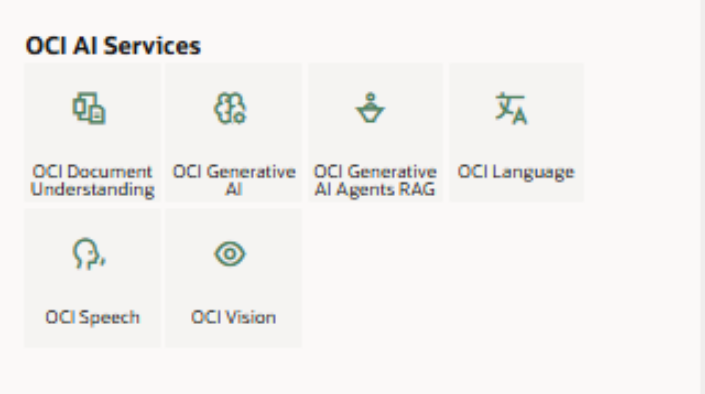
Enable smarter decisions, automate specialized tasks, and reduce manual intervention by adding intelligence to your automations. Directly connect from integrations to OCI AI Services and OpenAI machine learning models.

> **Note**
>
> AI features are available in select regions. See AI Feature Availability in *Provisioning and Administering Oracle Integration 3*.

Augment Integrations with OCI AI Services

Supercharge your integration with OCI AI Services. Add capabilities to automate document handling and processing, extract data from text, speech, and images, and generate text to provide custom responses or actions. Endless possibilities for bringing intelligence into every aspect of your automation, right within the integration canvas.

Just drag and drop the specific OCI AI Services action into your integration to incorporate OCI Document Understanding, OCI Generative AI, OCI Generative AI Agents RAG, OCI Language, OCI Vision, and OCI Speech.



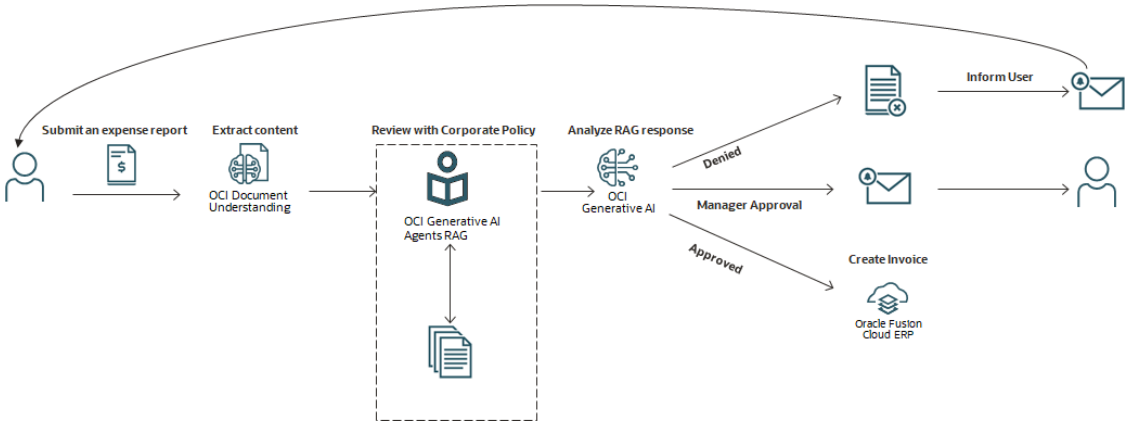
Watch a video:

Video

Use Case: Automate Expense Processing with OCI AI Services

You can use OCI AI Services to automate expense report processing. Use OCI Document Understanding to extract key data from expense reports such as expense amount and type, then use OCI Generative AI Agents RAG to evaluate the expense amount and type against company policies. Based on the analysis of company policies, OCI Generative AI then determines whether the expense report is approved, denied, or whether it requires human approval. If the expense is approved, an invoice is automatically created in Oracle ERP Cloud.

Copyright © 2022, 2026, Oracle and/or its affiliates.



Try it out:

- Extract Document Information with a Document Understanding Action
- Generate, Embed, or Rerank Text with a Generative AI Action
- Ask Questions and Receive Answers Based on Current Enterprise Data with a Generative AI Agents RAG Action
- Perform Text Analysis and Translation with a Language Action
- Transcribe Speech to Text and Synthesize Text to Speech with a Speech Action
- Analyze and Extract Information from Images with a Vision Action

Expand Integrations with OpenAI LLM Models

Use the OpenAI Adapter to incorporate OpenAI LLM Models into your integrations for advanced content generation, intelligent summarization, contextual classification, and conversation handling.

Here are some examples of what you can do:

- **Analyze text**: Understand and extract insights from textual data such as employee sentiment from appraisals or customer feedback from product reviews.
- **Categorize text**: Automatically classify documents or resumes based on content and predefined rules.
- **Draft emails**: Generate intelligent email responses for customer queries, HR communication, or marketing campaigns.
- **Translate**: Translate customer support responses or documentation into multiple languages.
- **Summarize**: Condense long documents or reports into digestible summaries for decision-makers.

Try it out:

- OpenAI Adapter: OpenAI Adapter Capabilities and Implement Common Patterns Using the OpenAI Adapter

Copyright © 2022, 2026, Oracle and/or its affiliates.

Expand Integrations with Anthropic LLM Models

Use the Anthropic Adapter to connect to Anthropic's Claude LLM models and enable text prompting and tools/function calling for conversational and AI-assisted workflows.

Here are some examples of what you can do:

- **Text and prompting**: Send instructions, questions, or conversation prompts to Claude for contextual and coherent responses.
- **Tools calling:** Trigger external tools or functions dynamically based on LLM output, enabling action-driven AI workflows.

Try it out:

- Anthropic Adapter: Anthropic Adapter Capabilities and Implement Common Patterns Using the Anthropic Adapter

Execute Queries Against an OpenSearch Cluster

Use the OpenSearch Adapter to execute queries against an OpenSearch cluster.

Here are some examples of what you can do:

- Execute queries against an OCI Search with OpenSearch cluster hosted by Oracle.
- Execute queries with or without the use of agentic AI.

Try it out:

- OpenSearch Adapter: Open Search Adapter Capabilities and Implement Common Patterns Using the Open Search Adapter.

Automate with Agentic AI

Use AI agents to orchestrate integrations into adaptive automation solutions. AI agents autonomously determine which agentic AI tools to use, when, and in which order to fit your business needs.

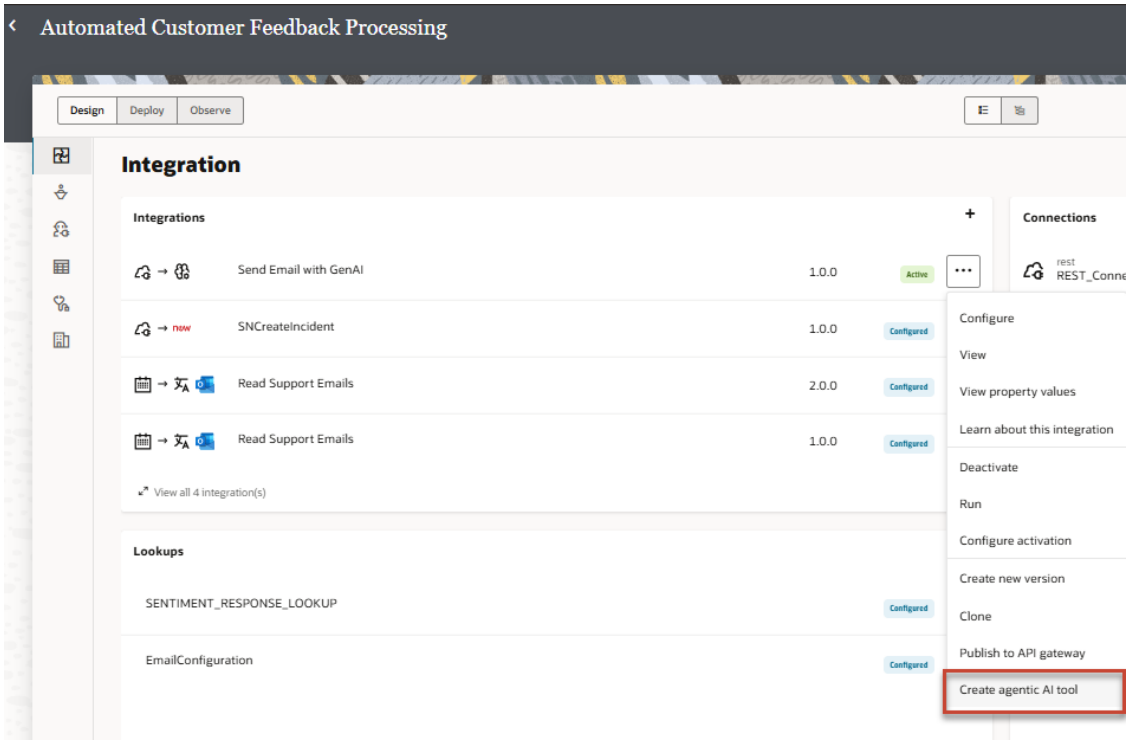
> **Note**
>
> AI features are available in select regions. See AI Feature Availability in *Provisioning and Administering Oracle Integration 3*.

Invoke Integrations from Any MCP Client

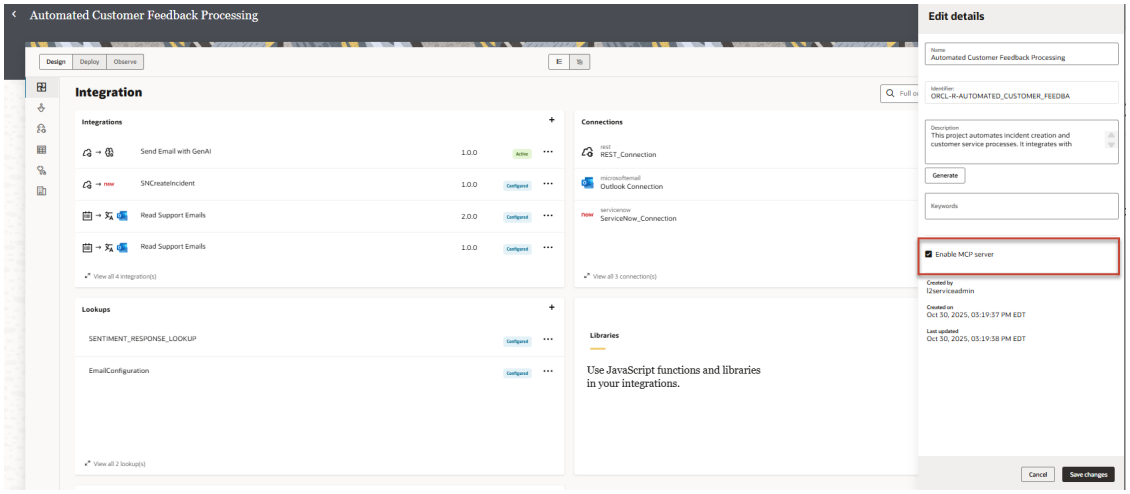
Register any integration as an agentic AI tool and enable Model Context Protocol (MCP) for the project. You can discover and use integrations as tools from any AI agent framework that supports MCP.

Register the integration as an agentic AI tool:

Copyright © 2022, 2026, Oracle and/or its affiliates.



Enable MCP for the project:



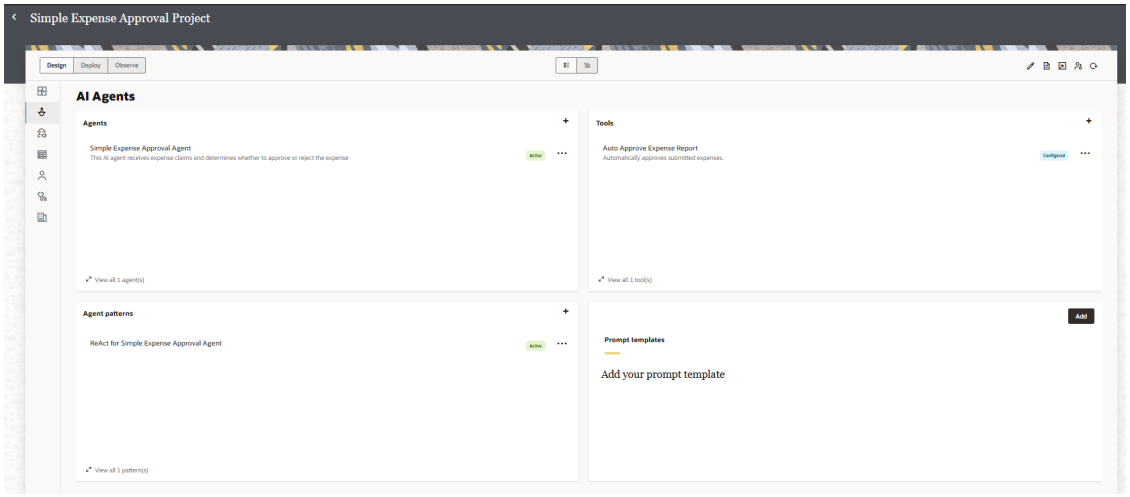
Try it out: Using Integrations as Tools in an MCP Server

Design and Build AI Agents

Design and build intelligent AI agents that act autonomously: analyzing data, making decisions, and orchestrating complex workflows. With integrations as agentic AI tools, seamlessly connect to any system and drive innovation and efficiency across your organization.

Create AI agents in projects:

Copyright © 2022, 2026, Oracle and/or its affiliates.



Try it out: Tutorial: Build and Run Your First AI Agent

Get Started with Observability

No matter your role or the number of integrations, agentic AI, B2B, decisions, healthcare, human in the loop, and robots that you're responsible for monitoring, you can find the information you need to keep your business on track in Oracle Integration.

Key Observability Features

Several sets of monitoring pages and customizable email messages provide you with the critical insights that you need to determine whether your business processes are running efficiently and error-free.

1. **Observability pages in Oracle Integration**

   To monitor all your integrations in one place, head to the Observability pages, which you open from the navigation pane. The pages provide information about all the components in your Oracle Integration instance, including integrations that are and aren't in projects.

   See About the Observability Pages.

2. **Observe page in a project**

   To monitor all automation solutions in a project, head to the Observe page in the project.

   See About the Observability Pages.

3. **Daily or hourly notification emails**

   Quickly identify whether integrations have experienced errors by configuring notification emails. The emails provide reports about message counts, including failed messages, as well as alerts about other failures, including when a connectivity agent becomes unavailable.

   See Send System Status Reports with Notification Emails.

Develop an Observability Strategy

An observability strategy helps you plan your monitoring work for the business processes that your organization has automated. Whether simple or complex, an observability strategy helps you focus your time and energy on the tasks that matter most.

Copyright © 2022, 2026, Oracle and/or its affiliates.

See Develop an Observability Strategy.

Explore the Observability Tasks

Ready to learn about all the monitoring tasks in Oracle Integration? See Monitor Integrations During Runtime.

Extend the Observability Pages to a Custom Application

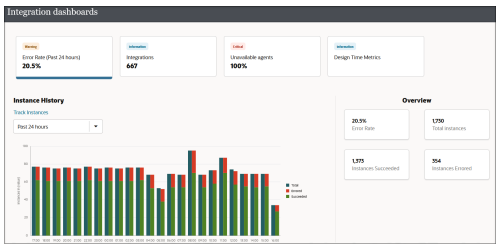
Want to see the observability information in your own monitoring tools? Use our REST APIs to access these capabilities. See About Building Your Own Observability Tools.

About the Observability Pages

Oracle Integration includes several sets of pages for monitoring integrations and their components, including the Observability pages, which you open from the navigation pane, and the Observe page within a project.

About the Observability Pages

The Observability pages, which you open from the navigation pane, provide information about all the components in your instance, including integrations that are and aren't in projects.

Page	Description	What you can do on the page
Dashboards	When you want to check the overall health of everything your organization has built, head to the interactive Dashboards page, which provides at-a-glance information about the messages that Oracle Integration processed and the integrations that ran.	• Understand the overall status and tracing levels for integrations. • See the number of active integrations you have and the patterns that they use. • Identify the connections that have the most errors. • Determine whether your connectivity agents are up and running as expected. • Understand the current status of the components your organization is developing, including projects, packages, integrations, connections, and more.  See View the Dashboard.
Design time audit	Who updated an integration or other component, and when did the update happen? Get this information on the Design time audit page.	• Review the actions that people took, the components they acted upon, the people who took the actions, and the dates and times when the actions occurred. See Check the Design Time Audit History for an Integration or Other Component.

Copyright © 2022, 2026, Oracle and/or its affiliates.

Page	Description	What you can do on the page
Integrations	The Integrations page is where to go when you want to see high-level information about your integrations and their messages.	• View the processing status for the messages that the each integration generated, including the total number of success and error messages. • Check performance over time of an integration, including viewing the fastest, slowest, and average run times for an integration, as well as the largest deviation from the average. • Drill down into an integration, such as determining why an integration was aborted. See Monitor the Message Processing Status of Integrations.
Subscriptions	If you have integrations that use the Event pattern, open the Subscriptions page to get status information about these integrations.	• Monitor the status of integrations that subscribe to events. • Retain an event subscription when deactivating an integration. See Monitor Integrations That Subscribe to Events.
Agents	When the Dashboards page tells you that one or more connectivity agents is unavailable, open the Agents page to identify the agent that is down.	• Identify the connectivity agents that are currently unavailable. See Monitor Agents.
Instances	When an integration runs, Oracle Integration creates an integration instance for it. Track the status of all your integration instances and view their message payloads on the Instances page.	• Collect information about the type and use of each integration. • Check the status of an individual integration instance. • View the tracing level and the activity stream for an integration instance, see how the message flow through the integration went, and download the logs for the activity stream. • Determine where an error is occurring in an integration. • Abort an integration instance. • Resubmit a failed integration instance. See Track Integration Instances.

Copyright © 2022, 2026, Oracle and/or its affiliates.

Page	Description	What you can do on the page
Errors	Like the Instances page, the Errors page lists integration instances. However, while the Instances page lists *all* integration instances, the Errors page lists only the integration instances that *failed*.	Many tasks from the Instances page, plus: • Quickly identify any failed integration instances. • Resubmit or abort a failed or in-progress integration instance. • Resubmit or abort multiple integration instances at one time. • View only recoverable or nonrecoverable integration instances. • View only the errored integration instances for a specific integration or connection, or for an integration that you have attempted to recover by resubmitting the integrations. See Manage Errors.
Future runs	On the Future runs page, view all the schedule integrations and the times they're scheduled to run. This page helps you understand the timeline of your integrations and the queue so that you don't schedule too many integrations to run at the same time.	• Understand the schedule integrations in the queue. • Determine whether your schedule integrations are clustered too closely. See View the Calendar of Schedule Integration Runs.
B2B tracking	If you use B2B for Oracle Integration, use the B2B tracking page to track information related to your B2B integrations.	• Track the B2B message interaction between trading partners, including viewing wire messages. Wire messages are raw messages that trading partners exchange. • Resubmit outbound messages. • View message logs, which are similar to the activity stream for an integration and which show messages from trading partners, the identity of the trading parters, the unpacking of the messages, the status of the acknowledgement messages, and more. See Track B2B Messages in *Using B2B for Oracle Integration 3*.
Fusion Applications	When an integration that consumes a published event from an Oracle Fusion Application fails, troubleshooting the failure can be difficult. In such cases, head to the Fusion Applications page, where you can determine the source of the error.	This page is only for integrations that use the Oracle ERP Cloud Adapter or the Oracle CX Sales and B2B Service Adapter (previously known as the Oracle Engagement Cloud Adapter). • When an event doesn't arrive as expected, determine whether the failure occurred in Oracle Fusion Applications or in Oracle Integration. • Review and manage issues related to business event integrations. See Diagnose and Manage Event-Based Oracle Fusion Applications Integrations.

Copyright © 2022, 2026, Oracle and/or its affiliates.

About the Observe Page in a Project

On the Observe page in a project, you can monitor the automation solutions in the project. Many tabs are identical to their counterpart pages in the Observability area, with one key difference: The Observe pages show information only for the automation solutions that are *in the project*.

See Monitor Integrations in a Project.

Develop an Observability Strategy

If you're responsible for monitoring automation solutions, consider developing an observability strategy so that you can meet your business objectives while streamlining your monitoring work.

On This Page

- Why You Need an Observability Strategy
- Tasks to Include
- Factors to Consider
- People Are Just as Important as Tasks
- Sample Observability Strategy

Why You Need an Observability Strategy

Your options for observability are numerous. But how do you plan your monitoring work so that you find issues as early as possible while still allowing you time to complete your other responsibilities? And how do you ensure that everyone in your organization is in agreement about your approach to monitoring?

The answer is to create an observability strategy. An observability ensures that your organization discovers any issues with its business processes in a timely manner.

When creating an observability strategy, consider your work, including:

- The tasks that you perform daily, weekly, monthly, and quarterly.
- The time when you perform the tasks.
- The person who performs the task.
- The reason you perform the tasks.
- The stakeholders to notify in the event of an issue, and when they should be notified.
- The steps you take to determine the source of an issue.
- The steps you take to resolve an issue.

Tasks to Include

Monitoring tasks typically fall into one of the following categories:

- Tasks you complete before an integration goes live.
- Daily monitoring tasks to make sure everything is running as expected.
- Occasional tasks to ensure operational excellence.

Copyright © 2022, 2026, Oracle and/or its affiliates.

For a collection of suggested tasks to potentially include in your observability strategy, see Workflow for Monitoring Integrations.

Factors to Consider

Consider an organization with business-critical integrations that run daily at 10 AM and 4 PM. You might choose to actively monitor these integrations at those times. For other integrations, you might rely upon an hourly notification email or periodic reviews of the Observability pages to inform you of issues.

When planning your observability strategy, consider your business requirements, including:

- Which integrations are critical to my organization?
- How soon do we need to know about issues that impact our critical integrations? What about less critical integrations?
- What are the business impacts if we don't addresss the issues in a timely manner?
- How often do we need to check in to make sure everything is running as expected?

People Are Just as Important as Tasks

Don't forget the critical step of getting stakeholders involved early in the planning of the observability strategy. Getting all stakeholders to contribute to and agree to an observability strategy is sometimes a challenging task. However, consensus on the strategy contributes to a stronger strategy that is likely to meet your organization's monitoring requirements.

Sample Observability Strategy

An observability strategy doesn't need to be lengthy to be effective. For example, consider the following simple plan as a starting place for your organization.

Time	Task	Owner	Details
9:00 AM	Start-of-day status check	Arya Backups: Melissa, Mark	Review the hourly notification emails that came in overnight or the Observability pages and determine the following information: • Did any integrations experienced errors overnight? • Are all connectivity agents are up and running?
10:00 AM	**Business-critical monitoring** Monitor the business-critical Order Processing integration	Rajan Backups: Arya, Sneha	This integration runs twice every weekday. Review the hourly notification email or the Observability pages and determine the following information: • Is the integration running without issues? • Does the performance of the integration meet your organization's requirements? • Are all connectivity agents are up and running?
11:00 AM	Hourly connectivity agent check	Arya Backups: Melissa, Mark	Determine whether all connectivity agents are up and running.
12:00 PM	Hourly connectivity agent check	Arya Backups: Melissa, Mark	Determine whether all connectivity agents are up and running.

Copyright © 2022, 2026, Oracle and/or its affiliates.

Time	Task	Owner	Details
1:00 PM	Midday status check and integration count check	Arya Backups: Melissa, Mark	Review the hourly notification email or the Observability pages and determine the following information: • Are all integrations running without issues? • Are all connectivity agents are up and running? • Have the expected 120 integration instances run so far today?
2:00 PM	Hourly connectivity agent check	Arya Backups: Melissa, Mark	Determine whether all connectivity agents are up and running.
3:00 PM	Hourly connectivity agent check	Arya Backups: Melissa, Mark	Determine whether all connectivity agents are up and running.
4:00 PM	**Business-critical monitoring** Monitor the business-critical Order Processing integration	Rajan Backups: Arya, Sneha	This integration runs twice every weekday. Review the hourly notification email or the Observability pages and determine the following information: • Is the integration running without issues? • Does the performance of the integration meet your organization's requirements? • Are all connectivity agents are up and running?
5:00 PM	End-of-day status check and integration count check	Arya Backups: Melissa, Mark	Review the hourly notification email or the Observability pages and determine the following information: • Are all integrations running without issues? • Are all connectivity agents are up and running? • Have the expected 260 integration instances run so far today?

About Building Your Own Observability Tools

The observability tools in Oracle Integration provide robust capabilities for monitoring your business. You can use Oracle Integration's REST APIs to provide the same monitoring capabilities in your own observability tools.

When to Build Your Own

The monitoring capabilities offer insights into only the automation solutions that you create in Oracle Integration. However, your organization might prefer a monitoring tool that tracks work across multiple business applications.

In such cases, be aware that all of the monitoring capabilities that are available in the user interface are also available from the Oracle Integration REST APIs. If needed, you can create custom monitoring tools, such as custom user interfaces or scripts that run reports.

Review the Monitoring REST APIs

See Monitoring REST Endpoints in *Developer API for Oracle Integration 3*.

About Oracle Integration

Oracle Integration is a complete, secure, but lightweight automation 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

Copyright © 2022, 2026, Oracle and/or its affiliates.

premises. Oracle Integration provides secure, enterprise-grade connectivity regardless of the applications you are connecting or where they reside.

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

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

Start Oracle Integration

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

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

To access Oracle Integration:

1. In a web browser, enter the URL of your instance that you received from your administrator.
2. On the login page, enter your user name and password.

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

   See Get Familiar with the Home Page in *Getting Started with Oracle Integration 3*.

Keyboard Shortcuts in the Integration Canvas

Use keyboard shortcuts to navigate and select options in the integration canvas in Oracle Integration.

> **Note**
>
> These keyboard shortcuts are available only in Oracle Integration 3. They are not supported in Oracle Integration Generation 2.

Table 2-1 General Keyboard Shortcuts

Task	Keyboard Shortcut
Open keyboard shortcuts reference	Shift + ?

Copyright © 2022, 2026, Oracle and/or its affiliates.

Table 2-1 (Cont.) General Keyboard Shortcuts

Task	Keyboard Shortcut
Focus canvas	Shift + C
Change focus between tool bars	Shift + T
Navigate toolbar	Tab or 

Table 2-2 Canvas Keyboard Shortcuts

Task	Keyboard Shortcut
Save integration	Shift + S
Open integrations page	Shift + B
Zoom in	+
Zoom out	-
Scroll	Alt +  fn +  Home Page Up Page Down End
Navigate: Focus action to the left	
Navigate: Focus action to the right	
Navigate: Focus action above	
Navigate: Focus action after	
Go to: First action	1
Go to: Last action	2
Go to: Selected action	3
Design: Select or deselect action	Enter
Design: Open action menu	M
Design: Insert action before	B
Design: Insert action after	A
Design: Insert action into	I
Design: Cut	Ctrl + X
Design: Paste	Ctrl + V
Design: Cancel cut and paste	Esc

Copyright © 2022, 2026, Oracle and/or its affiliates.

3

Best Practices

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

Topics:

- Avoid Common Integration Pattern Pitfalls
- Design Best Practices

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, synchronize journal transactions or upload employee records into Oracle HCM Cloud).

Copyright © 2022, 2026, Oracle and/or its affiliates.

Antipattern	Why an Antipattern?	Best Practice
Use an invoke activity within a looping construct to call external APIs for every record.	• Downstream applications are receiving a large number of atomic requests. This puts the entire system under duress. • A usage-based pricing model translates to high costs.	• Leverage application capabilities to accept multiple records in a single request: – Salesforce: 200 records, Oracle Engagement Cloud/Oracle ERP Cloud: 100 records, Oracle 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) to send the file to the destination at the end.

Scheduled Job that Never Stops Trying to Process

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

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

Import an Externally Updated IAR File

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

Copyright © 2022, 2026, Oracle and/or its affiliates.

Antipattern	Why an Antipattern?	Best Practice
Update the IAR file externally and then import it into Oracle Integration.	• This can lead to metadata inconsistency and validation failures. • Activation failures may occur.	• Use the import map feature in Oracle Integration. – Keeps metadata consistent and leverages validation. – The mapper can show an imported map and allow it to be edited.

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 designed with 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.	• Consider moving completely to an asynchronous integration (fire and forget, asynchronous response). – The integration platform provides an acknowledgment to clients upon receiving the message. – The integration 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: When developers create integrations, they define their own connections pointing to the same application. This leads to many duplicate connections.

Copyright © 2022, 2026, Oracle and/or its affiliates.

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

Read Files with Many Records

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

Antipattern	Why an Antipattern?	Best Practice
Read the entire file in memory using the read file option and process it record by record.	• Large amounts of memory are consumed and other system processing is impacted. • Sequential processing does not leverage built-in map reduction capabilities.	• Download the file to the staged location using the download file option. – Stage files are secured and deleted at the completion of the instance. • Use the read file with segments option. – You can still model record level processing inside the segment processing. – The platform automatically processes segments in parallel. – The platform brings in only portions of the file to memory, when needed.

Integrations Running Unchanged Despite Changing Business Needs

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

Copyright © 2022, 2026, Oracle and/or its affiliates.

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

Design Best Practices

Understand the following best practices and integration pattern pitfalls to avoid before you design an integration.

- Create Schedule Integrations Mindfully
- Follow Guidelines for Synchronous Integrations
- Design Long-Running Integrations as Asynchronous
- Understand Timeouts in Synchronous Integrations
- Send Data to Third-Party Systems Efficiently

Create Schedule Integrations Mindfully

When there are too many schedule integrations configured, instances can become backlogged waiting for resources to become available or previous integration runs to complete. This can

Copyright © 2022, 2026, Oracle and/or its affiliates.

cause processing delays where some instances are in a waiting state longer than they must be and schedules may not start at the scheduled time.

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

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

You can convert a schedule integration to a REST Adapter trigger-based application integration. See Convert a Schedule Integration to a REST Adapter-Triggered, Application Integration.

If you absolutely require a large number of schedule integrations, and you encounter the previously described issue, the following design changes as a solution are recommended:

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

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

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

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

See Schedule Integrations Are Not Running on Time and When a Schedule Integration Instance Gets Terminated.

Follow Guidelines for Synchronous Integrations

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 schedule integrations internally use an asynchronous request response. Therefore, a synchronous integration using a schedule integration is an antipattern.
- A synchronous integration calling multiple services that span more than five minutes is reported as a stuck thread.

Copyright © 2022, 2026, Oracle and/or its affiliates.

Design Long-Running Integrations as Asynchronous

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.

Understand Timeouts in Synchronous Integrations

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 is behind a fire wall, 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.

Send Data to Third-Party Systems Efficiently

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, including using parallel processing in outbound integrations.

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).

Copyright © 2022, 2026, Oracle and/or its affiliates.

4

Install and Manage the Connectivity Agent

If you need to exchange messages between Oracle Integration and locally installed application resources that are in an on-premises network or private cloud, you must install the connectivity agent. You must also create an agent group to hold the connectivity agent.

Topics:

- Install the Connectivity Agent
- Manage the Connectivity Agent

More Resources:

- About Creating Hybrid Integrations: Learn about conceptual details and patterns for the connectivity agent.
- Troubleshoot the Connectivity Agent: Get help if you experience problems with the on-premises connectivity agent.

Watch a video to learn more:

Video

Install the Connectivity Agent

Ensure that the local host and network meet the requirements, and then you can create an agent group and install the connectivity agent.

Topics:

- Meet the Requirements for the Connectivity Agent
- Plan the Connectivity Agent Setup
- Workflows for the Connectivity Agent
- Create an Agent Group
- Download and Install the Connectivity Agent

Meet the Requirements for the Connectivity Agent

Copyright © 2022, 2026, Oracle and/or its affiliates.

System Requirements

Before installing the on-premises connectivity agent, make sure that the on-premises host, often a virtual machine, meets the system requirements.

Area	Requirements
Operating system	Install the connectivity agent on one of the following certified operating systems: • Oracle Linux 6.*x* • Oracle Linux 7.*x* • Oracle Linux 8.*x* • Oracle Linux 9 • RedHat Enterprise Linux 6.6 • RedHat Enterprise Linux 7.*x* • RedHat Enterprise Linux 8.*x* • Suse Linux Enterprise Edition 12 SP2 • Windows Standard Edition 2016 • Windows 2019 **Note** • The connectivity agent is not certified on Kubernetes (K8s). • IBM or Open JDK are *not* supported.
Memory	Provide a minimum of 8 GB memory with 4 GB of heap size dedicated for the connectivity 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.
Software	Install and use JDK version 17. You do not need to purchase a JDK license for the sole purpose of installing and/or running the connectivity agent. See JDK license and support for OIC Connectivity Agent (Doc ID 2611142.1). The JDK installation can be the same one as used with other products installed on the same host. However, separate JDK licensing may be required for any use of Java beyond the use of the connectivity agent. Ensure that the JDK installation is *not* modified for use with these other products.

Network Requirements

Area	More information
General requirements	If your organization requires one or more connectivity agents, make sure that the connectivity agent is compatible with your organization's operating procedures and requirements. For example, the connectivity agent must establish an outbound call from your intranet to Oracle Integration. Therefore, your organization must allow outbound calls from your intranet. If your organization has a sensitive network, any outbound communication to outside the firewall might be restricted. See About the Connectivity Agent.

Copyright © 2022, 2026, Oracle and/or its affiliates.

Area	More information
Authentication	You must make sure that your organization allows the connectivity agent to use OAuth 2.0 token-based authentication when invoking Oracle Integration endpoints. See Create an Agent Group.
Requirements for the local host on which you install the connectivity agent	The connectivity agent must be able to access the application that you're integrating with. Therefore, the host that you install the connectivity agent on must also have access to the application.
Requirements for agent groups	To install an agent group, you must allow egress from the agent network to Oracle Integration design-time and runtime and the identity domain or Oracle Identity Cloud Service.
Requirements for the person who installs the connectivity agent	The person who installs the connectivity agent must have the required permissions and roles. See Agent Download and Installation.
Authentication for Windows hosts	When you install the connectivity agent on a Windows host, the connectivity agent does not work when a proxy uses NT LAN Manager (NTLM) authentication. If you are using the connectivity agent on a Windows host, you must use an alternate authentication method.

Requirements for the Oracle WebLogic JMS Adapter	When you use the Oracle WebLogic JMS Adapter and the connectivity agent, you must use Oracle WebLogic Server version 12.2.1.4 and higher. Prior versions of Oracle WebLogic Server are not supported with the connectivity agent and this adapter.
	To determine your version of Oracle WebLogic Server:
	1. Enter the Oracle WebLogic Server URL. For example:
	`http://abcweblogic.oracle.com:7001/console`
	2. At the bottom left, find the version. For example:
	`WebLogic Server Version: 12.2.1.4.0`

Plan the Connectivity Agent Setup

Use the connectivity agent to configure access to resources residing in an on-premises network. When planning the agent setup, determine the number of agent groups to create, the number of connectivity agents in each agent group, and the number of active agents in each group.

To learn more about the connectivity agent, see About Connecting to On-Premises Applications .

Decision	More information
Number of Agent Groups	If you are using the connectivity agent, you must create at least one agent group.
Number of Connectivity Agents in an Agent Group	You can associate either one or two connectivity agents with each agent group.
Number of Active Connectivity Agents	In each agent group, either one or two agents can be active at a time.

Copyright © 2022, 2026, Oracle and/or its affiliates.

Number of Agent Groups

When planning the number of agent groups to create, keep in mind several points.

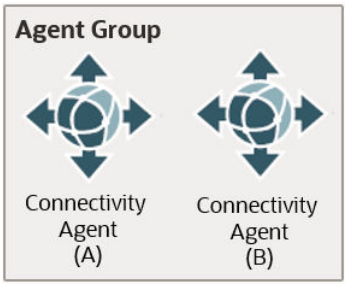
- You can associate up to two connectivity agents per agent group.
- You can associate each connectivity agent with only one agent group.
- You can create up to five agent groups per Oracle Integration service instance.

  For example, one service instance could have two agent groups, one with two associated agents, and one with one associated agent.

  Some organizations create separate agent groups for each line of business.
- If you have multiple Oracle Integration service instances, each service instance requires its own agent groups and connectivity agents.

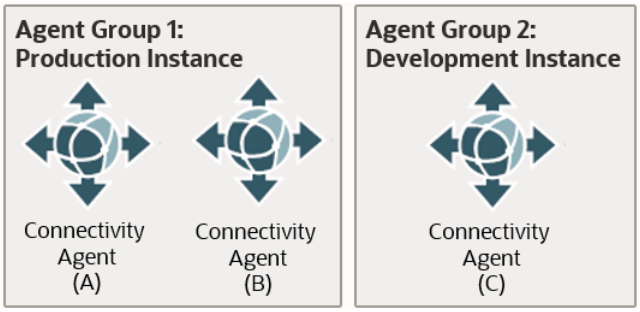
Number of Connectivity Agents in an Agent Group

You can associate one or two connectivity agents with each agent group. The number of connectivity agents to associate with an agent group depends upon your requirements for throughput and resiliency.

Setup	Why choose this configuration?
Two agents in an agent group (recommended) 	• The connectivity agents are highly available and offer better throughput. • You improve resiliency by eliminating the potential for a single point of failure. For instance, if a connectivity agent becomes unavailable, the second connectivity agent automatically picks up all of the work and your integrations continue running. If you opt for this configuration, you must also choose whether both agents are active at one time. See Number of Active Connectivity Agents.
One agent in an agent group (not recommended for production environments) 	This configuration introduces risk. A single connectivity agent creates a single point of failure. However, under the following circumstances, you might decide that this risk is acceptable: • Improved throughput and resiliency aren't required. • You want to limit the resources that you commit. • You want to maintain minimal complexity. For instance, you might opt for a single connectivity agent for a development environment while installing multiple connectivity agents in a production environment.

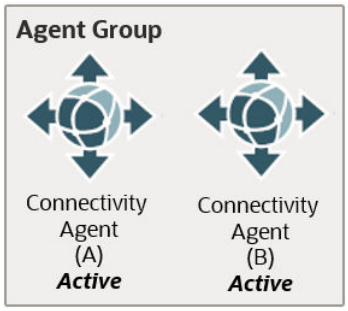
You can also mix and match. For instance, associate two connectivity agents with one agent group, and associate a different connectivity agent with a different agent group.

Copyright © 2022, 2026, Oracle and/or its affiliates.

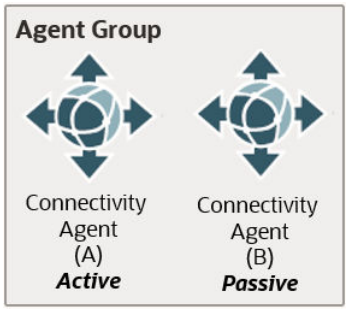


Number of Active Connectivity Agents

When you create an agent group that holds two connectivity agents, you must specify whether one or both of its connectivity agents are active at a time using the **Use only one agent at a time** setting.

Setting	Description	Performance impact
Use only one agent at a time is not enabled	Both agents remain active at all times, sharing incoming requests and polling responsibilities. 	This setup can result in improved performance when compared to agent groups in which **Use only one agent at a time** is enabled.

Copyright © 2022, 2026, Oracle and/or its affiliates.

Setting	Description	Performance impact
Use only one agent at a time is enabled	The first connectivity agent that you install or start is identified as the active agent. This agent handles all incoming requests and polling responsibilities. For example, a File Adapter polls for files in a specific directory. The second agent that you install or start is identified as the passive agent. This agent doesn't handle any incoming requests or polling responsibilities. Instead, it runs in standby mode, ready to take over if the primary agent fails. If the active agent fails, the passive agent becomes active and handles all incoming requests and polling responsibilities. This agent remains active even after the initially active agent recovers from failure. The recovered agent is now passive and remains in standby mode. 	This setup can result in lower performance when compared to agent groups in which **Use only one agent at a time** is *not* enabled.

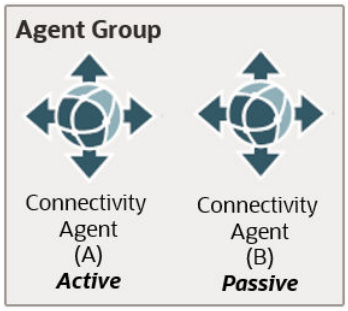
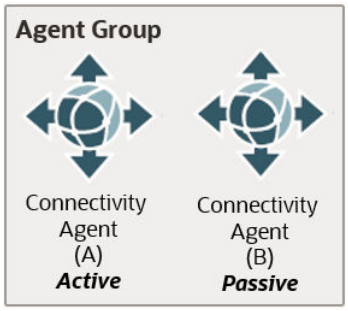
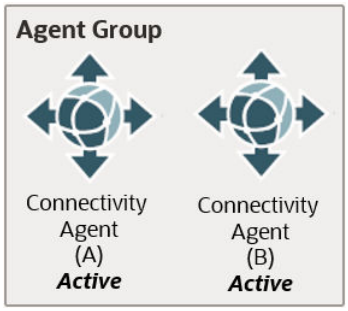
Again, you can create multiple agent groups with different active settings to meet your requirements.



Adapter-Specific Guidance

Several adapters provide an optimized experience when an agent group has only one active agent at a time. Additionally, one adapter requires only one active agent at a time.

Copyright © 2022, 2026, Oracle and/or its affiliates.

Adapter	Guidance
File Adapter and FTP Adapter	**Required setup for all service instances:** Enable **Use only one agent at a time**. Resiliency during agent failure is improved. These adapters can also work with a connectivity agent that is the only agent in an agent group.  See File Adapter Capabilities in *Using the File Adapter with Oracle Integration 3* and FTP Adapter Capabilities in *Using the FTP Adapter with Oracle Integration 3*.
MLLP Adapter	**Required setup for all service instances:** Enable **Use only one agent at a time**. This adapter can also work with a connectivity agent that is the only agent in an agent group.  The MLLP Adapter allows only one connectivity agent to be active at a time. Therefore, connections that are based on the MLLP Adapter must use an agent group that has enabled this setting, identifiable by the **Singleton** label.
All other adapters	**Preferred setup for all production service instances:** Leave **Use only one agent at a time** disabled. 

Workflows for the Connectivity Agent

Use the workflows to understand how to install and use the connectivity agent.

Copyright © 2022, 2026, Oracle and/or its affiliates.

Associate Multiple Connectivity Agents with an Agent Group

Follow these steps to install two on-premises connectivity agents and associate them with an agent group.

Step	Task	More information
1	Before installing the connectivity agent, ensure that the local host and network meet the requirements.	See Meet the Requirements for the Connectivity Agent.
2	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).
3	Create an agent group	You cannot use an existing group. See Create an Agent Group.
4	Download and install the connectivity agent on one host	Ensure that you specify the agent group identifier of the new agent group. See Download and Install the Connectivity Agent. See Download and Install the Agent.

Copyright © 2022, 2026, Oracle and/or its affiliates.

Step	Task	More information
5	Follow the same steps to download and install the connectivity agent on a second host	You can install multiple agents on the same host. However, to use high availability capabilities, install the second agent on a second host. That way, if one host goes down, agent processing continues running on the other host. Create a horizontal cluster to achieve high availability. Installing the agent on the same virtual machine (VM) does not guarantee high availability. • 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. • 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. **Note** Specify the *same* agent group identifier and Oracle Integration URL in the `InstallerProfile.cfg` file as with the first installation.
6	Confirm the installation	• After installation completes, go to the Agents page and note that two agents are associated with the same agent group. • Go to the Monitor agents page and note the same information. In this case, both agent instances are running.

Associate One Connectivity Agent with an Agent Group

Follow these steps to install the on-premises connectivity agent and associate a single agent with an agent group.

Copyright © 2022, 2026, Oracle and/or its affiliates.

Step	Task	More information
1	Before installing the connectivity agent, ensure that the local host and network meet the requirements.	See Meet the Requirements for the Connectivity Agent.
2	Create an agent group	See Create an Agent Group.
3	Download and run the on-premises connectivity agent installer on your host	During installation, you associate the on-premises connectivity agent with the agent group. See Download and Install the Connectivity Agent.

Use the Connectivity Agent in an Integration

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

Step	Task	More information
1	Create an adapter connection in Oracle Integration and associate the connection with the connectivity agent group	If an adapter supports using the connectivity agent, you must select the agent's group while creating a connection that is based on the adapter. See Create a Connection with an Agent Group. For adapters that don't support the connectivity agent, you don't select an agent group. See Create Connections.
2	Design an integration that uses this connection	See Create an Integration.
3	Activate the integration	See Activate an Integration.

Create an Agent Group

Before installing the connectivity agent, you must create an agent group in Oracle Integration. For a single Oracle Integration instance, you can create up to five agent groups. You can associate up to two connectivity agents with an agent group.

Creating the agent group also creates the necessary artifacts required for message exchange, including an OAuth client application in your identity domain environment. The OAuth client application allows the connectivity agent to use OAuth 2.0 token-based authentication when invoking Oracle Integration endpoints. You can edit the connection name after creation.

1. Plan your connectivity agent setup, including the number of agent groups to create and the number of agents in each group.

   See Plan the Connectivity Agent Setup.
2. In the navigation pane, click **Design**, then **Agents**.
3. Click **Create**.

   The Create agent group panel opens.
4. Enter the following information, then click **Create**.

Copyright © 2022, 2026, Oracle and/or its affiliates.

Field	Description
Name	Provide a meaningful name so that others can understand the agent name. The name must be unique among all agent names in the system. The name can consist of the following: • Letters (A-Z, a-z) • Numbers (0-9) • Spaces ( ) • Special characters (_ -) The name must not begin or end with a space and cannot be longer than 50 characters.
Identifier	Accept the default identifier value or change it, if necessary. The identifier is initially the same as the agent group name you provided, but in upper case. When you install the agent, the identifier value is automatically included in the `InstallerProfile.cfg` file. **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.
Description	Provide a meaningful description so that others can understand the responsibilities of the agent group.
Use only one agent at a time	If you plan to associate two connectivity agents with the agent group, you must specify whether *one* or *both* agents are active at any given time. The preferred setup for a production service instance is associating two agents with the agent group and leaving this setting *disabled*. However, if you use some specific adapters, you might need or want to enable this setting. Spend a few minutes planning your setup because you can't change this setting later. (But you can delete an agent group and create it again.) • For general guidance on your agent setup, see Plan the Connectivity Agent Setup. • For specific guidance on this setting, see Number of Active Connectivity Agents. **Notes:** • If you select **Use only one agent at a time** and then associate only one connectivity agent with the agent group, the agent group works as if you never selected the option. You must associate a second agent with that agent group to achieve active-passive agent functionality. • You cannot change the **Use only one agent at a time** setting after creating an agent group. However, you can delete the agent group and create a new agent group.

This action automatically creates an OAuth client application in the identity domain. The application name takes the following form:

agent_group_name-instance_identity_provider_name

This enables the connectivity agent to use OAuth 2.0 token-based authentication when invoking endpoints. In Oracle Integration 3, each agent group has its own OAuth client application.

If you enabled **Use only one agent at a time**, the label **Singleton** appears next to the agent group name.

Copyright © 2022, 2026, Oracle and/or its affiliates.

Download and Install the Connectivity Agent

Download and install the connectivity agent on a local host, which is typically a computer or virtual machine. If needed, you can run the agent installer as a background process and install a security certificate on the local host.

Installation Requirements

- Install the connectivity agent to the instance with the minimal privilege. Do not install it as `root`.
- Make sure you install the connectivity agent with the privileges that it needs to run.
- Configure the connectivity agent to scan only the specified directories at the specified frequency.
- Verify that ports or proxy are set up correctly to only allow for the connectivity agent connection to Oracle Cloud Infrastructure.
- After successfully installing the connectivity agent, delete the installation key.

Download and Install the Agent

During installation, associate the connectivity agent with the agent group identifier you generated when creating an agent group in Oracle Integration. Ensure that you restrict access to the folder in which you install the connectivity agent to only those users who need to stop and start the agent.

The connectivity agent comes preseeded with the credentials that it needs to access your Oracle Integration instance.

Prerequisites:

- Create one or more agent groups. See Create an Agent Group.
- If you restrict access to the internet based on the target IP address, you must allowlist the Oracle Integration runtime and design-time IP addresses and the identity domain IP address before installing the connectivity agent.
- You must ensure that the directory paths to poll/read/write files are accessible to any agents in an agent group.

To download and install the agent:

1. Ensure that you have the required roles and permissions to install the connectivity agent.
   - You must have the Java execute permission to install and restart the agent.
   - You must have the *Service*Administrator role to download the agent.

     See What Users Can Do in the Integrations Design Section by Role of *Provisioning and Administering Oracle Integration 3*.
2. Create a directory for connectivity agent installation on your on-premises host.

   Note the following restrictions:
   - Do not install the agent in a directory path that includes `/tmp`.

     The agent must never be installed in `/tmp`. Per the Filesystem Hierarchy Standard version 3.0, `/tmp` is meant for temporary files. Even though the install and agent work,

Copyright © 2022, 2026, Oracle and/or its affiliates.

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.

3. In the navigation pane, click **Design**, then **Agents**.
4. Click **Download**, then **Connectivity Agent**.
5. Download the connectivity agent installer to a directory on your on-premises host.
6. Extract `oic_conn_agent_installer.zip`.
7. 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 Connectivity Agent.

8. Hover over the agent group.
9. Click **Actions** ⋯, then select **Download config**.

   This step downloads a preconfigured `InstallerProfile.cfg` file to use during installation. The following message is displayed.

   ```
   BootStrap configuration downloaded for Agent Group agent_group_name.
   Replace the file in agent install location before proceeding with agent
   setup.
   ```

10. Replace the existing `InstallerProfile.cfg` template file in the `oic_conn_agent_installer` directory that was created when you extracted the agent installation file in Step 5 with the preconfigured `InstallerProfile.cfg` file you downloaded in Step 9.

    The preconfigured `InstallerProfile.cfg` file automatically includes values for all required parameters such as `oic_URL` and `agent_GROUP_IDENTIFIER` and OAuth 2.0 token-based authentication parameters such as client ID, client secret, and scope. This eliminates the need to manually specify values for these parameters.

    a. Optionally set proxy parameter values. These parameters are only required if the connectivity agent is used with a proxy in the on-premises environment.

    b. If you have multiple hosts that must be configured in a nonproxy host environment, you must separate each IP address or host with a pipe symbol (`|`) in the `proxy_NON_PROXY_HOSTS` parameter. For example:

    ```
    proxy_NON_PROXY_HOSTS=localhost|127.0.0.1|*.myorg.com
    ```

    c. If you use a proxy user for Windows, the user name must include the MS domain name in front of the user name, along with double backslashes before the user name (for example, *`MS_domain`*`\\`*`username`*). If you do not specify the double backslashes, you receive a `407 Proxy Authentication Required` error.

11. Set the `JAVA_HOME` property to the location of the JDK installation.

Copyright © 2022, 2026, Oracle and/or its affiliates.

12. Set the `PATH` property. For example, if `csh` is your shell:

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

13. Run the connectivity agent installer from the command prompt. If you want to copy and paste this command, ensure that it does not have any special characters.

    ```
    java –jar connectivityagent.jar
    ```

14. Wait for a successful installation message to appear.

    ```
    Done with Agent installation & configuration... Starting agent for message
    processing.
    Agent started successfully... listening for new messages...
    ```

    The agent group associated with this connectivity agent now has its own `InstallerProfile.cfg` file and OAuth client application in the identity domain. If you delete the agent group, the OAuth client application is also deleted in Oracle Identity Cloud Service.

    If errors occur, review the agent diagnostic logs.

    See Monitor Agents.

After Installation

After installing the connectivity agent, you can create a connection that uses the agent, run the agent as a service, and install a certificate.

- When you create a connection that uses a connectivity agent, associate its agent group with your adapter connection in the **Access type** section of the Connections page. See Create a Connection with an Agent Group.
- To run the connectivity agent as a service, see Run the Connectivity Agent as a Service.
- Some services require you to install a certificate on the agent host. If you need to install a certificate, see Install a Certificate on the Agent Host.
- There is an explicit association between the agent group and the preconfigured `InstallerProfile.cfg` file you downloaded in Step 9. You *cannot* replace the specific `InstallerProfile.cfg` file associated with one agent group with a different `InstallerProfile.cfg` file associated with a different agent group. Doing so results in an error during connectivity agent startup. At that point, you can also not revert back to the original `InstallerProfile.cfg` file.

  You must create a new agent group, download the connectivity agent, download the `InstallerProfile.cfg` file, replace the existing `InstallerProfile.cfg` file with the downloaded file, and install the connectivity agent. See Download and Install the Connectivity Agent.

  For more details about the specific errors, see Connectivity Agent Fails to Start After Changing the InstallerProfile.cfg File Associated with the Agent Group.

Copyright © 2022, 2026, Oracle and/or its affiliates.

Manage the Connectivity Agent

Get help managing a connectivity agent.

What Do You Need to Do?

Requirement or event	Next steps
You want to understand how updates work for the connectivity agent	Updates Are Automatic for the Connectivity Agent
You want to run the connectivity agent as a service	Run the Connectivity Agent as a Service
A service requires you to install a certificate on the agent host	Install a Certificate on the Agent Host
You need to restart the connectivity agent For example, it might have gone down, or a Support agent instructed you to restart it	Restart the Connectivity Agent
Configure the connectivity agent to use mTLS to access on-premises endpoints	See Configure the Connectivity Agent to Use mTLS to Access On-Premises Endpoints
The connectivity agent becomes unavailable due to a host failure	Recover the Connectivity Agent in Case of Host Failure
You created an agent group in error or need to remove an unused agent group	Delete an Agent Group
You upgraded from Oracle Integration Generation 2 and need to upgrade JDK on the agent host	Upgrade the JDK 11 Agent to the JDK 17 Agent
Something else	Troubleshoot the Connectivity Agent.

Updates Are Automatic for the Connectivity Agent

When an update becomes available for the connectivity agent, the local host for the agent gets the update automatically.

- About the Update Process
- Check the Status of the Connectivity Agent When It Restarts on Windows

About the Update Process

When Oracle Integration is updated, the connectivity agent is also updated within a four-hour window. Oracle doesn't send a separate alert for agent updates. 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.

The connectivity agent update 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 extracted.
4. A backup directory is created.
5. A backup copy is made of your existing installation in the new backup directory.
6. Older artifacts are replaced in the `agenthome` directory of your installation.

Copyright © 2022, 2026, Oracle and/or its affiliates.

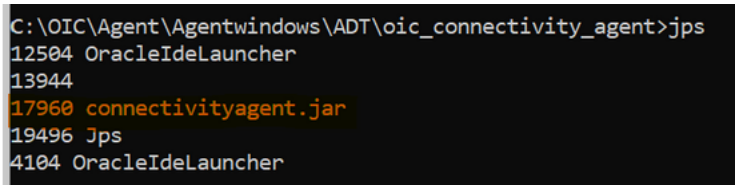
7. The endpoints are quiesced.
8. The agent is shut down and restarted.
   You are notified of update success.

> **Note**
>
> After agent update 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 update, the agent restart reloads the classes (after agent update) without shutting down the JVM. As a workaround, a manual restart of the agent is required. See Restart the Connectivity Agent.

Check the Status of the Connectivity Agent When It Restarts on Windows

The connectivity agent is automatically restarted after update completes. During the restart on Windows, the connectivity agent runs in the background.

To view connectivity agent restart progress, you can look in the PID file (`oic_connectivity_agent/pid`) for the process ID, run the `jps` command from the prompt (as shown below), or view the connectivity agent update logs located in `agenthome/logs/agent-diagnostic0.log` and `agenthome/logs/agent-upgrade.log`.



Run the Connectivity Agent as a Service

When you run the connectivity agent as a service (using `java -jar connectivityagent.jar`), the process is tied to the terminal window in which you are working and ends when the window is closed. If you want to run the process in the background, use one of the following options.

Run the Agent as a Service on Linux

- Perform one of the following tasks:
  - Run the agent process.

    For example:

    ```
    nohup java -jar connectivityagent.jar &
    ```

  - If you don't want to expose the password in `InstallerProfile.cfg`, perform the following steps:
    1. Enter `java -jar connectivityagent.jar` at the command prompt.
    2. Enter the user name and password when prompted.
    3. Enter `Ctrl+Z` to suspend the process.

Copyright © 2022, 2026, Oracle and/or its affiliates.

4. Enter `bg` (to run the process in the background).
5. Enter `jobs` to get the `jobid`.
6. Enter `disown -a %jobid` (from Step e) to disassociate the process from the owning shell.

Run the Agent as a Service on Windows

Running the connectivity agent as a service on Windows is *not* supported. You must be logged in to the Windows host. Otherwise, the agent stops working when you log off. Consider the following recommended options.

Option 1: Install the Agent Using a New Service Account

1. Create a new service account on Windows.
2. Use that account to log in to the Windows host and remain logged in with that account.
3. Install the connectivity agent on the Windows host.
4. Use that agent to communicate with the MS SQL Server or other endpoints accessed on Windows.

Option 2: Install on a Linux VM

Instead of installing the connectivity agent on a Windows host, install it on a Linux virtual machine that has access to the Windows systems with which that agent must interact.

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.p12`. Installing the certificate enables you to access hosts with self-signed certificates. A certificate is not typically needed.

> **Note**
>
> If you install a certificate *after* installing the connectivity agent, you must restart the agent.
> See Restart the Connectivity 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.

To install a certificate on the connectivity agent host:

1. Go to the `agenthome/agent/cert/` directory (`keystore.p12` is available here).
2. Run the following command:

   ```
   keytool -importcert -keystore keystore.p12 -storepass password -alias
   alias_name -noprompt -file certificate_file -trustcacerts
   ```

   Where:

Copyright © 2022, 2026, Oracle and/or its affiliates.

- `-storepass` *`password`*: The default, initial password for the agent keystore. Refer to your `keytool` documentation for the default `storepass` password. See keytool.
- `-alias` *`alias_name`*: Any name to uniquely identify the imported certificate in the keystore.
- `-file` *`certificate_file`*: Absolute path of the certificate file.

Restart the Connectivity 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 terminate it.
2. Wait at least 45 seconds before restarting the agent. This is because the agent monitoring framework waits for 45 seconds before marking the agent status as being down. If you start the agent before 45 seconds have completed, agent restart fails with the following error message:

   ```
   Agent is already running for this particular instance
   ```

3. Restart the agent based on your environment and production load size.

Copyright © 2022, 2026, Oracle and/or its affiliates.

Environment	Enter the Following Command

Production environments	It is recommended that you restart the agent with the `-XX:+HeapDumpOnOutOfMemoryError` parameter:
	
	`java -XX:+HeapDumpOnOutOfMemoryError -jar connectivityagent.jar`
	
	If the connectivity agent runs out of memory, this parameter by default ensures that the heap dump is stored in a `java_pid`*`pid`*`.hprof` file in the directory where the agent application is run.
	Based on production loads, it may sometimes be necessary to allocate a larger amount of heap size for the agent process. If you determine that the process must be allocated a larger heap size, tune the `-Xms` and `-Xmx` parameters accordingly.
	• `-Xms`*`heap_size`*`G`
	• `-Xmx`*`heap_size`*`G`
	For example, assume you want to assign a minimum of 2 GB and a maximum of 8 GB to the agent JVM.
	
	`-Xms2G -Xmx8G`
	
	Start the agent with the following parameters.
	
	`java -Xms2G -Xmx8G -jar connectivityagent.jar`
	
	You can specify the `-XX`, `-Xms`, and `-Xmx` parameters at the same time, if needed.

Nonproduction environments	`java –jar connectivityagent.jar`

You can restart the agent as a background process. See Run the Connectivity Agent as a Service.

Configure the Connectivity Agent to Use mTLS to Access On-Premises Endpoints

You can enable connectivity agent-based, mutual transport layer security (mTLS) to access on-premises endpoints with the REST Adapter.

See Configure Connectivity Agent-Based, mTLS Communication with the REST Adapter in *Using the REST Adapter with Oracle Integration 3*.

Copyright © 2022, 2026, Oracle and/or its affiliates.

Recover the Connectivity Agent in Case of Host Failure

If the host on which the connectivity agent is running fails and makes your environment unavailable, follow these steps to recover the connectivity agent. The failed host can be an on-premises machine or in Oracle Cloud Infrastructure Compute.

Note

- Perform regular backups of *AGENT_INSTALL_DIR*.
- 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 failure.

If your host fails and makes the environment unavailable, do the following;

1. Copy the *AGENT_INSTALL_DIR* directory from your last backup to the new host.
2. Change directories to the *AGENT_INSTALL_DIR* directory.
3. 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 backup and recovery are performed on a different host, the agent instance name does not change to reflect the new host on which the agent is running.

Copyright © 2022, 2026, Oracle and/or its affiliates.

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 deletes the internal artifacts that the agent relies on after agent group creation. The OAuth client application in the identity domain is also deleted.

1. Before deleting an agent group, ensure that you terminate the running agent at the operating system command prompt:

   ```
   kill -9 agent_PID_number
   ```

   The *agent_PID_number* can be obtained from the *AGENT_INSTALL_DIRECTORY*/pid file.
2. In the navigation pane, click **Design**, then **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 click the number (it can be 1 or 2) and first delete those agent instances. If any connections are using the agent, you cannot delete the agent instance.
4. Hover over the agent to delete.
5. Click **Actions** ⋮, then select **Delete**.
6. Select **Delete** when prompted to confirm.

Configure the Connectivity Agent to Use mTLS to Access On-Premises Endpoints

You can enable connectivity agent-based, mutual transport layer security (mTLS) to access on-premises endpoints with the REST Adapter.

See Configure Connectivity Agent-Based, mTLS Communication with the REST Adapter in *Using the REST Adapter with Oracle Integration 3*.

Upgrade the JDK 11 Agent to the JDK 17 Agent

If you need to upgrade from JDK 11, perform the following steps.

1. Install JDK 17.
2. Stop the existing connectivity agent.
3. Set `JAVA_HOME` and `PATH` to `JDK17`.
4. Restart the connectivity agent.

Copyright © 2022, 2026, Oracle and/or its affiliates.

5

Design and Monitor Integrations in Projects

You can design, manage, and monitor integrations from a single, self-contained workspace known as a project. This eliminates the need to navigate to other parts of the Oracle Integration user interface (which is required with standalone integrations or integrations in packages).

Topics:

- Design a Project
- Manage a Project
- Monitor Integrations in a Project

Before using integration projects, be familiar with the conceptual details. See Get Started with Projects.

Watch a video to learn more:

Video

Design a Project

Create and design an automation solution, including all of its components, in a project.

Topics:

- Control Who Can Edit, View, and Monitor in a Project
- Create or Import a Project
- Create a Project By Selecting an Accelerator or Recipe that the AI Suggests
- Copy an Integration into a Project
- Use AI to Create an Integration
- Use AI to Edit an Integration
- Use AI to Create Integrations Based on Integrations from the Integration Store
- Add and Share a Connection Across a Project
- Add and Share a Lookup Across a Project
- Create, Publish, and Subscribe to Events in a Project
- Create an AI-Powered Event Description
- Create an AI-Powered Project Description
- Create Robots in a Project
- Create Healthcare Integrations in a Project
- Create B2B Integrations in a Project
- Create Decision Models in a Project

Copyright © 2022, 2026, Oracle and/or its affiliates.

- Create Human in the Loop Workflows in a Project
- Use AI to Generate Project Documentation
- Use AI to Generate Integration Documentation

Control Who Can Edit, View, and Monitor in a Project

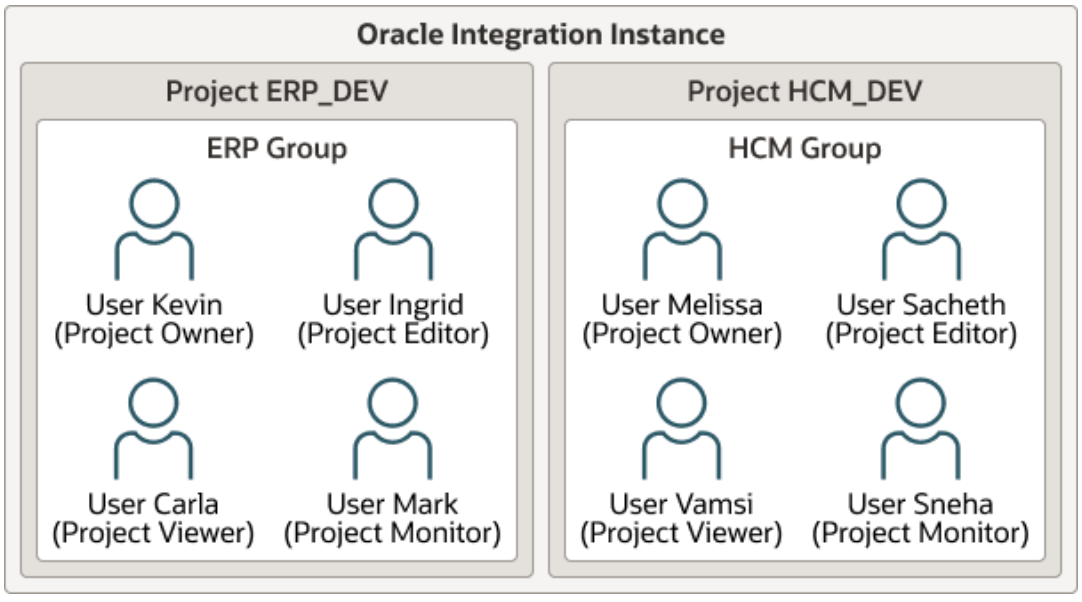
You can control the users and groups that edit, view, and monitor a project with role-based access control (RBAC). You select who can access a set of project resources while limiting (or hiding) those same resources from others (for example, providing an HCM group of users with access to some project resources while restricting an ERP group of users from accessing those same project resources).

- Capabilities
- RBAC-Enabled Projects FAQ
- Select Who Can Edit, View, and Monitor a Project
- Best Practices
- RBAC-Enabled Project Use Case

Capabilities

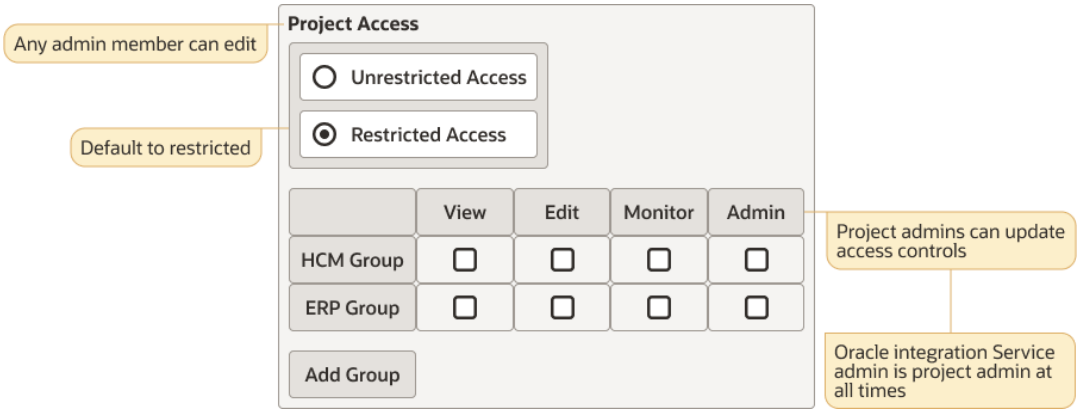
RBAC-enabled projects provide the following capabilities:

- Support project isolation
  Project isolation enables multiple users and groups to work in the same Oracle Integration instance, with access to only the projects to which they are assigned. This enables multiple groups (for example, ERP and HCM) to work on the same Oracle Integration instance and only edit, view, and monitor project resources assigned to their group.



- Support for setting restriction levels within a project
  You can assign users and groups to specific project permissions based on their needs. For example, you assign project editor permissions to some users and groups to create and edit projects and invoke integrations, while you assign project monitor permissions to other users and groups to only monitor integration instances at runtime.

Copyright © 2022, 2026, Oracle and/or its affiliates.



Project permissions are similar to the Oracle Integration service roles (for example, ServiceDeveloper, ServiceMonitor, ServiceInvoker, and others) that the user with the ServiceAdministrator role first assigns to other users and groups from the Oracle Cloud Console in an identity domain environment. See Manage Access and Assign Roles in *Provisioning and Administering Oracle Integration 3*.

Project permissions are essentially a layer of permissions on top of service roles.

Once you assign these service roles to users and groups, the user with the ServiceAdministrator role or the project owner can assign these users and groups with project permissions inside a specific project. Project access is restricted based on a user's service role, plus their project permissions.

The following project permissions are available:

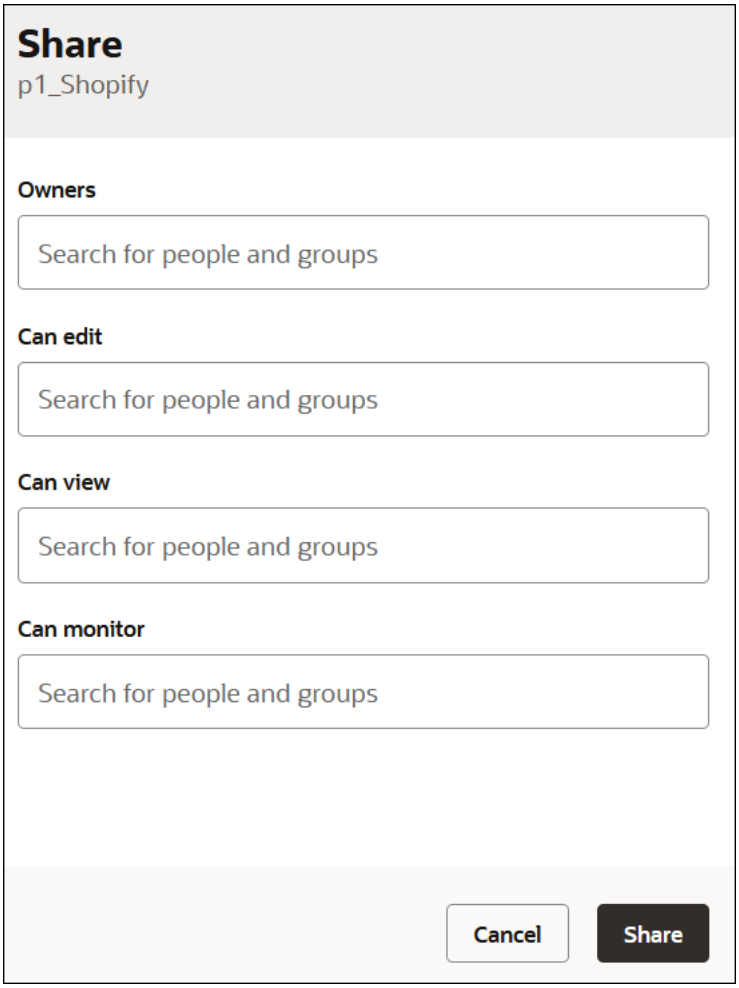
Project Permission	Description
Owner	Enables you to perform all available actions on the project (as a project-level super user), including associating groups/users with each project permission. The project creator (owner) is automatically set as a project-level super user. Modification of project permissions is restricted to the user with the ServiceAdministrator role (system super user) and the project owner (project-level super user). Only a user with the ServiceAdministrator role or a ServiceDeveloper who is an owner/editor of a project can export the project to another Oracle Integration instance.

Copyright © 2022, 2026, Oracle and/or its affiliates.

Project Permission	Description
Editor	Enables you to edit certain metadata in a project, create projects, and create, remove, edit, and run project resources. You can view runtime metrics and perform actions against runtime instances. You cannot modify the assigned project permissions.
Viewer	Enables you to view certain metadata in a project and discover and view project resources. You cannot add, remove, edit, run, export, or import a project or project resources.
Monitor	Enables you to monitor runtime metrics in a project and perform actions (for example, abort, retry, and discard) against runtime instances. You cannot access the **Design** and **Deploy** tabs of a project.
None	A user *without* any project permissions (owner, editor, viewer, monitor) can see the project name when viewing the main Projects page. However, the user cannot do the following: – Access the project details page to retrieve any details or query details with a REST API. – View instance monitoring details (for example, integration statistics, instances, errors, and more) outside the project under the **Observability** tab.

- Support RBAC for projects created prior to version 23.06
  After version 23.06 patching, projects created in versions 23.04 and earlier automatically include the Share panel inside the project. These earlier projects automatically inherit the **Everyone** project permission. This setting makes everyone with the correct service role an owner who can edit, view, and monitor project resources. You can then restrict access if necessary by assigning users and groups to each project permission. If you are happy with your pre-23.06 permissions, you can also leave things as they are. Assigning specific project permissions is not required.
- Enforce the rule that a user's assigned service roles (ServiceDeveloper, ServiceMonitor, ServiceInvoker, and others) *always* takes precedence over their assigned project permissions. For example:
  - Assume a user with the ServiceMonitor service role is assigned the editor project permission in the **Can edit** field.

Copyright © 2022, 2026, Oracle and/or its affiliates.



However, because this user is only assigned the ServiceMonitor service role, their permissions in the project are not elevated to allow them to create, update, or delete project resources; they can only monitor the project.

- Support projects on a single Oracle Integration instance that are both unrestricted and restricted for use by one or more groups.
  When you create a project, import a project, or convert a package to a project, you can select the **Anyone can edit, view, and monitor** check box if you want to make a project unrestricted to all users with the correct service roles. See Create or Import a Project and Convert a Package to a Project.
- Allow users with the ServiceAdministrator role to perform all actions in a project, regardless of project permissions.
- Filter the ability to call REST APIs.
  For example, a user in an ERP group who invokes the `/ic/api/integration/v1/monitoring/instances` API only returns flow instance data from integrations attached to their group.
- Support having zero or more project owners.
  If zero, then project owner defaults to the user with the ServiceAdministrator role. Only administrators have the unique privilege to update project-level access control. Project owners are essentially project-level super users.

Copyright © 2022, 2026, Oracle and/or its affiliates.

RBAC-Enabled Projects FAQ

- **Can you import a standalone integration into an RBAC-enabled project?**
  Only a user with the ServiceAdministrator role or a user with the ServiceDeveloper role that has the owner or editor project permission can import a standalone integration into an RBAC-enabled project.
- **Are project permissions moved from development to test to production environments in a CICD pipeline?**
  Project permissions are not moved to test and production. The users and groups have different permissions in different environments. However, if a user sets up permissions in production, they are carried forward for future project updates. When a project is imported for the first time, it is owned by whoever imported it. That user must set the project permissions. If a project is re-imported, the permissions are left alone.
- **Do standalone (non-project) integrations use RBAC?**
  Integrations created outside of a project do not support RBAC. Non-project or global resources are restricted by existing service roles.
- **Are there limits on the number of users and groups you can assign to a project?**
  You can assign a maximum of five users and/or groups (any combination) to each of the project roles.
- **Do I need to use RBAC with my projects**

  No, using RBAC is optional. If you do not want to use it, you can ignore **Share** in the project.
- **Can members of a project see the other members and their project permissions?**
  Only a user with the ServiceAdministrator role or a ServiceDeveloper who is the project owner can see other assigned members. For these two conditions only, **Share** is editable.
- **Do project permissions take precedence over service roles?**
  No. Service roles (ServiceDeveloper, ServiceMonitor, ServiceInvoker, and others) *always* takes precedent over the assigned project permissions. For example, if a user with the ServiceMonitor service role is assigned the editor project permission, they cannot access the **Design** and **Deploy** tabs of a project.
- **Can restrictions be enforced at the REST API level? For example, can lookup update/delete only be assigned to a specific user and restricted from another?**
  No. If both users have editor permissions on one type of resource, they have permissions on all types of resources.
- **If you have the editor permission in project A and want to invoke a child integration (set as publicly available) in project B, but you only have the monitor permission (or perhaps, no permission) in project B, can you do so? Or do you need to update your permissions in project B to match those in project A?**
  You can discover and invoke the child integration in project B without setting any additional project permissions.
- **Any special permissions for project deployment? Can only a user with the edit permission create a deployment?**
  To create a project deployment, you must have the ServiceAdministrator role or the ServiceDeveloper role plus the project owner/editor permission.
- **Can a user create groups and assign permission to groups rather than assigning users?**
  Yes, the entries can be identity domain users or groups.
- **Can a user with the ServiceMonitor service role see all integrations?**

Copyright © 2022, 2026, Oracle and/or its affiliates.

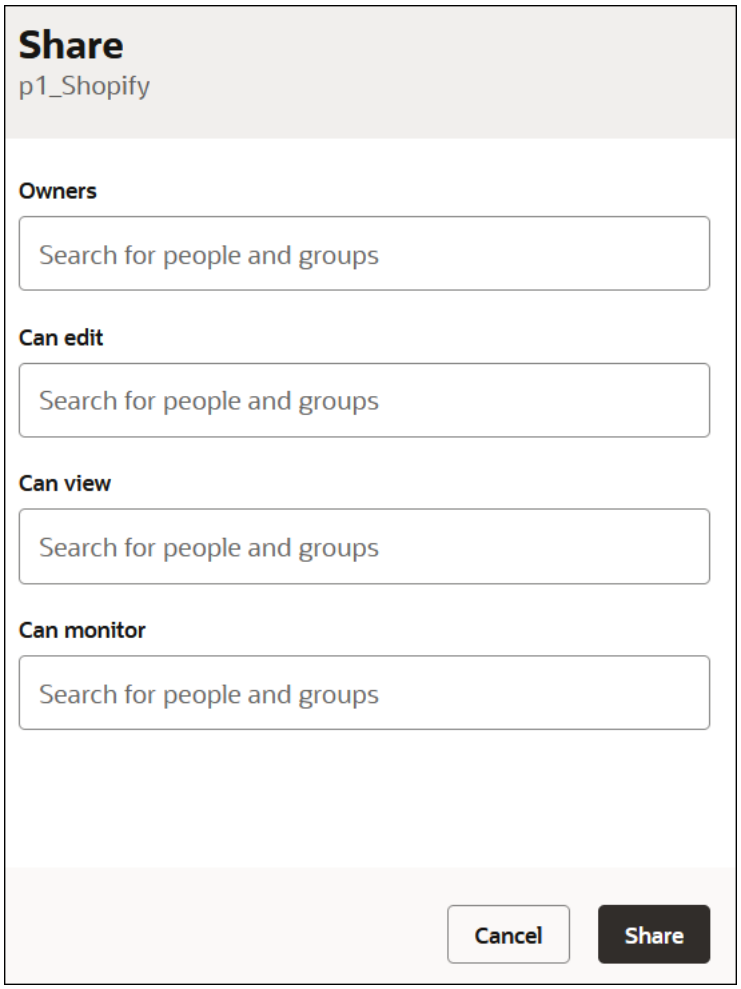
From monitoring pages under **Observability**, the user with the ServiceMonitor service role cannot see all the integrations. They cannot see integrations that are part of a project on which they do not have any permissions. The same applies to integration instances.

- **Are project roles applied in a production system?**
  Project roles are still applicable. For example, the HCM_monitor group can monitor HCM projects, but not the finance project.
- **What can a user with no project permissions do?**
  They can see the existence of the project on the main Projects page, but cannot perform any actions and are not allowed to access the project details page. They also cannot see monitoring resources (for example, integration statistics, instances, errors, and others) for the project under the **Observability** tab.

Select Who Can Edit, View, and Monitor a Project

The user with the ServiceAdministrator role or the project owner can select the users and groups who can view, edit, and monitor the page.

1. In the navigation pane, click **Projects**.
2. Click the project name or click .
3. In the upper right corner, click **Share** .
   The Share panel is displayed.



Copyright © 2022, 2026, Oracle and/or its affiliates.

4. Assign users and groups to the project permissions, and click **Share** .

> **Note**
>
> You must have already created users and groups to which you assigned those users before they are visible for selection in this dialog. See Manage Access and Assign Roles in *Provisioning and Administering Oracle Integration 3*.

Field	Description
Owners	Begin typing to search for users and groups to own the project. By default, the user who created the project is an owner. The user with the ServiceAdministrator service role is also automatically an owner. The owner can perform all available actions, including associating users and groups with each of the project permissions.
Can edit	Begin typing to search for users and groups to which to assign the editor permission. If you select a user or group that includes users that do not have the ServiceDeveloper role, those users are unable to edit the project.
Can view	Begin typing to search for users and groups to which to assign the viewer permission.
Can monitor	Begin typing to search for users and groups to which to assign the monitor permission.

Best Practices

- Provide project access only to users that need it.
- Give users the minimal permissions necessary to perform their responsibilities. For example, if a user only needs to monitor integrations, don't give them the permission to edit integrations.
- Give projects nonsensitive names. For example, don't specify names such as employeeSalaries, employeeRatings, and so on. Even if users don't have access to projects, they can still see the project names listed on the Projects page.

RBAC-Enabled Project Use Case

This use cases provides a high level example of creating an RBAC-enabled project. It demonstrates what users can and cannot do based on their assigned project permissions.

1. Neeharika, the user with the ServiceAdministrator role, performs the following tasks:
   a. Creates the following user and service role assignments in the Oracle Cloud Console:
      - Vijaya: ServiceDeveloper
      - Bipin: ServiceDeveloper
      - Sumit: ServiceDeveloper
   b. Goes to the Projects page, and clicks **Add**.

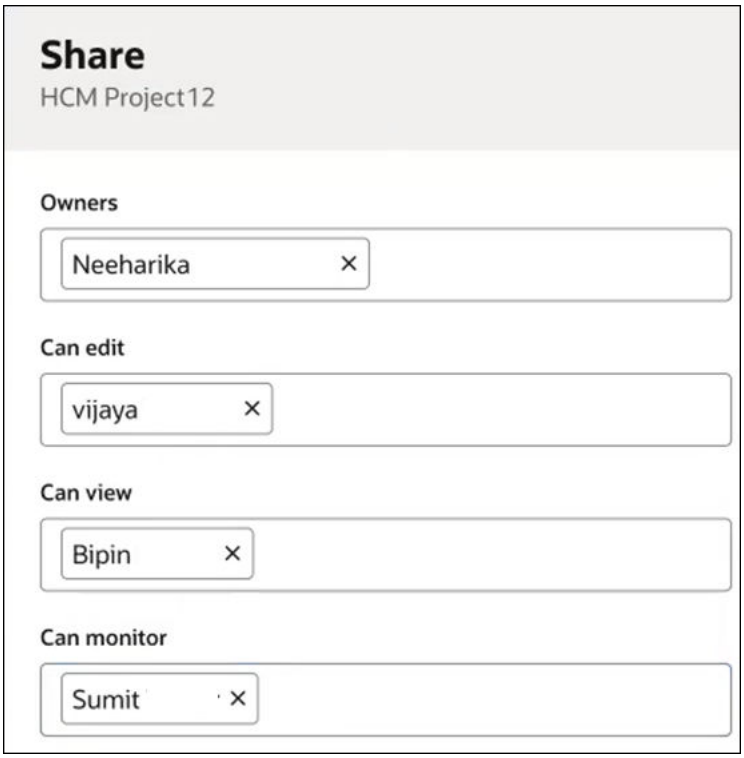
Copyright © 2022, 2026, Oracle and/or its affiliates.

c. Creates two new projects:
   - **HCM Project12**
   - **FinancialServiceLocalInvoke**

d. Opens **HCM Project12**.

e. In the upper right corner, clicks **Share**.

f. Assigns the following users to each project permission. By default, Neeharika is the owner because she created the **HCM Project12** project.

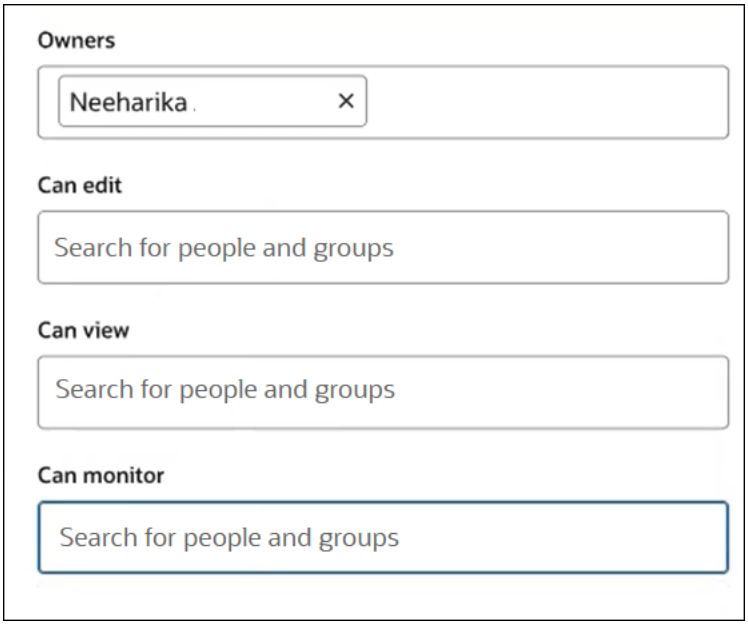


This configuration creates the following similarities between the service roles and the project permissions:

User	Service Role	Project Permission
Vijaya	ServiceDeveloper	Editor
Bipin	ServiceDeveloper	Viewer
Sumit	ServiceDeveloper	Monitor

g. Does not make any project permission updates under **Share** of the **FinancialServiceLocalInvoke** project.
Because Neeharika created this project, she is listed as the owner. The **Can edit** (for the editor permission), **Can view** (for the viewer permission), and **Can monitor** (for the monitor permission) fields remain empty.

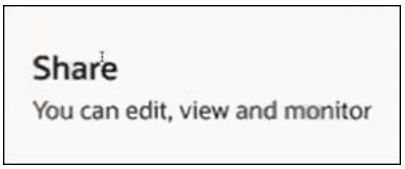
Copyright © 2022, 2026, Oracle and/or its affiliates.



With project permission assignments complete, Vijaya (editor permission), Bipin (viewer permission), and Sumit (monitor permission) log in and see their capabilities and restrictions.

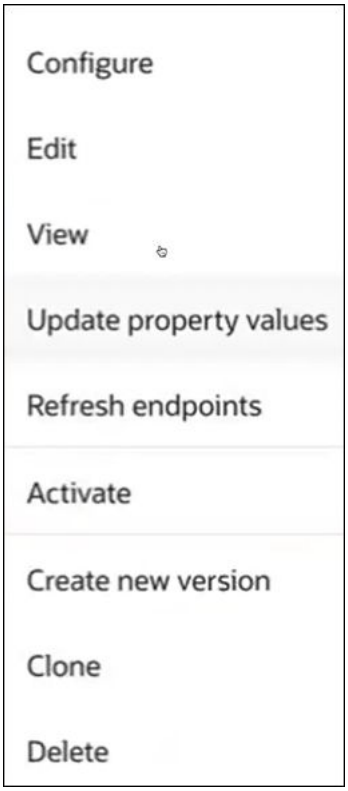
2. What the Vijaya (ServiceDeveloper with the editor project permission) can do:

   a. Opens the **HCM Project12** project and goes to **Share**. Because Vijaya has editor permissions, she can edit, view, and monitor resources in the project.



   b. Clicks **Actions** in the **Integrations** section and notes that all edit actions are listed and can be performed.

Copyright © 2022, 2026, Oracle and/or its affiliates.



c. Moves around the project and successfully performs assorted editor tasks without any restrictions, such as the following:
   - Adds new project keywords in the **Details** section.
   - Imports a schedule integration and updates the schedule.
   - Deletes a library.

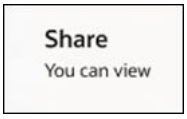
d. Exits the **HCM Project12** project.

e. Attempts to open the **FinancialServiceLocalInvoke** project. This is the project described in Step h to which no project permissions were assigned.
Because Vijaya is not assigned any permissions on this project, the following error is displayed:

```
User Vijaya does not have sufficient privilege to perform this action.
```

3. What Bipin (ServiceDeveloper with the viewer project permission) can do:

   a. Opens the **HCM Project12** project and, clicks **Share** . Because Bipin has viewer permissions, he can only view resources in the project.



   b. Clicks **Actions** ... in the **Integrations** section and notes that no edit actions are listed; only the **View** and **Schedule** (for viewing the schedule) actions can be performed.

Copyright © 2022, 2026, Oracle and/or its affiliates.



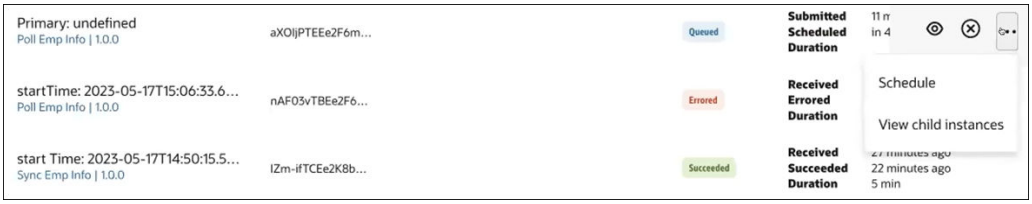
c. Clicks the **Deploy** tab.

d. Clicks **Actions** ⋮ and notes that no edit actions are listed; only the **View** action can be performed.



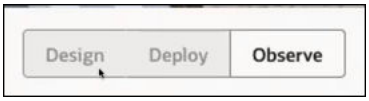
e. Clicks the **Observe** tab.

f. Clicks **Actions** ⋮ and notes that no edit actions such as abort or retry are listed; only the **Schedule** (to view a schedule) and **View child instances** actions can be performed.



4. What Sumit (ServiceDeveloper with the monitor project permission) can do:

a. Opens the **HCM Project12** project. Because Sumit has monitor permissions in the project, only the **Observe** tab is accessible. The **Design** and **Deploy** tabs are both disabled.

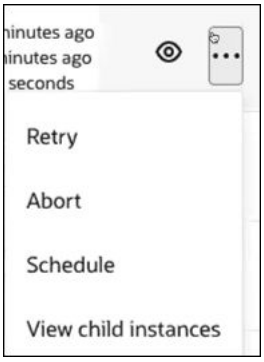


b. Clicks **Observe**.

c. Clicks **View details** to view the activity stream.

d. Click **Actions** ⋮ and notes that **Retry**, **Abort**, **Schedule**, and **View child instances** can all be performed.

Copyright © 2022, 2026, Oracle and/or its affiliates.



e. Clicks the **Future runs** and **Audit** tabs to view future scheduled runs and audit details, respectively.



f. In the navigation pane, clicks **Observability**, then **Dashboards** for a global view of aggregated data for all integrations in the instance (whether or not they are in a project). All integrations are visible to Sumit regardless of his project permissions.

g. In the navigation pane, clicks the **Integrations** tab to view all integration instances *except* for instances in projects in which Sumit does not have project permissions. For example, if Sumit clicks **Filter**, selects a project on which he does not have any permissions, and clicks **Apply**, the following error is displayed:

```
User Sumit does not have sufficient privilege to perform this action.
```

h. Clicks **Actions** ... to view the instance statistics for an integration instance that is part of the **HCM Project12** project.



i. Selects the **Edit schedule** option, but receives the following error:

```
User Sumit does not have sufficient privilege to perform this action.
```

The restriction occurs because Sumit's monitor permissions do not support this task.

j. In the navigation pane, clicks the **Instances** tab to view all tracking instances *except* for instances in projects in which Sumit does not have project permissions.

k. Performs monitoring tasks successfully:

- Views the activity stream.

Copyright © 2022, 2026, Oracle and/or its affiliates.

- Aborts and retries instances.

l. In the navigation pane, clicks the **Errors** tab to view all instances in error *except* for instances in projects in which Sumit does not have project permissions.

m. Performs monitoring tasks successfully:

- Views the activity stream.
- Aborts and retries instances.

Create or Import a Project

You can create a project in which to design, manage, and monitor integrations. You can also import user-developed, accelerator, or recipe projects into an instance from your local host or from a GitHub repository.

- Create or Import a Project
- Understand an Imported Oracle Accelerator Project
- Add a Newer (Extended) Version of an Accelerator Project
- Understand an Imported Recipe Project

See Learn About Recipes and Accelerators in *Using Recipes and Accelerators in Oracle Integration 3*.

Create or Import a Project

1. In the navigation pane, click **Projects**.
2. Click **Add**.
3. Select an option:
   - Create a new project.
   - Import a project from your local host.
   - Import a project from your GitHub repository.
4. To create a new project, perform the following steps.
   a. Click **Create**.
   b. On the Start a new project page, you can specify the project detail (of the project you want to create) using natural language, and create a project by selecting an accelerator or recipe that matches your natural language text. Or, if you want to create your own project and build exactly what you need, perform the following steps.
      i. In the **New project** option, click **Choose**.
      ii. Enter the following information.

Field	Description
Name	Provide a meaningful name so that others can understand the project. You can include English alphabetic characters, numbers, underscores, and dashes in the identifier. **Note**: You cannot create a project name containing `-BA-` or `_BA_`. These conventions are reserved for accelerator projects.

Copyright © 2022, 2026, Oracle and/or its affiliates.

Field	Description
Identifier	Accept the default identifier value. The identifier is the same as the project name you provided, but in upper case.
Keywords	Enter keywords (tags) to identify the project. You can search for projects on the Projects page using keywords. Keywords are useful for filtering and discovery.
Description	Provide a meaningful description so that others can understand the project.
Anyone can edit, view, and monitor	• Select this option to allow anyone to edit, view, and monitor resources in the project. You can always restrict access at a later time. • Don't select this option if you want to restrict access to editing, viewing, and monitoring resources in this project. After project creation, you can select the specific groups and users to which to provide access. See Control Who Can Edit, View, and Monitor in a Project.

iii. Click **Create**.
Your new project page is displayed. You can now add integrations, connections, lookups, and JavaScript libraries to the project.

5. To import a project (`.CAR` file) from your local host, perform the following steps. This enables you to import a project with all assets and metadata from one system to another. You can import user-developed, accelerator, or recipe projects.
Ensure that you understand connection behavior when importing a project. See Export a Project.

   a. Click **Import**.

   b. Click the field to browse for a project or drag and drop a project into the field.

   c. With the **Anyone can edit, view, and monitor** check box, select the level of access you want to provide to other users:

      - Select this option to allow anyone to edit, view, and monitor resources in the project.
        You can always restrict access at a later time.
      - Don't select this option if you want to restrict access to editing, viewing, and monitoring resources in this project. After import, you can select the specific groups and users to which to provide access. See Control Who Can Edit, View, and Monitor in a Project.

   d. Click **Import**.
   The project connections contain information that is specific to the processing environment. Connections in the imported project omit this potentially sensitive data. You must reconfigure connections before they can operate properly.

6. To import a project or a project deployment from your GitHub repository, perform the following steps. To use this option, you must first configure access to your GitHub repository. See Integrate Projects and Project Deployments with a GitHub Repository.

   a. Click **Download**.

   b. Click the tab for the type of resource to import:

Copyright © 2022, 2026, Oracle and/or its affiliates.

- **Projects**
- **Deployments**

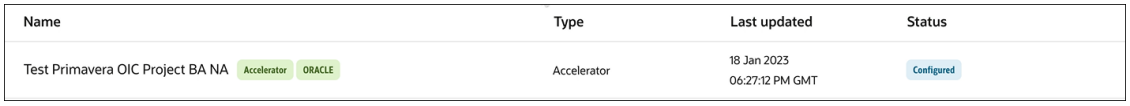
c. Select the file, then click **Import**.

Watch a video to learn more:

Video

Understand an Imported Oracle Accelerator Project

If you import an Oracle accelerator project into your project, the project includes the labels **Accelerator** and **Oracle** on the Projects page. The **Type** column also includes the **Accelerator** label.



Note the following accelerator project restrictions:

- You cannot edit the details that were specified when the project was created, such as name, project identifier, description, and keyword selections. These details are in read-only mode.
- The **Edit**, **Clone**, and **Delete** tasks are not visible under the **Actions** ⋯ menu on the **Deploy** tab. The only task visible under this menu is **View**.
- You can extend (customize) an integration in the accelerator project by selecting **Actions** ⋯, then **Extend**. See Extend an Integration in an Accelerator Project.

Note the following restrictions for individual assets in accelerator projects

Copyright © 2022, 2026, Oracle and/or its affiliates.

Integrations	Connections	Lookups	JavaScript Libraries	Deployments	Certificates
Allowed: • View • Activate • Deactivate • Override default property values Restricted: • Add • Delete • Clone • Create new version Allowed in extended accelerator: • Add a scope or stitch action before an invoke • Merge a scope or stitch action from the next lower version of the integration Restricted in extended accelerator: • Update primary information	Allowed: • View • Test • Refresh metadata • Edit and save • Save and reactivate Restricted: • Add connections • Delete connections • Update primary information •	Allowed: • View • Edit and save • Add or remove rows and update values in rows Restricted: • Add lookups • Delete lookups • Update domain name • Add or remove domain • Update primary information	Allowed: • View Restricted: • Add libraries • Delete libraries	Allowed: • Uninstall • View • Export ** • Activate • Deactivate Restricted: • Create, update, or delete the deployment	No restrictions.

** Export as an accelerator or extended accelerator only.

Add a Newer (Extended) Version of an Accelerator Project

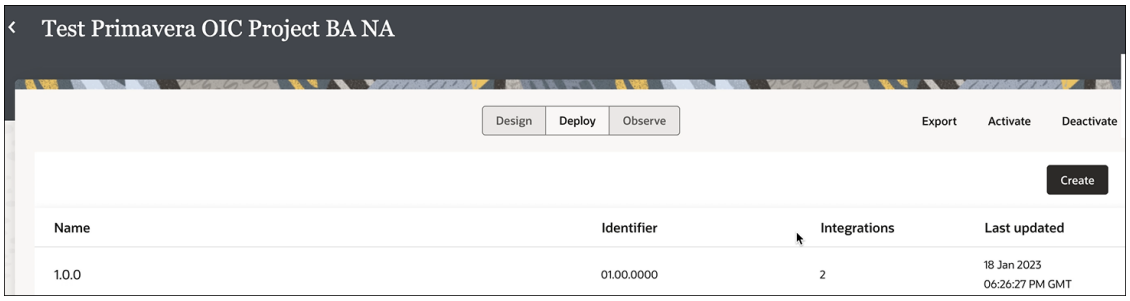
Assume a newer version of an accelerator project becomes available for you to import.

For this example, assume the first version of the accelerator project included integrations Oracle ERP 1 and Oracle ERP 2, each with versions of 1.0.0.

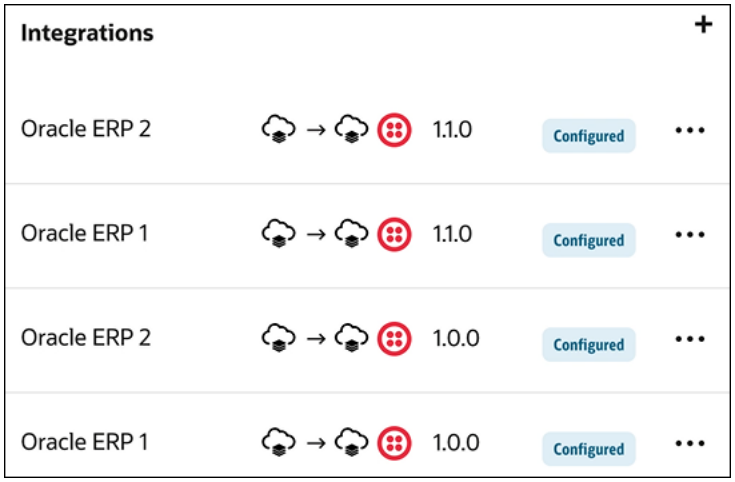
Copyright © 2022, 2026, Oracle and/or its affiliates.



The two integrations and their versions are included in a project deployment named 1.0.0 that you previously imported into your instance.

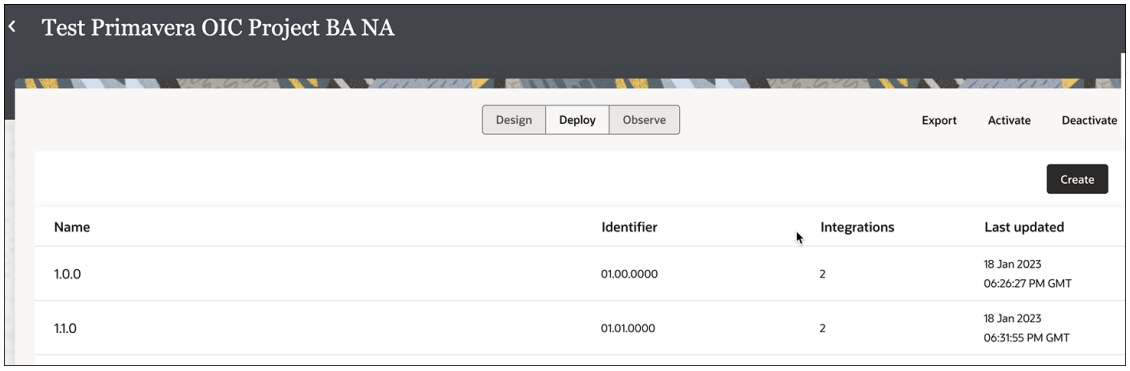


A newer version of the accelerator project then becomes available. This version now includes two versions of the two integrations: Oracle ERP 1 versions 1.1.0 and 1.0.0 and Oracle ERP 2 versions 1.1.0 and 1.0.0.



All four integrations and versions are included in a project deployment named 1.1.0.

Copyright © 2022, 2026, Oracle and/or its affiliates.

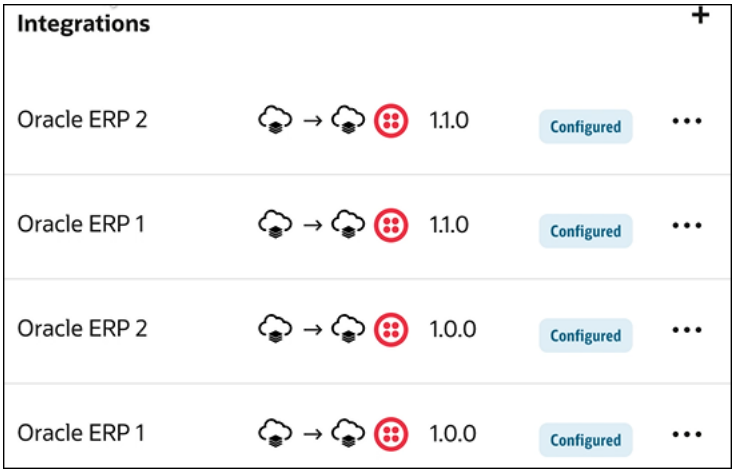


1. In the navigation pane, click **Projects**.
2. Click **Add**.
3. Click **Import** and upload the latest accelerator project.
   The following message is displayed:

   ```
   Project has been successfully imported.
   ```



4. Click the project name or click  .
   Note that the latest accelerator project includes Oracle ERP 1 versions 1.1.0 and 1.0.0 and Oracle ERP 2 versions 1.1.0 and 1.0.0.



Understand an Imported Recipe Project

When you import a recipe project, it is displayed as **Developed** in the **Type** column on the Projects page.

Copyright © 2022, 2026, Oracle and/or its affiliates.



Name	Type	Last updated	Status
Oracle R Primavera Sync	Developed	18 Jan 2023 06:33:37 PM GMT	Configured

Create a Project By Selecting an Accelerator or Recipe that the AI Suggests

You can create a project using natural language by selecting an accelerator or recipe from the accelerators or recipes that the AI suggests. The AI provides these suggestions based on the natural language text that you specify.

> **Note**
>
> AI features are available in select regions. See AI Feature Availability in *Provisioning and Administering Oracle Integration 3*.

1. Decide where to start:
   - Home page:
     a. On the Oracle Integration Home page, click **Start a new project** in the upper right corner.
     The Start a new project panel appears.
   - Projects page:
     a. In the navigation pane, click **Projects**.
     b. Click **Add**.
     The Start a new project panel appears.
2. In the Start a new project panel, click **Create**.
3. On the Start a new project page, enter the project detail using natural language and press **Enter**. For example: `Create sales orders in Oracle ERP Cloud`. You can also enter keywords. For example: `salesforce`.
   Based on the project detail or keyword that you entered, the AI shows accelerators and recipes that match your project detail/keyword.
4. Click **Choose** to select the accelerator or recipe you want to install.
   The accelerator or recipe you selected starts installing. Once the installation is complete, the Configuration Editor page appears.
5. On the Configuration Editor page, perform the following steps.
   a. On the Connections subpage, you can edit or replace connections to configure them. Hover over the connection you want to edit or replace and perform one of the following.
      - Click **Edit** to configure the connection as needed.
      - Click **Actions** ⋯, then select **Replace** to select a configuration from the list of shared connections across your projects. After selecting a shared connection, click **Save**. If you don't want to select a connection from the list of shared connections, click **Cancel**.
   b. Click the **Lookups** tab in the upper right corner to configure lookups (if available) on the Lookups subpage. If you don't have any lookups, you can proceed to the next step.

Copyright © 2022, 2026, Oracle and/or its affiliates.

c. Click the **Deployments** tab in the upper right corner to deploy the integration on the Deployments subpage:

   i. Hover over the row you want to activate, then click **Activate**.
   The Activate integration panel opens.

   ii. Select the level of tracing as needed, and click **Activate**.

   You can also deploy the integration using the following steps instead:

   i. Click **Back** to go to the **Design** tab.

   ii. In the **Integrations** section, hover over the specific integration to activate.

   iii. Click **Actions**, then select **Activate**.
   The Activate integration panel opens.

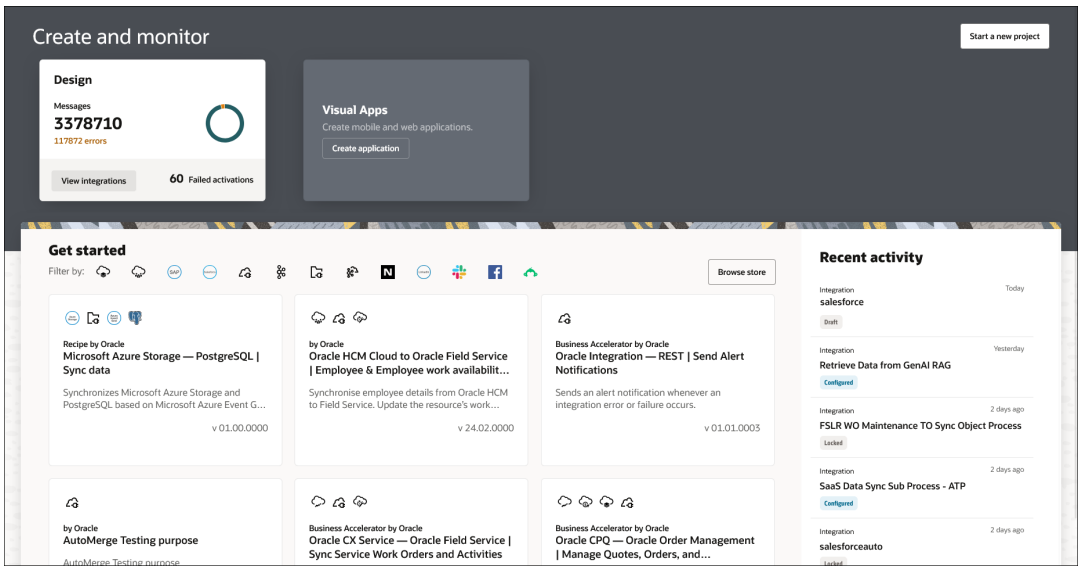
   iv. Select the level of tracing as needed, and click **Activate**.

> **Note**
>
> If you have multiple integrations, then you can deploy them from the **Deploy** tab. See Create a Deployment.

Example

Let's create a project to create sales orders in Oracle ERP Cloud when opportunities are won in Salesforce. To create this project, specify the project detail or purpose using natural language on the Start a new project page and select an accelerator or recipe to install from the list of accelerators or recipes that the AI recommends.

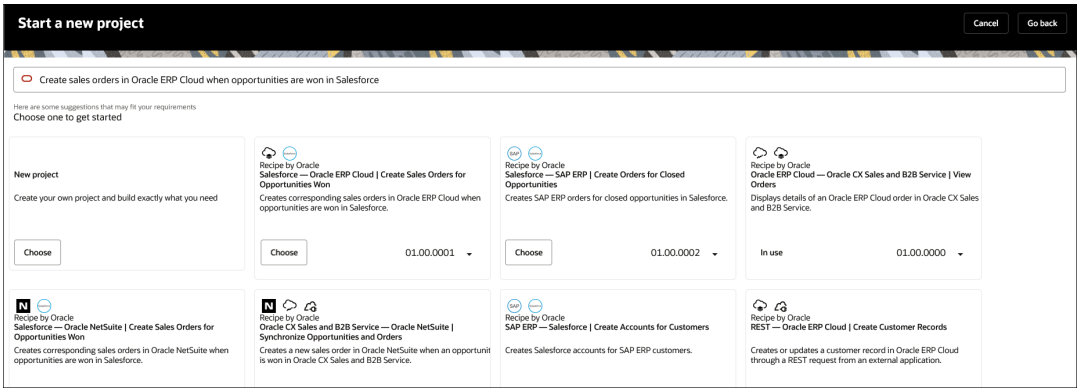
1. On the Oracle Integration Home page, click **Start a new project** in the upper right corner.



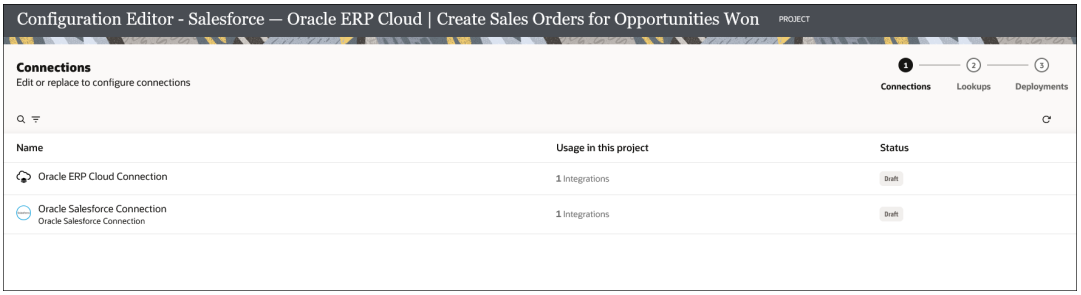
2. In the Start a new project panel, click **Create**.

3. On the Start a new project page, enter `Create sales orders in Oracle ERP Cloud when opportunities are won in Salesforce` and press **Enter**.
The AI shows accelerators and recipes that match the above project detail.

Copyright © 2022, 2026, Oracle and/or its affiliates.



4. Click **Choose** to select the accelerator or recipe that closes or exactly matches the project you want to create. In this example, select **Create Sales Order for Opportunities Won** recipe.
The recipe starts installing. Once the installation is complete, the Configuration Editor page appears.

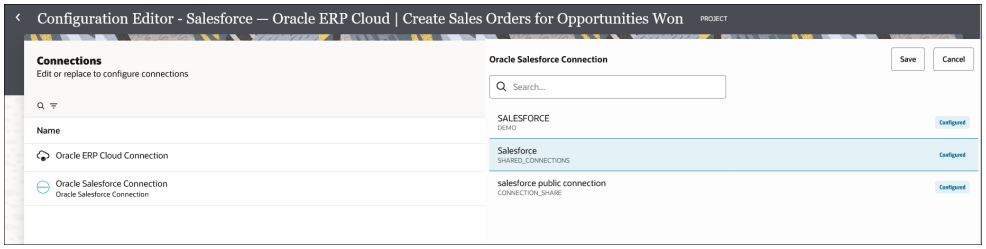


5. On the Configuration Editor page, perform the following steps.
   a. On the Connections subpage, the following connections are shown:
      - **Oracle ERP Cloud Connection**
      - **Oracle Salesforce Connection**

      Configure each of the connections by performing the following steps.
      i. Hover over the connection you want to configure, click **Actions** ⋮, and then select **Replace** to select a configuration from the list of shared connections across your projects.
      ii. After selecting a shared connection, click **Save**.

> **Note**
>
> If you don't want to configure the connections using shared connections, click **Edit** ✎ to configure the connections as needed.

Copyright © 2022, 2026, Oracle and/or its affiliates.



b. As this project does not have any lookup data, you don't need to configure any lookups on the Lookups subpage. You can proceed to the next step.

c. Click the **Deployments** tab in the upper right corner to deploy the integration on the Deployments subpage:

   i. Hover over the row to activate, then click **Activate** .
   The Activate integration panel opens.

   ii. Select the level of tracing as needed, and click **Activate**.

Copy an Integration into a Project

You can copy integrations into a project. The integration to copy can be either in a different project (as long as it is marked as globally available) or outside of any project.

1. In the navigation pane, click **Projects**.
2. Click the project name or click .
3. In the **Integrations** section, click **Add** if no integrations currently exist or **+** if integrations already exist.

   The Add integration panel opens.
4. Copy an integration into a project.

   You can copy an integration into a project. The integration to copy can be available in either of the following locations:

   - In another project.

     a. Ensure that the **Search in projects** check box is selected. This is the default selection.

     b. Search for or select the project that includes the integration to copy. Besides searching the listed projects, all available projects in Oracle Integration are searched on the basis of full name, partial name, keywords, and description to fetch the data from the server side.

     c. Search for or select the integration to copy in that project. Only the listed integrations are searched. That is, the integrations exposed outside of the selected project are listed upon selection of the project. (Data is only fetched from the client side.)
     Even though the integration to copy is in a project, it must be publicly available. That is, you selected the **Available to other projects** check box during integration

Copyright © 2022, 2026, Oracle and/or its affiliates.

creation or editing under **Primary Info**  in the integration canvas. Otherwise, you receive the following error:

```
Either there is no integration available or visible outside the
project.
```

Integrations in projects in which the **Available to other projects** check box is *not* selected are not available for selection.

d. Click **Copy**.

- Globally available (that is, outside any projects). Besides searching for listed integrations, all globally available integrations in Oracle Integration are searched on the basis of full name, partial name, keywords, and description.

  a. Deselect the **Search in projects** check box.

  b. Search for or select the integration to copy.

  c. Click **Copy**.

Use AI to Create an Integration

You can use AI to create an integration (within a project) based on natural language text. Using natural language, you can specify the purpose of the integration.

On the integration canvas, specify what the integration needs to do, and click **Generate the integration skeleton based on the user's goal** . See Best Practices.

For example, you can specify to create an integration that receives customer information from Salesforce and updates NetSuite with that customer information if the customer is a VIP customer, sends a notification to the customer after updating NetSuite, and logs a message if the customer is not a VIP customer.

> **Note**
>
> - Creation of an integration using AI is only supported within a project.
> - AI features are available in select regions. See AI Feature Availability in *Provisioning and Administering Oracle Integration 3*.

Here's what happens after you click **Generate the integration skeleton based on the user's goal** .

- The AI determines the nodes to be created for the request specified by the user and provides the list of nodes in a sequential order along with the type of adapter to be used for the trigger/invoke.
- The connection needed for the integration is automatically selected from the available connections within the project and a trigger/invoke is created. If there is no connection available within the project, the AI will try to create a draft connection and then create a trigger/invoke using it.
- The nodes are created too.

Copyright © 2022, 2026, Oracle and/or its affiliates.

- If multiple connections are available within the project, then the AI tries to identify the appropriate connection and then creates the trigger/invoke.
- Make sure you have preconfigured the necessary connections correctly and specified their usage clearly in the natural language text you provided. The AI might automatically configure the trigger/invoke by selecting the operation and business object needed for the trigger/invoke based on the natural language text about what the trigger/invoke needs to do and the configured connections specified for use. The automatic configuration of trigger/invoke by AI is designed to deliver the best possible results. It is attempted automatically, however successful configuration cannot be guaranteed in every case.

> **Note**
>
> The automatic configuration of trigger/invoke is supported only for the following adapters and under these conditions:
>
> – REST Adapter: Supported only when configured with the Swagger/OpenAPI catalog.
>
> – Oracle ERP Cloud Adapter, Oracle CX Sales and B2B Service Adapter, and Oracle HCM Cloud Adapter: Supported only for CRUD operations, generic/non-custom business objects, and generic/non-custom business events.
>
> – Oracle NetSuite Adapter: Only invoke is supported, and only for basic/search operations and single generic/non-custom business objects.
>
> – Salesforce Adapter:
>
> * Trigger is supported only on the available platform events for the configured salesforce application.
>
> * Invoke is supported only for create, update, or delete operations using generic/non-custom business objects.
>
> – SOAP Adapter: Supported only for WSDL URL or file upload use cases.
>
> – Oracle Autonomous Transaction Processing Adapter: Only invoke is supported, and only for operations on tables (for connections without an agent) and stored procedures.

The AI creates a skeleton integration.

In the **Create/updated integration flow** option, click **Accept AI generated actions** to accept the integration flow created by AI or click **Remove all AI generated actions** to remove the integration flow created by AI.

If you clicked **Remove all AI generated actions**, the integration flow created by AI is removed. You can modify your request or specify a new request indicating what the integration needs to do.

If you clicked **Accept AI generated actions**, the integration flow created by AI is retained and displayed on the canvas. You can modify this skeleton integration to configure the integration as needed. If needed, you can edit the connections and the triggers/invokes. You can activate the integration after you've configured it.

To learn how you can edit an integration using AI, see Use AI to Edit an Integration.

Copyright © 2022, 2026, Oracle and/or its affiliates.

Best Practices

- Specify the exact system to be called (for example: Salesforce, NetSuite, or others) so that the trigger/invoke is created with the same adapter.
- In order to create the trigger/invoke with correct connections, provide the connection name (for example: `Call service A using connection {connection_name in system}`).
- To add a fault handler, specify conditional logic too (for example: `Notify the user if any failures occur during the process` or `Handle failures/errors during the process`).

> **Note**
>
> On the integration canvas, you can collapse the option used to specify natural language text or expand it, as needed.

Examples

- Example 1
- Example 2
- Example 3

Example 1

Let's use AI to create an integration flow that creates a sales order. You need to specify what the integration needs to do clearly.

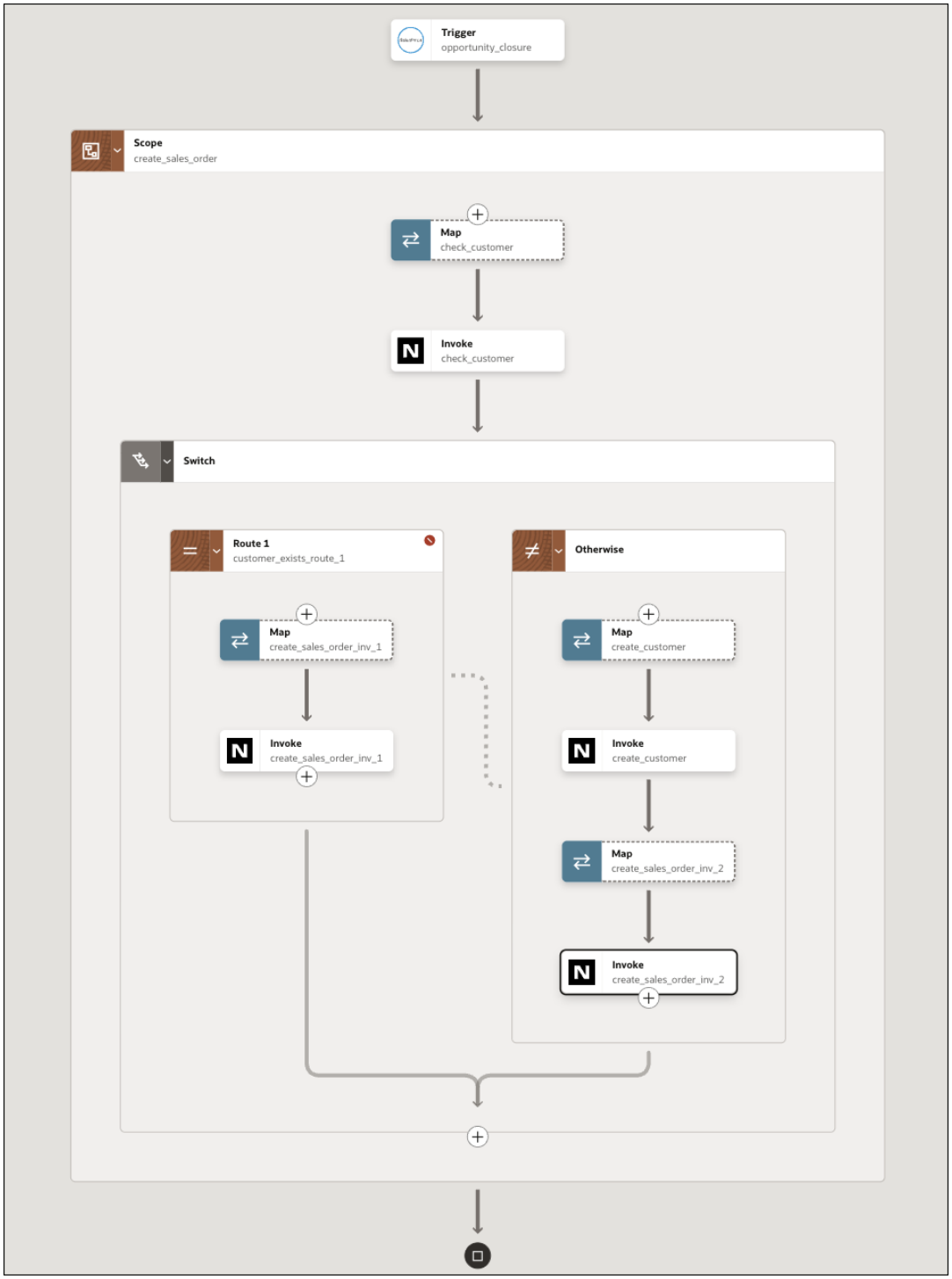
1. Create a Salesforce connection named **SalesforceConnection**.
2. Create a NetSuite connection named **NetSuiteConnection**.
3. Create an application integration.
4. On the integration canvas, enter the following natural language text:
   `Create a sales order in NetSuite using NetSuiteConnection on closure of an opportunity in Salesforce (that uses SalesforceConnection). Before creating the sales order, create a customer in NetSuite using NetSuiteConnection if the customer does not exist. Notify the user about errors if there are any failures.`
5. Click **Generate the integration skeleton based on the user's goal** .
   The AI automatically configures the trigger/invoke by selecting the operation and business object needed for the trigger/invoke based on the natural language text and the configured connections specified for use in the natural language text, and creates an integration flow.
   - The trigger is created with the Salesforce connection named **SalesforceConnection** fully configured with:
     - Action Group: `Streaming API`
     - Streaming Event Type: `Subscribe to platform events published on Salesforce.com`
     - Event: `Opportunity_Closure_Event`
     - Custom HTTP Request Header: `api_key`

Copyright © 2022, 2026, Oracle and/or its affiliates.

- The invoke is created with the NetSuite connection named **NetSuiteConnection** fully configured with:
  - Processing Mode: `Sync`
  - Operation Type: `Basic`
  - Operation: `Add`
  - Business Object: `Customer`
- The invoke is created with the NetSuite connection named **NetSuiteConnection** fully configured with:
  - Processing Mode: `Sync`
  - Operation Type: `Basic`
  - Operation: `Add`
  - Business Object: `SalesOrder`
- The invoke is created with the NetSuite connection named **NetSuiteConnection** fully configured with:
  - Processing Mode: `Sync`
  - Operation Type: `Basic`
  - Operation: `Add`
  - Business Object: `SalesOrder`

6. In the **Create/updated integration flow** option, click **Accept AI generated actions**  to accept the integration flow created by AI.
   The integration canvas shows the integration flow that you accepted.

Copyright © 2022, 2026, Oracle and/or its affiliates.



7. Configure the actions as needed.
8. Activate the integration.

Example 2

Let's use AI to create an integration flow that performs operations such as create, fetch, update, upsert, and delete. You need to specify what the integration needs to do clearly.

1. Create a Salesforce connection named **SalesforceConnection**.

Copyright © 2022, 2026, Oracle and/or its affiliates.

2. Create a REST connection named **RestTrigger** with an appropriate OpenAPI URL.
3. Create an application integration.
4. On the integration canvas, enter the following natural language text:
   `Create an integration with a REST trigger using the RestTrigger connection, which receives an account ID, and creates, fetches, updates, upserts, and deletes the account from Salesforce using the same RestTrigger connection. Add a fault handler to log the payload for troubleshooting.`
5. Click **Generate the integration skeleton based on the user's goal**  .
   The AI automatically configures the trigger/invoke by selecting the operation and business object needed for the trigger/invoke based on the natural language text and the configured connections specified for use in the natural language text, and creates an integration flow.
   - The trigger is created with the **RestTrigger** connection fully configured with:
     - Resource URI: `/accounts`
     - Verb: `GET`
     - Query parameters: `AccountId`
     - Sample JSON response:

       ```
       {
         "BillingCountry" : "BillingCountry",
         "BillingCity" : "BillingCity",
         "BillingStreet" : "BillingStreet",
         "ShippingPostalCode" : "ShippingPostalCode",
         "ShippingCountry" : "ShippingCountry",
         "BillingPostalCode" : "BillingPostalCode",
         "ShippingCity" : "ShippingCity",
         "ShippingState" : "ShippingState",
         "BillingState" : "BillingState",
         "AccountName" : "AccountName",
         "ShippingStreet" : "ShippingStreet"
       }
       ```

   - The invoke is created with the Salesforce connection named **SalesforceConnection** fully configured with:
     - Action Group: `Create, Update Or Delete Information`
     - Operation: `Create`
     - Business Object: `Account`
   - The invoke is created with the Salesforce connection named **SalesforceConnection** fully configured with:
     - Action Group: `Query Information`
     - Operation: `Retrieve`
     - Business Object: `Account`
   - The invoke is created with the Salesforce connection named **SalesforceConnection** fully configured with:
     - Action Group: `Create, Update Or Delete Information`
     - Operation: `Update`

Copyright © 2022, 2026, Oracle and/or its affiliates.

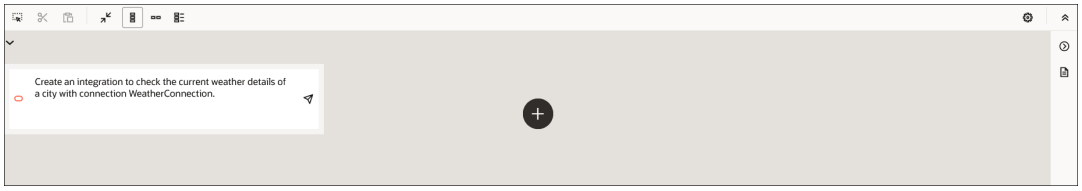
  - Business Object: `Account`
- The invoke is created with the Salesforce connection named **SalesforceConnection** fully configured with:
  - Action Group: `Create, Update Or Delete Information`
  - Operation: `Upsert`
  - Business Object: `Account`
- The invoke is created with the Salesforce connection named **SalesforceConnection** fully configured with:
  - Action Group: `Create, Update Or Delete Information`
  - Operation: `Delete`
  - Business Object: `Account`

6. In the **Create/updated integration flow** option, click **Accept AI generated actions** to accept the integration flow created by AI.
   The integration canvas shows the integration flow that you accepted.
7. Configure the actions as needed.
8. Activate the integration.

Example 3

Let's use AI to create an integration flow that checks the current weather details of a city.

1. Create a REST connection named **WeatherConnection** with an appropriate OpenAPI URL.
2. Create an application integration.
3. On the integration canvas, enter the following natural language text:
   `Create an integration to check the current weather details of a city with connection WeatherConnection.`



4. Click **Generate the integration skeleton based on the user's goal** .
   The AI automatically configures the trigger/invoke by selecting the operation and business object needed for the trigger/invoke based on the natural language text and the configured connections specified for use in the natural language text, and creates an integration flow.
   - The trigger is created with the **WeatherConnection** REST connection fully configured with:
     - Resource URI: `/current`
     - Verb: `GET`
     - Query parameters: This includes parameters (such as `cityId`, `long`, and so on).

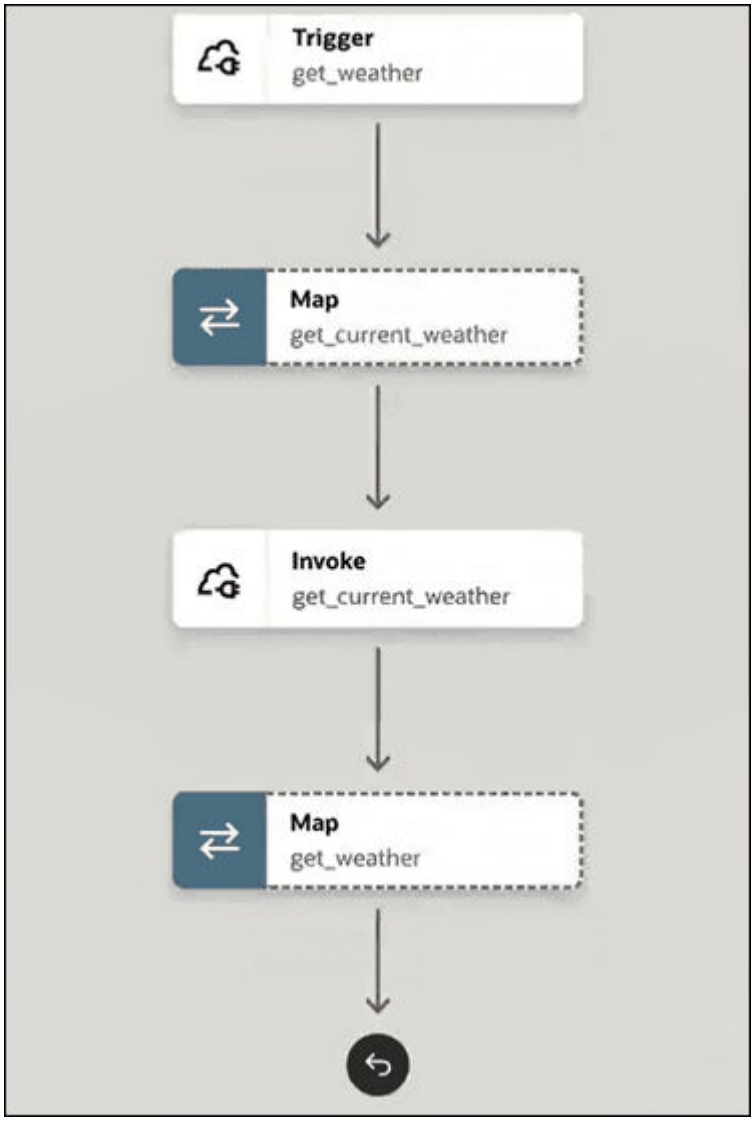
Copyright © 2022, 2026, Oracle and/or its affiliates.

- Example of a response body:

```
{
    "data": [
        {
            "location": {
                "city": "Raleigh",
                "state": "NC",
                "country": "US",
                "latitude": 38.0,
                "longitude": -78.25,
                "timezone": "America/New_York"
            },
            "time": "2017-03-15 T13:11:00",
            "station": "KU",
            "temperature": 13.9,
            "humidity": 30,
            "pressure": 1013.12,
            "wind": {
                "speed_mps": 5.85,
                "direction": 125,
            },
            "precipitation": 2.0,
            "cloud_coverage": 42,
            "visibilitykm": 10,
            "uv": 6.5,
            "airquality_index": 50
        }
    ],
    "count": 1
}
```

- The invoke is created with the **WeatherConnection** REST connection fully configured with:
  - Resource URI: `/current`
  - Verb: `GET`
  - Query parameters: This includes parameters (such as `cityId`, `long`, and so on).

Copyright © 2022, 2026, Oracle and/or its affiliates.



5. In the **Create/updated integration flow** option, click **Accept AI generated actions** to accept the integration flow created by AI.
   The integration canvas shows the integration flow that you accepted.
6. Configure the actions as needed.
7. Activate the integration.

Copyright © 2022, 2026, Oracle and/or its affiliates.

Use AI to Edit an Integration

You can use AI to edit an integration that was created manually or using AI. You can also incrementally edit the integration flow when creating it using AI. On the integration canvas, you can specify the purpose of the integration using natural language.

> **Note**
>
> - This feature is available only in projects.
> - AI features are available in select regions. See AI Feature Availability in *Provisioning and Administering Oracle Integration 3*.

1. Decide where to start:
   - Edit an existing integration:
     a. In the navigation pane, click **Projects**.
     b. Select the project name.
     c. Click **Integration**.
     d. In the **Integrations** section, hover over the integration to edit.
     e. Click **Actions**, then select **Edit**.
   - Perform incremental updates when creating a new integration:
     a. Create an integration. On the integration canvas, specify what the integration needs to do.
     b. Click **Generate the integration skeleton based on the user's goal**.
     c. The AI creates a skeleton integration.
2. To edit the integration, specify the updates using natural language clearly (with details such as node names, connection names, routes, and so on), and then click **Generate the integration skeleton based on the user's goal**.

Copyright © 2022, 2026, Oracle and/or its affiliates.

> **Note**
>
> **You should specify the node on which the edit operation needs to be performed clearly, otherwise the edit operation might not be performed correctly. You should also specify any conditions or routes clearly. Examples:**
>
> - After `node_name`, `edit_action_to_perform`.
>   **Example**: After receiving order from NetSuite, add a call to Salesforce using connection `connection_name` to get order line items. Iterate over the line items and update the line item in NetSuite.
> - Before `node_name`, `edit_action_to_perform`.
>   **Example**: Before receiving an order from NetSuite, add a call to check the existence of the customer in NetSuite using connection `connection_name`. If the customer does not exist, create the customer record using connection `connection_name`, otherwise log the customer details.
> - Inside `route_name` or `for_each_name` or `while_name`, `edit_action_to_perform`.
>   **Example:** Inside the success route (the route when customer exists), add a call to get orders of the customer, and log the order details.

Repeat this step if you need to make further updates to the integration flow.

Each time you update the integration using AI, review and save the integration.

3. In the **Create/updated integration flow** option, perform one of the following steps:

   - Click **Accept AI generated actions**  to accept the integration flow created by AI.
     The integration flow created by AI is retained and displayed on the canvas. You can modify this skeleton integration to configure the integration as needed. If needed, you can edit the connections and the triggers/invokes. You can activate the integration after you've configured it.
   - Click **Remove all AI generated actions**  to remove the integration flow created by AI.
     The integration flow created by AI is removed. You can modify your request or specify a new request indicating what the integration needs to do.

> **Note**
>
> The AI might automatically configure the trigger/invoke based on the natural language text you entered. The automatic configuration of trigger/invoke by AI is designed to deliver the best possible results. It is attempted automatically, however successful configuration cannot be guaranteed in every case. For more information about automatic configuration of trigger/invoke, see Use AI to Create an Integration.

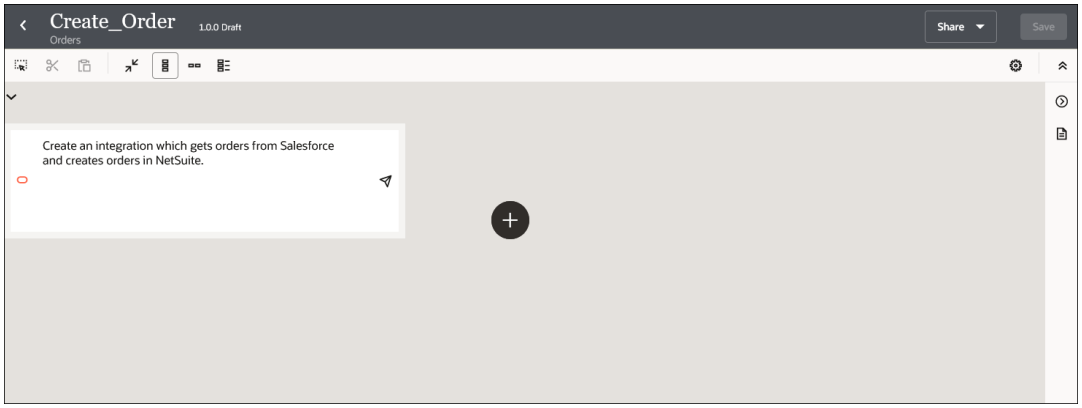
Example

Let's use AI to perform incremental updates when creating a new integration. Note that you can also edit an existing integration. In this particular example, we are creating an integration and performing edits.

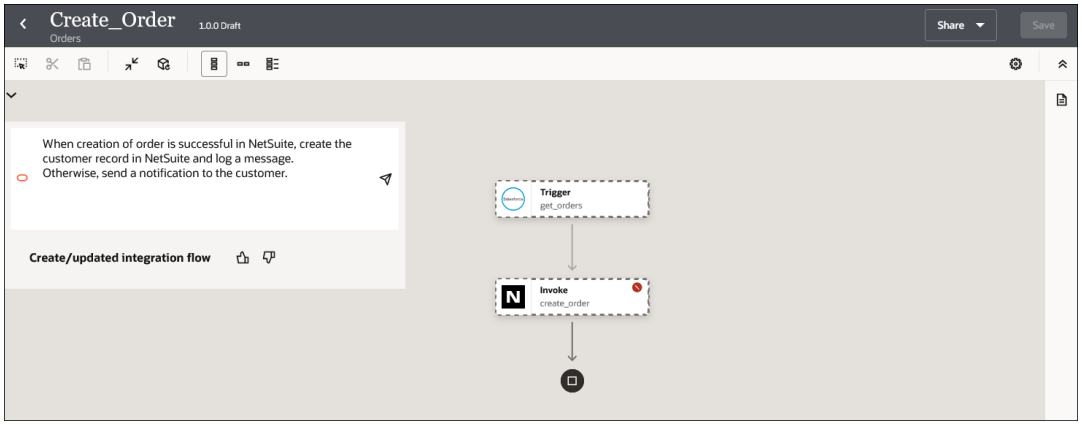
1. Create an application integration.

Copyright © 2022, 2026, Oracle and/or its affiliates.

2. On the integration canvas, enter the following natural language text, and click **Generate the integration skeleton based on the user's goal**.
```
Create an integration which gets orders from Salesforce and creates orders in NetSuite.
```

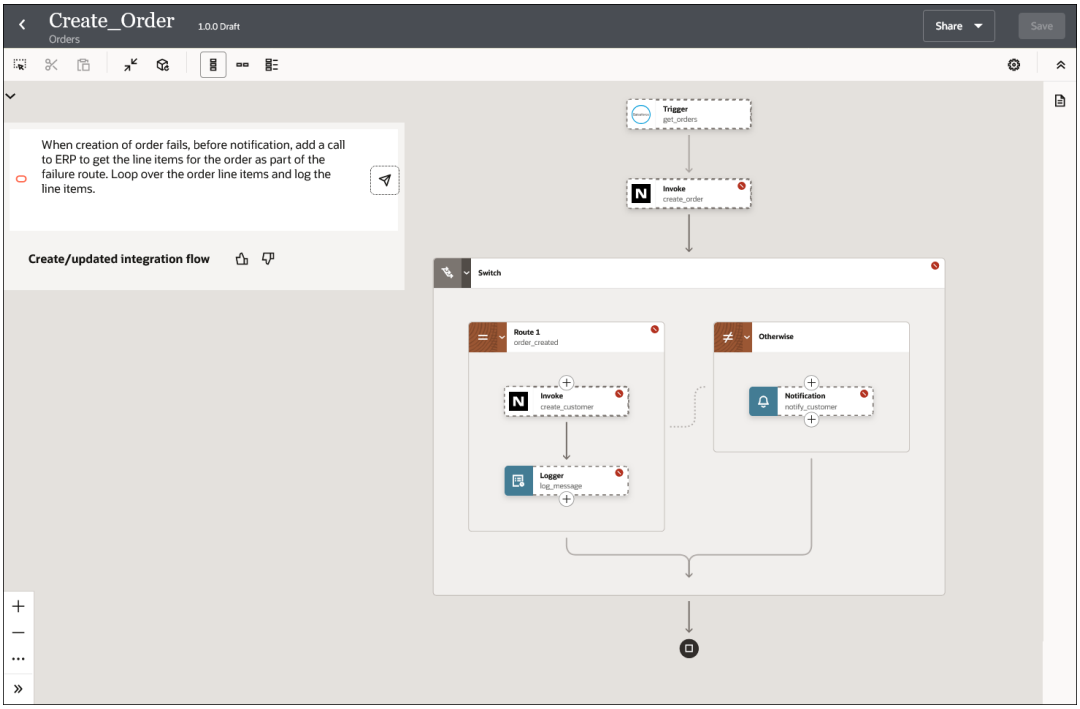


3. Review and save the integration before proceeding to modify the integration flow using AI.
4. Enter the following natural language text to edit the integration flow, and click **Generate the integration skeleton based on the user's goal**.
```
When creation of order is successful in NetSuite, create the customer record in NetSuite and log a message. Otherwise, send a notification to the customer.
```



5. Review and save the integration before proceeding to modify the integration flow using AI.
6. Enter the following natural language text to edit the integration flow, and click **Generate the integration skeleton based on the user's goal**.
```
When creation of order fails, before notification, add a call to ERP to get the line items for the order as part of the failure route. Loop over the order line items and log the line items.
```

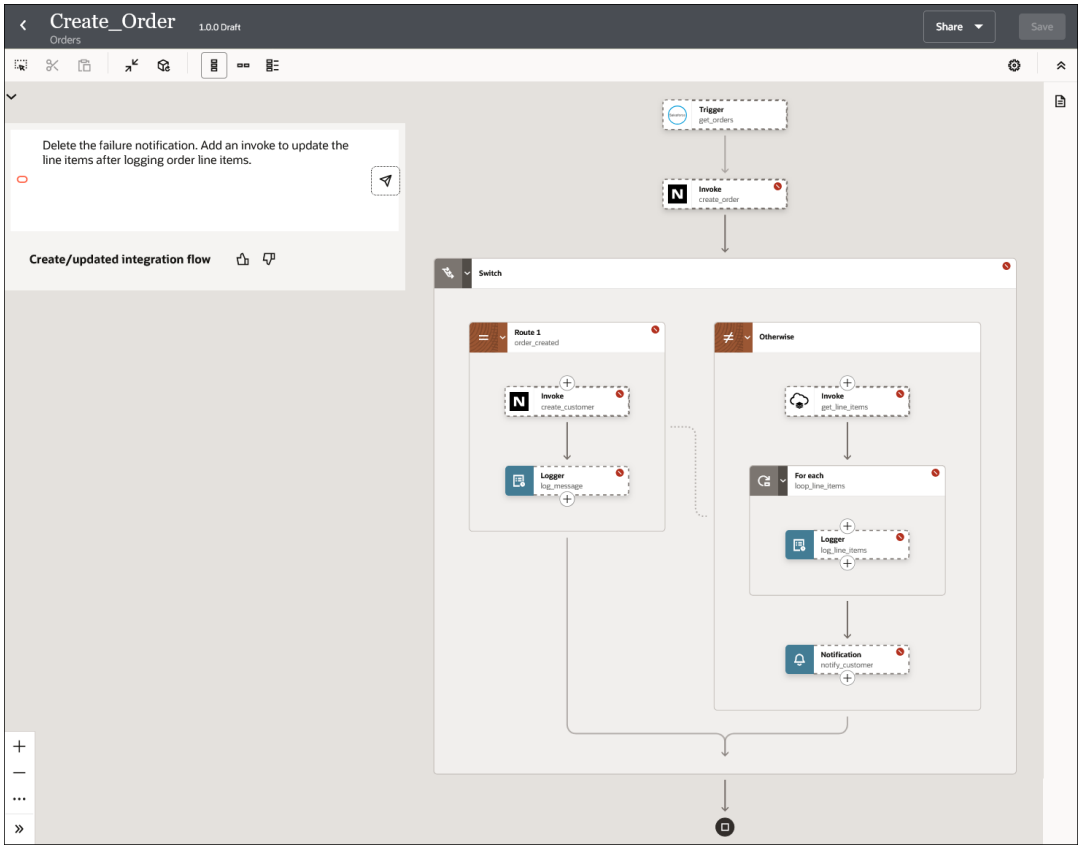

Copyright © 2022, 2026, Oracle and/or its affiliates.



7. Review and save the integration before proceeding to modify the integration flow using AI.
8. Enter the following natural language text to edit the integration flow, and click **Generate the integration skeleton based on the user's goal** .

```
Delete the failure notification. Add an invoke to update the line items after logging order line items.
```

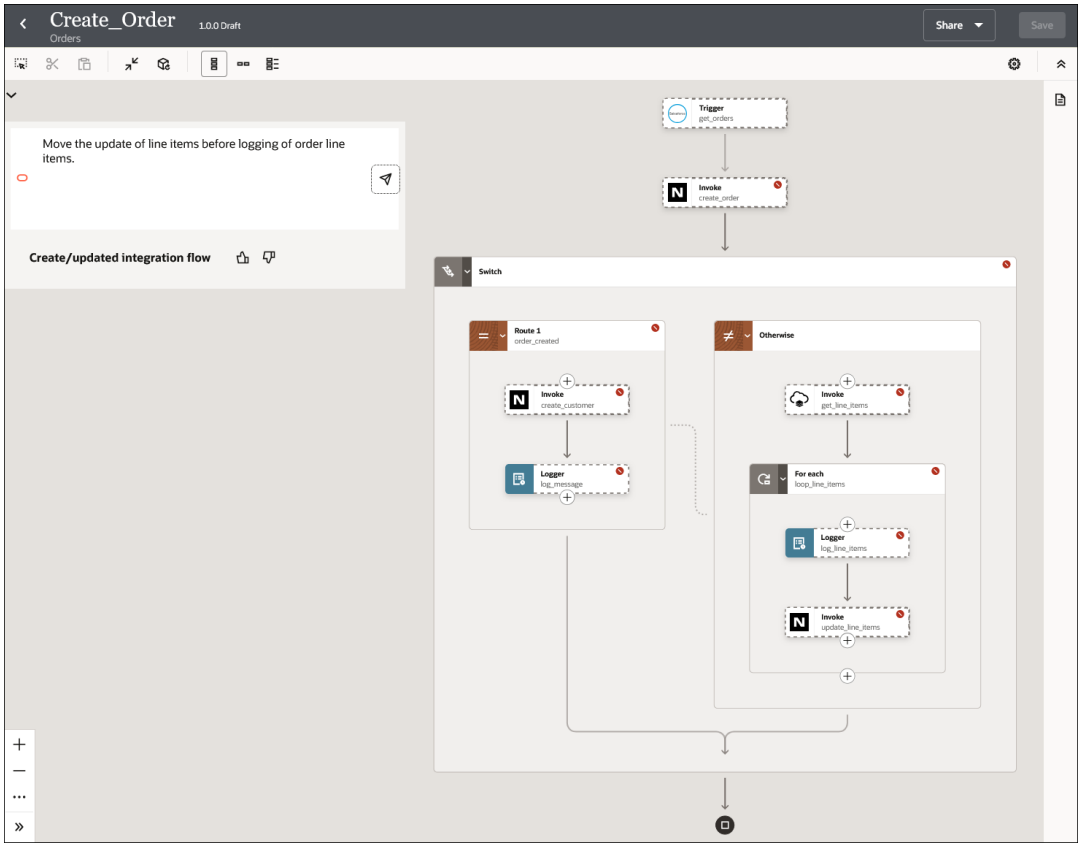

Copyright © 2022, 2026, Oracle and/or its affiliates.



9. Review and save the integration before proceeding to modify the integration flow using AI.
10. Enter the following natural language text to edit the integration flow, and click **Generate the integration skeleton based on the user's goal** .

```
Move the update of line items before logging of order line items.
```

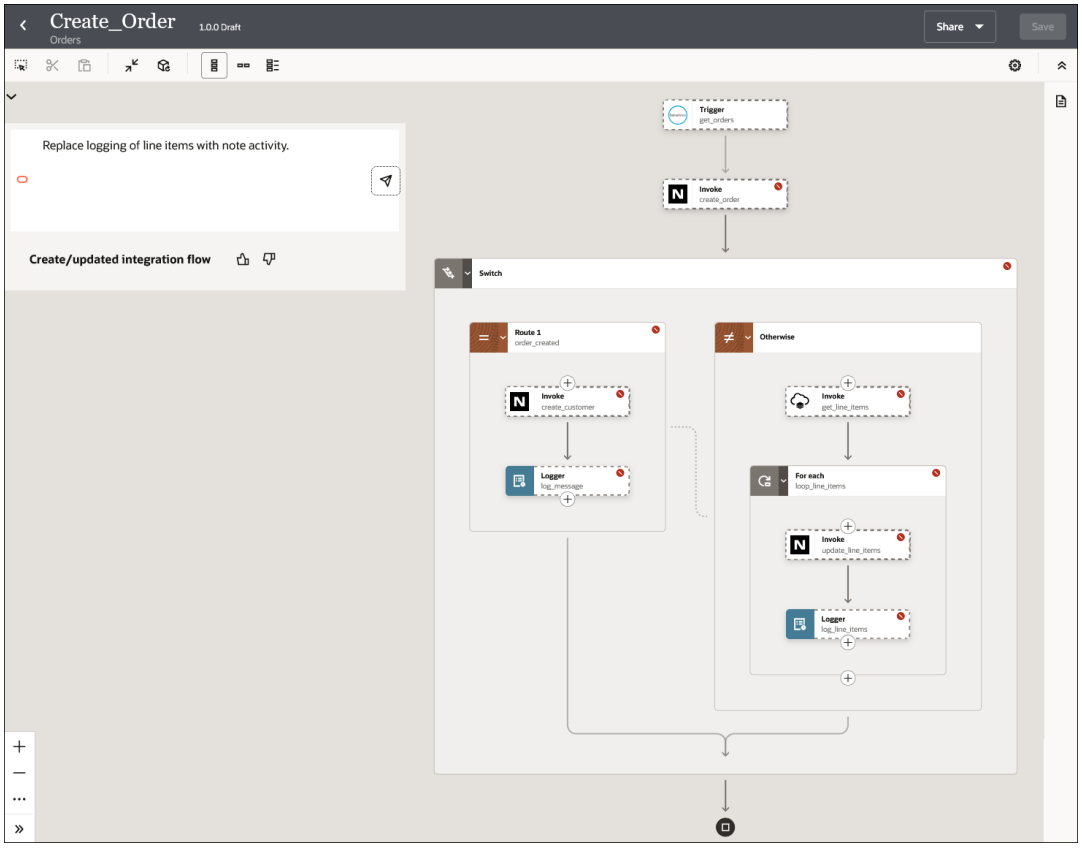

Copyright © 2022, 2026, Oracle and/or its affiliates.



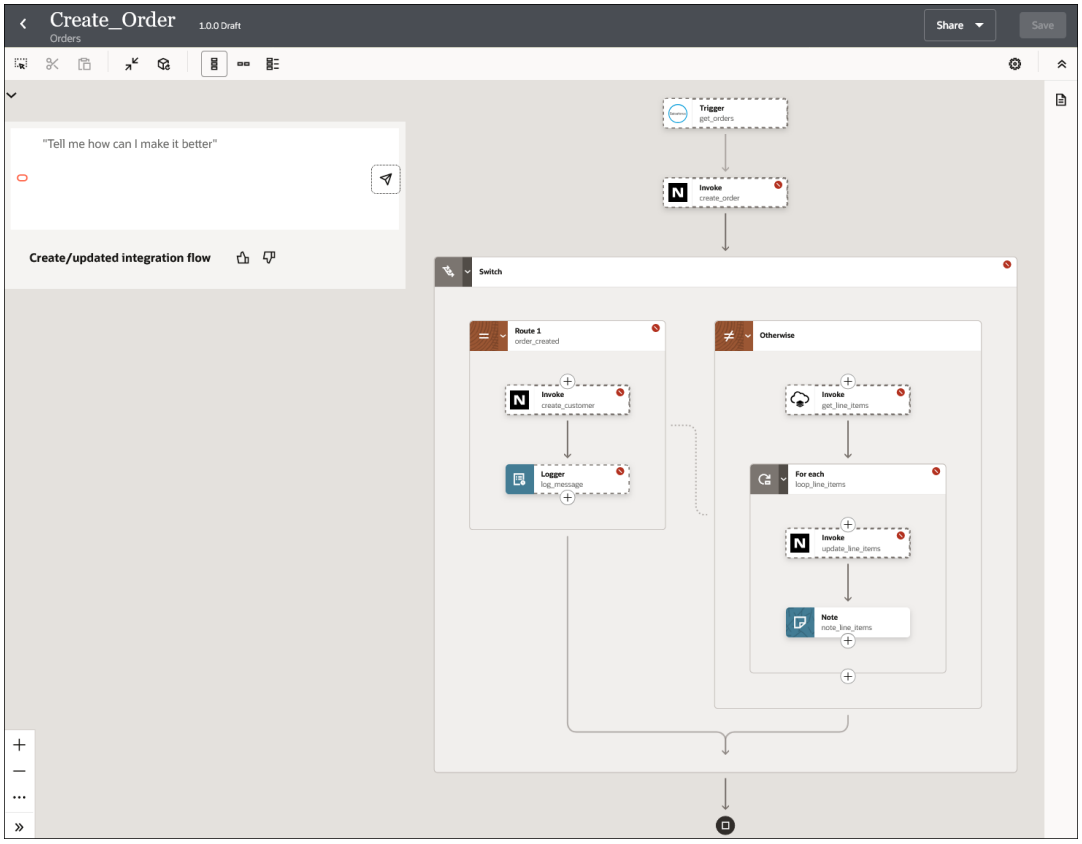
11. Review and save the integration before proceeding to modify the integration flow using AI.
12. Enter the following natural language text to edit the integration flow, and click **Generate the integration skeleton based on the user's goal** .

```
Replace logging of line items with note activity.
```


Copyright © 2022, 2026, Oracle and/or its affiliates.

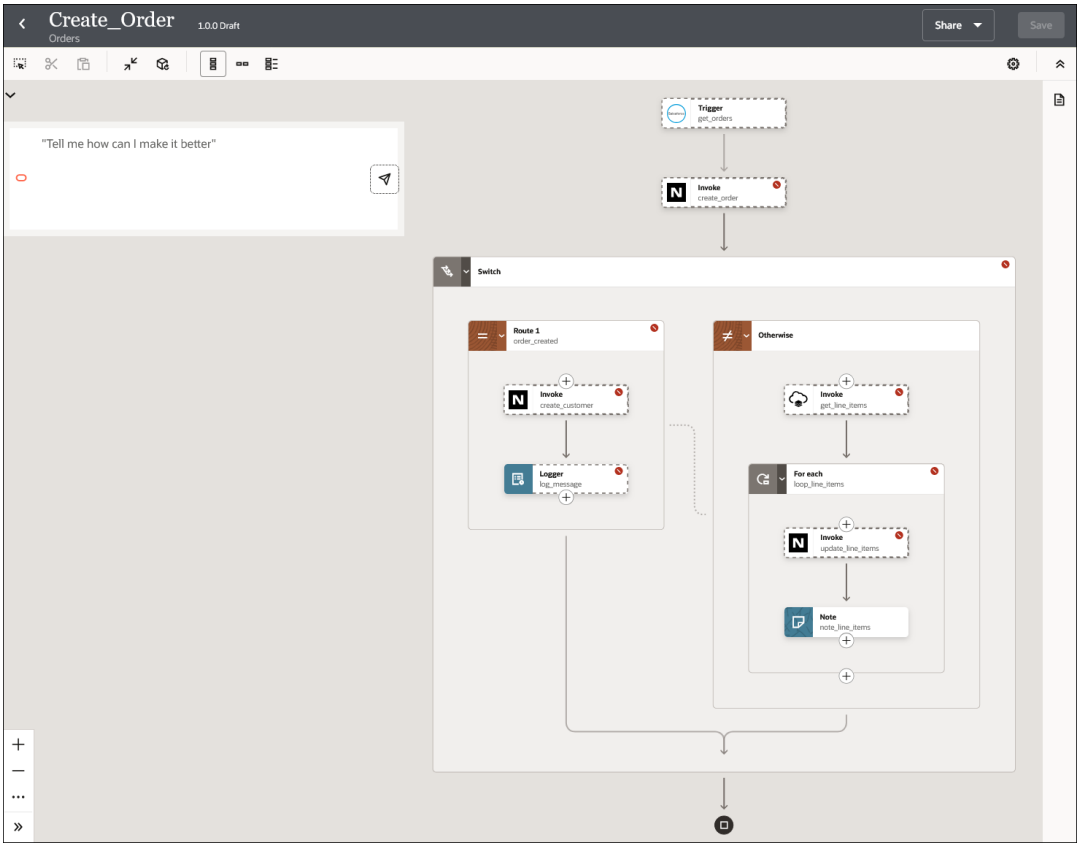


13. Review and save the integration.



Copyright © 2022, 2026, Oracle and/or its affiliates.

14. In the **Create/updated integration flow** option, click **Accept AI generated actions** to accept the integration flow created by AI.
The integration flow created by AI is retained and displayed on the canvas.



You can modify this skeleton integration further as needed. You can activate the integration after you've configured it.

Use AI to Create Integrations Based on Integrations from the Integration Store

You can use AI to quickly create integrations based on prebuilt integrations from the Integration Store. After AI creates the integration skeleton, you can configure it further or modify it, as needed.

> **Note**
>
> - This feature is available only within a project.
> - AI features are available in select regions. See AI Feature Availability in *Provisioning and Administering Oracle Integration 3*.

Typically, when creating integrations manually, you might need to perform several steps. If there are integrations in the Integration Store that match your requirements, you can reuse one

Copyright © 2022, 2026, Oracle and/or its affiliates.

that is most relevant to quickly create a new integration automatically. Once AI creates the integration skeleton, you can customize it, as needed. This reduces the number of manual steps that you need to perform when creating an integration.

> **Note**
>
> This feature does not replace the existing integration when reusing it. It will create a new integration.

1. In the navigation pane, click **Projects**.
2. Select the project name.
3. Click **Integration** .
4. In the **Integrations** section, click **Add** if no integrations currently exist or **+** if integrations already exist.
   The Add integration panel opens.
5. Click **Start with a prebuilt integration**.
   The Start with a prebuilt integration page opens.
6. On the Start with a prebuilt integration page, enter the purpose of the integration (a short description) using natural language and press **Enter**. For example: `Create sales orders in Oracle ERP Cloud`. Based on the integration detail that you entered, the AI shows a list of recipes that have integrations that match your natural language text.
7. Perform one of the following tasks.
   To reuse an integration from the results that are displayed:
   a. Click **Choose** to select the recipe that closely matches your needs.
      The Create integration panel opens.
   b. The values for name and description are prepopulated. Edit the name and description, as needed.
   c. Click **Finish**.
      The integration from the selected recipe is imported with the new name you specified, and the Configuration Editor page appears. On the Configuration Editor page, you can edit the connections, if needed.
   d. Click **Go back** to go to the **Design** tab.
   e. In the **Integrations** section, click the integration you created.
      The integration canvas opens.
   f. Review the integration and modify it as needed.

   To create your own integration, and build exactly what you need (if you don't want to reuse an integration from the recipes that are listed):
   a. On the Start with a prebuilt integration page, click **Create my own integration** in the upper right corner of the page.
      The Create integration panel opens.
   b. Enter a name, an optional description, and a detailed description (describing the integration flow you want to create).
   c. Click **Create**.
      The AI identifies the type of integration and creates a skeleton integration based on the detailed description you provided. The integration canvas displays the skeleton

Copyright © 2022, 2026, Oracle and/or its affiliates.

integration that was created. You can modify it as needed. See Use AI to Create an Integration.

Add and Share a Connection Across a Project

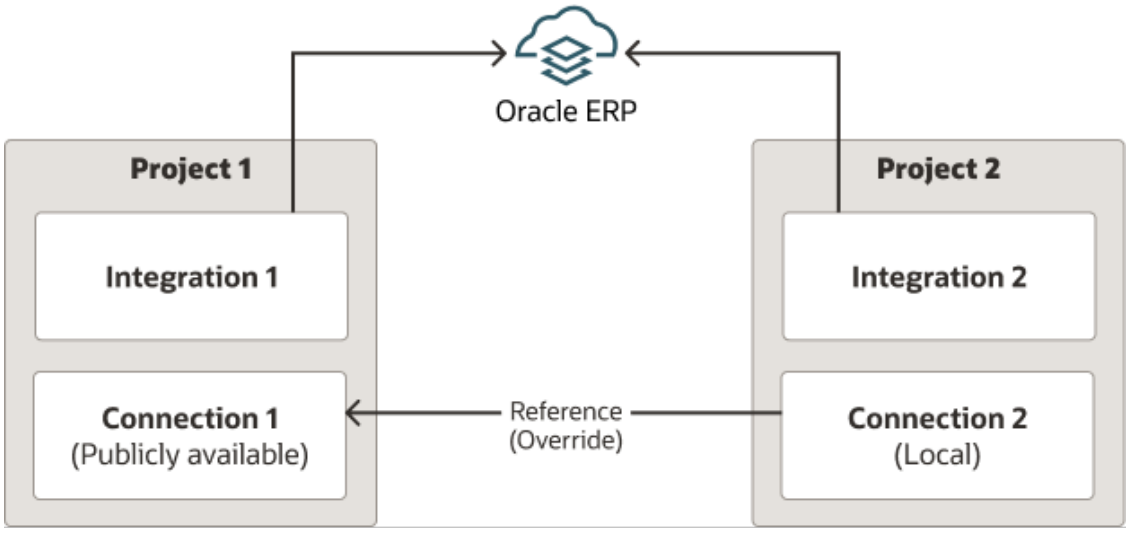
You can add a connection to use in an integration in a project. During creation, you can also select to make the connection available across projects. This selection allows you to share the connection resources from one project with other projects. This feature also eliminates the need to create separate connections in each project.

- Share Connection Resources Across Projects
- Refresh Endpoints Using Connection Sharing
- Share Connection Resources in Accelerator Projects

Share Connection Resources Across Projects

You can share adapter connection resources across projects. For example, you may have two projects that need to integrate with a common system, such as Oracle ERP Cloud. You create and configure the following connection types:

- Publicly available connection: You create, configure, and maintain a single Oracle ERP Cloud Adapter connection that you select to share with other projects. This selection makes the connection publicly available to adapters of the same adapter type and role in other projects.
- Local connections: You create additional connections in other projects that reference the publicly available connection. Referencing the connections means that you don't explicitly configure connection properties, security properties, or access types for these connections. You override the local connection settings by referencing the settings of the publicly available connection. This reference enables you to share resources across each project.



Copyright © 2022, 2026, Oracle and/or its affiliates.

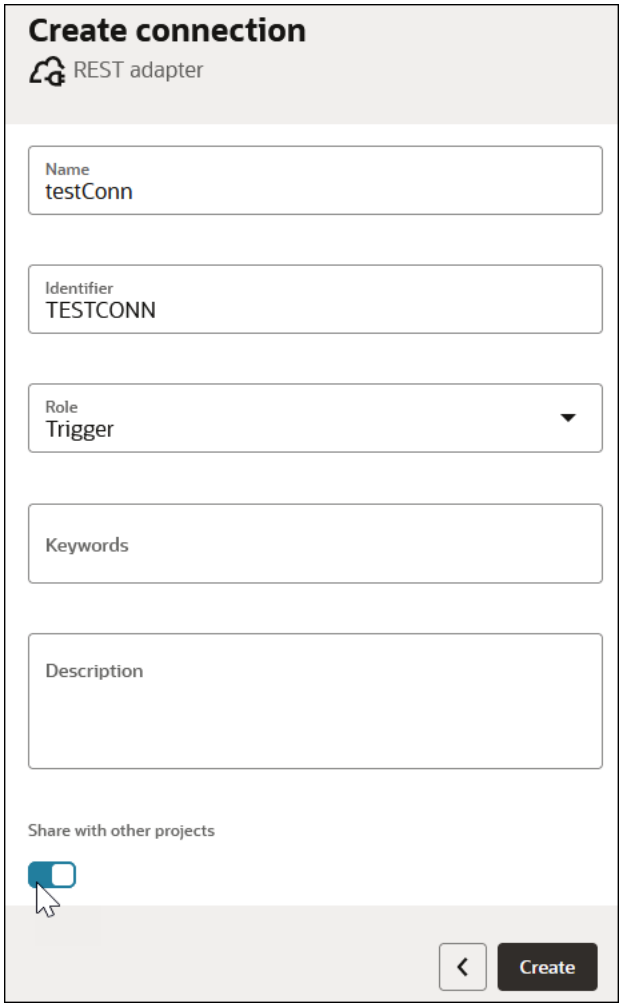
> **Note**
>
> Note the following connection sharing restrictions:
>
> - If you update the connection configuration settings for a publicly available connection, the activated integrations for any local connections that are referencing the publicly available connection must be reactivated to capture those updates.
> - Lookup and JavaScript library resources cannot be shared across projects.
> - You cannot track failed integration instances by the publicly-available connection.

The following example describes how to configure connection sharing:

- You create and configure a publicly available connection.
- You create a local connection that references the configuration settings of the publicly available connection.

1. Open a project.
2. In the **Connections** section, click **Add** if no connections currently exist or **+** if connections already exist.
3. Create an adapter connection. For this example, a REST Adapter is selected. See Create a Connection.
4. Select a role from the **Role** list (for this example, **Trigger** is selected) and click **Share with other projects** to make this connection publicly available for use across projects.

Copyright © 2022, 2026, Oracle and/or its affiliates.

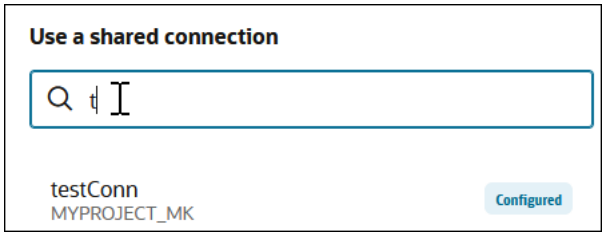


5. Configure the connection properties, connection security, and access control for the publicly available adapter.
6. Test and save the connection, then exit the Connections page.
   The connection is displayed as **Configured** in the project.

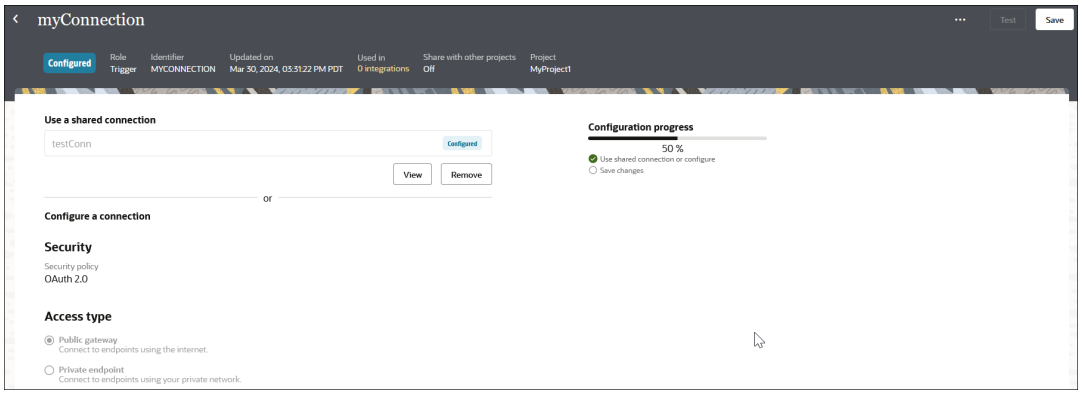


7. Open a different project to create the local connection.
8. Create a connection of the same adapter type (REST Adapter) and same role (**Trigger**) to match the publicly available adapter.
   The Connections page is displayed. Note the **Use a shared connection** field at the top.
9. Enter a letter that is part of the name of the publicly available connection (for this example, `t` is entered).
   A filtered list of connections is returned that shows only publicly available connections of the same adapter type (REST Adapter) and same role (**Trigger**) that include that letter.

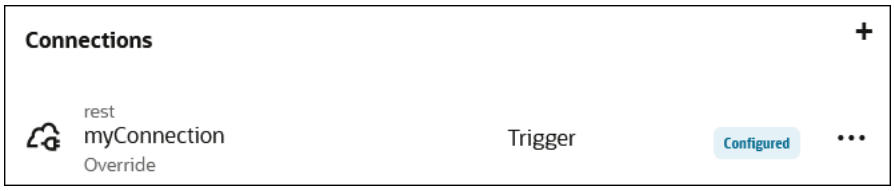
Copyright © 2022, 2026, Oracle and/or its affiliates.



10. Select the publicly available connection (for this example, **testConn**). If the project of the publicly available connection is configured with role-based access control (RBAC) settings, you must have at least view permissions in that project.
11. Click **Save**.
    The local connection automatically references the connection configuration settings of the publicly available connection.

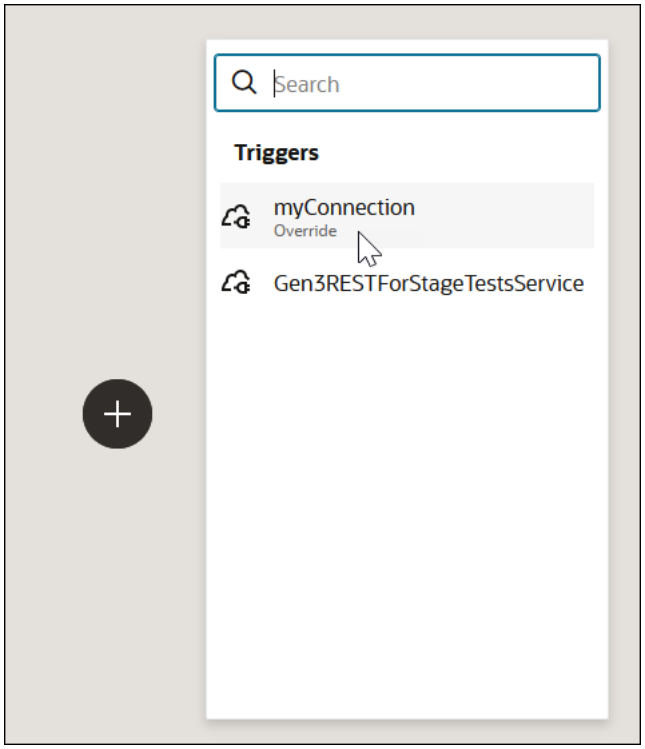


12. Save the connection, then exit the Connections page.
    The connection is displayed as **Configured** in the project. The **Override** label is displayed, meaning this connection references the resources of the publicly available connection.



13. Create an integration.
    The local connection is available for selection in the integration canvas. The **Override** label indicates that this connection is referencing the resources of the publicly available connection in the other project.

Copyright © 2022, 2026, Oracle and/or its affiliates.

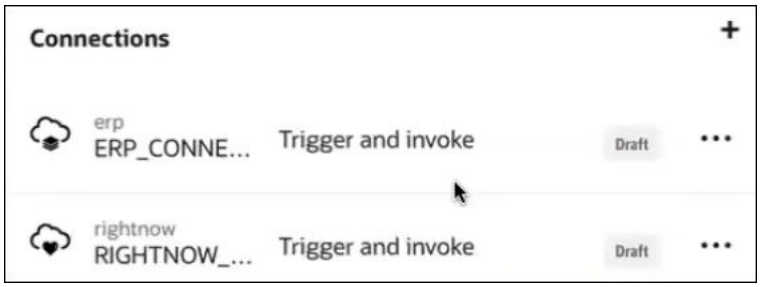


14. Design the remaining parts of your integration.
15. Activate and run the integration.
16. View the activity stream and note that the integration instance was run by referencing the resources of the publicly available connection.

Refresh Endpoints Using Connection Sharing

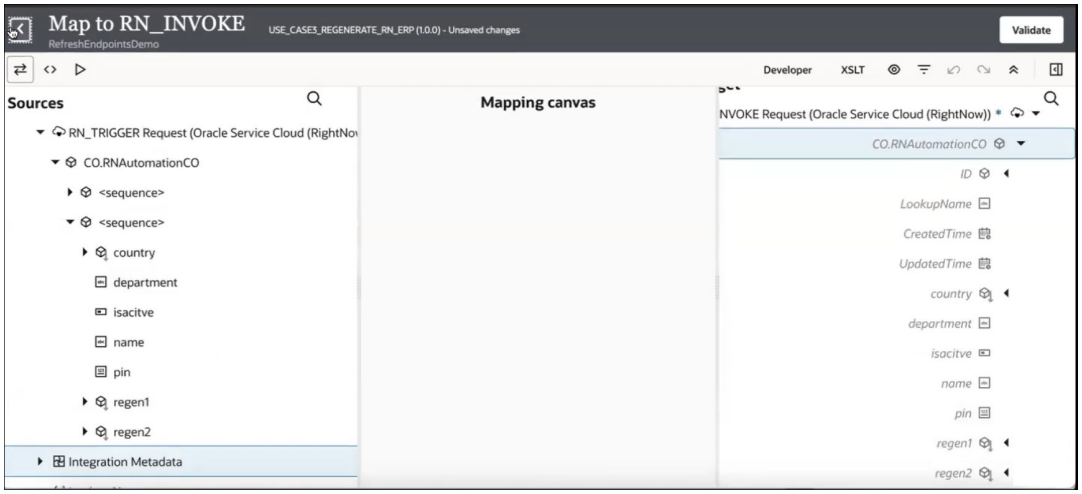
You can override a local connection in a draft state with a publicly-available connection in a configured state, and then regenerate the endpoints for the local connection. This section describes this capability with a brief example.

1. Create a local connection and leave it in a draft state (for this example, the connection is named **RIGHTNOW_CONN**).

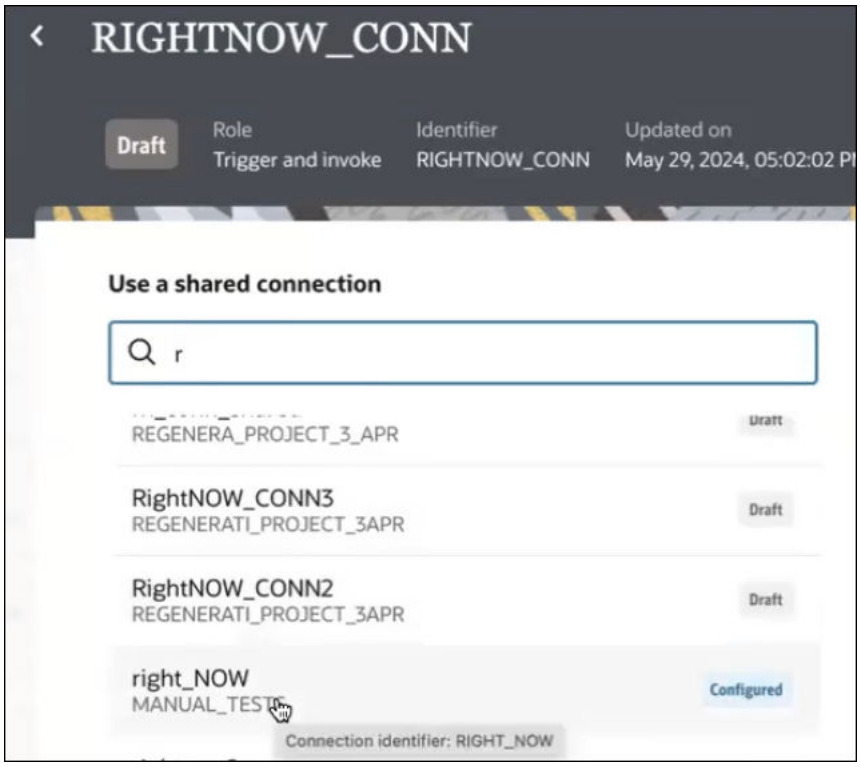


2. Go to the integration canvas and view the map action for the connection. This shows the contents of the mapper prior to endpoint refreshing. Note that there are source and target elements named **regen 1** and **regen 2**.

Copyright © 2022, 2026, Oracle and/or its affiliates.

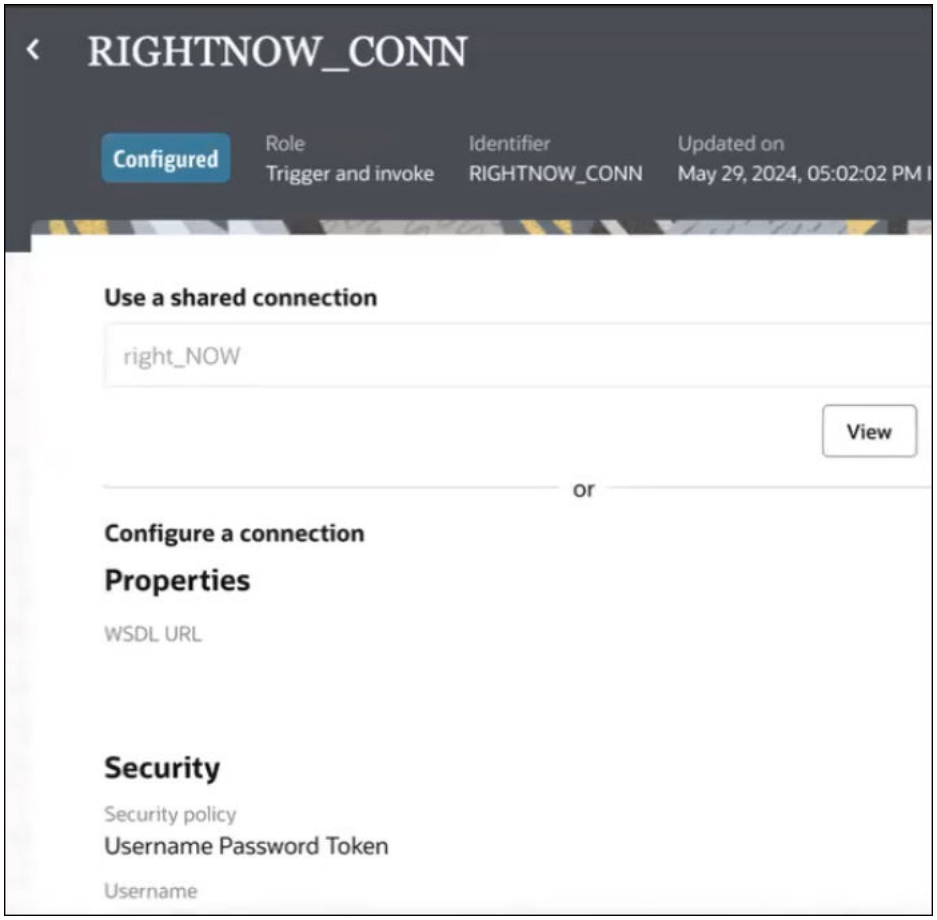


3. Return to the **Connections** section and open the connection (**RIGHTNOW_CONN**) for editing.
4. In the **Use a shared connection** field, enter the beginning letter to filter the list, and select the shared connection to use (for this example, named **right_NOW**).



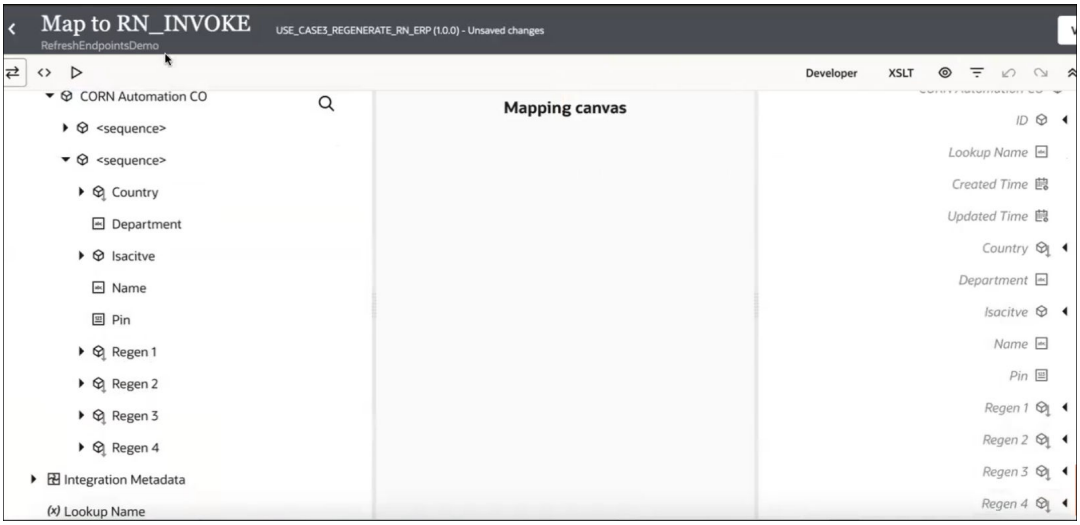
5. Save the connection.
The connection state is changed to configured.

Copyright © 2022, 2026, Oracle and/or its affiliates.



6. Go to the **Integrations** section of the project.
7. Hover over the row of the integration, and select **Actions** ⋮, and then **Refresh endpoints**.
   The Refresh endpoints panel opens.
8. Select the connection to refresh (**RIGHTNOW_CONN**), then click **Refresh**.
9. Wait for the refresh to complete.
10. Return to the mapper for the connection. Note that there are now source and target elements named **Regen 1**, **Regen 2**, **Regen 3**, and **Regen 4**.

Copyright © 2022, 2026, Oracle and/or its affiliates.



11. Validate the mappings.
12. Save and exit the integration.

Share Connection Resources in Accelerator Projects

When you install an accelerator project, all connections are initially in a draft state. You can use connection sharing for these connections instead of configuring each connection individually.

For example, if the accelerator project is using Oracle ERP Cloud Adapter and Twilio Adapter connections, you can use connection sharing if the following conditions are met:

- There are publicly available connections for Oracle ERP Cloud Adapter and Twilio Adapter.
- The roles are the same.
- You have view permissions at a minimum on the project that includes the publicly available connections.

After you configure connection sharing for these adapters, you can activate the integration in the accelerator project.

Watch a video to learn more:

Video

Add and Share a Lookup Across a Project

When you create a lookup in a project, you can select to make it available across other projects. This selection allows you to share the lookup resources of one project with other projects. This capability eliminates the need to create separate lookups with the same resources in each project.

Capabilities

- You can share the lookup resources of one project with other projects.
- When you clone a lookup that is shared across projects, the new lookup inherits the shared references from its origin.
- You can update a lookup that is shared across projects, and those changes are inherited by any project referencing the lookup.

Copyright © 2022, 2026, Oracle and/or its affiliates.

- You can clone a project that includes a shared lookup, and those settings are inherited by the new project.
- Lookups created prior to release 25.10 are by default not selected to be shareable with other projects. You must manually select those lookups to be shared across projects.
- You can export and import projects that share lookups into another instance. However, you must first import the project that includes the lookup before you can successfully import the project that references the lookup.
- There is no limit to the number of projects with which to share a lookup.

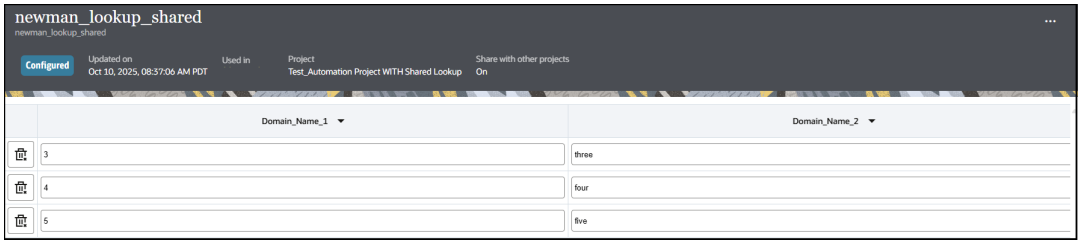
Restrictions

- A project with a sharable lookup is not backwards compatible with versions older than 25.10. For example, if you create an integration using a shareable lookup in 25.10, then export and import it to an earlier version (for example, 25.06), activation fails because that version does not include that feature.
- You cannot change a lookup from shared to not shared if it is currently referenced by at least one project.
- You cannot delete a lookup in the project context if it's used by any integration, whether local to that project or in another project. You must remove those dependencies before you can delete the lookup.
- Lookups used in any integration (even in a different project) cannot be deleted. You must first remove the lookup from the integration.
- Multilevel lookup hierarchies are not supported. When a shared lookup is referenced by another project, no local copy of that shared lookup is created in the consuming project. Instead, the reference maintains a direct link to the source shared lookup. A shared lookup, however, can be directly referenced by multiple projects concurrently, ensuring reusability and consistency without hierarchical chaining.

Share Lookup Resources Across Projects

This section provides an overview of how to share a lookup across projects.

1. Open a project.
2. Create a lookup and select to share it across projects. See Create a Lookup.
   For this example, the lookup consists of the following content. Note that **Share with other projects** is set to **On**.



3. Open another project.
4. Design an integration.
5. Add a log action from which to reference the lookup.
   The Configure logger panel is displayed.
6. Click **Functions**, then expand **Integration Cloud**.

Copyright © 2022, 2026, Oracle and/or its affiliates.

7. Select **sharedLookupValue**. You configure this function to reference the resources of the shared lookup created in the other project. This function consists of six parameters:

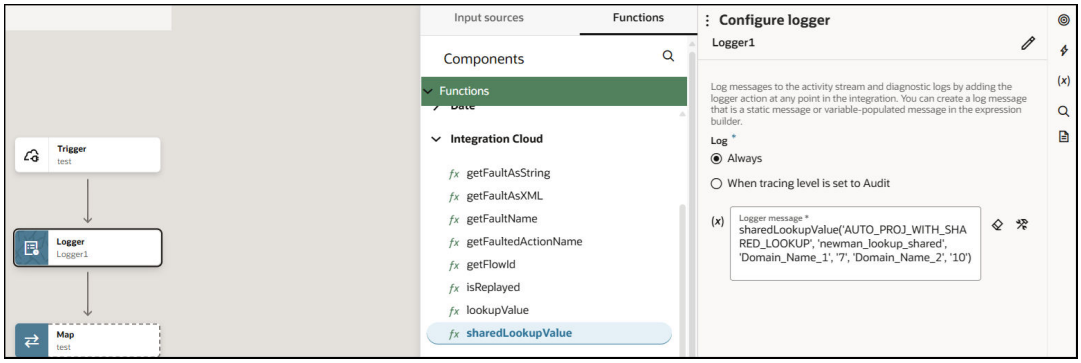
   ```
   dvm:sharedLookupValue('Project_Identifier', 'Lookup_Name',
   'source_column', srcValuePlaceHolder, 'target_column', 'default_value')
   ```

   Where:

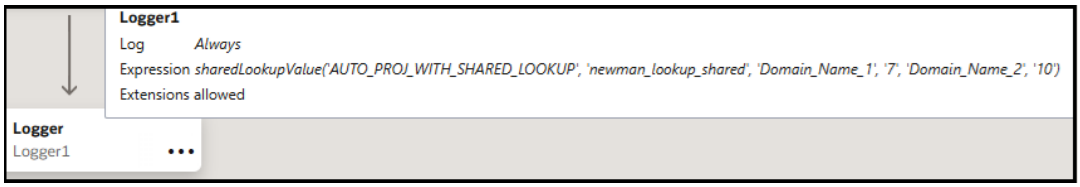
   - *Project_Identifier*: The identifier value of the project in which you created the lookup to share across projects.
   - *Lookup_Name*: The name of the lookup that you selected to share across projects.
   - *source_column*: The name specified for the first column when creating the lookup. The name can be the default value of Domain_Name_1, an edited domain name you specified, or an adapter that you selected to replace the domain name.
   - *srcValuePlaceHolder*
   - *target_column*: The name specified for the second column when creating the lookup. The name can be the default value of Domain_Name_2, an edited domain name you specified, or an adapter that you selected to replace the domain name.
   - *default_value*

   For this example, the function is defined in the log action as follows:

   ```
   sharedLookupValue('AUTO_PROJ_WITH_SHARED_LOOKUP', 'newman_lookup_shared',
   'Domain_Name_1', '7', 'Domain_Name_2', '10')
   ```



8. Close the Configure logger panel and save the configuration.
9. Hover over the logger. The expression is displayed that enables this second project to reference the resources of the shared lookup created in the first project.



10. Complete and activate your integration.

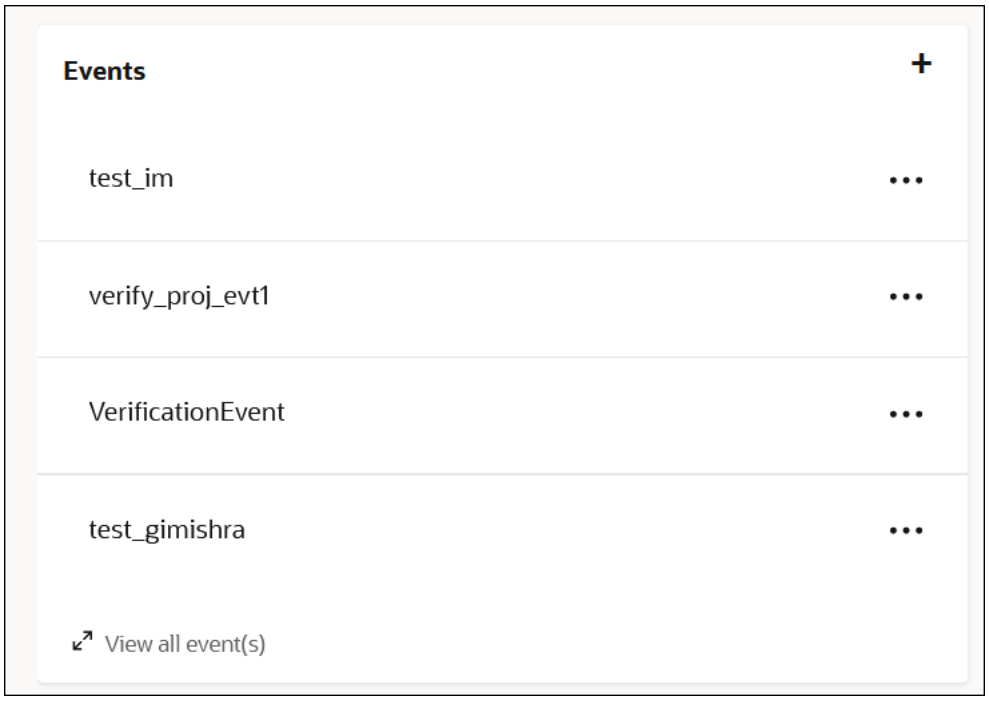
Copyright © 2022, 2026, Oracle and/or its affiliates.

Create, Publish, and Subscribe to Events in a Project

You can create events that you publish and subscribe to in integrations in a project.

The process you follow to create, publish, and subscribe to an event in a project is the same as outside of a project. The event to which to subscribe can be in the same project or in a different project. The following steps provide an overview of the process and cross-reference other sections of this guide for more specific procedures.

1. In the navigation pane, click **Projects**.
2. Click the project name or click .
3. Scroll down to the **Events** section at the bottom.
4. In the **Events** section, click **+** to create an event.
   The Define new event panel opens.
5. Create the event payload structure. You can also select to share the event with other projects. See Step 3 in Create Events to Publish and Subscribe to in Integrations.
6. After you create events, you can view and perform additional actions in the **Events** section of the projects.



7. Click **Actions** ... to access a menu with the following options:

Option	Description
Clone	Enables you to clone the event.
Edit	Enables you to edit the event payload structure.
Delete	Enables you to delete an event. You cannot delete an active event.

Copyright © 2022, 2026, Oracle and/or its affiliates.

Option	Description
View	Enables you to view the event payload structure, the integrations publishing the event, and the integrations subscribing to the event.

8. Go to the **Integrations** section of the project.
9. Create an application or schedule integration to publish the event. During integration design, you drag a **Publish event** action into the integration canvas and select the event to publish. See Publish Events in an Integration. The event can be in the same project or in a different project. See Share Events Across Projects.
10. Return to the **Integrations** section of the project.
11. Create an event integration to subscribe to the event. You must select an **Event** integration pattern. During integration design, you select the event to which to subscribe. The event can be in the same project or in a different project. See Subscribe to Events in an Integration and Share Events Across Projects.
    The subscribing integration event is triggered when the integration publishing the event is run. Multiple integrations can subscribe to the same published event.

Create an AI-Powered Event Description

You can use AI to generate a new description or replace an existing description of an event with a single click.

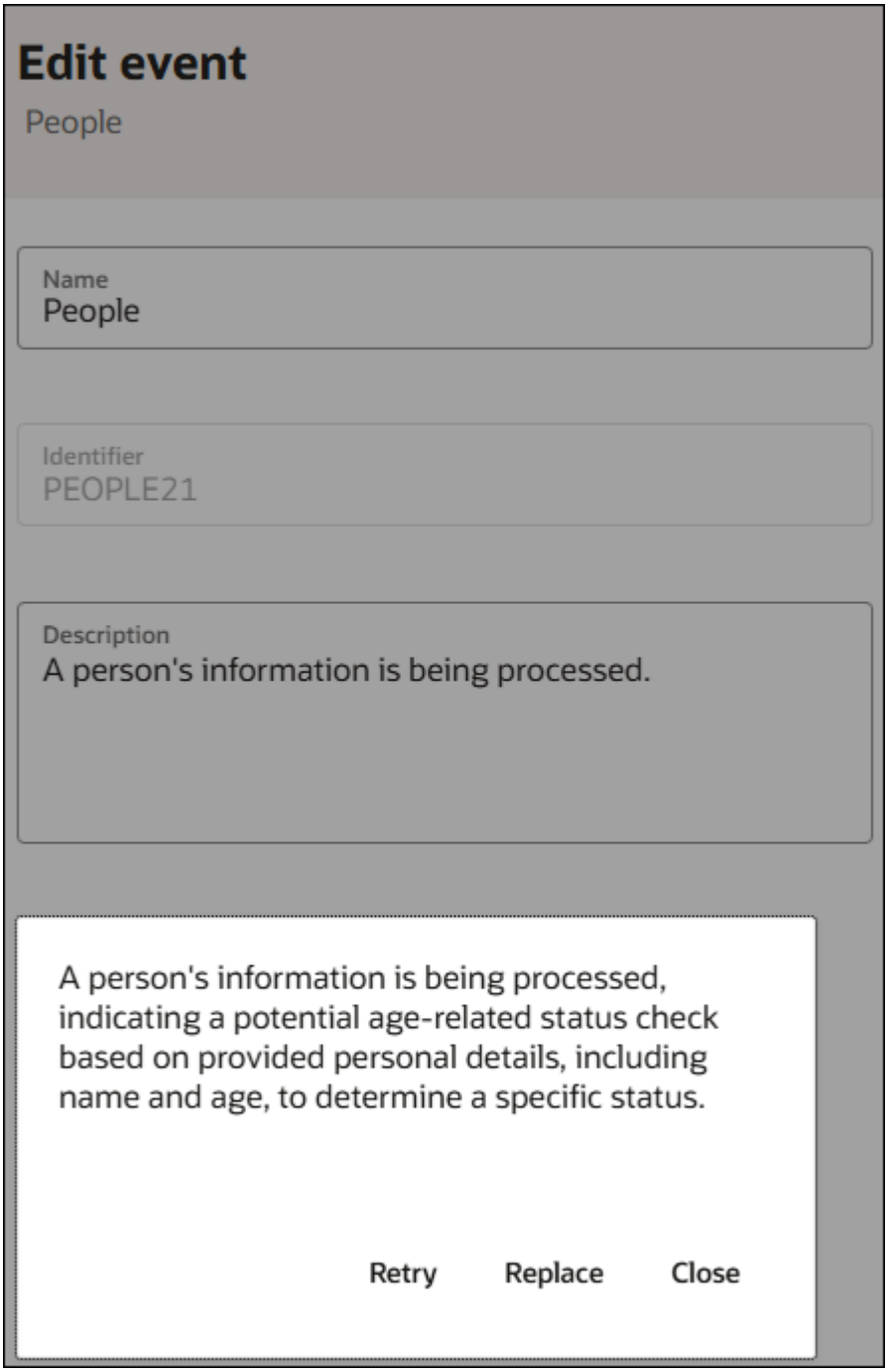
> **Note**
>
> AI features are available in select regions. See AI Feature Availability in *Provisioning and Administering Oracle Integration 3*.

1. In the navigation pane, click **Projects**.
2. Select the project name.
3. Click **Integration**.
4. In the **Events** section, hover over the event for which to create an AI-generated description.
5. Click **Actions**, then select **Edit**.
6. In the Edit Event panel, click **Generate**.

   The **Description** field can be empty or already include a description.

   - Click **Retry** to regenerate the description. If you retry, the newer description varies slightly in terms of the words used, but the meaning remains the same.
   - Click **Replace** to add the generated event description to the **Description** field. If the **Description** field is empty, the generated event description is added to the field. If the **Description** field already includes an event description, the current description is replaced by the generated description. For this example, a description already existed. **Replace** is only selectable if the event is editable.

Copyright © 2022, 2026, Oracle and/or its affiliates.



7. Click **Continue**.
8. In the Define event structure panel, click **Continue**.
9. Click **Save**.

Create an AI-Powered Project Description

You can use AI to generate a new description or replace an existing description of a project with a single click.

See AI Innovation and Oracle Integration.

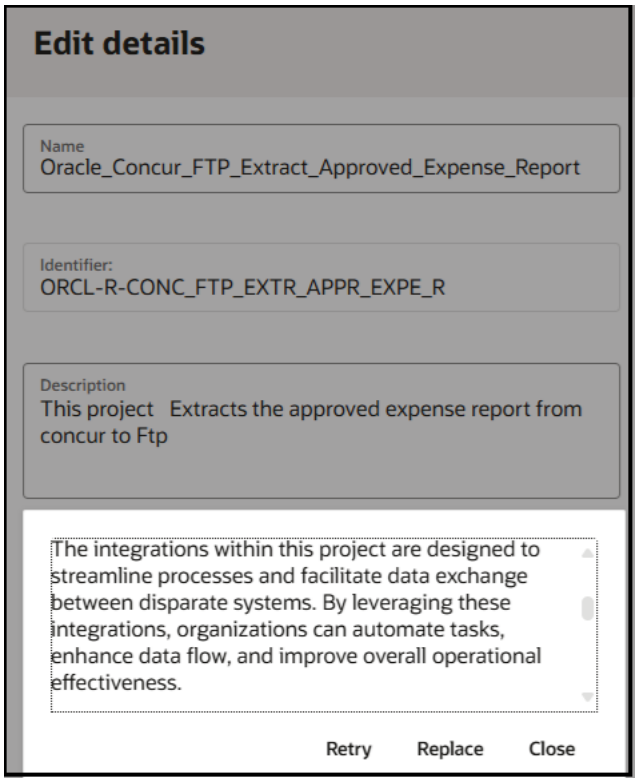
Copyright © 2022, 2026, Oracle and/or its affiliates.

1. In the navigation pane, click **Projects**.
2. Select the project name.
3. In the upper right corner, click  .
4. Click **Generate**.

> **Note**
>
> - If you have not configured an AI engine, an error message is displayed when you click **Generate**.
> - If you have reached the limit on your number of API keys, you receive an unauthorized key error.

The **Description** field can be empty or already include a description.

- If the **Description** field is empty, the generated project description is added to the field.
- If the **Description** field already includes a project description, the updated description is displayed as a popup. For this example, a description already existed.



5. If the **Description** field already includes a description, perform one of the following steps:
   a. Click **Retry** to regenerate the description. If you retry, the newer description varies slightly in terms of the words used, but the meaning remains the same.
   b. Click **Replace** to replace the current description in the **Description** section. **Replace** is only selectable if the project is editable.

Copyright © 2022, 2026, Oracle and/or its affiliates.

6. Click **Save Changes**.

Create Robots in a Project

The only place to build robots and their related components is in a project.

All tasks related to robots and robotic process automation are documented in a separate guide. The following table provides links to this guide.

Goal	More information
Enable robots in your instance	Start Building Robots Today! in *Using Robots in Oracle Integration 3*
Learn about robots	Learn About Robots in *Using Robots in Oracle Integration 3*
Build a robot	Quick Start for Building Robots in *Using Robots in Oracle Integration 3*

Create Healthcare Integrations in a Project

You can create HL7 messages, create schemas, import FHIR profile packages, and create integrations using the healthcare action in a project if your Oracle Integration instance includes the Healthcare edition.

This table provides an overview of healthcare tasks in a project and provides references to relevant documentation.

Functionality	See
Install or upgrade to the Healthcare edition. To create healthcare integrations in a project, your Oracle Integration instance must include the Healthcare edition.	• Create an Oracle Integration Instance in *Provisioning and Administering Oracle Integration 3* • Edit the Edition, License Type, Message Packs, and Custom Endpoint of an Instance in *Provisioning and Administering Oracle Integration 3*

Copyright © 2022, 2026, Oracle and/or its affiliates.

Functionality	See
You can perform the following healthcare-related tasks in projects: • Create, clone, edit, view, and delete HL7 messages and schemas. • Select the HL7 message when configuring the healthcare action in an integration in the Integrations section of the project. • Import FHIR profile packages. • Export a project that includes healthcare resources and import that project into another instance. • Clone a project that includes healthcare resources. • Control the users and groups that edit, view, and monitor a project that includes healthcare resources with role-based access control (RBAC). • View HL7 messages and healthcare schemas in the dependency diagram. • Create, clone, edit, and view Clinical Document Architecture (CDA) documents • Ensure messages are processed in the received order .	• Tour of the Projects Workspace • Create Oracle Integration for Healthcare Schemas and Documents in *Using Oracle Integration for Healthcare in Oracle Integration 3* • Import FHIR Profile Packages in *Using Oracle Integration for Healthcare in Oracle Integration 3* • Convert HL7 Messages with a Healthcare Action in *Using Oracle Integration for Healthcare in Oracle Integration 3* • Create Clinical Document Architecture (CDA) Documents in *Using Oracle Integration for Healthcare in Oracle Integration 3* • Ensure Messages are Processed in the Received Order in *Using Oracle Integration for Healthcare in Oracle Integration 3*

Create B2B Integrations in a Project

You can create B2B messages, schemas, and integrations using the B2B action in a project.

This table provides an overview of B2B tasks in a project and provides references to relevant documentation.

Copyright © 2022, 2026, Oracle and/or its affiliates.

Functionality	See
You can perform the following B2B-related tasks in projects: • Control the users and groups that can edit, view, and monitor B2B resources in a project with role-based access control (RBAC). • Create, clone, edit, view, and delete B2B documents and schemas. • Select the B2B document when configuring the B2B action in an integration in the **Integrations** section of the project. • Create trading partners, define transports, create agreements, and export and import a trading partner. • Export a project that includes B2B resources and import that project into another instance. • Activate and deactivate a project deployment that includes B2B resources. • Clone a project that includes B2B resources. • Delete a project that includes B2B resources. Not all B2B features are currently supported in projects. See Design B2B Integrations in Projects or in Standalone Environments.	• Design B2B Integrations in Projects or in Standalone Environments in *Using B2B for Oracle Integration 3* • Tour of the Projects Workspace • B2B Schemas and B2B Documents in *Using B2B for Oracle Integration 3* • Manage B2B Trading Partners in *Using B2B for Oracle Integration 3* • Use the B2B Action in Standalone Mode in *Using B2B for Oracle Integration 3*

Create Decision Models in a Project

You can only create decision models in a project.

Functionality	See
Create decision models	Create Decision Models in *Using Decisions in Oracle Integration 3*
Create decisions and subdecisions in the models	Add Decisions in *Using Decisions in Oracle Integration 3*
Define the input data and type for decisions	Define Decision Input and Type in *Using Decisions in Oracle Integration 3*
Create associated services to use decision models in integrations	Connect Nodes on the Canvas in *Using Decisions in Oracle Integration 3*
Configure the logic for a decision node	Configure a Decision's Logic in *Using Decisions in Oracle Integration 3*
Update, clone, edit, view, deactivate, or delete a decision model	Manage Decision Models in *Using Decisions in Oracle Integration 3*

Create Human in the Loop Workflows in a Project

You can only create human in the loop workflows and their related components in a project.

Functionality	See
Create and configure workflows	Model Workflows in *Using Human in the Loop in Oracle Integration 3*
Create forms	Design Forms in *Using Human in the Loop in Oracle Integration 3*

Copyright © 2022, 2026, Oracle and/or its affiliates.

Functionality	See
Manage workflows and forms	Manage Workflows and Forms in *Using Human in the Loop in Oracle Integration 3*
Use workflows with agentic AI	Use Human Approval Workflows with Agentic AI in *Using Human in the Loop in Oracle Integration 3*

Use AI to Generate Project Documentation

You can use AI to create documentation that describes a project.

> **Note**
>
> AI features are available in select regions. See AI Feature Availability in *Provisioning and Administering Oracle Integration 3*.

1. In the navigation pane, click **Projects**.
2. Perform one of the following steps:
   a. On the Projects page, hover over the project for which to generate documentation using AI.
   b. Click **Actions** ⋯, then select **Learn about this project**.
   or
   c. On the Projects page, click the project name for which to generate documentation using AI.
   d. In the upper right corner, click **Learn about this project**.

   The Learn about this project panel opens.
3. In the Learn about this project panel, you can view the documentation generated by AI. The documentation generated by AI might usually include information such as purpose of the solution, overview, general details, applications and access, technologies in this project (which might include details about integrations, healthcare, B2B, robots, decisions, and so on), shared resources, unused resources, deployments, change history, testing suggestions, and so on. You can also perform the following in the Learn about this project panel.
   - Use the search field to search for information in the documentation generated by AI. Enter the text or keyword to search for in the search field, and press **Enter**. The text you enter is highlighted in the documentation wherever it appears. For example, if you want to search for healthcare-related information, enter `healthcare` in the search field. The text `healthcare` is highlighted wherever it appears in the documentation. Press **Enter** to see the next instance where the search text appears in the documentation.
   - Use the **Content** navigation pane to navigate to the various sections of the documentation.
   - Click **Save as PDF** to save the documentation as a PDF.
   - Click **Regenerate documentation** to regenerate the documentation by AI based on the current data and replace the previous documentation. This option is helpful

Copyright © 2022, 2026, Oracle and/or its affiliates.

when you make changes to the integrations, other artifacts, or any other information/ configurations in the project and want to regenerate the documentation.

- In the **Content** navigation pane, click **Technologies in this project**. In the **Technologies in this project** section, in the **Integrations** subsection, click **Learn about this integration** (next to an integration that is listed) to view the integration documentation by AI for that integration. When you click **Learn about this integration** next to the integration, the Learn about this integration panel appears. For information about integration documentation by AI, see Use AI to Generate Integration Documentation. To go back to the Learn about this project panel, click the **Learn about this project** link in the upper left corner of the Learn about this integration panel.
- In the content, if you want to regenerate the documentation for a specific section, click **Regenerate <section name>** next to the section name. The regeneration options in the panel are disabled until the section documentation is regenerated. They are enabled after the section documentation is regenerated.

4. Click **Close** to close the Learn about this project panel.

Use AI to Generate Integration Documentation

You can use AI to create documentation that describes an integration flow.

> **Note**
>
> AI features are available in select regions. See AI Feature Availability in *Provisioning and Administering Oracle Integration 3*.

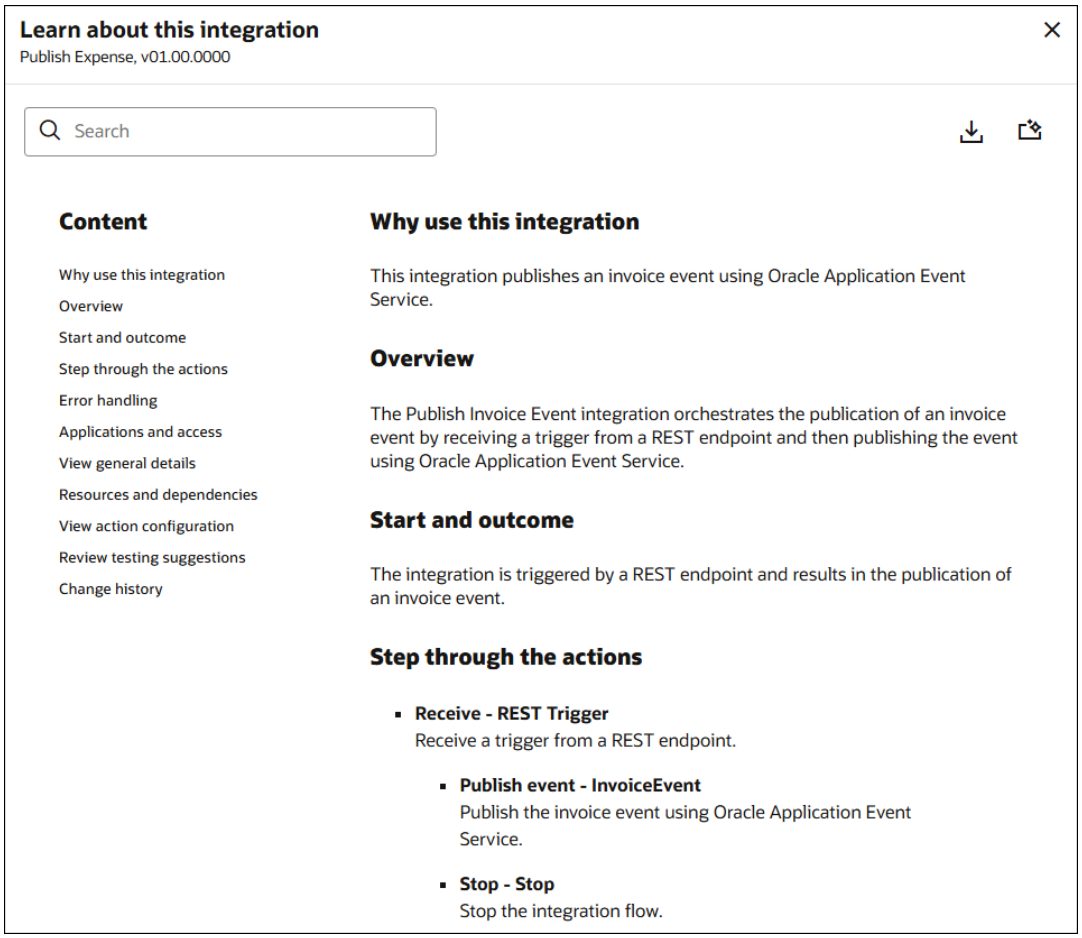
1. In the navigation pane, click **Projects**.
2. Select the project name.
3. Click **Integration**.
4. Perform one of the following steps:
   a. In the **Integration** section, hover over the integration for which to generate documentation using AI based on the current integration flow.
   b. Click **Actions**, then select **Learn about this integration**.
   or
   c. In the **Integration** section, click the integration for which to generate documentation using AI.
   The integration canvas opens.
   d. On the side of the canvas, click **Learn about this integration**.

   The Learn about this integration panel opens.
5. In the Learn about this integration panel, you can view the documentation generated by AI. The documentation generated by AI might usually include information such as purpose of the integration, overview, how the integration is triggered, outcome, actions, error handling, applications and access needed, general details, resources and dependencies, testing suggestions, and so on. You can also perform the following in the Learn about this integration panel.

Copyright © 2022, 2026, Oracle and/or its affiliates.

- Use the search field to search for information in the documentation generated by AI. Enter the text or keyword to search for in the search field, and press **Enter**. The text you enter is highlighted in the documentation wherever it appears. For example, if you want to search for trigger-related information, enter `trigger` in the search field. The text `trigger` is highlighted wherever it appears in the documentation. Press **Enter** to see the next instance where the search text appears in the documentation.
- Use the **Content** navigation pane to navigate to the various sections of the documentation.
- Click **Save as PDF** to save the documentation as a PDF.
- Click **Regenerate documentation** to regenerate the documentation by AI based on the current integration flow and replace the previous documentation. This option is helpful when you make changes to the integration flow and want to regenerate the documentation.

6. Click **Close** to close the Learn about this integration panel.



Manage a Project

You manage a project from a single workspace, including creating project deployments, activating and deactivating project deployments or individual integrations, exporting projects,

Copyright © 2022, 2026, Oracle and/or its affiliates.

extending integrations in an accelerator project, upgrading accelerator projects, setting tracing levels on integrations, testing integrations, deleting projects, exporting projects, and more.

Topics:

- Create and Manage a Project Deployment
- Activate or Deactivate a Project
- Edit and Replace Dependent Resources in a Project
- Export a Project
- Clone a Project
- Integrate Projects and Project Deployments with a GitHub Repository
- Invoke Child Integrations Inside or Outside of Projects
- Manage Accelerator Projects
- Deploy Integration Endpoints to Oracle Cloud Infrastructure API Gateway
- Clone an Integration in a Project
- Update the Tracing Level of Integrations in a Project
- Test REST Adapter Trigger Connection-Based Integrations in Projects
- View the Dependent Relationships Between Project Resources
- View Run Details About Integrations in Projects
- View Project Status
- Edit Project Description Details
- Delete a Project
- Add Retry Logic to Invoke Connections in Projects

Create and Manage a Project Deployment

You can create and activate a project deployment consisting of integrations, B2B, healthcare, decisions, and robots. When you create the project deployment, you select the versions to include. You can also perform project deployment management tasks on user-developed deployments, such as editing, cloning, and deleting the deployment.

Understand the Integration Versioning Life Cycle in a Project Deployment

When you create a project deployment, you must select the integrations and their versions to include. For example, you can select integration A/version 1 and integration A/version 2. You cannot select two or more minor versions of the same major version. By default, the latest version of the integration is selected.

- Life Cycle of a Project Deployment
- Life Cycle of a Project Deployment that Requires a Rollback

Life Cycle of a Project Deployment

The following example describes the life cycle of a project deployment. The following project initially includes two integrations with the following versions prior to creation of the project deployment:

Copyright © 2022, 2026, Oracle and/or its affiliates.

- Project
  - Integration_A
    - Version 01.00
    - Version 01.01
    - Version 02.00
  - Integration_B
    - Version 01.00
    - Version 02.00

Because the project is all-inclusive and integrations are versioned, this project can be deployed in a number of different configurations.

The initial project deployment delivery may include the following integrations and versions:

- Project
  - Integration_A
    - Version 01.00
  - Integration_B
    - Version 01.00

After updates are made to Integration_A, the project deployment is subsequently delivered as follows:

- Project
  - Integration_A
    - Version 01.01
  - Integration_B
    - Version 01.00

At a later time when Integration_A/Version 02.00 is complete, but Integration_B/Version 02.00 requires further development, the project is delivered as follows:

- Project
  - Integration_A
    - Version 02.00
  - Integration_B
    - Version 01.00

In this scenario, development of Integration_B/Version 02.00 can continue without disruption, as can delivery of the project (with Integration_B/Version 01.00).

Life Cycle of a Project Deployment that Requires a Rollback

Assume an error is detected in one of the integrations in a project deployment (for example, in project deployment version 3, which consists of 12 integrations). If you revert to a prior project deployment version, the entire set of integrations in the deployment is impacted. For example:

- Deactivate project deployment version 3. This action deactivates the new minor version that caused the issue, plus all of the other integrations.
- Re-activate project deployment version 2. This action activates the 12 integrations.

Copyright © 2022, 2026, Oracle and/or its affiliates.

While this approach may be reasonable in a test environment, it is not feasible in a production environment. It is recommended that you instead perform a rollback at the individual impacted integration level to avoid impacting the unaffected integrations.

Let's look at a further example of this approach:

- Project deployment version 1 includes 10 integrations and is deployed to a production environment.
- Project deployment version 2 includes 12 integrations (the original 10 integrations, plus two new integrations) and is also deployed to a production environment.
- Project deployment version 3 includes 13 integrations (the 12 integrations, plus a new minor version of one of the integrations) and is also deployed to a production environment. However, once deployed, a bug is identified in the new minor version of that integration.

To resolve this issue, it is recommended that you roll back the single faulty integration in the deployment project:

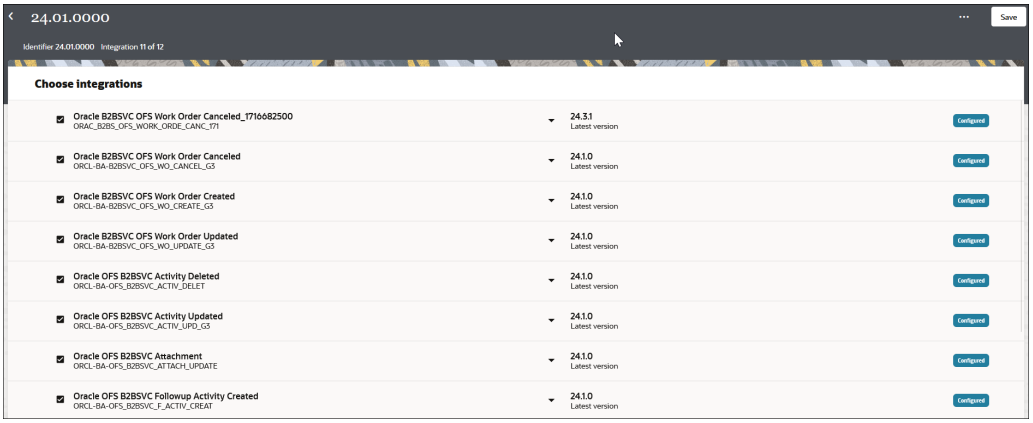
1. Identify the faulty integration version introduced in the latest project deployment (for this example, in version 3).
2. Deactivate *only* the faulty minor version of the integration in version 3.
3. Activate the prior stable version of the same integration in version 2.
4. Fix and validate the faulty integration with testing in the development and test environments.
5. Create and activate a new project deployment (for this example, version 4) that includes the fixed version of the integration.
   This approach minimizes overall impact while preserving availability.

Create a Deployment

1. In the navigation pane, click **Projects**.
2. Click the project name or click  .
3. Click the **Deploy** tab.
4. Click **Create**.

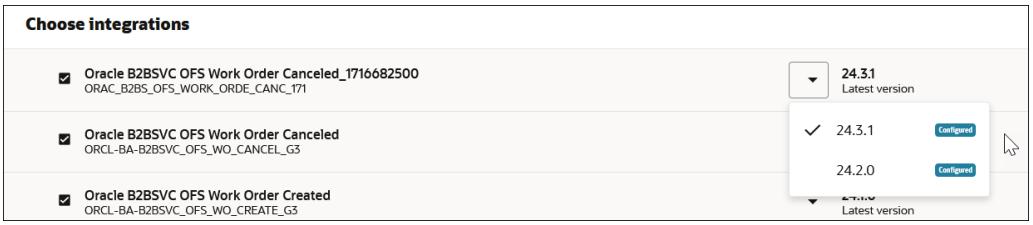
   The Create deployment panel appears.
5. Enter a name, identifier, and optional description. The name is required for activating the project deployment.
6. Select the integrations to include in the deployment.
   a. Scroll through the list of integrations included in the project and select the ones to include in the deployment.

Copyright © 2022, 2026, Oracle and/or its affiliates.



The available versions of each integration are displayed in the second column. By default, the most recently-created versions are selected when you create a deployment.

b. Select the version to include for each integration in the deployment. For this example, select one of the two available versions.



7. Select the robots to include in the deployment.

   Even if you selected an integration that calls a robot, the robot is not automatically included in the deployment. You must manually select all the robots to include.

   a. In the left toolbar, select **Robot** .

   b. Scroll through the list of robots included in the project and select the ones to include in the deployment.

      The available versions of each robot are displayed in the second column. By default, the most recently-created versions are selected when you create a deployment.

   c. Select the version to include for each robot in the deployment.

8. Click **Save**, then exit the Choose integrations page.

9. Click **Activate** .

   The Activate project panel is displayed. The name of the deployment is displayed automatically and cannot be changed.

10. Select the tracing level:

    - **Production**: All activities outside loops and invoke/logger activities inside loops (up to a 1000 iterations) are shown in the activity stream and the data is retained for 32 days.
    - **Audit**: In addition to the production settings, wire payloads of trigger/invoke(s) are also shown in the activity stream and the data is retained for eight days.

11. Click **Activate**.

Copyright © 2022, 2026, Oracle and/or its affiliates.

Manage Project Deployments

The **Deploy** tab provides additional management tasks.

1. In the navigation pane, click **Projects**.
2. Click the project name or click  .
3. Click the **Deploy** tab.
4. Select **Actions** ⋮ for a user-developed project deployment, and note the following options. For accelerator projects imported into your instance, only the **View** option is visible. See Understand an Imported Oracle Accelerator Project.

Action Option	Description
Edit	Select to edit a project deployment.
View	Select to view a project deployment in read-only mode.
Clone	Select to clone a project deployment.
Export	Select to export a project deployment.
Delete	Select to delete a project deployment. **Note**: If you delete an individual integration, it is removed from all deployments in which it is a included.

Activate or Deactivate a Project

You can activate or deactivate a project. Project activation or deactivation results in bulk processing of the selected integrations in the project.

- Activate a Project
- Deactivate a Project

Activate a Project

Before you can activate a project, you must create a project deployment and explicitly select the integrations and their versions to include in the deployment. During activation, each contained asset is visited and activated. It's possible that an asset may already be in the necessary state, in which case no action is taken. Project activation is ideally all or nothing, although those assets activated individually are taken into consideration. Assets can be activated in any order. See Create and Manage a Project Deployment.

1. In the navigation pane, click **Projects**.
2. Click the project name or click  .
3. In the upper right corner, click **Activate**.
   The Activate project panel opens.
4. Select a project deployment name.

Copyright © 2022, 2026, Oracle and/or its affiliates.

> **Note**
>
> If there are no deployment names available, create a project deployment under the **Deploy** and select the specific integrations and integration versions to include.

5. Select a tracing level:
   - **Production**: All activities outside loops and invoke/logger activities inside loops (up to a 1000 iterations) are shown in the activity stream and the data is retained for 32 days.
   - **Audit**: In addition to the production settings, wire payloads of trigger/invoke(s) are also shown in the activity stream and the data is retained for eight days.
6. Click **Activate**.

Deactivate a Project

A project is deactivated in its entirety. All assets are deactivated regardless of whether an asset was activated within the context of a project or individually. Assets can be deactivated in any order.

1. In the navigation pane, click **Projects**.
2. Click the project name or click .
3. In the upper right corner, click **Deactivate**.
   The Deactivate project panel opens.
4. Click **Deactivate**.

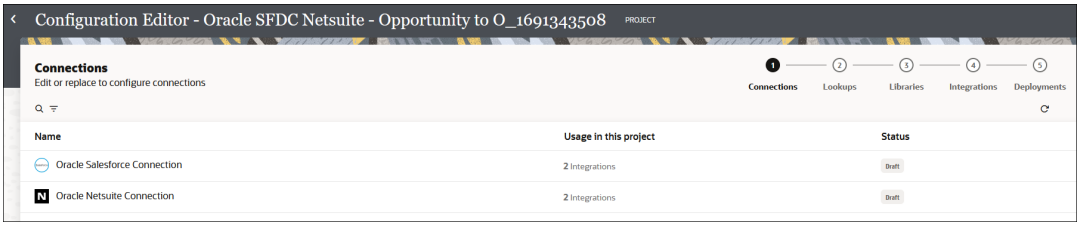
Edit and Replace Dependent Resources in a Project

You can edit dependent resources (connections, libraries, lookups, and JavaScript libraries) in a single integration in a project or all integrations in a project with a wizard that's accessible from the Configuration Editor page.

- Edit Dependent Resources at the Project Level
- Edit and Replace Dependent Resources at the Integration Level in a Project

Edit Dependent Resources at the Project Level

1. In the navigation pane, click **Projects**.
2. Click **Actions** , and select **Configure**.
   The Configuration Editor page opens. A wizard is displayed at the top. This wizard guides you through configuration of all potential resources in a project (regardless of whether resources have currently been defined).



Copyright © 2022, 2026, Oracle and/or its affiliates.

3. If you want to edit the connection properties and security properties, hover over the row and click **Edit** .
   a. Edit the connection.
   b. Click **Test**, then **Save**.
      The active and inactive integrations using the connection are displayed.
   c. Click **Save** to save your changes or **Save & Reactivate** if the connection you edited is being used in a currently activated integration.
   d. Click **Back** .
4. Click the **Lookups** icon. A green icon is displayed for the **Connections** icon.
   a. If you want to edit a lookup, click **Edit** .
   b. Edit the lookup.
   c. Click **Save**.
   d. Click **Back** .
5. Click the **Libraries** icon. A green icon is displayed for the **Lookups** icon.
   a. If you want to edit a library, click **Edit** .
   b. Edit the library.
   c. Click **Save**.
   d. Click **Back** .
6. Click the **Integrations** icon. A green icon is displayed for the **Libraries** icon.
   a. Hover over the row of an integration.
   b. If you want to edit the integration, click **Actions** , then **Edit**.
   c. If you want to activate an integration, click **Activate** . You can also deactivate currently activated integrations.
      If all connections are configured, the Activate integration panel is displayed. If you have incomplete connections, you are prompted to first configure the inactive connections and try activation again.
   d. Select a tracing level, then click **Activate**.
7. Click the **Deployments** icon. A green icon is displayed for the **Integrations** icon.
   a. Hover over the row of the project.
   b. If you want to edit the deployment, click **Actions** , then **Edit**.
   c. If you want to activate the project deployment, click **Activate** . If you have not created a deployment, you are prompted to create one.

Edit and Replace Dependent Resources at the Integration Level in a Project

1. In the navigation pane, click **Projects**.
2. Click the project name or click **Edit** .
3. In the **Integrations** section, find the specific integration to configure.

Copyright © 2022, 2026, Oracle and/or its affiliates.

4. Click **Actions** ⋮, and select **Configure**.
   The Configuration Editor page opens. A wizard is displayed at the top. This wizard guides you through configuration of all current resources in an integration. If resources have not been defined, the wizard does not display steps for configuring them.

5. If you want to replace a connection with a different one, click **Replace** .
   Note the following connection replacement guidelines:
   - Only connections of the same role type (trigger, invoke, or trigger and invoke) are displayed for selection. You can only replace a connection resource with another resource of the same role.
   - The connection status must be displayed as **Configured**. You cannot replace a connection with a connection whose status is **Draft**. If a compatible connection resource does not exist, a message is displayed.
   - You cannot replace connections used in integrations with the following state:
     - Active
     - Activation in progress
     - Deactivation in progress
     - Locked
   - If you replace a connection, a **Revert** icon is displayed in the row that enables you to change back to the previous connection.

6. If you want to edit the connection properties and security properties, click **Edit** .
   a. Edit the connection.
   b. Click **Test**, then **Save**.
      The number of active and inactive integrations using the connections are displayed.
      - If there are no active integrations, reactivation of integrations is not available.
      - If you made changes, click **Save & Reactivate** to apply the connection changes to active integrations. Schedule integrations with an active schedule are excluded. Any queued/in-progress schedule integration instances are canceled.
   c. Click **Back** .

   If your integration includes lookups, the **Lookups** icon appears at the top.

7. Click the **Lookups** icon.
   - If the connection selected in the previous step is in a configured state, a green icon is displayed for the **Connections** icon.
   - If the connection selected in the previous step is *not* in a configured state, an error icon is displayed for the **Connections** icon. Hover over the icon to show information about the error.

   a. If you want to edit a lookup, click **Edit** .
   b. Edit the lookup.
   c. Click **Save**.
   d. Click **Back** .

   If your integration includes libraries, the **Libraries** icon appears at the top.

Copyright © 2022, 2026, Oracle and/or its affiliates.

8. Click the **Libraries** icon. A green icon is displayed for the **Lookups** icon.
   a. If you want to edit a library, click **Edit** .
   b. Edit the library.
   c. Click **Save**.
   d. Click **Back** .
9. Click the **Activation** icon. A green icon is displayed for the **Libraries** icon.
   a. Hover over the row of an integration.
   b. If you want to edit a deactivated integration, click **Actions** , then **Edit**.
   c. If you want to activate an integration, click **Activate** . You can also deactivate currently activated integrations.
   d. Select a tracing level, then click **Activate**.

See Edit and Replace Dependent Resources.

Export a Project

You can export a project for import into another instance. This action exports any integration, robot, B2B, process, and healthcare resources. You can export a project to your local host or to a GitHub repository. You can export accelerator projects in simplified mode or non-simplified mode.

Understand Project Export Behavior

Note the following behavior when exporting and importing projects.

- A project export artifact (`.CAR` file) does not include any remote shared connections.
- A local connection's override configuration or association is not exported.
- An imported project does not replace an already existing resource. Therefore, the override configuration on the target system is not lost.
- You cannot export an individual integration within a project. All integrations in the project are exported. Instead, create a project deployment that includes the specific integration that you want to export. When you export the project deployment, you get a `.CAR` file that includes that integration. See Create and Manage a Project Deployment.
- When you export a project, you don't need to remain on this page and wait for the export to complete. You can navigate throughout the user interface and perform other tasks. When the export completes, a message is displayed. However, during the export process, do not click **Refresh** .

See Create or Import a Project.

Export a Project

1. Export a project in either of two ways:
   a. In the navigation pane, click **Projects**.
   b. Click the project name or click .
   c. In the upper right corner, click **Export** .

Copyright © 2022, 2026, Oracle and/or its affiliates.

The Export project panel opens.

Or

a. In the navigation pane, click **Projects**.

b. Hover over the project to export.

c. From the **Actions** ⋮ menu, select **Export**.
The Export project panel opens.

2. Select the location to which to export the project:

- **Export CAR file**: Export the project file to your local host.
- **Export to Repository**: Export the project file to the GitHub repository branch that you configured on the Repository page. See Integrate Projects and Project Deployments with a GitHub Repository.

When you select a location, the **Export as** list shows the type of project to export (for example, accelerator, extended accelerator, or developed). In most cases, this list cannot be edited. If you initially exported an accelerator project as a developed project, imported it into a new instance, and are now exporting it again, this field is enabled for selection.

3. Select a project deployment.

> **Note**
>
> If there are no project deployments available, create one under the **Deploy** tab and select the specific integrations and integration versions to include. See Create and Manage a Project Deployment.

4. If you are exporting an accelerator or extended accelerator, the following fields are displayed. These fields are not displayed if you are exporting a developed project.

a. Select the mode for the project deployment to export:
This toggle is not enabled in many cases. If you initially exported an accelerator project as a developed project, imported it into a new instance, and are now exporting it again as an accelerator, you can enable this toggle.

Accelerator projects in simplified or nonsimplified mode provide you with different extension capabilities.

Copyright © 2022, 2026, Oracle and/or its affiliates.

Simplified Mode	Nonsimplified Mode
The toggle is displayed as follows:  	The toggle is displayed as follows: 
Simplified mode provides the following capabilities: • If the exported integration includes one extensible endpoint (for example, an invoke connection, or a log, reply, or notification action), an extension group with a data stitch action is automatically created when you first edit/extend the exported integration. This data stitch action provides the following features: – View-only mapper – Variable/values filtering – Source tree filtering for values and variables – Design/develop view on the source tree • If the exported integration includes multiple extensible endpoints, a data stitch action is automatically created inside an extension group for the first endpoint in the exported integration when you first edit/extend the exported integration. This data stitch action provides the following features: – View-only mapper – Variable/values filtering – Source tree filtering for values and variables – Design/develop view on the source tree	Nonsimplified mode provides the following capabilities: • If the exported integration includes one extensible endpoint (for example, an invoke connection, or a log, reply, or notification action), an extension group with a data stitch action is automatically created when you first edit/extend the exported integration. • If the exported integration includes multiple extensible endpoints, *no* extension groups are automatically created when you first edit/extend the exported integration.

b. If you want to view configuration details, select **Can view configuration details**.

c. From the **Cloning** field, select how to export the project. Your selection determines how the project can be exported at a later time.

- **As Developed, recipe or accelerator**: Exports the project as a developed, recipe, or accelerator project. If you import this project into another instance and decide to export it again, **Developed**, **Recipe**, or **Accelerator** are all selectable from the **Export as** list. After import, the word **Oracle** is not displayed next to the project name.
- **As accelerator**: Exports the project as an accelerator. If you import this project into another instance and decide to export it again, **Accelerator** is displayed in the **Export as** list and cannot be deselected.
- **Not allowed**: After the accelerator project is exported and imported into another instance, it cannot be cloned.

5. Click **Export**.
If you exported a project or project deployment to a GitHub repository, the following occurs:

Copyright © 2022, 2026, Oracle and/or its affiliates.

If You Export a ...	Then ...
Project	The project CAR file is exported into a `projects` folder under the GitHub repository branch you configured in Integrate Projects and Project Deployments with a GitHub Repository.
Project deployment	The project deployment CAR file is exported into a `deployments` folder under the GitHub repository branch you configured in Integrate Projects and Project Deployments with a GitHub Repository.

Clone a Project

You can clone a project for consumption by other teams that require access to similar integration, robot, B2B, decision, and healthcare resources. When you clone a business accelerator project, you can convert it to another business accelerator project or to a developer project to gain full control over its resources.

Note the following project cloning behavior:

- Project deployments are cloned.
- Runtime data visible under the **Observe** tab is not cloned.
- Role-based access control (RBAC) settings are not cloned.
- All cloned connections are in a draft state with no connection property or security information defined.

1. In the navigation pane, click **Projects**.
2. Hover over the row of the project to clone.
3. Click **Actions** ⋯, then select **Clone**.
   The Clone project panel opens.
4. Enter a name, unique identifier, optional keyword, and optional description.
5. If you are cloning a business accelerator project, the **Clone as** section is displayed.
   a. Select an option based on your business requirements:

This Option...	Enables You To ...
Accelerator	Create a cloned business accelerator project with the following capabilities: • Labeled as an **Accelerator** in the **Type** column. • Retains its business accelerator capabilities. • Can be extended (customized). • Can be updated without losing your extensions when a newer version of the business accelerator is installed.
Developed project	Create a cloned developer project with the following capabilities: • Labeled as a **Developer** in the **Type** column. • Is completely decoupled from the original business accelerator. • Cannot be extended. • Cannot be updated when a newer version of the business accelerator project is installed.

   b. In the **Tag** field, enter a name. This name is appended to the end of the original business accelerator name and identifier.

Copyright © 2022, 2026, Oracle and/or its affiliates.

6. Click **Clone**.
   A message appears indicating your cloning request was submitted.

7. Click **Refresh** .
   Over time, accelerators can be updated and newer versions published to the Integration Store. For cloned business accelerators or extended accelerators, you can install the latest updates directly from the Projects page.

8. Hover over the cloned business accelerator or extended accelerator project, and click **Actions** , then **Get latest** to install any updates.
   Note that the original business accelerator from which you created the cloned accelerator cannot be updated from the **Actions** menu and does not include a **Get latest** option. To update that version, you must instead go directly to the Integration Store.

See Upgrade an Accelerator Project and Merge Extensions.

.

Integrate Projects and Project Deployments with a GitHub Repository

You can configure access to your GitHub repository in Oracle Integration. This access enables you to export (push) and import (pull) projects and project deployments between your Oracle Integration service instance and the GitHub repository.

- Capabilities
- Restrictions
- Prerequisites
- Configure Access to the GitHub Repository
- Export a Project or a Project Deployment to a GitHub Repository
- Import a Project or a Project Deployment from a GitHub Repository

Capabilities

A GitHub repository enables you to store and manage code, files, and file change histories. Integrating your GitHub repository with Oracle Integration enables you to:

- Export (push) a project or project deployment file from Oracle Integration to the GitHub repository.
- Import (pull) a project or project deployment file from the GitHub repository into Oracle Integration.

You can use a GitHub repository for the following scenarios with Oracle Integration:

- Continuous integration/continuous delivery (CI/CD) for integration/project development management. For example, you can export a project to a GitHub repository and run automation testing to ensure that it passes your business requirements.
- Version control for integrations/projects.
- Backup and recovery.
- The movement of projects between Oracle Integration service instances. For example, moving a project from a development environment on one service instance to a production environment on another service instance.

Restrictions

- Integration with a GitHub repository is only available in projects.

Copyright © 2022, 2026, Oracle and/or its affiliates.

- The only repository supported is GitHub (`github.com`).
  - Enterprise GitHub (`github.com/enterprise`) is not supported.
  - VBScript (VBS)-hosted GitHub is not supported.

Prerequisites

Before you can configure access to the GitHub repository, you must perform the following tasks outside of Oracle Integration:

- Create a GitHub repository. See https://github.com/.
- Create a branch in the repository.
- Get the API key of the GitHub repository.
  1. Go to **Profile**, and then **Settings**.
  2. In the left side menu on the Settings page, go to **Developer Settings**.
  3. On the Developer Settings page, click **Personal access tokens** and then **Fine-grained tokens**.
  4. On the Fine-grained personal access tokens page, create a token.
     a. Specify the GitHub owner name, expiration time for the token, and repository access for the user.
     b. Select the repositories in which the token must be generated and provide access to the user's token in the repository permissions.
     c. Select repository permission **Contents** as **read** and **write**.
     d. Create the token and provide a token name.
     e. Copy and save the token for use in Configure Access to the GitHub Repository.

Configure Access to the GitHub Repository

You must configure access to your GitHub repository on any Oracle Integration service instance between which you plan to export and import projects and project deployments. Only a user with the ServiceAdministrator role can configure access to the GitHub repository.

1. In the navigation pane, click **Settings**, then **Repository**.
2. Click **Edit**.
3. Enter the following details, then click **Test**.

Element	Description
User name	Enter the user name for accessing your GitHub repository. **Note**: You cannot enter an email account as the user name (for example, `joesmith@mycompany.com`). The period (`.`) and at sign (`@`) characters cause the following error: `Invalid branch name. Please enter a valid branch name`
Repository name	Enter the name of the GitHub repository.

Copyright © 2022, 2026, Oracle and/or its affiliates.

Element	Description
Branch name	Enter the branch name in the GitHub repository in which to commit the project or project deployment file.
Token ID	Enter the developer token ID you created in Prerequisites.

4. If the connection test is successful, click **Save**.
The GitHub repository is now configured for exporting and importing projects and project deployments. Otherwise, correct any connection errors and test again.

Export a Project or a Project Deployment to a GitHub Repository

You export a project or a project deployment to the GitHub repository branch you specified during GitHub repository configuration. See Export a Project.

Import a Project or a Project Deployment from a GitHub Repository

You export a import or a project deployment from the GitHub repository branch to your Oracle Integration service instance. See Create or Import a Project.

Invoke Child Integrations Inside or Outside of Projects

You can configure a parent integration to invoke a child integration in the same project, a child integration in a different project, or a child integration that is not part of any project (known as an integration that is globally available).

- Capabilities
- Restrictions
- Set a Project Integration as Public
- Prerequisites to Dynamically Invoke a Child Integration from a Parent Integration After Import into a Project

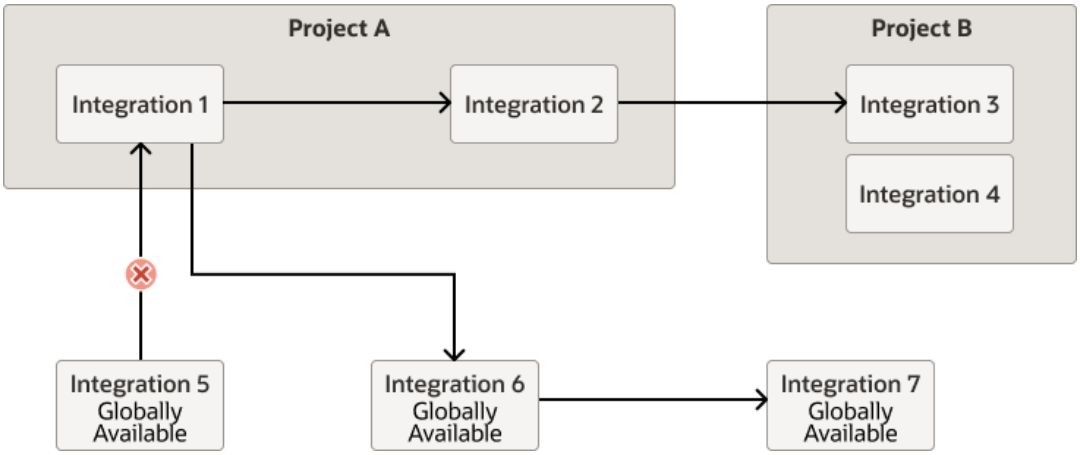
Capabilities

- A parent integration in a project can be configured to invoke a child integration located:
  - In the same project.
  - In a different project, if that integration is set as public. See Set a Project Integration as Public.
  - Globally available (that is, the integration is not included in any project).

  During runtime, all integration behavior is consolidated into a single activity stream for viewing, regardless of the location of all the integrations.

  As an example, the following parent-to-child invokes are supported:
  - Parent Integration 1 in Project A invokes child integration 6 that is globally available (integration is not in any project). Child integration 6 then invokes child integration 7, which is also globally available.
  - Parent Integration 2 in Project A invokes child integration 3 in Project B.
  - Parent Integration 5, which is globally available, cannot invoke integration 1 in Project A because this type of invocation is not allowed.

Copyright © 2022, 2026, Oracle and/or its affiliates.



- A parent integration in a project can dynamically invoke either of two child integrations at runtime, regardless of location. For example, both child integrations can be in the globally available or one child integration can be globally available and the other child integration can be in another project. See Dynamically Invoke a Child Integration.

Restrictions

- Only REST, SOAP, and schedule integration endpoints are supported.
- Only activated integrations are available when selecting a child integration in the Integration Adapter Wizard. See Invoke a Child Integration from a Parent Integration.
- A parent integration that is globally available (outside a project) *cannot* invoke a child integration inside any project. Parent integrations that are globally available can only invoke child integrations that are also globally available.
- If you select to activate or deactivate a project deployment, all child integrations outside that project (whether in another project or globally available) must already be activated or deactivated, respectively. The project deployment cannot activate or deactivate integrations outside that project. See Create and Manage a Project Deployment.
- When exporting a project deployment, none of the integrations from outside that project are included in the exported file. You *must* separately export those integrations and import them along with the referencing projects into the target instance.
- References to an integration in another project may specify an integration that itself has external references. A chain of references can span multiple projects.

Set a Project Integration as Public

Setting a project integration as public enables that integration to be invoked by a parent integration in a different project. If you don't set a project integration as public, it can only be invoked by a parent integration in the same project.

Set a project integration as public in either of two ways:

Create a new integration:

1. In the **Integrations** section, click **Add** if no integrations currently exist or **+** if integrations already exist.
2. Select the type of integration to create (application or schedule).
3. Select the **Available to other projects** check box.

Update an existing integration:

Copyright © 2022, 2026, Oracle and/or its affiliates.

1. Go to the integration canvas for the integration to make public.
2. Click **Primary Info** .
3. Select the **Available to other projects** check box.

When selected, the **Available to other projects** check box enables an integration in a different project to appear for selection in the Integration Adapter Wizard that is invoked when you drag the **Integration** action into the integration canvas. See Invoke a Child Integration from a Parent Integration.

Prerequisites to Dynamically Invoke a Child Integration from a Parent Integration After Import into a Project

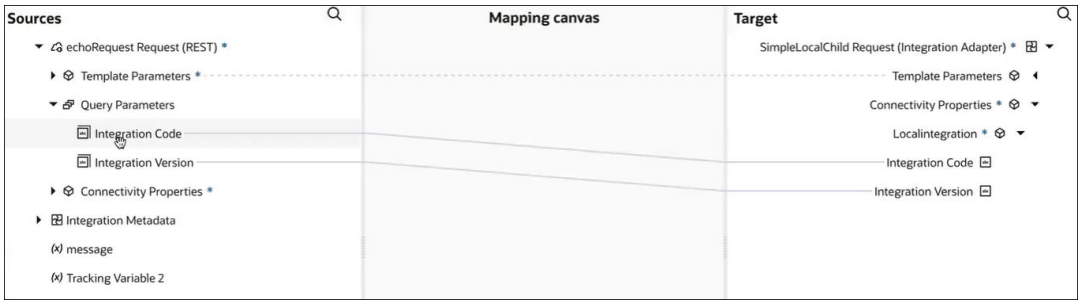
To dynamically invoke a child integration from a parent integration in a project, you need the project ID, integration code, and integration version. This differs from globally-available integrations (integrations not included in any project), where you can dynamically invoke a child integration from a parent integration with only the integration code and integration version.

If you import a globally-available parent integration and its child integrations into a project and then attempt to dynamically call the second child integration from the parent integration, the child integration is not called from the project. Instead, the globally-available child integration is called. This occurs because only the integration code and integration version are available to configure in the mapper, and not the project code.

You must create and map a project code query parameter in the mapper for dynamic invocation of the second integration in a project to succeed.

Assume you perform the following steps:

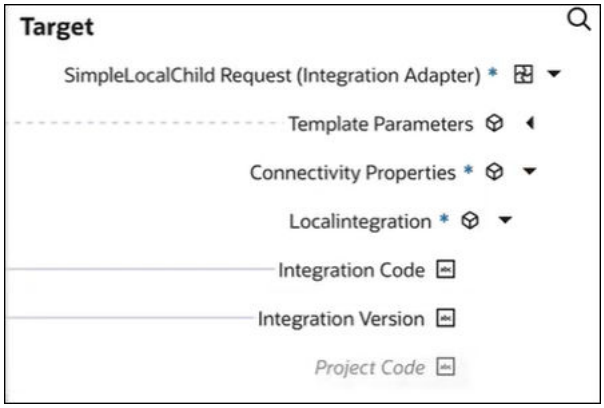
1. Configure a parent integration to invoke two child integrations. All three integrations are globally available, meaning they are not part of a specific project.
2. Export the three integrations and import them into a project.
3. Edit the mapper of the imported parent integration.
   Note that there is no value for the target **Project Code** node under **Localintegration** to configure.



Solution: Perform the following steps:

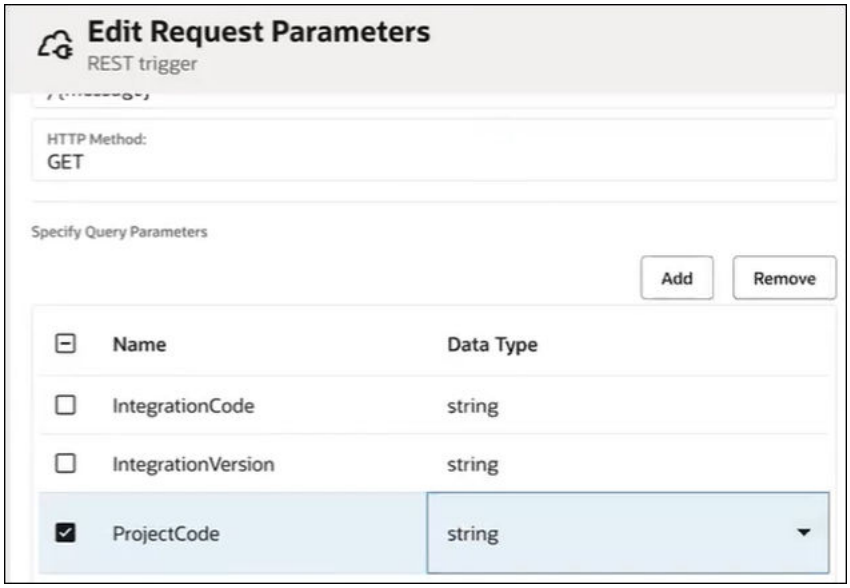
1. In the integration canvas, select to edit the integration action.
2. Click **Continue** on the pages of the Integration Adapter Wizard without making changes, then click **Done** on the final page. This action creates the project code entry in the mapper.
3. Open the mapper and note that the target **Project Code** node for the project under **Localintegration** is now visible.

Copyright © 2022, 2026, Oracle and/or its affiliates.



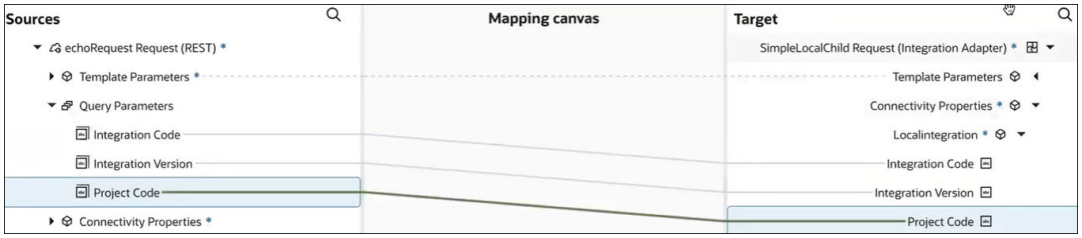
You must now create a target query parameter and map it to **Project Code**.

4. Exit the mapper.
5. Open the REST Adapter trigger connection for editing in the Adapter Endpoint Configuration Wizard.
6. On the Request Parameters page, create a new query parameter. For this example, `ProjectCode` is created.

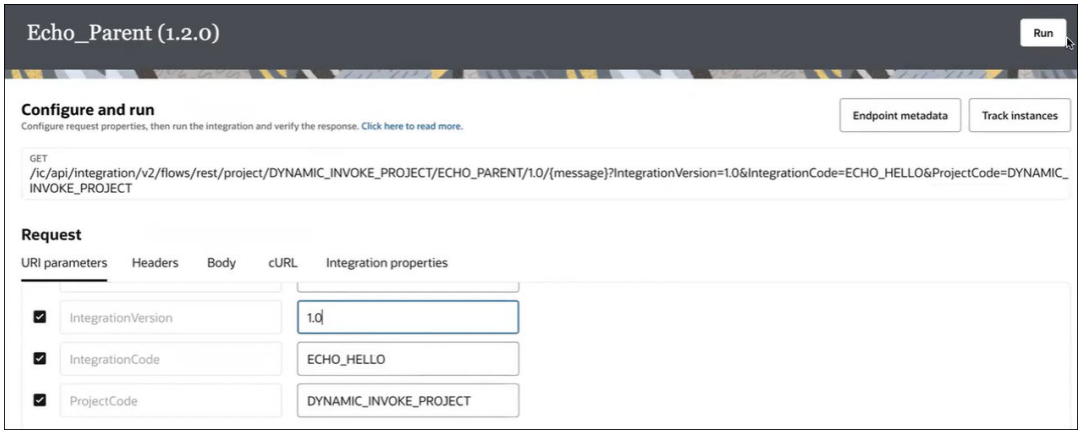


7. Click through the remaining pages of the wizard, then click **Done**.
8. Return to the mapper.
   The new source query parameter is now visible.
9. Drag the source **Project Code** query parameter to the target **Project Code**.

Copyright © 2022, 2026, Oracle and/or its affiliates.



When you later run the parent integration from the Configure and run page, the integration version, integration code, and project code parameters are available.



10. Click **Run**. The child integration in the project is now invoked, instead of the globally-available version.

Information about dynamically invoking a child integration is provided. See Dynamically Invoke a Child Integration.

Manage Accelerator Projects

You can install accelerator projects from the Integration Store. You can extend (customize) integrations in the accelerator project and later upgrade an accelerator project to a newer version automatically without losing any extensions you made to the previous version.

Topics:

- Extend an Integration in an Accelerator Project
- Upgrade an Accelerator Project and Merge Extensions
- Behavior When You Activate an Updated Accelerator Project Deployment with Newer Integration Versions

Watch a video to learn more:

Video

Extend an Integration in an Accelerator Project

You can extend (customize) an integration in an accelerator project by adding and configuring an extension group. An extension group enables you to extend your integration by adding invoke connections; data stitch, for-each, switch, map, robots, and integration actions; and

Copyright © 2022, 2026, Oracle and/or its affiliates.

global variables to the integrations in your accelerator project. When Oracle uploads newer versions of accelerator projects to the Integration Store, you can upgrade an accelerator project to this newer version automatically without losing your extensions.

- Learn About Accelerators Available in the Integration Store
- Extend an Integration Before or After an Invoke Connection or a Robotic Process Automation Action
- Add a Global Variable to an Extended Integration
- Add an Extension Group Before a Stage File Action Configured with a Write File Operation
- Add Connections to an Extended Integration
- Add Lookups and JavaScript Libraries to an Extended Integration
- Extend an Integration Before or After a Log or Notification Action
- Add an Extension Group Before or After a Parent Integration that Invokes a Child integration
- Merge Extension Group Changes into Updated Accelerator Projects

Learn About Accelerators Available in the Integration Store

The Integration Store provides accelerator projects that you can install and extend in a project. Newer, updated versions of accelerator projects can also become available in the Integration Store for you to re-install, as necessary.

See Find Recipes and Accelerators and Install a Recipe or Accelerator in *Using Recipes and Accelerators in Oracle Integration 3*.

Extend an Integration Before or After an Invoke Connection or a Robotic Process Automation Action

You can extend an integration to add the following before or after an invoke connection or a robotic process automation action:

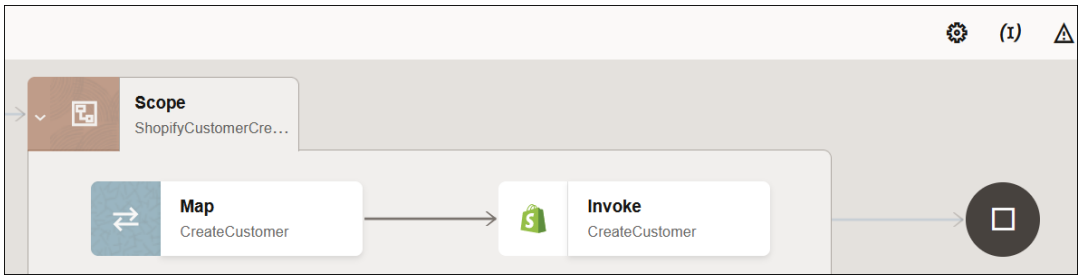
- Data stitch action
- For-each action
- Switch action
- Map action
- Integration action (and associated map) for invoking a child integration
- Invoke connection (and associated map)
- Robot flow action (and associated map)

1. In the navigation pane, click **Projects**.
2. Find the accelerator project to extend. The **Accelerator** and **Oracle** labels identify these types of projects. You can also use the **Filter** to display a specific project type (all, user-developed, accelerator, extended accelerator, or recipe). For example:



Copyright © 2022, 2026, Oracle and/or its affiliates.

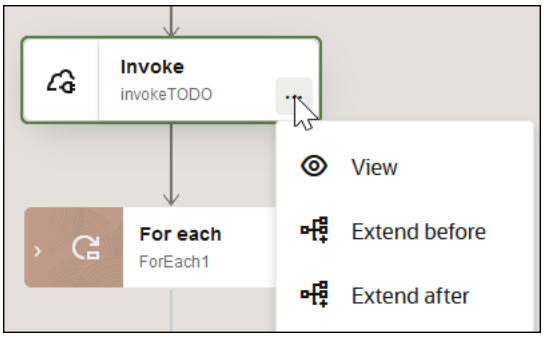
3. Click the project name or click .
4. In the **Integrations** section, find the specific integration to extend.
5. Click **Actions** , and select **Extend**. The **Extend** option is only available in accelerator projects. This option behaves similar to the **Edit** option in non-accelerator integrations. The integration is displayed in the canvas.



6. Perform one of the following actions before or after an invoke action or a robotic process automation action.
   a. Click **Add** at the location where you want to add the extension, then select **Extension Group**.

   or

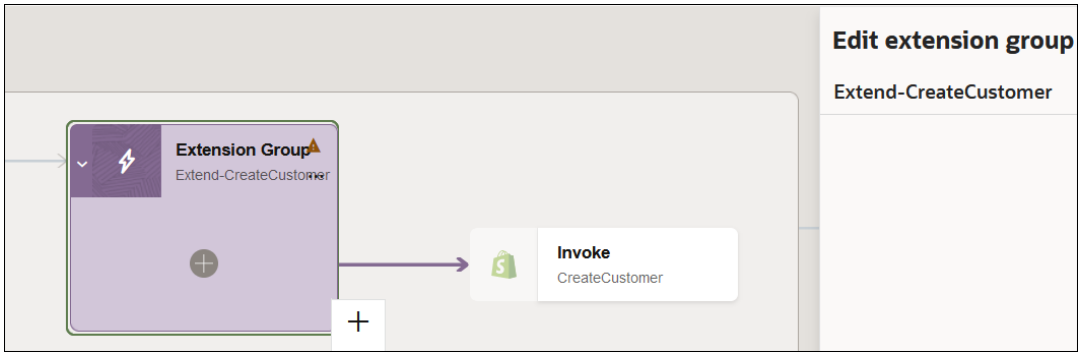
   a. Click **Actions** inside the invoke action or the robotic process automation action.
   b. Based on where you want to place the extension, select **Extend before** or **Extend after**. Both options are available if you have not yet added an extension to either side of the invoke action or the robotic process automation action.



> **Note**
>
> If **Extend before** and **Extend after** are not available for selection and you have not added an extension to either side of the invoke, your administrator has restricted the use of extensions on the invoke connection. You cannot extend restricted invokes.

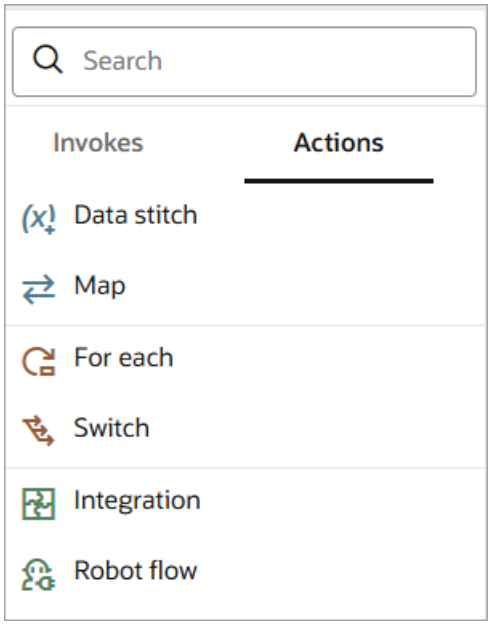
The extension group is added to the integration.

Copyright © 2022, 2026, Oracle and/or its affiliates.



7. If you want to update the name, click **Edit** in **Edit extension group**.

8. Click **Integration actions** and drag an invoke or action into the integration or click **Add** inside the extension group to add an invoke or action to the integration. For example, if you click **Add**, the following menu is displayed.



See the following sections for details:

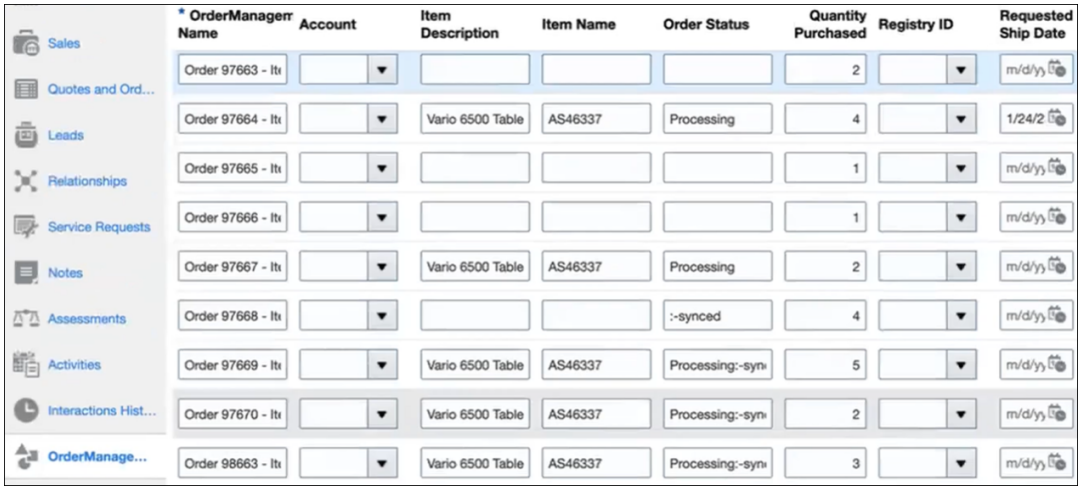
- Add a Data Stitch Action to an Extended Integration
- Add an Ad-Hoc Map Action to an Extended Integration
- Add a For-Each Action to an Extended Integration
- Add a Switch Action to an Extended Integration
- Add an Integration Action to an Extended Integration
- Add a Robot Flow Action to an Extended Integration
- Add an Invoke Connection and Associated Map to an Extended Integration

Copyright © 2022, 2026, Oracle and/or its affiliates.

Add a Data Stitch Action to an Extended Integration

You can add a data stitch action to an extended integration. This section provides a use case.

1. If you select **Actions**, then **Data stitch**:
   You can incrementally build a message payload from one or more existing payloads with a data stitch action. For example, a data stitch enables you to customize the sales order created in an inbound application. Assume you have a use case in which whenever a sales order is created in an inbound application, your integration must retrieve and push information in a custom object to an outbound application. In the inbound application, the custom object consists of the following attributes.



   The Configure stitch panel opens.

   a. Configure the data stitch. This action enables you to customize the integration with mappings to or from attributes of custom objects. For this example, a data stitch variable is created to process the order status attribute in the custom object.



   See Build Complex Assignment Statements with a Data Stitch Action.

   b. When integration design is complete, click **Save**.
   Assume you later need to further customize the custom objects in the inbound application that triggers your integration. For example, an additional field to handle the **Item Number** is added to the custom object.

   This use case is not about installing a newer version of the accelerator project available from the Integration Store. Rather, this is an informal user update to the

Copyright © 2022, 2026, Oracle and/or its affiliates.

custom objects. You can handle this use case by refreshing the endpoints to pick up the new object and editing the data stitch action to process the new object. You must use a new identifier and/or version. You can also set the name, keywords, and description.



2. Click **Actions** ..., then select **Refresh Endpoints**. This action fetches the latest attributes added to the custom object.
3. Open the data stitch action in the extension group and add the latest custom attribute mapping to the extension group. For this example, a second stitch variable is created to process the item number attribute in the custom object.

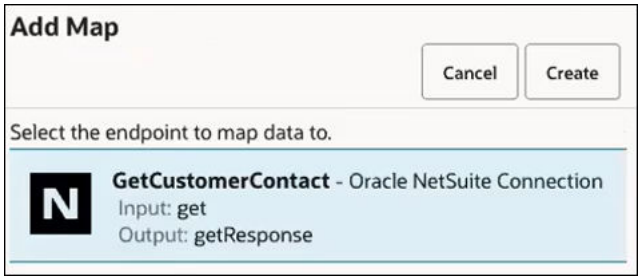


Add an Ad-Hoc Map Action to an Extended Integration

You can add an ad-hoc map to an extended integration.

1. If you select **Actions**, then **Map**:
   The Add Map panel opens to show the endpoints available to which to map. Endpoints both inside and outside the extension group are shown.

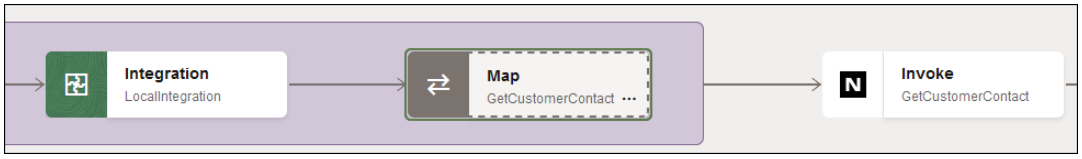
Copyright © 2022, 2026, Oracle and/or its affiliates.



If there are no endpoints available to which to map, the following message is displayed:

```
No outputs available to map to
```

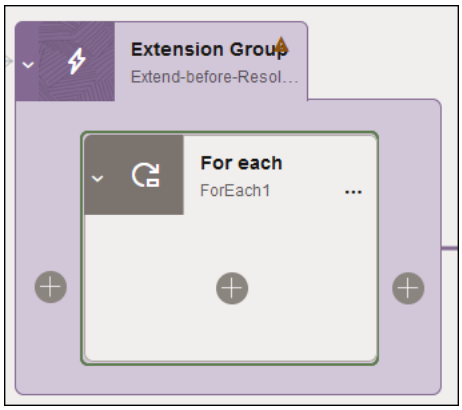
2. Select the endpoint, and click **Create**.
   The mapper opens.
3. Map appropriate source elements to target elements.
   A map action is visible in the **Extension Group**.



Add a For-Each Action to an Extended Integration

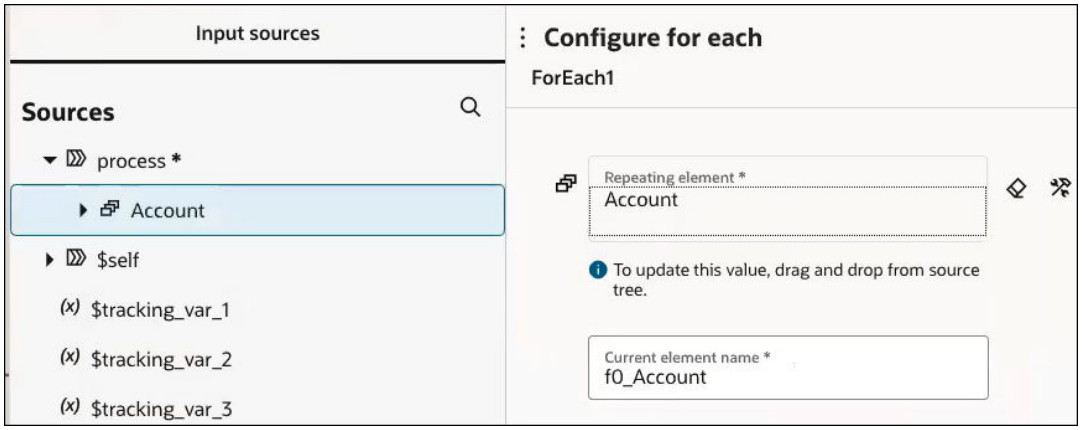
You can add a for-each action to an extended integration.

1. If you select **Actions**, then **For each**:
   A for-each action is added to the **Extension Group**.

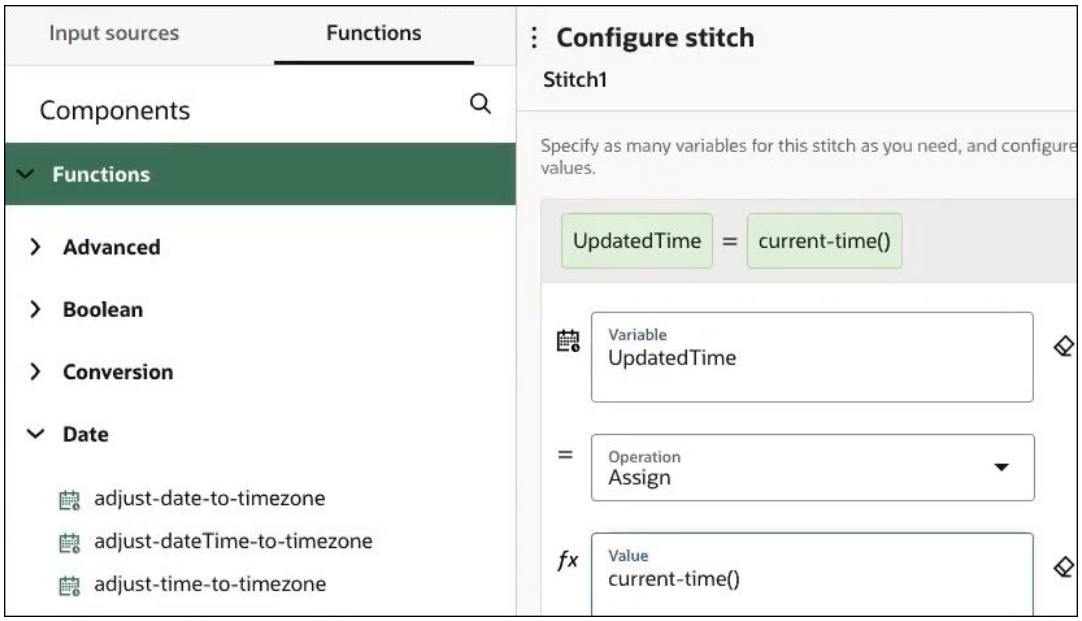


2. Configure the for-each action. For this example, an **Account** element is defined as the repeating action.

Copyright © 2022, 2026, Oracle and/or its affiliates.



3. Perform further configuration, as necessary. For this example, a data stitch action is added. For each iteration of the **Account** element, a value is updated to the current time.



The extension group design looks as follows:



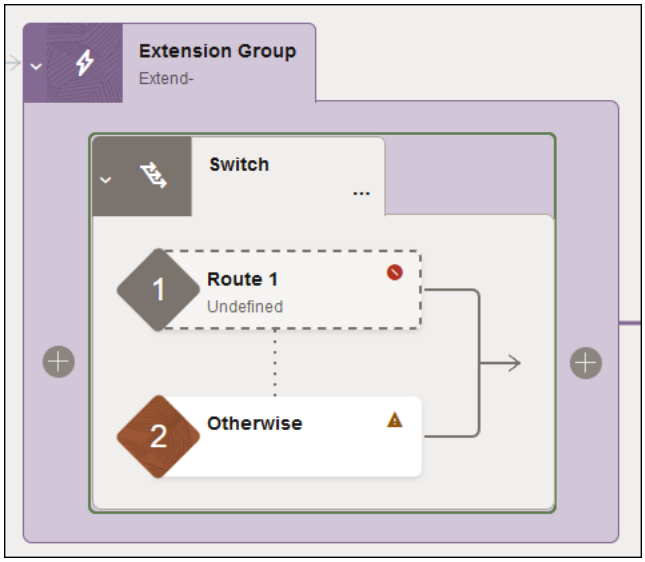
Copyright © 2022, 2026, Oracle and/or its affiliates.

Add a Switch Action to an Extended Integration

You can add a switch action to an extended integration.

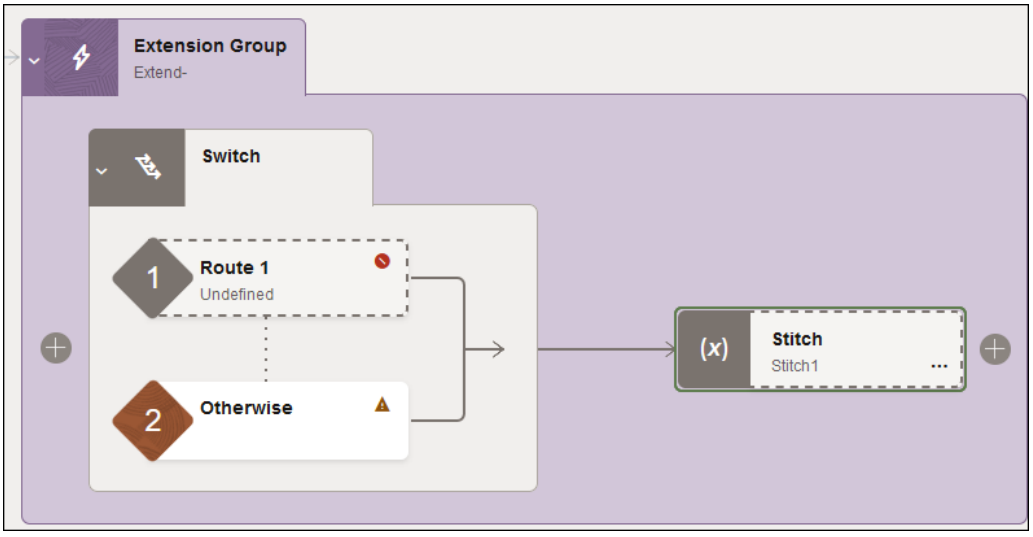
1. If you select **Actions**, then **Switch**:
   A switch action is added to the **Extension Group**.

   You can add routing expressions in an integration with a switch action.



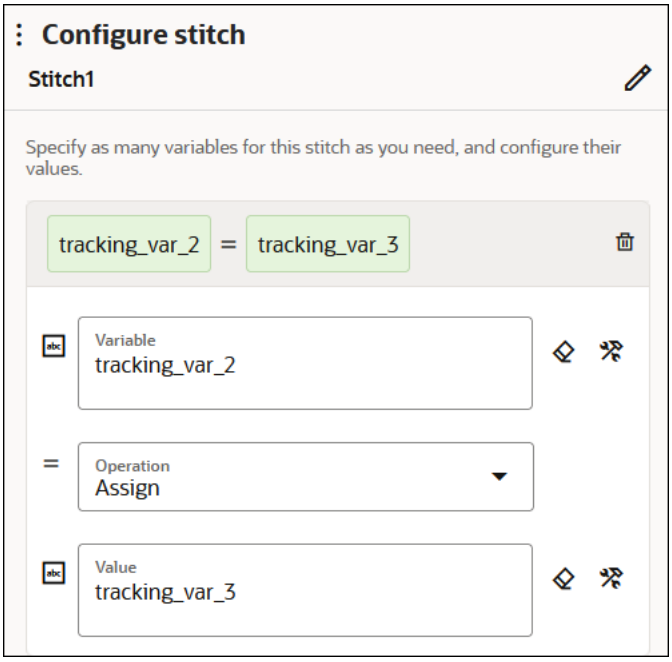
   You can add a data stitch action inside the switch action.

   a. Click +, then select **Stitch**.

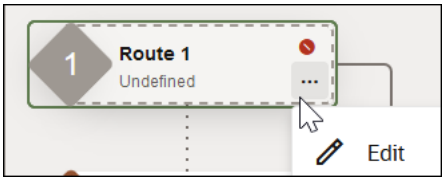


   b. Configure the data stitch action. For example:

Copyright © 2022, 2026, Oracle and/or its affiliates.

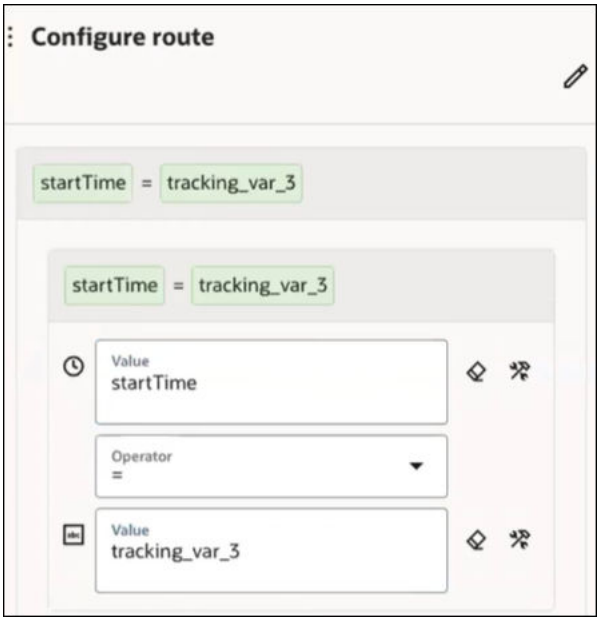


c. Inside **Route 1**, click **Actions** ..., then select **Edit**.



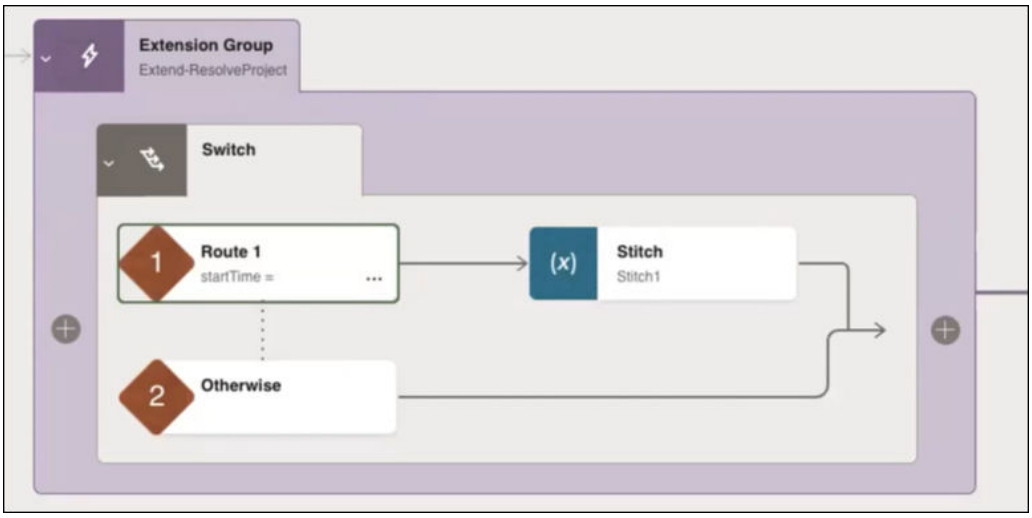
The Configure route panel opens.

d. Configure the switch action. For example:



Copyright © 2022, 2026, Oracle and/or its affiliates.

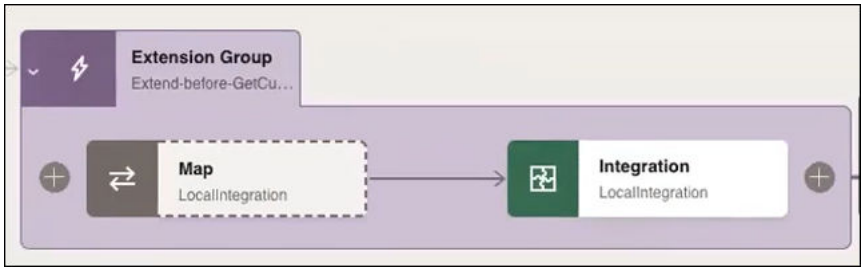
The extended portion of the integration looks as follows.



Add an Integration Action to an Extended Integration

You can add an integration action to an extended integration.

1. If you select **Actions**, then **Integration**:
   The Integration Adapter Wizard is displayed.
2. Click through the pages of the wizard to select the child integration to invoke.
   An integration action and associated map action are added to the **Extension Group**.



Add a Robot Flow Action to an Extended Integration

You can add a robot flow action to an extended integration.

1. If you select **Robot flow**, then configure the robot's basic information:
   - **What do you want to call your endpoint?:** Enter descriptive text for the robot, such as the robot's name or information about its business process.
   - **What does this endpoint do?:** Provide a brief description of the robot.
2. Select **Continue**.
3. From the **Flow Name** drop-down, select the robot that the integration calls, including the appropriate version.

   The drop-down lists all of the robots that are in the same project as the integration.

Copyright © 2022, 2026, Oracle and/or its affiliates.

4. Select **Continue**.
5. Review the summary, and select **Finish**.

   A Map object appears before the Robot process automation action.
6. Pass information from the integration to the robot.
   a. Point to the **Map** action that is before the **Robot process automation** action, select ⋯, and select **Edit**.

      The mapper opens.
   b. In the Sources list on the left, expand the tree until you find the data element that you need to pass to the robot.
   c. In the Target list on the right, expand the tree until you find the input that needs to receive the data.
   d. Drag the source element to the target element.

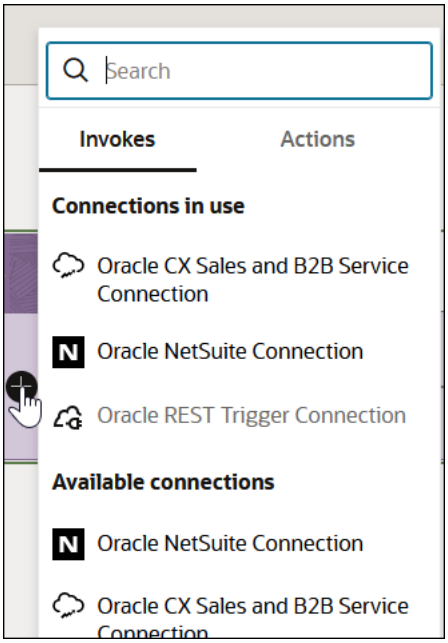
      For more help working in the mapper, see About Mapping Data Between Applications and Map Data in *Using the Oracle Mapper with Oracle Integration 3*.
   e. If you need to pass more information to the robot, map additional elements.
   f. After you finish mapping elements, test your mappings.

      See Test Your Mappings in *Using the Oracle Mapper with Oracle Integration 3*.
   g. Select **Go back <** to return to the canvas.

Add an Invoke Connection and Associated Map to an Extended Integration

You can add an invoke connection and associated map to an extended integration.

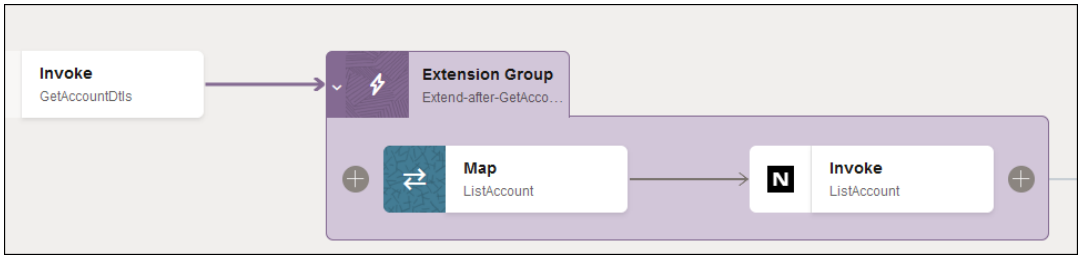
1. If you select **Invokes**, then select an invoke adapter connection:



   The Adapter Endpoint Configuration Wizard is displayed for the selected invoke connection.
2. Click through the pages of the wizard and configure the invoke connection.

Copyright © 2022, 2026, Oracle and/or its affiliates.

3. When complete, the configured invoke connection and an associated map action are added to the **Extension Group**.

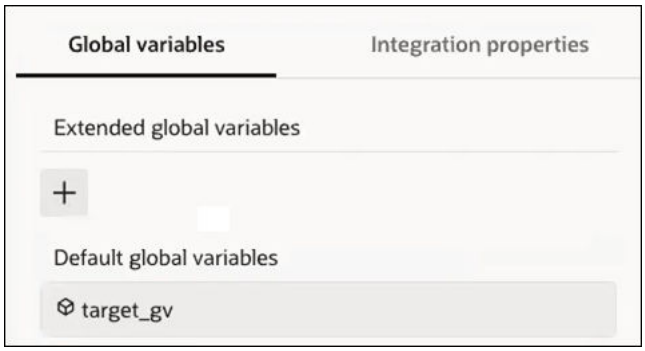


4. Double-click the map action and map appropriate source elements to target elements.

Add a Global Variable to an Extended Integration

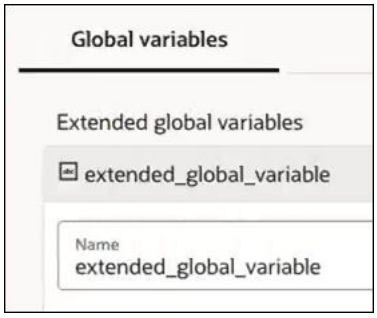
You can add a global variable to an extended integration.

1. In the **Integrations** section, find the specific integration to extend.
2. Click **Actions** ..., and select **Extend**.
   The integration is displayed in the canvas.
3. In the right pane, click **Global Variables** (x).
   There are two types of global variables:
   - **Extended global variables**: Variables that you can create to extend the accelerator for your business requirements.
   - **Default global variables**: Variables that are defined by the producer of the extended accelerator. You cannot edit these variables.

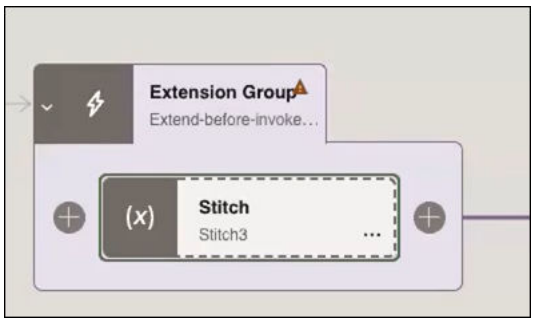


4. Click + to add your own extended global variable. For this example, `extended_global_variable` is added.

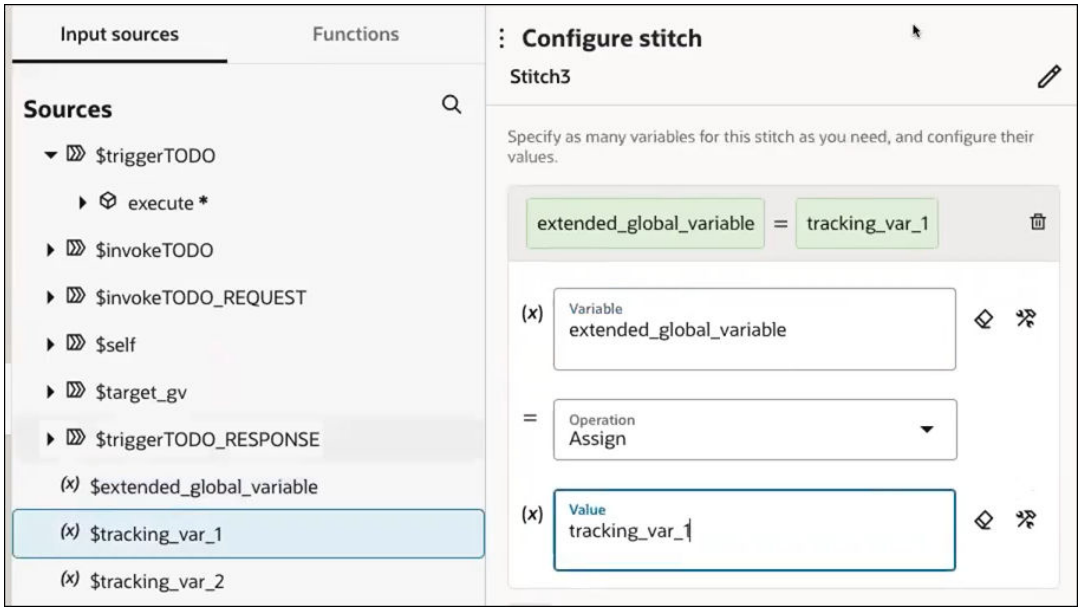
Copyright © 2022, 2026, Oracle and/or its affiliates.



5. Add an **Extension Group** to the integration.
6. Add an action in which to use the extended global variable. For this example, a data stitch action is added.



7. Configure the data stitch action to use the extended global variable.



Copyright © 2022, 2026, Oracle and/or its affiliates.

> **Note**
>
> If there is a naming conflict with global variables (for example, two global variables have the same name), the entire merge is canceled. This cancellation occurs because a global variable can be used in any extension group. For example, if you create an extended global variable named **conflict_gv**, and a default global variable of the same name already exists, the merge is canceled, and an error is displayed.



Add an Extension Group Before a Stage File Action Configured with a Write File Operation

You can add an extension group before a stage file action configured with a write file operation.

You can add an extension group before a stage file action that is configured to write to a file. This is the only stage file action operation to which you can add an extension. This operation includes a preceding map action. For example, you may want to extend the contents of the map action with a data stitch action.

1. Perform one of the following actions before the stage file action.
   a. Click **Add** at the location where you want to add the extension, then select **Extension Group**.

   or

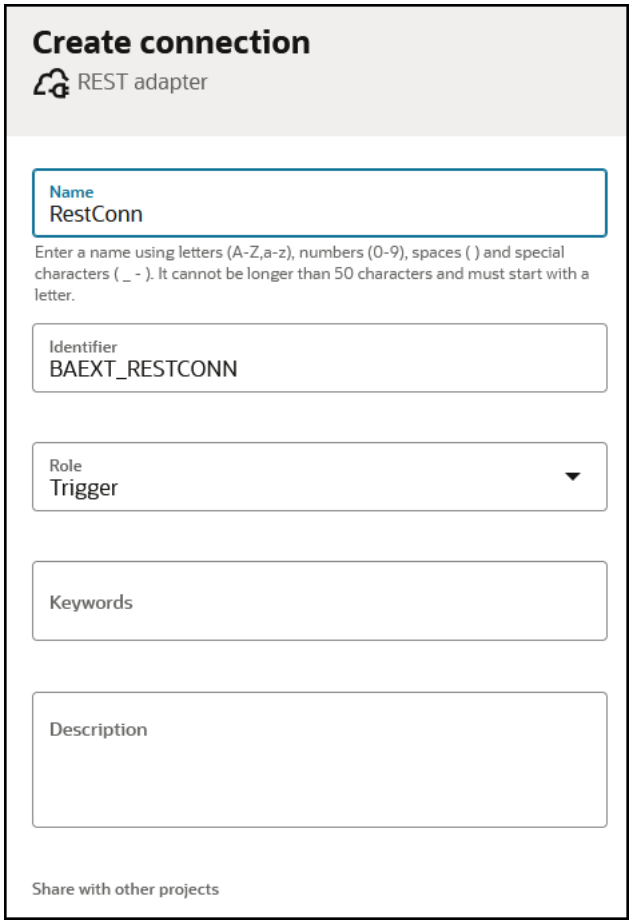
   a. Click **Actions** inside the stage file action and select **Extend before**.
2. Add the appropriate action.

Add Connections to an Extended Integration

You can add connections to an extended integration.

1. In the navigation pane, click **Projects**.
2. Find the accelerator project to extend.
3. In the **Connections** section, click **Add** .
4. Select an adapter and click **Create**.
5. Enter a name. Note that the prefix `BAEXT_` is automatically added to the name in the **Identifier** field to indicate that this is an extended connection.

Copyright © 2022, 2026, Oracle and/or its affiliates.



6. Click **Create**, then define your connection properties and security.
   Once complete, the extended connection appears in the **Connections** section, and can be used in integrations.

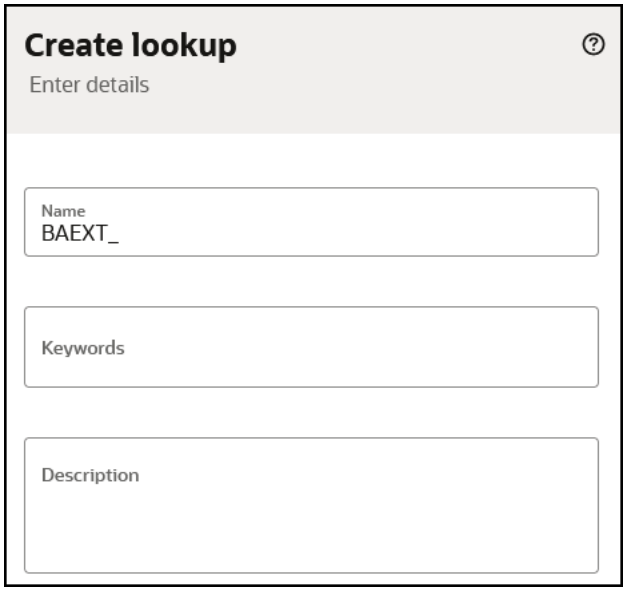
Add Lookups and JavaScript Libraries to an Extended Integration

You can add lookups and Javascript libraries to an extended integration.

To add a lookup to an extended integration.

1. In the navigation pane, click **Projects**.
2. Find the accelerator project to extend.
3. In the **Lookups** section, click **Add** +, then click **Create**.
   Note that the prefix `BAEXT_` appears by default in the **Name** field, An extended lookup name must be prefixed with this value. If you delete the `BAEXT_` prefix and attempt to create the lookup, you receive a warning.

Copyright © 2022, 2026, Oracle and/or its affiliates.

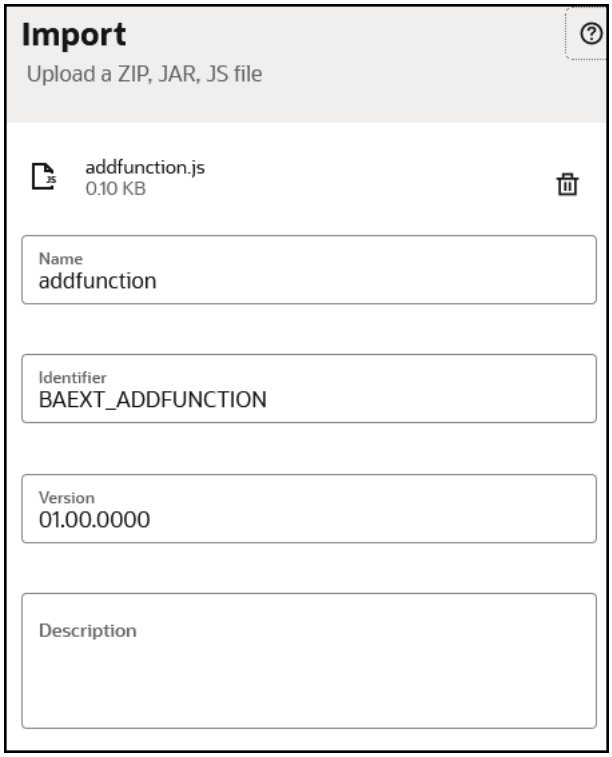


4. Add a name after the prefix `BAEXT_`.
5. Enter or select an existing keyword, and an optional description.
6. Click **Create**.
7. Design your lookup.
   Once complete, the extended lookup appears in the **Lookups** section, and can be used in integrations.

To add a JavaScript library to an extended integration.

1. In the navigation pane, click **Projects**.
2. Find the accelerator project to extend.
3. In the **Libraries** section, click **Add** +.
4. Drag a JavaScript library file into the Import panel.
   The **Name**, **Identifier**, and **Version** fields are prepopulated with details from the file you imported. Note that the prefix `BAEXT_` is automatically added to the function name in the **Identifier** field to indicate that this is an extended JavaScript library.

Copyright © 2022, 2026, Oracle and/or its affiliates.



5. Click **Import**.
6. Complete design of your JavaScript library.
   Once complete, the extended JavaScript library appears in the **Libraries** section, and can be used in integrations.

Extend an Integration Before or After a Log or Notification Action

You can extend an integration before or after a log or notification action. The log or notification action can either be inside of a fault handler or outside of a fault handler.

1. Perform one of the following actions before or after a log or notification action.

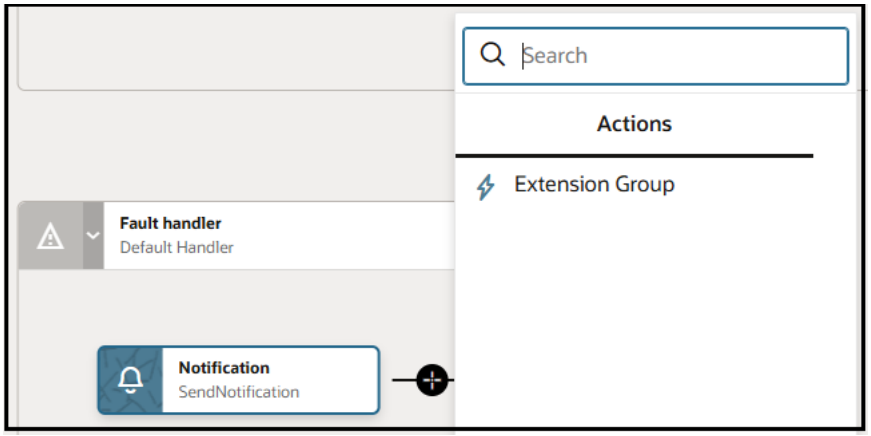
   a. Click **Add** at the location where you want to add the extension, then select **Extension Group**.

   or

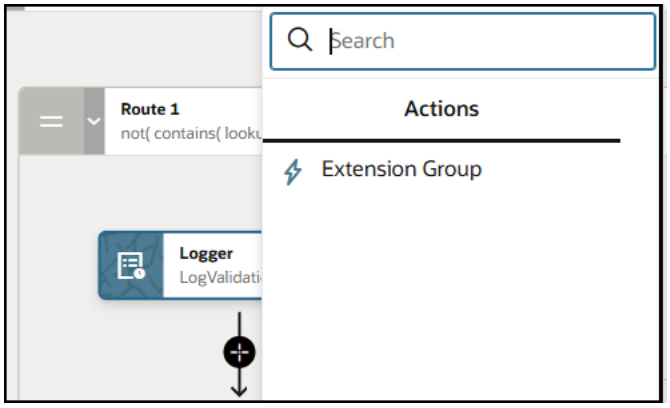
   a. Click **Actions** inside the action and select **Extend before** or **Extend before**.

   For example, to add an extension group after a notification action inside a fault handler:

Copyright © 2022, 2026, Oracle and/or its affiliates.



Or before a log action outside a fault handler:



An extension group is added to the integration.

2. Add the appropriate action in the extension group:
   - Add a Data Stitch Action to an Extended Integration
   - Add an Ad-Hoc Map Action to an Extended Integration
   - Add a For-Each Action to an Extended Integration
   - Add a Switch Action to an Extended Integration
   - Add an Integration Action to an Extended Integration
   - Add a Robot Flow Action to an Extended Integration
   - Add an Invoke Connection and Associated Map to an Extended Integration

Add an Extension Group Before or After a Parent Integration that Invokes a Child integration

You can add an extension group before or after a parent integration that invokes a child integration.

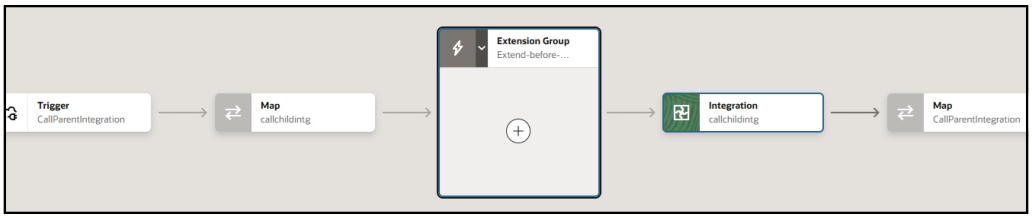
1. In the **Integrations** section, find the specific parent integration to extend.
2. Click **Actions** ⋮, and select **Extend**. The **Extend** option is only available in accelerator projects. This option behaves similar to the **Edit** option in non-accelerator integrations.
3. Perform one of the following actions before or after the integration action.

Copyright © 2022, 2026, Oracle and/or its affiliates.

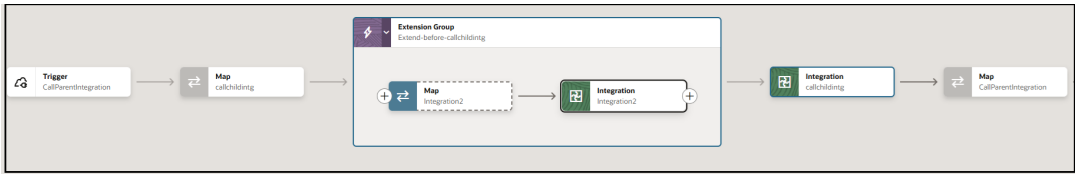
a. Click **Add** before or after the integration action, then select **Extension Group**.

or

a. Click **Actions** inside the integration action and select **Extend before** or **Extend after**.



4. Click and select the action to extend the integration.
For this example, an active integration action is selected and configured in the extension group in front of the integration action.



See the following sections for the actions and invoke connections you can add:

- Add a Data Stitch Action to an Extended Integration
- Add an Ad-Hoc Map Action to an Extended Integration
- Add a For-Each Action to an Extended Integration
- Add a Switch Action to an Extended Integration
- Add an Integration Action to an Extended Integration
- Add a Robot Flow Action to an Extended Integration
- Add an Invoke Connection and Associated Map to an Extended Integration

Merge Extension Group Changes into Updated Accelerator Projects

Oracle can periodically update and upload newer versions of accelerator projects to the Integration Store for your consumption. You can upgrade an accelerator project to this newer version automatically without making manual changes to the existing installation.

During upgrade, you are prompted to automatically merge any customizations you performed in the extension group of the previous version into the newer version of the accelerator project. See Upgrade an Accelerator Project and Merge Extensions.

on an invoke connection also includes options for **Extend before** or **Extend after**. This enables you to install the supported accelerator actions before or after the invoke connection.

Upgrade an Accelerator Project and Merge Extensions

You can install and then extend (customize) accelerator projects in your instance. Oracle can periodically update and upload newer versions of these accelerator projects to the Integration

Copyright © 2022, 2026, Oracle and/or its affiliates.

Store for your consumption. You can upgrade an accelerator project to this newer version automatically without losing any extensions you made to the existing installation.

- Understand the Rules for Merging Extensions
- Automatically Merge Extensions
- Manually Merge Extensions
- Create and Extend a Cloned Business Accelerator Project

Understand the Rules for Merging Extensions

Before upgrading an extended accelerator project, it is important to understand the rules for merging extensions.

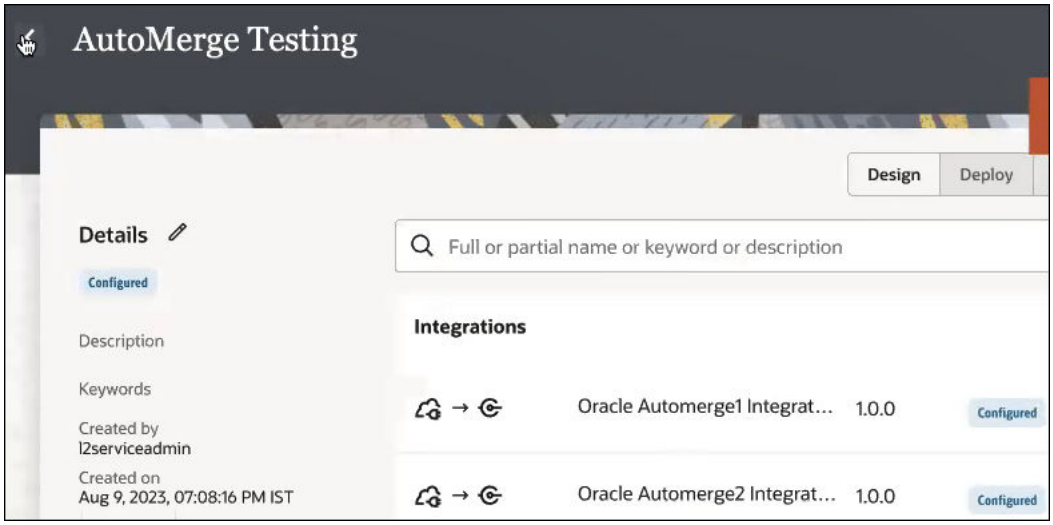
Rule	Example
When you have multiple extended (customized) versions of an accelerator project installed, and select to install a newer version, the installer checks for extensions in the previously installed highest version. If that version includes extensions, they are automatically merged into the newer version.	You currently have the following accelerator project versions installed: • Versions 2.0.0 (which includes extensions) • Versions 4.0.0 (which includes extensions) If you install version 6.0.0, the extensions made in version 4.0.0 (previously installed highest version) are automatically merged into version 6.0.0. The extensions in version 2.0.0 are ignored.
When you have multiple versions of an accelerator project installed, and select to install a newer version, the installer checks for extensions in the previously installed highest version. If that version does *not* include extensions, nothing is merged into the newer version.	You currently have the following accelerator project versions installed: • Version 2.0.0 (which includes extensions) • Version 4.0.0 (which does *not* include extensions) If you install version 6.0.0, the installer checks for extensions in the previously installed highest version. In this case, no extensions are applied because version 4.0.0 included no extensions. The extensions in version 2.0.0 are ignored.
When you have a single version of an accelerator project installed, and select to install a newer version, the installer checks for extensions in the previously installed highest version. If that version includes extensions, they are automatically merged into the newest version.	You currently have the following accelerator project version installed: • Version 1.0.0 (which includes extensions) If you install a higher version of the accelerator project (for example, version 2.0.0, 3.0.0, 4.0.0, or higher), the extensions are applied to the newly installed version.
When you have multiple extended versions of an accelerator project installed, and select to install an *older* version, the installer does *not* check for extensions because you are installing a lower version.	You currently have the following accelerator project versions installed: • Versions 2.0.0 (which includes extensions) • Versions 6.0.0 (which includes extensions) If you install version 4.0.0, the extensions made in version 6.0.0 are *not* merged into version 4.0.0 because you are installing a lower version. The extensions in version 2.0.0 are ignored.

Automatically Merge Extensions

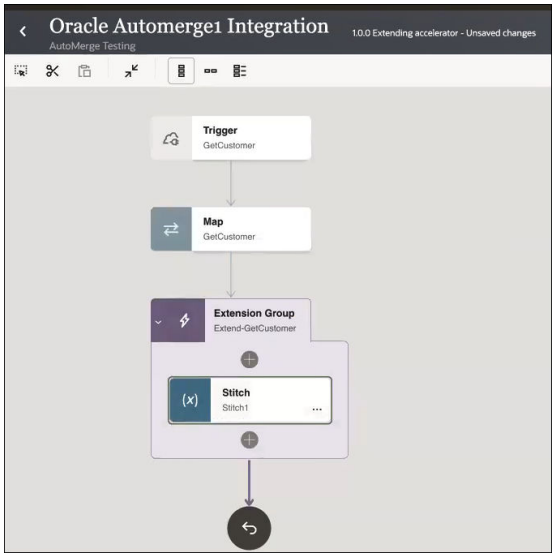
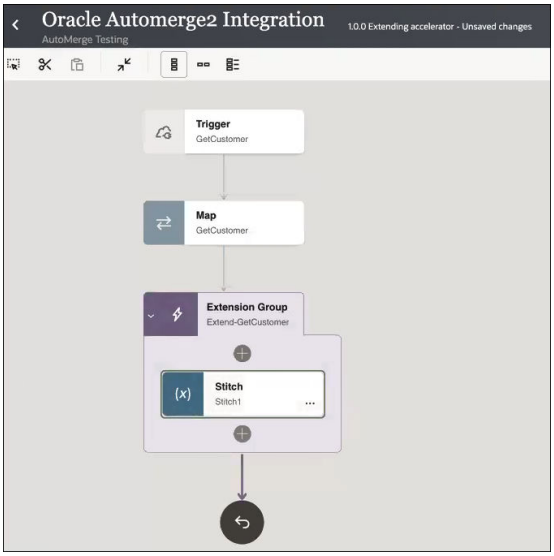
This section provides an overview of how to upgrade to a newer extended accelerator project and automatically merge your extensions (customizations). This process automatically merges the extensions in all integrations in the project.

Assume you have an accelerator project that includes two integrations on version 1.0.0.

Copyright © 2022, 2026, Oracle and/or its affiliates.



Both integrations include an extension group with a stitch action that you added.

Oracle Automerge1 Integration	Oracle Automerge2 Integration
	

Upgraded versions of this accelerator project are then uploaded to the Integration Store. You want to install a newer version and not lose the extensions you made to your current version (1.0.0).

1. Access updated accelerators in either of the following ways:
   a. In the navigation pane, click **Projects**.
   b. Hover over the accelerator project, and click **Actions** ⋮.

   Or

   a. On the Oracle Integration Home page, click **Browse store**.

Copyright © 2022, 2026, Oracle and/or its affiliates.

b. Find the accelerator project to upgrade. You can use the **Filter** to search for accelerators. The **Get latest** label indicates that a new version of the accelerator project is available for installation.

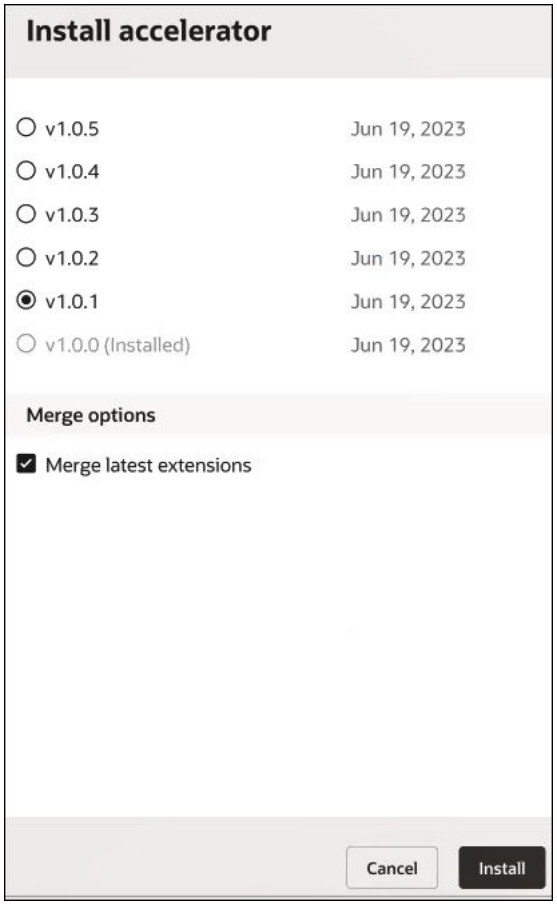
2. Click **Get latest**.
You can also configure and delete installations from the Integration Store.

   The Install accelerator panel opens and shows the latest accelerator versions to which you can upgrade.

3. Select the version to which to upgrade (for this example, **v1.0.1** is selected).

4. If you want to merge the extensions you added (for this example, the stitch actions you added to both version 1.0.0 integrations), then click **Merge latest extensions**.

5. Click **Install**.



6. Click the link in the Confirmation message that is displayed. This takes you to your project to view installation progress.

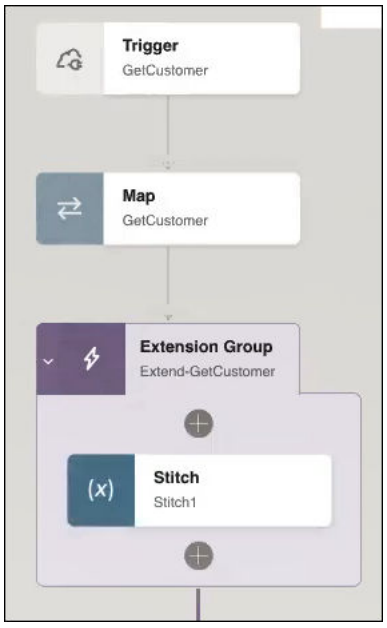
7. Click **Refresh** periodically.

8. Note that the latest installed versions (1.0.1) are successfully merged above the initial integration versions (1.0.0).

Copyright © 2022, 2026, Oracle and/or its affiliates.

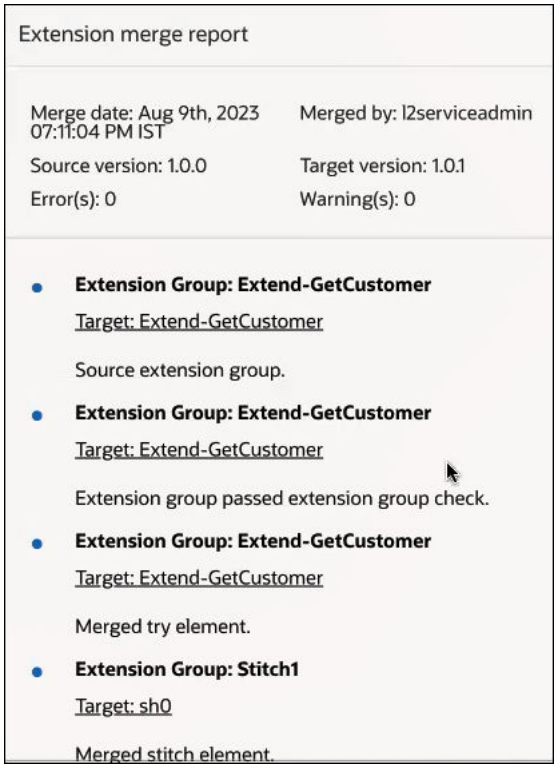


9. Click the latest integrations (1.0.1) to see that your customizations have been merged.



10. Click **Merge report**  to view a report about the merge.

Copyright © 2022, 2026, Oracle and/or its affiliates.



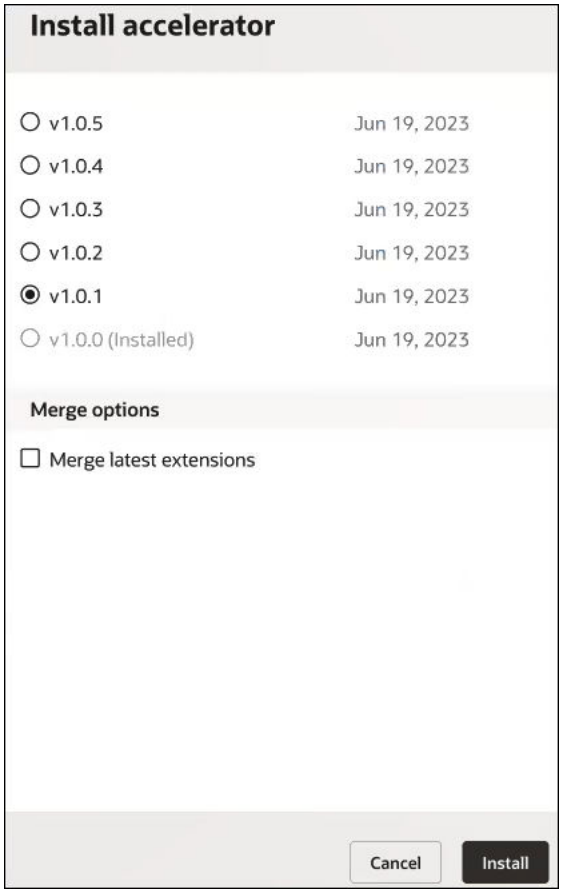
11. Exit the integration.
12. Click **Save** when prompted.

Manually Merge Extensions

If you do not select the **Merge latest extensions** check box described in Step 5, your customizations are not applied. However, you still have the opportunity to manually merge your extensions.

1. Select the version to install, but do not click **Merge latest extensions**.

Copyright © 2022, 2026, Oracle and/or its affiliates.

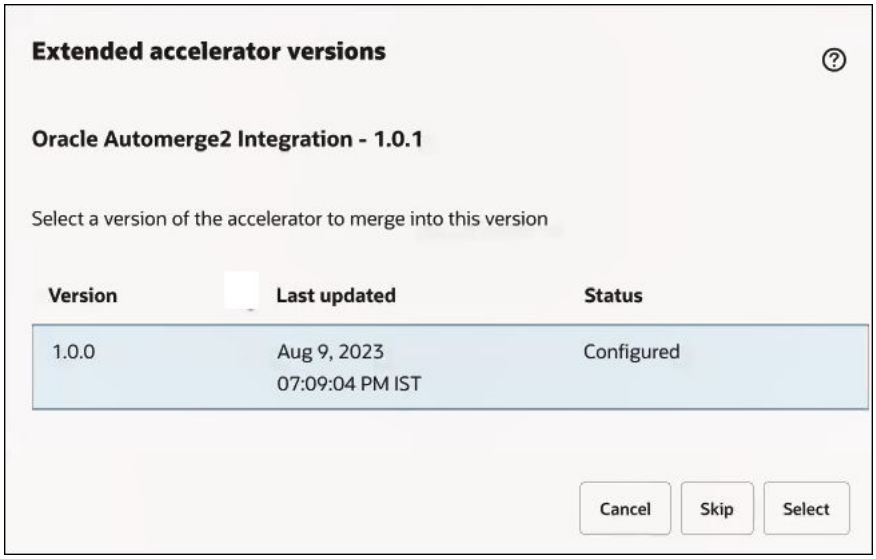


2. Go to the project.
3. In the **Integrations** section, click the newly installed version. Note that the status label is listed as **Configured**, and not **Merge Successful** for the two newly-installed integrations.



The Extended accelerator versions dialog prompts you to select to merge your extensions.

Copyright © 2022, 2026, Oracle and/or its affiliates.



4. Click **Select** to merge your extensions. You can also click **Skip** if you do not want to merge your extensions.
5. Repeat these steps for any remaining integrations that include extensions you want to manually merge.

Create and Extend a Cloned Business Accelerator Project

When you clone a business accelerator project on the Projects page, you can select **Accelerator** in the **Clone as** field. This option enables you to create a cloned business accelerator that you can then extend. A cloned business accelerator project includes the following capabilities:

- Labeled as an **Accelerator** in the **Type** column.
- Retains its business accelerator capabilities.
- Can be extended (customized).
- Can be automatically updated without losing any extensions when a newer version of the business accelerator is installed.

When you install a newer version of the business accelerator from the **Actions** ⋯ menu or the Integration Store, your business accelerator is upgraded without losing your extensions.

See Clone a Project.

Behavior When You Activate an Updated Accelerator Project Deployment with Newer Integration Versions

When you activate a newer project deployment version of an accelerator (for example, 1.0.1), the active integrations in the previously installed project deployment version of that accelerator (for example, 1.0.0) may be deactivated or remain activated based on the versions of the new integrations. Examples of this behavior are described below.

Let's look at several examples that involve importing different versions of an accelerator project deployment. The same behavior applies if you install accelerator project deployments from the Integration Store.

- Activate Project Deployments

Copyright © 2022, 2026, Oracle and/or its affiliates.

- Deactivating Project Deployments

Activate Project Deployments

1. Import and activate accelerator project deployment version 1.0.0. This version consists of the following integration versions, which are activated when the project deployment is activated:
   - Integration1 1.0.0
   - Integration2 1.0.0
   - Integration3 1.0.0

   At a later date, accelerator project deployment version 1.0.1 becomes available. This version consists of the following updated integration versions:
   - Integration1 1.0.1
   - Integration2 1.0.1
   - Integration3 1.0.1
2. Import version 1.0.1 of the accelerator project deployment.
   The Deployment page for the projects now shows both versions.
   - Deployment version 1.0.0 of the accelerator project
   - Deployment version 1.0.1 of the accelerator project
3. Activate version 1.0.1 of the accelerator project deployment.
   Note the following behavior of the integrations in both deployments when accelerator project deployment version 1.0.1 is activated:

Accelerator Project Deployment Version 1.0.0	Accelerator Project Deployment Version 1.0.1
• Integration1 1.0.0 is deactivated.	• Integration1 1.0.1 is activated.
• Integration2 1.0.0 is deactivated.	• Integration2 1.0.1 is activated.
• Integration3 1.0.0 is deactivated.	• Integration3 1.0.1 is activated.

   At a later date, accelerator project deployment version 1.0.2 becomes available. This version consists of the following updated and new integration versions:
   - Integration1 1.0.1
   - Integration2 1.1.0
   - Integration3 2.0.0
   - Integration4 1.0.0
4. Import version 1.0.2 of the accelerator project deployment.
   The Deployment page for the projects now shows all three versions.
   - Deployment version 1.0.0 of the accelerator project
   - Deployment version 1.0.1 of the accelerator project
   - Deployment version 1.0.2 of the accelerator project
5. Activate version 1.0.2 of the accelerator project deployment.
   Note the following behavior of the integrations in both deployments when project deployment version 1.0.2 is activated:

Copyright © 2022, 2026, Oracle and/or its affiliates.

Accelerator Project Deployment Version 1.0.1	Accelerator Project Deployment Version 1.0.2
• Integration1 1.0.1 shows a warning message because it's already active. This message can also appear if integration activation is in progress or not ready for activation. This message appears because Integration1 1.0.1 was active in the previous project deployment and cannot be activated again in the new project deployment.	• Integration1 1.0.1 remains activated.
• Integration2 1.0.1 is deactivated	• Integration2 1.1.0 is activated. This is considered a minor version update. Therefore, Integration2 1.0.1 is deactivated.
• Integration3 1.0.1 remains activated	• Integration3 2.0.0 is activated. This is considered a major version upgrade from 1.0.1 to 2.0.0. Therefore, the previous version of Integration3 1.0.1 is not deactivated. The new Integration3 2.0.0 version is also activated. Both major versions are active at the same time.
• N/A	• New Integration4 1.0.0 is activated.

At a later date, accelerator project deployment version 1.0.3 becomes available. This version consists of the following integration versions:

- Integration1 1.0.1
- Integration2 1.1.0
- Integration3 2.0.0

6. Import version 1.0.3 of the accelerator project deployment.
The Deployment page for the projects now shows four versions.
   - Deployment version 1.0.0 of the accelerator project
   - Deployment version 1.0.1 of the accelerator project
   - Deployment version 1.0.2 of the accelerator project
   - Deployment version 1.0.3 of the accelerator project
7. Activate version 1.0.3 of the accelerator project deployment.
Note the following behavior of the integrations in both deployments when project deployment version 1.0.3 is activated:

Accelerator Project Deployment Version 1.0.2	Accelerator Project Deployment Version 1.0.3
• Integration1 1.0.1 remains activated. The warning message appears because the integration was already active and cannot be activated again in the new project deployment.	• Integration1 1.0.1 remains activated.
• Integration2 1.1.0 remains activated. The warning message appears because the integration was already active and cannot be activated again in the new project deployment.	• Integration2 1.1.0 remains activated.

Copyright © 2022, 2026, Oracle and/or its affiliates.

Accelerator Project Deployment Version 1.0.2	Accelerator Project Deployment Version 1.0.3
• Integration3 2.0.0 remains activated. The warning message appears because the integration was already active and cannot be activated again in the new project deployment.	• Integration3 2.0.0 remains activated.
• New Integration4 1.0.0 remains activated. If an active integration is not updated or included in the latest project deployment, it is not deactivated. Excluded integrations are not deactivated if none of their versions are included in the newest project deployment.	• N/A

Deactivating Project Deployments

Let's review the behavior for deactivating project deployments. Based on the actions above, the status of each project deployment is as follows:

Project Deployment	Status
1.0.0	Deactivate option is unavailable. When project deployment 1.0.1 was activated, all the integrations of project deployment 1.0.0 were deactivated and replaced with newer activated versions in project deployment 1.0.1. There are no active integrations in project deployment 1.0.0, which is why there is no option to deactivate.
1.0.1	Deactivate option is available because there are active integrations: • Integration1 1.0.1 • Integration3 1.0.1
1.0.2	Deactivate option is available because there are active integrations: • Integration1 1.0.1 • Integration2 1.1.0 • Integration3 2.0.0 • Integration4 1.0.0
1.0.3	Deactivate option is available because there are active integrations: • Integration1 1.0.1 • Integration2 1.1.0 • Integration3 2.0.0

If a single integration of a project deployment is active, the deployment is also active and includes the option to deactivate.

1. Hover over the row for project deployment 1.0.3, and select **Actions** ⋯, then **Deactivate**. This action deactivates the following integrations:

   - Integration1 1.0.1
   - Integration2 1.1.0
   - Integration3 2.0.0

   Because these three integration versions also appear in project deployment 1.0.2, they are also deactivated in that deployment. When a project deployment is deactivated, all

Copyright © 2022, 2026, Oracle and/or its affiliates.

integrations in that deployment are deactivated, even if those integrations are also part of other deployments.

2. Hover over the row for project deployment 1.0.2, and select **Actions** ⋮, then **Deactivate**. Integration4 1.0.0 is deactivated. However, warning messages indicate that Integration1 1.0.1, Integration2 1.1.0, and Integration3 2.0.0 are not active and cannot be deactivated. This is because each deactivation occurred when project deployment 1.0.3 was deactivated. When deactivating project deployment 1.0.2, an attempt is again made to deactivate the already inactive integrations.
3. Hover over the row for project deployment 1.0.1, and select **Actions** ⋮, then **Deactivate**. Integration3 1.0.1 is deactivated. However, warning messages indicate that Integration1 1.0.1 is not active and cannot be deactivated. This is because that deactivation occurred when project deployment 1.0.3 was deactivated.

See Create and Manage a Project Deployment.

Deploy Integration Endpoints to Oracle Cloud Infrastructure API Gateway

You can deploy individual integration endpoints as routes to Oracle Cloud Infrastructure API Gateway. An Oracle Cloud Infrastructure API Gateway instance supports a maximum of 20 deployments. Each deployment can contain up to 50 routes (each routing to individual endpoints). This provides you with a capacity of 1000 integration endpoints to which to deploy.

See Overview of API Gateway.

- Restrictions
- Create a Dynamic Group and Policy to Grant Gateway Access
- Create a Virtual Cloud Network
- Create a Subnet, Gateway, Routing Rules, and Security Rules for the Virtual Cloud Network
- Create an Oracle Cloud Infrastructure API Gateway and Deployment in the Oracle Cloud Console
- Deploy an Integration to the API Gateway

Watch a video to learn more:

Video

> **Note**
>
> To perform tasks in the Oracle Cloud Console, you must have the ServiceAdministrator role.

Restrictions

- Deployment to Oracle Cloud Infrastructure API Gateway only works in identity domain-enabled environments.
- Only integrations in a project can be deployed to Oracle Cloud Infrastructure API Gateway. The integration must be publicly available. That is, the **Available to other projects** check box must be selected for this integration.
- The integration must be activated.
- You can only deploy a REST Adapter trigger-based integration.

Copyright © 2022, 2026, Oracle and/or its affiliates.

- REST Adapter trigger connections that expose multiple entry points to a single integration are *not* supported. See Receive Requests for Multiple Resources in a Single REST Adapter Trigger with a Pick Action.

See API Limits.

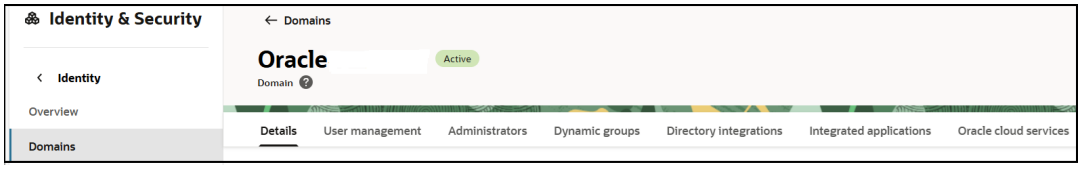
Create a Dynamic Group and Policy to Grant Gateway Access

To deploy integration endpoints to Oracle Cloud Infrastructure API Gateway, you must satisfy the following prerequisites in the Oracle Cloud Console.

- Create a dynamic group.
- Create a policy to grant access to Oracle Cloud Infrastructure API Gateway.

You create the required dynamic group and assign a policy to that group to allow your Oracle Integration instance to access Oracle Cloud Infrastructure API Gateway. The policy defines the permissions for the dynamic group and determines which operations the dynamic group can perform in the Oracle Cloud Infrastructure API Gateway. You must first obtain the client ID of the OAuth application for the Oracle Integration instance.

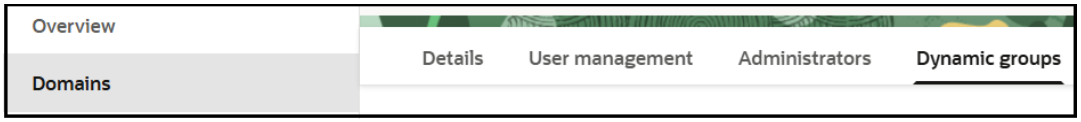
1. Log in to the Oracle Cloud Console.
2. In the navigation pane, select **Identity & Security**.
3. Under **Identity**, select **Domains**.
4. In the upper right corner, click the **Profile** icon, then select the identity domain.
5. In the menu bar, click **Oracle cloud services**.



6. In the **Name** column, click your service instance.
7. Click **OAuth configuration**.
8. Scroll down to the **General Information** section and copy the client ID value to use to create your dynamic group.
9. Scroll to the breadcrumbs at the top and click **Oracle cloud services** to return to the main page for the identity domain.



10. In the menu bar, click **Dynamic groups**.



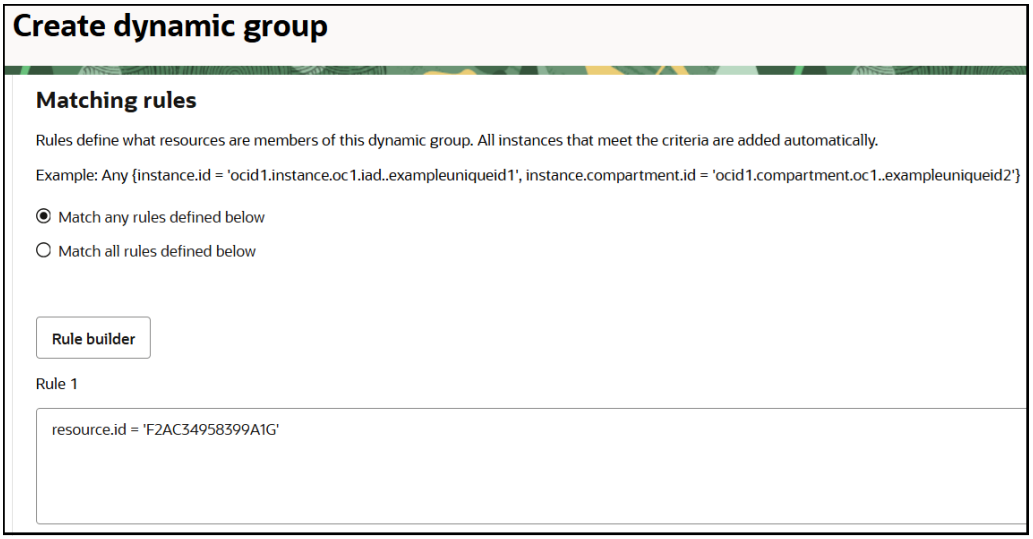
11. Click **Create dynamic group**.
12. Enter the following details:

Copyright © 2022, 2026, Oracle and/or its affiliates.

a. In the **Name** and **Description** fields, enter values. These fields are required.

b. In the **Matching rules** section, enter the required rule, then click **Create**. The resource ID you specify must match the client ID of the OAuth application of your Oracle Integration instance. Ensure that you enclose the value in single quotes. For example:

```
resource.id = 'client_ID'
```



13. In the left navigation pane, click **Policies**.
14. Click **Create Policy**.
15. Select the compartment in which to create the policy.
16. Enter the following details:

    a. In the **Name** and **Description** fields, enter values. These fields are required. The **Compartment** field shows the compartment you previously selected.

    b. In the **Policy Builder** section, click **Show manual editor** to build the required policy for the dynamic group. For example:

    ```
    allow dynamic-group dynamic_group to manage api-gateway-family in
    compartment compartment_name
    ```

    Where:

    - *dynamic_group*: Is the dynamic group name you created.
    - *compartment_name*: Is the compartment in which your Oracle Integration instance is located.

    This enables the Oracle Integration instance associated with the dynamic group to call Oracle Cloud Infrastructure API Gateway in this particular compartment.

17. Click **Create**.

Create a Virtual Cloud Network

You must create a virtual cloud network before you can create an Oracle Cloud Infrastructure API Gateway.

Copyright © 2022, 2026, Oracle and/or its affiliates.

These steps describe how to manually create the virtual cloud network. For additional instructions, see Creating a VCN.

1. In the navigation menu, go to **Networking**.
2. Click **Virtual cloud networks**.
3. Click **Create VCN**.
   While not shown here, you can also select a wizard from the **Actions** menu to create a virtual cloud network. See Create and Configure a Virtual Cloud Network for a tutorial and videos.
4. Enter the following information, then click **Create VCN**.

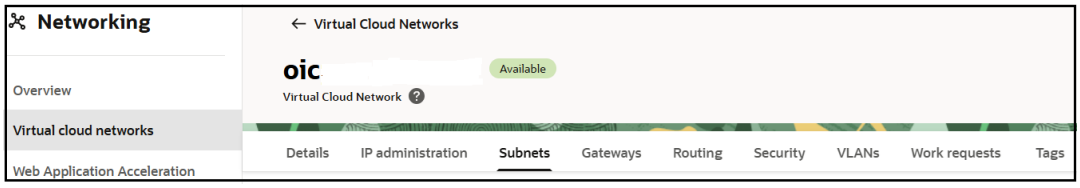
Element	Description
Name	Enter a virtual cloud network name.
Create in Compartment	Displays the compartment you previously selected.
IPv4 CIDR Blocks	Assign up to five IPv4 CIDR blocks to a VCN. At least one is required. See VCN and Subnet Management.
Use DNS hostnames in this VCN	If you plan to use VCN DNS or a third-party DNS, this is required for instance hostname assignment. This selection cannot be changed after the VCN is created. See DNS in Your Virtual Cloud Network
DNS Label	This value is generated from the virtual cloud network name if not specified.
DNS Domain Name	This value is generated from the virtual cloud network name if not specified.
IPv6 Prefixes	Assign a single Oracle-allocated IPv6 /56 prefix to this VCN, or a BYOIPv6 prefix or ULA prefix to the VCN.
BYOIPv6 Prefix	Add BYOIPv6 prefixes. See IPv6 Addresses.
Tags	Add tags to organize your resources.

The details page for the virtual cloud network is displayed.

Create a Subnet, Gateway, Routing Rules, and Security Rules for the Virtual Cloud Network

You must create a subnet, internet gateway, routing rules, and security rules for the virtual cloud network.

1. In the menu bar, click **Subnets**.



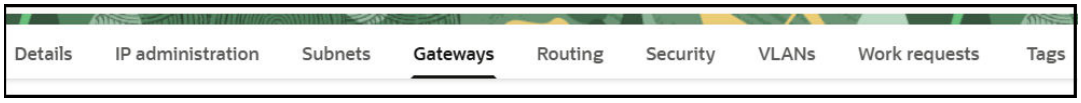
You now create a subnet in the VCN. A subnet is a logical subdivision of the VCN. Each subnet consists of a contiguous range of IP addresses that don't overlap with other subnets in the VCN. For additional instructions, see Creating a Subnet.

Copyright © 2022, 2026, Oracle and/or its affiliates.

2. Click **Create Subnet**.
3. Enter the following information, then click **Create Subnet**.

Element	Description
Name	Enter a subnet name.
Create in Compartment	Displays the compartment you previously selected.
Subnet Type	Select a subnet type: • **Regional (Recommended)**: Instances in the subnet can be created in any availability domain in the region. This is useful for high availability. • **Availability Domain-specific**: Instances in the subnet can only be created in one availability domain in the region.
IPv4 CIDR Block	Enter the IPv4 CIDR block.
IPv6 Prefixes	If you want to assign IPv6 prefixes, click **Assign an Oracle allocated IPv6 /64 prefix** and enter the values.
Route Table compartment	Select the route table compartment.
Route Table	Select the route table. See VCN Route Tables.
Subnet Access	Select an access type: • **Private Subnet**: Prohibit public IP addresses for instances in this subnet. • **Public Subnet**: Allow public IP addresses for instances in this subnet.
Use DNS hostnames in this Subnet	If you plan to use VCN DNS or a third-party DNS, this is required for instance hostname assignment. This selection cannot be changed after the VCN is created.
DNS Label	This value is generated from the virtual cloud network name if not specified.
DNS Domain Name	Enter the DNS domain name in the format shown in the field.
Dhcp Options compartment	Select the default DHCP compartment.
Dhcp Options	Select the DHCP options. See DHCP Options.
Security List compartment	Select the security list compartment.
Security List	Select the security list. You can associate up to five network security lists with the subnet. See Security Lists.
Resource Logging	Select to enable resource logging to allow resource tracking, troubleshooting, and data insights.
Tags	Add tags to organize your resources.

4. Click the subnet breadcrumb at the top, then select **Gateways**.



5. Scroll down and click **Create Internet Gateway**.

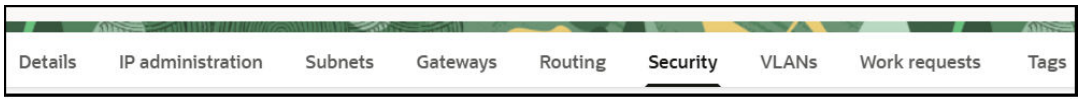
Copyright © 2022, 2026, Oracle and/or its affiliates.

An internet gateway is an optional gateway you can add to a virtual cloud network to enable direct connectivity to the internet. The gateway supports connections from within the virtual cloud network (egress) and connections from the internet (ingress).

6. Enter the following information, then click **Create Internet Gateway**. For additional information, see Creating an Internet Gateway.

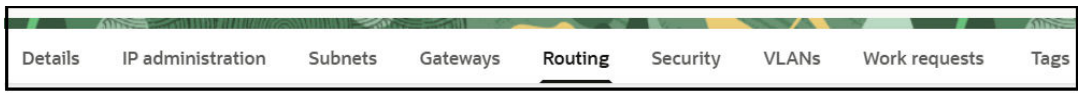
Element	Description
Name	Enter an internet gateway name.
Create in Compartment	Displays the compartment you previously selected.
Advanced options	Click if you want to associate a route table with the gateway. You can modify the rules in the current route table or replace it with another route table. See VCN Route Tables.

7. In the menu bar at the top, select **Security**.



Security rules control traffic. For additional information, see Security Rules.

8. In the **Name** column, click the default security list.
9. Click the **Security rules** tab.
10. Click **Add Ingress Rules**.
    a. Specify the CIDR source type and IP protocol.
    b. Leave the **Source Port Range** field blank.
    c. Enter 443 in the **Destination Port Range** field.
    d. Click **Add Ingress Rules**.
11. Return to the details page for the virtual cloud network you created.
12. In the menu bar at the top, click **Routing**.
    A virtual cloud network uses route tables to send out traffic (for example, to the internet, an on-premises network, or a peered virtual cloud network). For additional information, see VCN Route Tables and Working with VCN Route Tables and Route Rules.



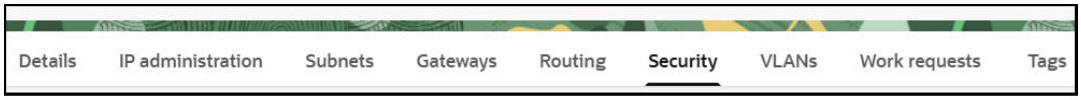
13. In the **Name** column, click the default route rule.
14. Click the **Route Rules** tab.
15. Enter the following information, then click **Add Route Rules**.

Element	Description
Protocol Version	Select the protocol version to use. • **IPv4** • **IPv6**

Copyright © 2022, 2026, Oracle and/or its affiliates.

Element	Description
Target Type	Select **Internet Gateway**.
Destination CIDR Block	Enter the destination CIDR block.
Target Internet Gateway compartment	Select the target internet gateway compartment.
Target Internet Gateway	Select the API gateway.
Description	Enter an optional description.

16. In the menu bar at the top, click **Security**.



17. Scroll down to **Network Security Groups**, and click **Create Network Security Group**. For additional instructions, see Creating a Network Security Group.
18. Enter the following information, then click **Create**.

Element	Description
Name	Enter a name.
Create in Compartment	Leave the compartment as is.
Tags	Add tags to organize your resources.

19. Scroll down to **Add Security Rules** and expand **Rule**.
20. Enter the following information, then click **Create**.

Element	Description
Direction	Select **Ingress**.
Source Type	Select **CIDR**.
Source CIDR	Add an IP address or a range of IP addresses.
IP Protocol	Select **TCP**.
Source Port Range	Specify `All`.
Source CIDR	Specify the value.
Destination Port Range	Specify 443.

21. In the breadcrumbs at the top of the page, click the link for the virtual cloud network that you created.
22. In the menu bar, click **Security**.
    If egress rules do not exist, you must define them.
23. Click **Create Security List**.
24. Scroll down to the **Allow Rules for Egress** section and add egress rules.

Create an Oracle Cloud Infrastructure API Gateway and Deployment in the Oracle Cloud Console

You must create an Oracle Cloud Infrastructure API Gateway. Each gateway instance supports a maximum of 20 deployments. Each deployment can handle up to 50 routes. This means that one gateway instance can protect up to 1000 APIs. You can select an existing deployment, enabling a new route to be created in that deployment. You can also create a new deployment.

1. In the navigation menu, go to **Developer Services**.

Copyright © 2022, 2026, Oracle and/or its affiliates.

2. Under **API Management**, select **Gateways**.
3. Select the compartment to use for deployment.
4. Click **Create Gateway**.
5. Enter the following information, then click **Create Gateway**.

Element	Description
Name	Enter a gateway name.
Type	Select **Public**.
Compartment	Displays the compartment you previously selected.
Network	Select the following networking details: • **Virtual cloud network**: Select a virtual cloud network. • **Subnet**: Select a VCN with at least one regional subnet added. See Networking Overview.
Enable network security groups	Select the check box, then select a compartment with at least one network security group.
Certificate	Select an SSL/TLS certificate that has been added to Oracle Cloud for use with a custom DNS configuration or use the default certificate provided by the gateway. See Setting Up Custom Domains and TLS Certificates.

The new gateway is displayed in the **Name** column on the Gateways page.



6. Click the gateway name.
7. Under **Resources** in the left navigation pane, click **Deployments**.
8. Click **Create deployment**.
   The Create deployment wizard is displayed.

   You can create deployments based on categories appropriate to your business environment. For example, you may want to create separate deployments for applications, functional areas within an application, client requirements (for example, all APIs for a client-facing portal), and so on. You then deploy the integration endpoints to the appropriate deployment category.
9. Enter the following information, then click **Next**.

Element	Description
Name	Enter a deployment name. For this example, `NetSuite` is entered.
Path prefix	Enter a prefix. For this example, `/netsuite` is entered.

Copyright © 2022, 2026, Oracle and/or its affiliates.

Element	Description
Compartment	Displays the compartment you previously selected.
API request policies	Select API request policies as required for your environment: • **Mutual-TLS**: Select to enable mTLS. • **CORS**: Configure CORS access. • **Rate limiting**: Configure rate limiting. • **Usage plans**: Configure usage plans.
API logging policies	Select a logging level. See Adding Logging to API Deployments.
Tags (under **Show advanced options**)	Add tags to organize your resources.

10. Select an **Authentication** method, and specify additional details, then click **Next**

    - **No Authentication**: Any client that has network access to the gateway can make requests to all routes in this deployment.
    - **Single Authentication**: Configure integration with a single identity provider. You can optionally limit access for all routes to authenticated clients only.
      If you select this option, authentication, validation, and other security fields are displayed for configuration.
    - **Multi-Authentication**: Configure integration with one or more identity providers. You can optionally limit access for all routes to authenticated clients only.
      If you select this option, additional authentication fields are displayed for configuration.

    Route 1 is displayed.

    Individual integration endpoints are deployed as routes to an Oracle Cloud Infrastructure API Gateway deployment.

11. Enter the following information, then click **Next**.

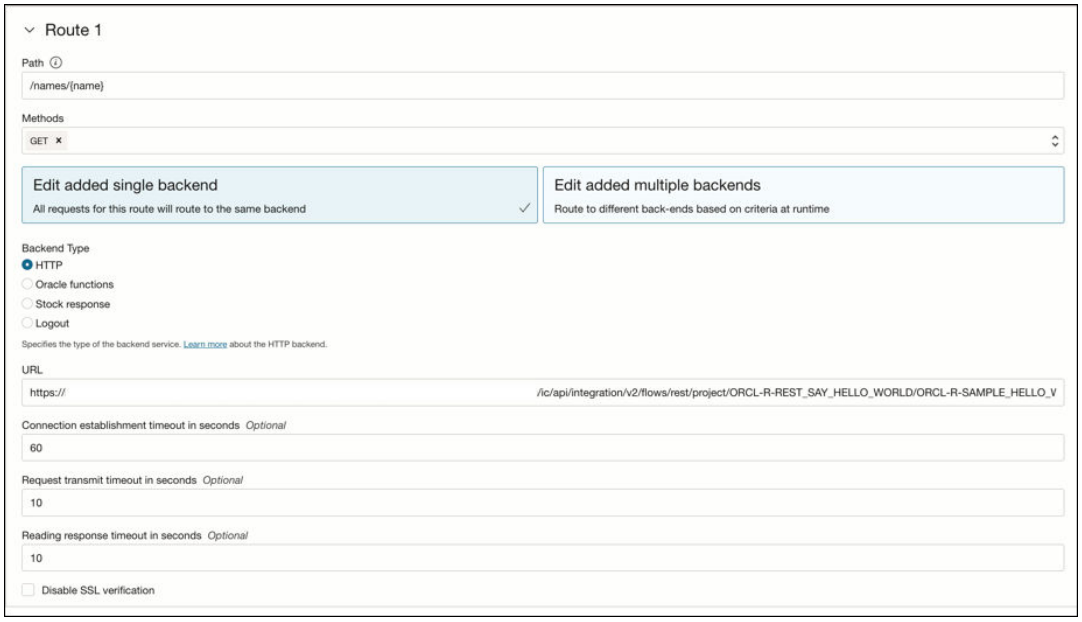
Element	Description
Path	Enter a path (for example, `/order`).
Methods	Select one or more methods based upon your requirements (for example, **GET**, **POST**, **PUT**, or others).
Backend Type	This option enables all requests for this route to be sent to the same backend. Select a backend type. Each selection causes additional fields to be displayed for you to configure. • **HTTP** • **Oracle functions** • **Stock response** • **Logout**

Copyright © 2022, 2026, Oracle and/or its affiliates.

Element	Description
Add multiple backends	Route to different backends based on criteria at runtime. To configure your route to support multiple backends, first define the request context element to use as the selector. The gateway uses the selector at runtime to choose the backend based on a matching rule defined for the backend. **Selector**: Select the request context table. Some selections cause additional fields to be displayed for you to configure. • **Auth** • **Headers** • **Host** • **Path** • **Query parameters** • **Subdomain** • **Usage plan ocid** **Backends**: Add one or more backends for your route. Each backend needs a matching expression that the gateway uses to match based on the request context in the selector at runtime.
Define backend	Click to define a backend. You can also expand and specify polices as required for your environment. • **Show route request policies** • **Show route response policies** • **Show response caching policies** • **Show route logging policies**

12. Review your selections on the Deployment page, then click **Create**.
Route 1 is configured.
13. Follow Step 8 through Step 12 to create any additional deployments and routes. For this example, a deployment (**Oracle Rest Say Hello World**) is created that includes a single route:

Copyright © 2022, 2026, Oracle and/or its affiliates.



When complete, the deployments are listed on the details page for the gateway.

You are now ready to deploy individual integration endpoints as routes to Oracle Cloud Infrastructure API Gateway.

Deploy an Integration to the API Gateway

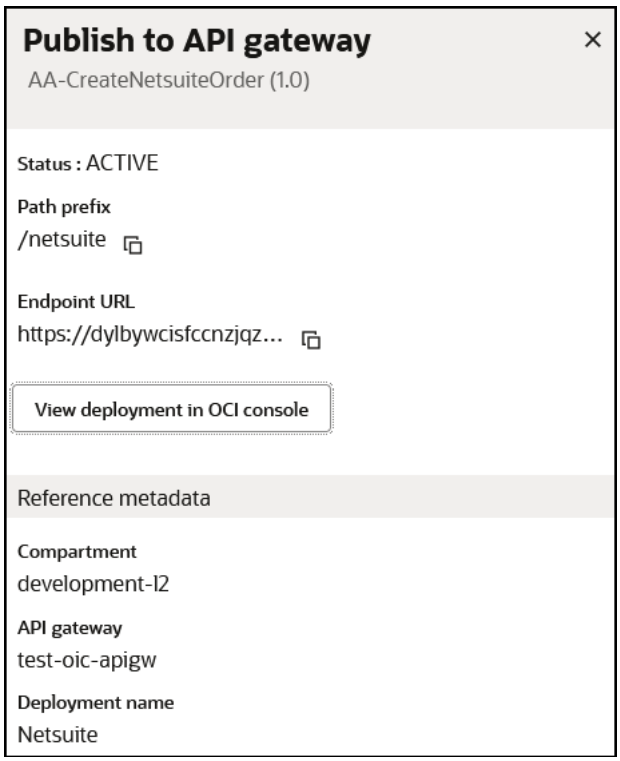
After completing all prerequisites and gateway and deployment configuration tasks in the Oracle Cloud Console, you can deploy individual integration endpoints as routes from a project to Oracle Cloud Infrastructure API Gateway. You perform deployment from Oracle Integration.

1. In the navigation pane, click **Projects**.
2. Click the project name or click .
3. In the **Integrations** section, find the already-activated integration to publish. Only active REST Adapter, triggered-based integrations can be deployed.
4. Click **Actions** , and select **Publish to API Gateway**.
   The Publish to API gateway panel opens.
5. Enter the following information:

Element	Description
REST endpoint	Displays the endpoint of the integration. This endpoint cannot be deselected.
Search or select compartment	Select the compartment created in the Oracle Cloud Console that includes Oracle Cloud Infrastructure API Gateway.
Search or select API gateway	Select the Oracle Cloud Infrastructure API Gateway instance created in that compartment.
Select or create API gateway deployment	Select a deployment inside the gateway instance. You can also create a gateway from this field.

For this example, the **test-oic-apigw** gateway and **NetSuite** deployment created in Create an Oracle Cloud Infrastructure API Gateway and Deployment in the Oracle Cloud Console are selected.

Copyright © 2022, 2026, Oracle and/or its affiliates.



6. Click **Publish**.
   The deployment is visible in Oracle Cloud Infrastructure API Gateway. Use the **Observe** tab in the project to check if the API calls are being reached when invoked by the Oracle Cloud Infrastructure API Gateway deployment URL.
7. If you want to unpublish your API gateway deployment, click **Undo publish**.

Clone an Integration in a Project

Cloning an integration in a project creates a new copy with identical connections and data mappings. The cloned version must have a unique identifier and/or version. You can also specify a name, keyword, and description. You cannot specify a package name. The remaining configuration is the same. You can reconfigure the clone after you create it. You can also clone a locked integration.

1. In the navigation pane, click **Projects**.
2. Click the project name or click .
3. In the **Integrations** section, find the integration to clone.
4. Click **Actions** , then select **Clone**.

   The Clone integration panel opens.
5. Enter a name, unique identifier, version number, keywords, 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`.
6. Click **Clone**.

Copyright © 2022, 2026, Oracle and/or its affiliates.

Update the Tracing Level of Integrations in a Project

You can change the tracing level of an active integration without having to deactivate and reactivate it. You can also set the tracing level globally for all integrations in a project.

1. In the navigation pane, click **Projects**.
2. Click the project name or click .
3. In the **Integrations** section, hover over the active integration on which to set tracing.
4. Click **Actions** and select **Configure activation**.
5. Follow the steps to set the tracing level. See Manage Tracing Levels on Integrations.

Test REST Adapter Trigger Connection-Based Integrations in Projects

You can test application integrations in projects that are designed with a REST Adapter trigger connection.

1. In the navigation pane, click **Projects**.
2. Click the project name or click .
3. In the **Integrations** section, hover over the specific integration to run.
4. Click **Actions** and select **Run**. The **Run** option is only available for REST Adapter trigger connection-based integrations.

   The Configure and run page is displayed.
5. Follow the steps to run the integration. See Test Integrations from Outside the Integration Canvas.

View the Dependent Relationships Between Project Resources

You can view the dependent relationships between all integrations, connections, lookups, JavaScript libraries, events, queues, HL7 messages, decisions, B2B documents and schemas, and healthcare schemas in a project, in another project, or outside a project.

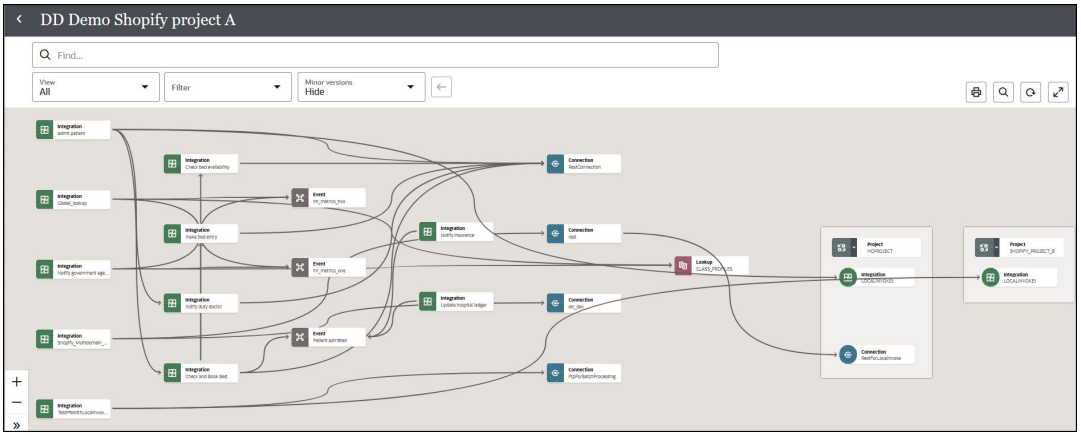
- View Project Dependencies
- View a Project Summary Report
- View Lookup Usage
- View the Dependency Diagrams of Different Integration Versions
- View Integrations that are Shared Between Projects

View Project Dependencies

1. In the navigation pane, click **Projects**.
2. Click the project name or click .
3. Click **View dependencies** .
   A view of the dependencies between all resources in the current project and other projects is displayed. If you are invoking an integration in a different project, that integration is shown in a container with the label **Project** and the project name as the container title. If

Copyright © 2022, 2026, Oracle and/or its affiliates.

the diagram is dependent upon an integration that is outside of any project, the container is labeled **Not in project**. An option for printing the dependency diagram is available in the upper right corner.

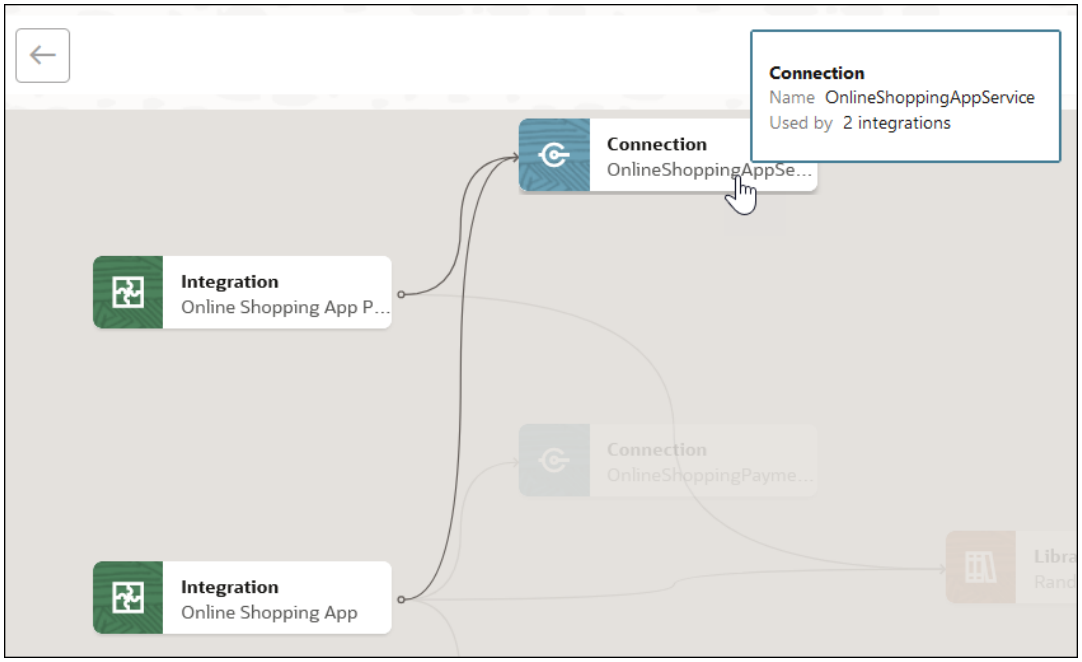


4. Hover over a resource to display its dependencies.
For this example, the integration uses one connection and three integrations, including one in another project. Hovering over the lines connecting the nodes also shows the relationship between the nodes in the tool tip.

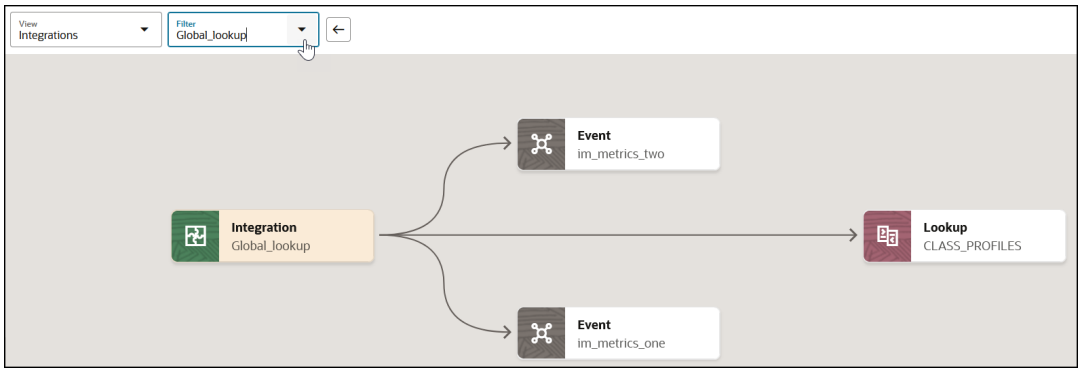


5. Hover over a connection to see its dependent resources. For this example, the connection is used in two integrations.

Copyright © 2022, 2026, Oracle and/or its affiliates.



6. Click the nodes to drill down into the resource-specific view, and keep clicking to drill down further.
7. Click **Back** ← to return to the previous view until you return to the beginning of the navigation.
8. From the **View** list, select to view specific resources (integrations, connections, lookups, JavaScript libraries, HL7 messages, decisions, healthcare schemas, and events). By default, **All** is selected to show all resources in the project.
9. From the **Filter** list, select to view a specific integration or connection and its dependencies. When you select a specific integration or connection, the **View** list changes to show the same resource type. For example:



View a Project Summary Report

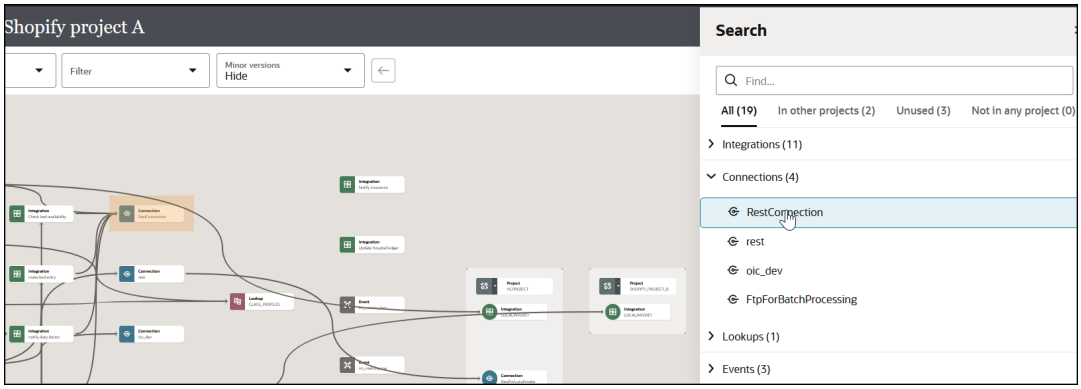
1. Click **View dependencies** .
2. Click **Project summary** .
   The Summary panel shows tabs for the dependent resources in the current project, the dependent resources being used in other projects, the unused dependent resources, and

Copyright © 2022, 2026, Oracle and/or its affiliates.

the dependent resources being used that are not in projects. You can search on the name of a resource to display it.



3. Click a tab, and expand the resource category.
4. Click a specific resource to highlight its location in the dependency diagram.



View Lookup Usage

You can view the specific action in an integration in which a lookup is being used.

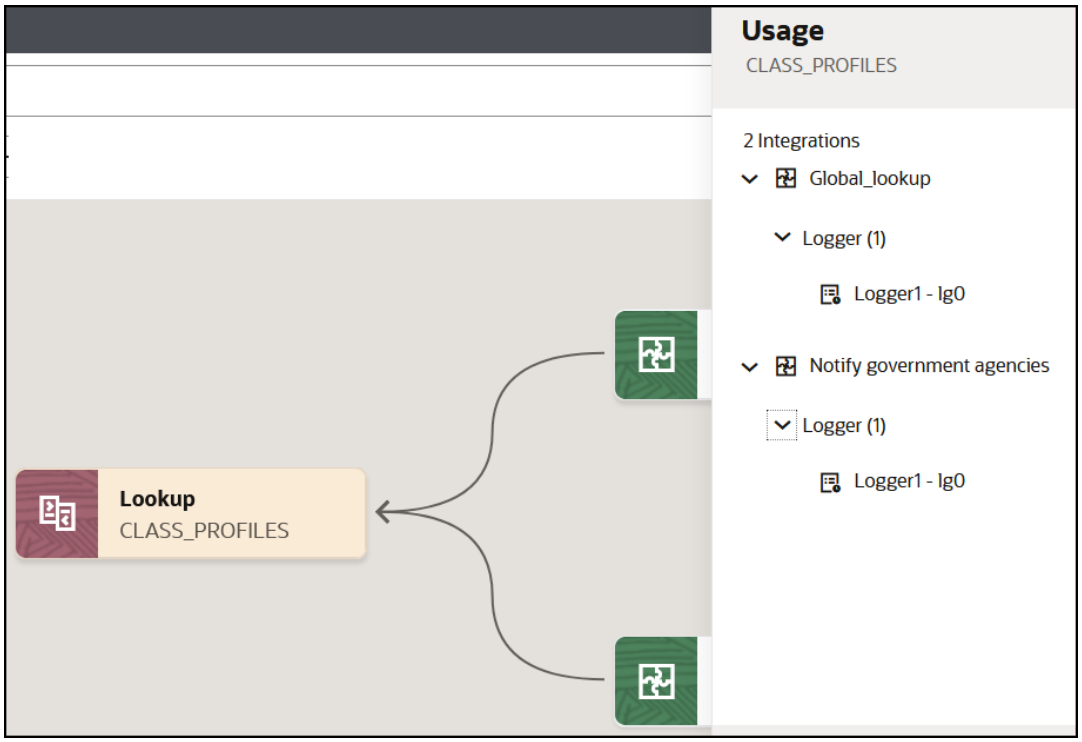
1. Click **View dependencies** .
2. Hover over a lookup in the dependency diagram to see the integrations in which it is used.



3. Click the lookup for a closer view.
4. In the lookup, click **Actions** , and select **Usage**.
The Usage panel is displayed.

Copyright © 2022, 2026, Oracle and/or its affiliates.

5. Expand the integrations in the panel to see the actions using the lookup. For this example, log actions in two integrations are using the lookup.



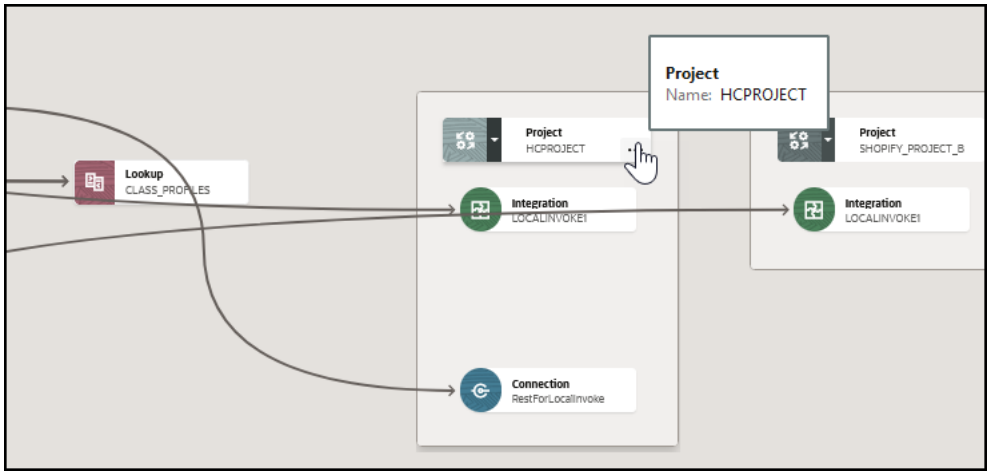
6. Hover over a log action name to display a copy icon for copying the name.
7. Go to the specific integration and click **Search** to open the Search panel.
8. Enter the name to find its location in the integration.

Navigate to the Dependency Diagrams of Remote Projects

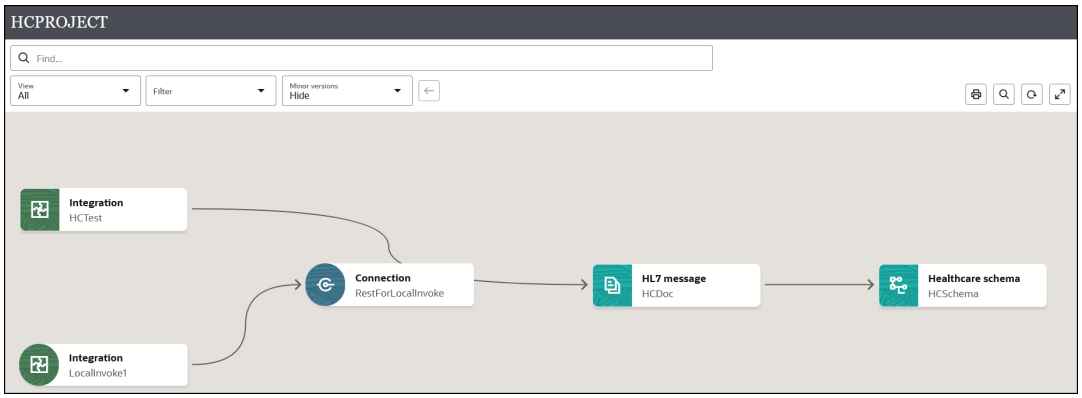
If your dependency diagram includes dependent resources in other (remote) projects, you can access the dependency diagrams for those projects.

1. Click **View dependencies**.
2. Hover over a remote project in the dependency diagram. These projects are identified by the word **Project** and the project name.
3. Click **Actions**, and select **Open**.

Copyright © 2022, 2026, Oracle and/or its affiliates.



The remote project opens in a second browser tab.



4. View resources in the second project, then close the browser tab when done.

View the Dependency Diagrams of Different Integration Versions

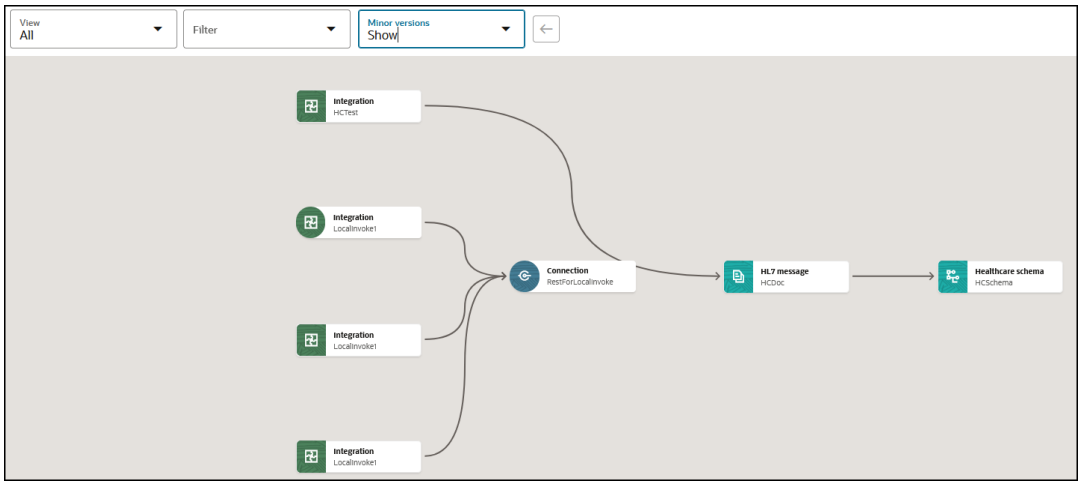
The latest active minor version of an integration is shown by default in the dependency diagram. You can also view all minor versions of an integration in the dependency diagram.

1. Click **View dependencies** for a project with multiple integration versions. For this example, **LocalInvoke1** has three minor versions.

Copyright © 2022, 2026, Oracle and/or its affiliates.

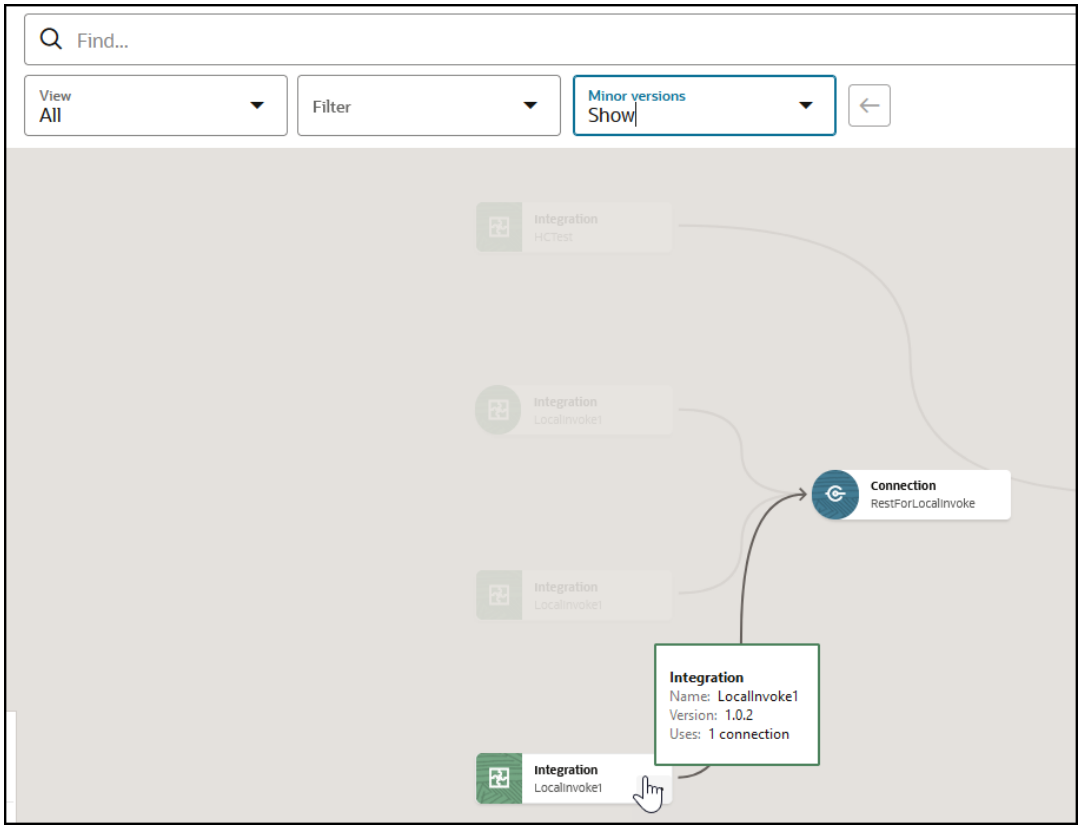


2. From the **Minor versions** list, select **Show**.



3. Hover over the integrations to see the versions and their dependent resources. For this example, minor version 1.0.2 is shown.

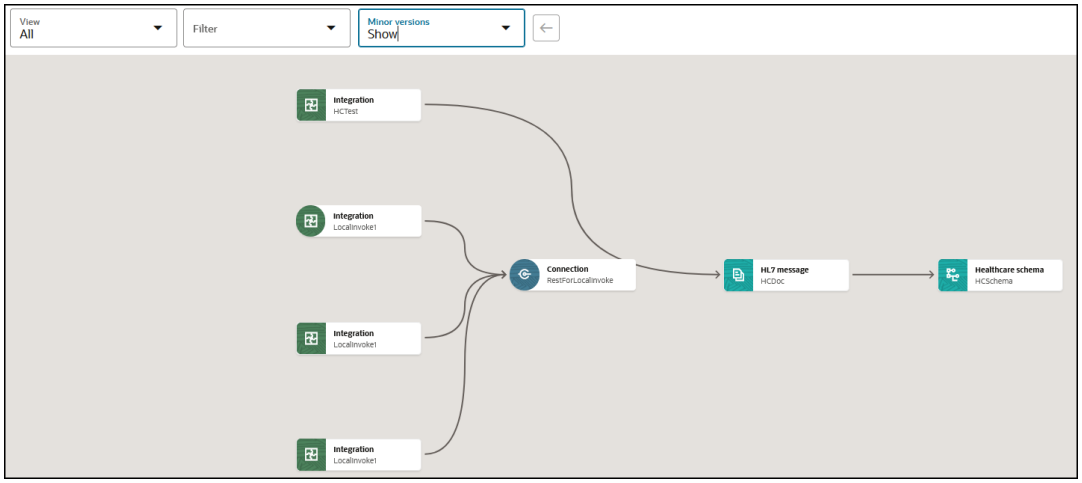
Copyright © 2022, 2026, Oracle and/or its affiliates.



View Integrations that are Shared Between Projects

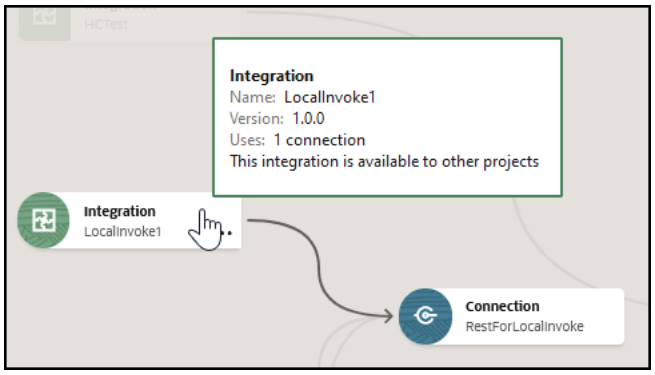
Integrations that are shared between projects are identified by a green circle in the dependency diagram.

1. Click **View dependencies** .
2. View the round circle for the second integration in the dependency diagram. This circle indicates that this integration is shared between projects. Integrations that are not shared with other projects are identified by a green square.



Copyright © 2022, 2026, Oracle and/or its affiliates.

3. Hover over the shared integration and note that the tool tip indicates this integration is available to other projects.



View Run Details About Integrations in Projects

You can view run details about active application integrations in projects.

1. In the navigation pane, click **Projects**.
2. Click the project name or click .
3. In the **Integrations** section, hover over the specific integration to view.
4. Click **Actions** and select **Run details**. The metadata URL of the integration is displayed.

View Project Status

The overall status of a project is based on the status of its integrations.

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

   The **Type** column indicates the project type is **Developed**, meaning it was created by a user.

2. View the overall status of the project in the **Status** column.

   Overall project status is based on the status of each integration in your project. The following table provides example of how the overall status is determined. For this example, assume you have three integrations in your project.

If Integration 1 Status Is...	If Integration 2 Status Is...	If Integration 3 Status Is...	Then Project Status Is...
Draft	Draft	Draft	Draft
Draft	Draft	Configured	Draft
Draft	Configured	Configured	Draft
Configured	Configured	Configured	Configured
Configured	Configured	Active	Configured
Configured	Active	Active	Configured
Active	Active	Active	Active
Draft	Configured	Active	Draft

Copyright © 2022, 2026, Oracle and/or its affiliates.

Edit Project Description Details

You can edit naming description details in user-developed projects that you specified during creation, such as the name, description, and keyword selections. You *cannot* change the project identifier value.

1. In the navigation pane, click **Projects**.
2. Click the project name or click .
3. In the upper right corner, click **Edit details** .
4. Optionally change the project name, description, and selected keywords.
5. Click **Save changes**.

Delete a Project

You can delete a project. However, you cannot delete a project if an integration is active or any resource is locked.

1. In the navigation pane, click **Projects**.
2. Hover over the project to delete.
3. Click **Actions** , then select **Delete**.

Add Retry Logic to Invoke Connections in Projects

You can add retry logic to invoke connections in projects that enables you to automatically retry failed invocations a specified number of times within a specified time period. Both retry settings are configured with default values, but you can update these settings, if necessary.

- Capabilities
- Add Retry Logic to an Invoke Connection at Design Time
- Run an Integration Configured with Retry Logic

Capabilities

Retry logic provides the following capabilities:

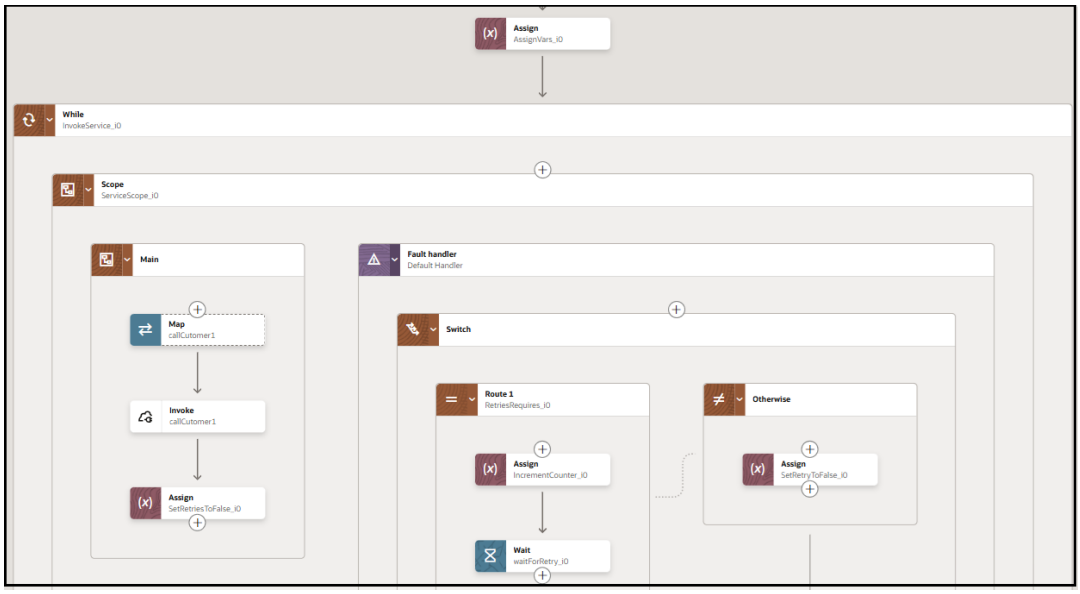
- The number of invoke connection retries and the time between retries are automatically configured for you. You can also change these values, if necessary.
- Is available only in integrations in projects.
- Both business and runtime faults are retried. For example, a business fault may be a duplicate customer error in Netsuite or a runtime fault may be an `HTTP 500` error.
- All faults from invoke connections are currently included as part of a retry. You cannot select the type of faults to retry or ignore.
- Retry logic can be added to as many invoke connections in an integration as you feel necessary. Retry logic is not handled at the integration level.

Add Retry Logic to an Invoke Connection at Design Time

1. Open a project, and then open the integration in which to add retry logic.

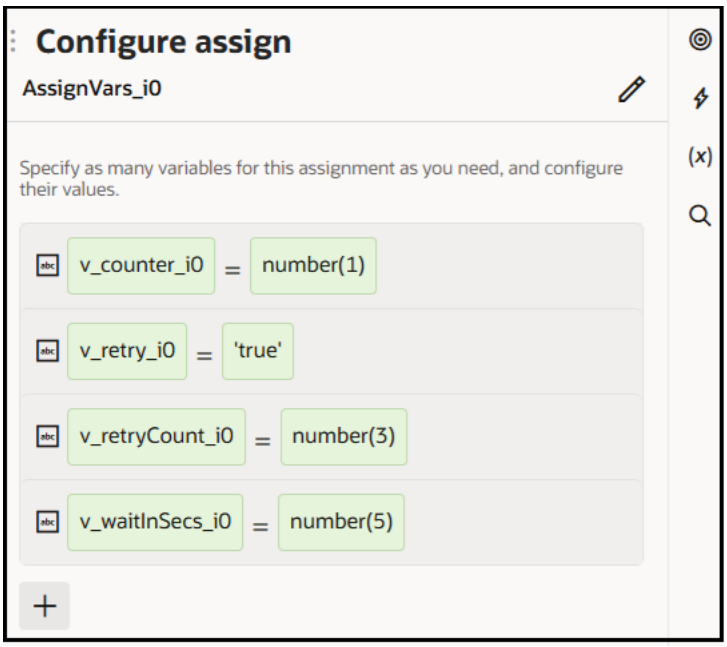
Copyright © 2022, 2026, Oracle and/or its affiliates.

2. Click the **Actions** ⋮ menu of an invoke connection and select **Add Retry Logic**.
3. Click **Add** when prompted to add new canvas actions and move the selected invoke connection and its associated map action.
Your integration is updated with retry logic.



The retry logic consists of the following:

- An assign action is added immediately above the while action that controls the default number of retries (set to 3) and the time between retries (set to 5 seconds). You can update these values, as necessary.



Copyright © 2022, 2026, Oracle and/or its affiliates.

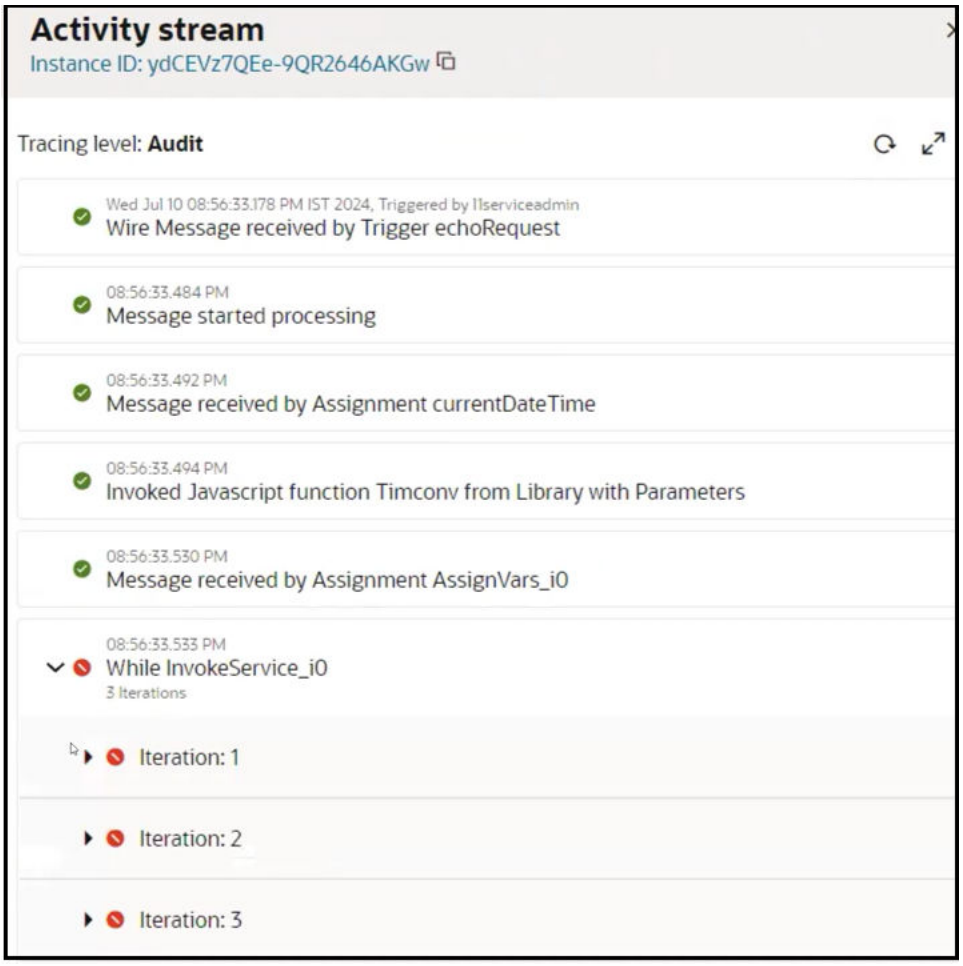
> **Note**
>
> Do not set your number of retries to a large value. This can impact performance.

- The invoke connection and its associated map action are moved inside a scope action of the while action (for this example, the invoke connection **callCustomer1** and map action **callCustomer1**). An assign action is also added to the scope. If your invoke connection did not include a map action, only the invoke connection is moved inside the scope action. A new map action is not created.
- A fault handler with a switch action is added.

Run an Integration Configured with Retry Logic

This section provides a view of activity stream output when invoke connection failures occur.

1. Run an integration on which you added retry logic.
2. Expand the activity stream and note that three retries (iterations) were run (the default setting).



When the invoke connection fails, it is caught by the fault handler in the scope action. A variable increments the fault. Regardless of the error code (for example, an HTTP error or an XPath expression error), a retry is performed.

Copyright © 2022, 2026, Oracle and/or its affiliates.

3. Expand each failed iteration to view the timestamp between retries. Note the five second delay between the first and second iterations (the default setting).



Monitor Integrations in a Project

You can track instances and monitor errors in a project from a single workspace during runtime. You can also access and view the activity stream that shows details about the movement of messages through integrations.

Topics:

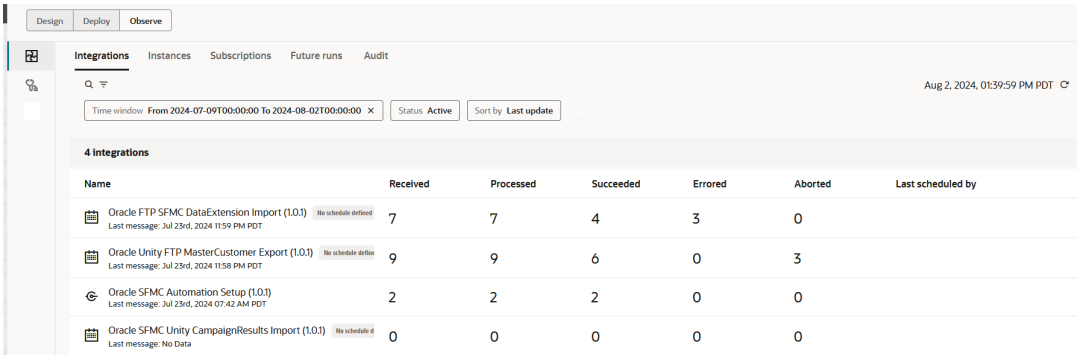
- Monitor the Message Processing Status of Integrations in Projects
- Track Instances in a Project
- View Subscribed Instances in a Project
- Manage Errors in a Project
- View the Schedule Integration Runs Calendar in a Project
- View the Design-Time Audit Log of a Project
- Monitor Robots in a Project
- View the Relationship Between Parent and Child Integration Tracking Instances in a Project
- Monitor MLLP Adapter Connections and Queues

Monitor the Message Processing Status of Integrations in Projects

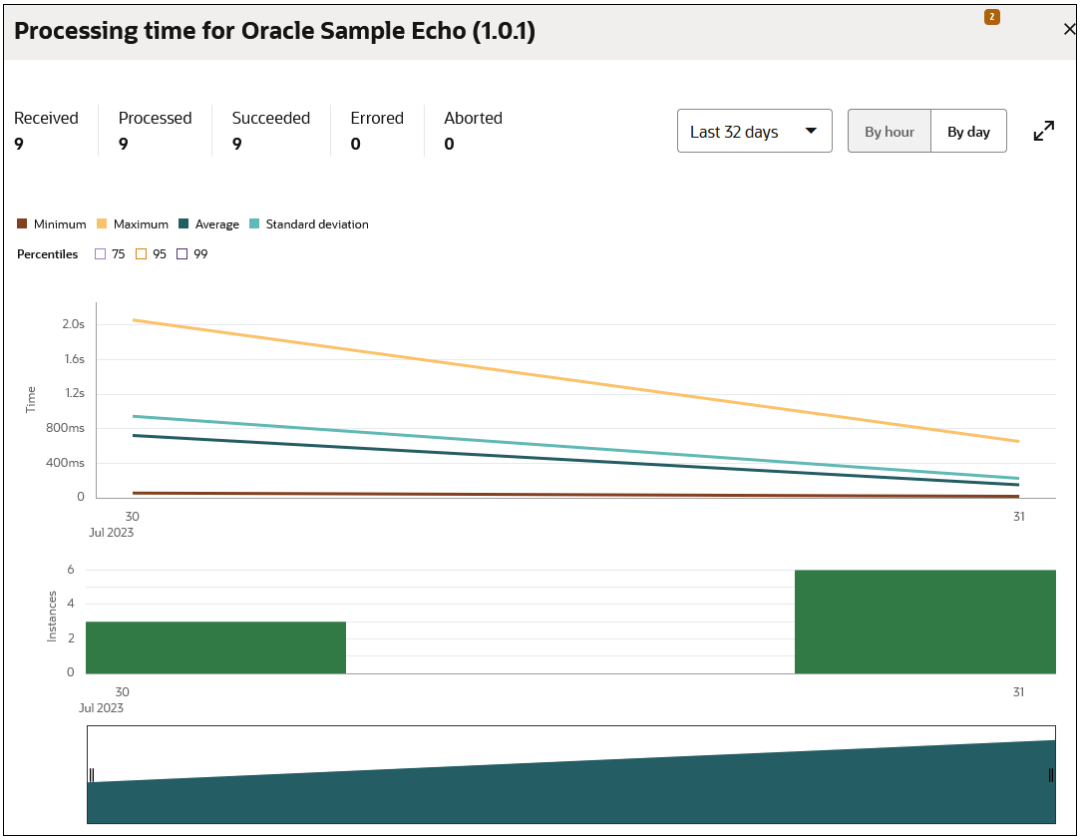
You can view the message processing status of your running integrations in a project, including how many messages have been received and processed, how many successful messages and errors have occurred, and how many messages have been canceled. You can also view aggregated instance metrics for the selected integration.

1. In the navigation pane, click **Projects**.
2. Click the project name or click **Edit** .
3. Click **Observe**.
4. Click **Integrations**.

Copyright © 2022, 2026, Oracle and/or its affiliates.



5. Click **Filter** , then select the time window, integration status (active, configured, activating, failed, or all), and integration style (application, schedule, or event). You can also specify a custom time range to filter the search for integrations under **Date range** with the **From** and **To** buttons.
6. Click numbers in the **Received**, **Processed**, **Succeeded**, **Errored**, and **Aborted** columns to access the list of instances in those states on the Instances tab of the project. If you click the **Received** number, instances are displayed based on their creation date. All other clickable numbers are sorted by their last updated date.
7. Hover over a row for an integration and click **View statistics** to show aggregated instance metrics for the selected integration. For information about using the Statistics page, see Step 6 in Monitor the Message Processing Status of Integrations.



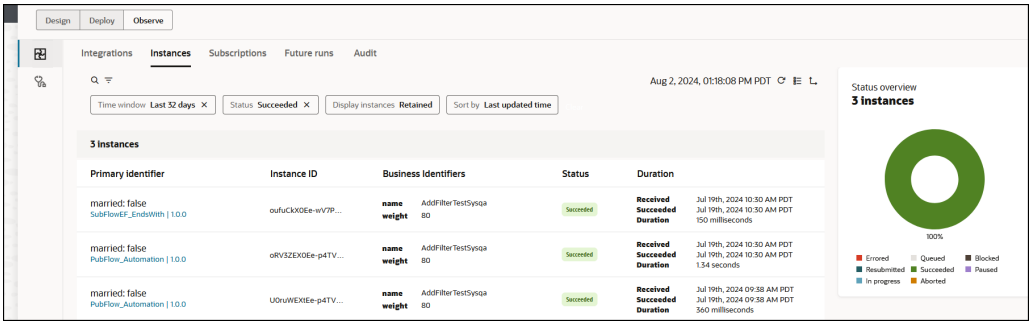
Copyright © 2022, 2026, Oracle and/or its affiliates.

Track Instances in a Project

You can filter and track the status of integration instances through business identifiers. You can also view the activity stream and message payload of an instance you are tracking and replay integration instances configured as replayable.

1. In the navigation pane, click **Projects**.
2. Click the project name or click **Edit** .
3. Click **Observe**.
4. Click **Instances**.

   The status of each integration instance is displayed in a graph. Each potential integration status is identified by a specific color.
5. View a specific status of your integration instances (for this example, **Succeeded**) in either of the following ways:
   a. View the graph to identify if there are successfully completed instances and click that part.
   b. Select **Succeeded** from the **Status** field in the **Filter** icon.



6. Hover over the row of an integration instance and click **View Details** to access the activity stream.
7. Click **Download** to download the logs. The Oracle Cloud identifier (OCID) value of your service instance is also included in the logs.

   When you download logs from the activity stream, this action is visible in the audit logs.

   If you view the message payload data in the activity stream of an integration instance in which the tracing level is set to debug or audit, then the time, your user name, and a message indicating that you viewed message payload data are written to the audit log. If the tracing level is set to production, your user name is not logged to the audit log because message payload data is not visible.
8. Hover over the row of an integration instance and click **Actions** , and then **Run again** to replay REST Adapter-triggered and AS2 Adapter-triggered integration instances configured as replayable. This feature is available both inside and outside of projects. See Capabilities and Restrictions and Understand the Differences Between Replaying and Recovering Instances.

Copyright © 2022, 2026, Oracle and/or its affiliates.

9. Hover over the row of an integration instance and click **Actions** ..., and then **View Child Instances** to display child instances invoked by parent instances.
10. To the right of the time stamp, click **Chart View**.
    a. View time lines for instances grouped by integration. The scale is shown in minutes by default.

       The following icons are displayed:

       - **Bar** icon: Indicates an instance has completed.
       - **Diamond** icon: Indicates an instance is incomplete (for example, has not started or is still running).

    b. Hover over either icon to display details such as instance ID and time stamp.
    c. Click **Seconds** to show the scale in seconds.
    d. Click **Expand** to show where there is overlap between completed instances and incomplete instances.
    e. Click **Collapse**.
11. Click **Back** to return to the **Observe** tab.

See Track Integration Instances.

View Subscribed Instances in a Project

You can monitor the number of integrations that subscribe to events, and their status, on the Subscriptions page in a project.

1. In the navigation pane, click **Projects**.
2. Click the project name or click **Edit**.
3. Click **Observe**.
4. Click **Subscriptions**.

   The Subscriptions page shows the active event subscriptions in the project.
5. Click **Filter** to filter by event subscription integrations or retained events (events published, but not yet received by the event subscription integrations).

Manage Errors in a Project

You can manage errors that occur in integrations in a project.

1. In the navigation pane, click **Projects**.
2. Click the project name or click .
3. Click **Observe**.
4. Click **Instances**.
5. Identify if your project has errors in either of the following ways:
    a. View the graph to identify if there are errors and click that part. Each potential integration status is identified by a specific color.
    b. Select **Errored** from the **Status** field in the **Filter** icon.

Copyright © 2022, 2026, Oracle and/or its affiliates.

The page is refreshed to display tracking information about errors.

6. Hover over the row of an integration instance in error and click **View Details** to access the activity stream.

   You can view error messages and the message payload in the activity stream of failed integration instances. You can also download the activity stream logs and the payload of the failed section of an integration instance. See View Error Messages and the Message Payload in the Activity Stream of Failed Instances.

7. Hover over the row of an integration instance in error and click **Actions** to display other options:

If You Select This Option...	Then ...
Resubmit	Respond when prompted to resubmit the message. See Resubmit Failed Messages.
Abort	Respond when prompted to abort the message. See Discard Errors.
Schedule	The Schedule and future runs page is displayed. See View Future Schedule Integration Runs.
View Child Instances	View the child instances invoked by parent instances.

8. Click **Back** to return to the **Observe** tab.

See Manage Errors.

View the Schedule Integration Runs Calendar in a Project

You can view a calendar of all runs scheduled to trigger in the project in the future, across all schedule integrations whose schedules have started.

1. In the navigation pane, click **Projects**.
2. Click the project name or click .
3. Click **Observe**.
4. Click **Future runs**. See View the Calendar of Schedule Integration Runs.

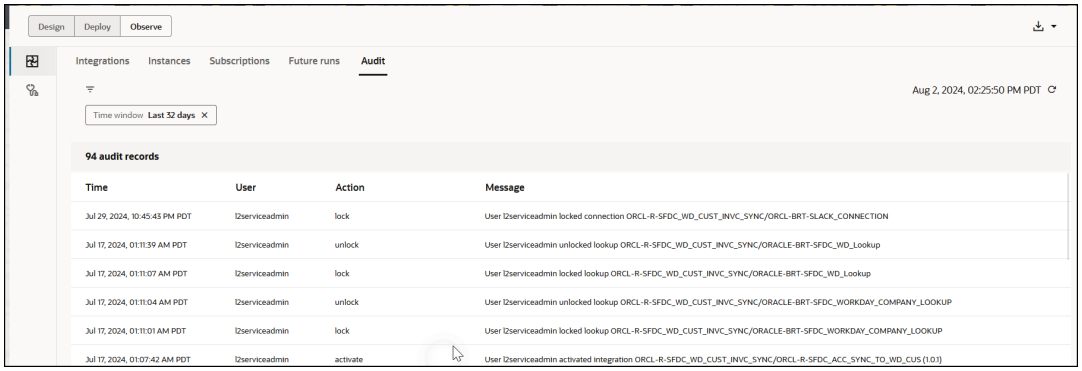
View the Design-Time Audit Log of a Project

You can view and download the design-time audit log of actions performed in a project.

1. In the navigation pane, click **Projects**.
2. Click the project name or click .
3. Click **Observe**.
4. Click **Audit**.

   Each row represents a record. The ZIP file can contain a maximum of 10,000 audit records.

Copyright © 2022, 2026, Oracle and/or its affiliates.



5. Click **Filter** to filter data by time window, integration name, user name, or actions such as activate and deactivate. You can click the **Data range** option to specify a date range that is greater than 32 days. The maximum date range is six months. If you are using projects, the project identifier is also included in the audit logs. Use this identifier to audit instance and error details.
6. Click **Download all** to download a ZIP file of the entire audit log.

   For example:

   ```
   [2023-03-23 06:20:11.554 Z] [userID: myserviceadmin]
   [myserviceadmin,CREATE,ICS_AppInstance,OCI_NATIVE_FUNCTIONS_CONNECTION,OCI_
   NATIVE_FUNCTIONS_CONNECTION,null] User myserviceadmin created Connection
   OCI_NATIVE_FUNCTIONS_CONNECTION
   [2023-03-10 18:05:48.559 Z] [userID: myserviceadmin]
   [myserviceadmin,CREATE,ICS_AppInstance,APPLICATION_EVENT_CONNECTION,APPLICA
   TION_EVENT_CONNECTION,null] User myserviceadmin created Connection
   APPLICATION_EVENT_CONNECTION
   [2023-03-10 18:05:48.349 Z] [userID: myserviceadmin]
   [myserviceadmin,CREATE,ICS_AppInstance,STAGE_FILE_CONNECTION,STAGE_FILE_CON
   NECTION,null] User myserviceadmin created Connection STAGE_FILE_CONNECTION
   [2023-03-10 18:05:48.158 Z] [userID: myserviceadmin]
   [myserviceadmin,CREATE,ICS_AppInstance,PRESEEDED_B2B_TRANSLATOR_CONN_1851,B
   2B,null] User myserviceadmin created Connection B2B
   [2023-03-10 18:05:47.750 Z] [userID: myserviceadmin]
   [myserviceadmin,CREATE,ICS_AppInstance,PRESEEDED_COLLOCATED_CONN_1741,Local
    Integration,null] User myserviceadmin created Connection Local Integration
   ```

7. Click **Download filtered** to download records based on the currently-selected filters that you configured with the **Filter** option (data range, integration name, user name, and action).

Monitor Robots in a Project

You can monitor robot instances directly within a project.

All information about robots is in a separate guide. See Monitor an Automation Solution in *Using Robots in Oracle Integration 3*.

Copyright © 2022, 2026, Oracle and/or its affiliates.

View the Relationship Between Parent and Child Integration Tracking Instances in a Project

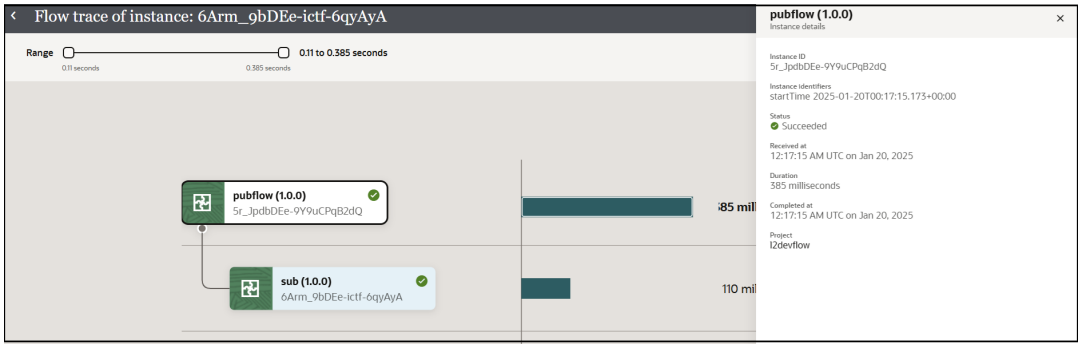
You can view an end-to-end representation of runtime dependencies between parent and child integration tracking instances in a project.

A parent integration can call a child integration, which can then call other child integrations. When the parent integration with a child integration executes, it creates its own tracking instance and also tracking instances for the child integrations, which may create more child integration instances. You can view this hierarchy of integrations to see all parent-child relationships between tracking instances.

> **Note**
>
> This feature is only available in projects.

1. Open the project with runtime tracking instances to view.
2. Click the **Observe** tab, then click the **Instances** tab.
3. Hover over the integration instance to view, then click **Actions** ⋯, and then **View instance hierarchy**.

   For this example, a parent integration (**pubflow**) invoked a child integration (**sub**). Both integration instances completed successfully. The instance IDs for both integrations are displayed in the instance icons. A graph shows the time each integration instance took to complete. The panel on the right shows the instance ID of the parent integration, the instance status, the overall duration of the integration flow, and other details. A slider at the top shows the overall time required for the parent and child integration instances to complete. You can move the slider to view different moments in the overall integration flow.



4. Hover over either integration instance and click **Actions** ⋯ to access additional information about each integration instance:
   - **View activity stream**
   - **View processing time statistics**
   - **View instance details** (displayed by default for the parent integration when you initially selected **View instance hierarchy**)
   - **Copy instance ID**

Copyright © 2022, 2026, Oracle and/or its affiliates.

Monitor MLLP Adapter Connections and Queues

From projects, you can monitor the TCP/IP-based MLLP Adapter connections required for sending and receiving HL7 messages and monitor queues in which HL7 messages are stored to ensure that they are processed in the order received.

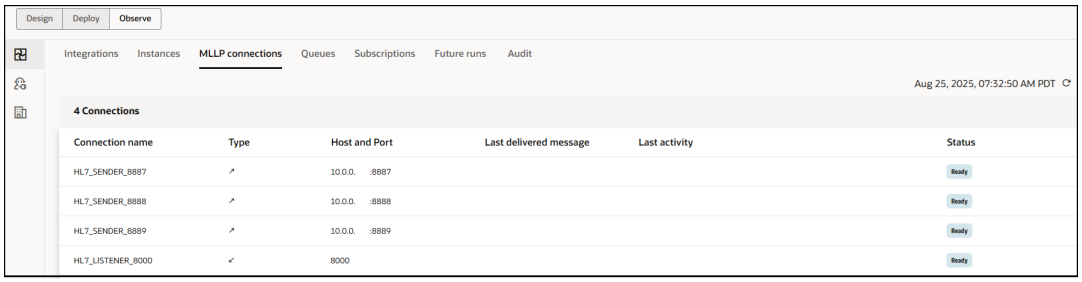
> **Note**
>
> The **MLLP connections** and **Queues** tabs are only visible under the **Observe** tab if you have installed the Healthcare edition. See Create an Oracle Integration Instance in *Provisioning and Administering Oracle Integration 3*.

1. In the navigation pane, click **Projects**.
2. Click the project name or click .
3. Click **Observe** to monitor MLLP connections and queues.
   - Monitor MLLP Adapter Connections
   - Monitor Queues

Monitor MLLP Adapter Connections

You can monitor the status for inbound and outbound TCP/IP-based MLLP Adapter connections from the **Observe** tab in a project. This monitoring helps you identify network or system issues that may require human intervention to restore MLLP Adapter connections.

1. Click **MLLP connections**.
   The **Type** column shows the outbound connection (↗) for sending messages or inbound connection (↙) for listening for incoming messages. The host and port are shown for outbound connections and the listener port is shown for inbound connections. The time at which the message was last delivered or received by this connection is also shown, along with a description of the last activity performed by that connection (for example, a message was sent, a message was received, the network is down, and others). The status TCP/IP socket connection is shown as either ready for use (it has not yet sent or received any messages), is open and in use, or is closed and not in use.



Monitor Queues

To ensure that HL7 messages are delivered to target endpoints in the order in which they are received, Oracle Integration implements ordered message processing. Messages are stored in queues to ensure that they are processed in the order received. You can monitor the status of queued messages at runtime from the **Observe** tab in a project.

Copyright © 2022, 2026, Oracle and/or its affiliates.

1. Click **Queues**.
   See Monitor and Manage Queued Messages at Runtime in *Using Oracle Integration for Healthcare in Oracle Integration 3*.

See Introduction to Oracle Integration for Healthcare.

Copyright © 2022, 2026, Oracle and/or its affiliates.

6

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 and Outbound Endpoints and Transport Layer Security Server Version Support
- Create a Connection
- Create a Connection with an Agent Group
- Configure Connection Properties and Security Properties
- Use OAuth 2.0 in Oracle Integration
- Test the Connection
- Edit a Connection
- Clone a Connection
- Delete a Connection
- Unlock a Connection
- Refresh Integration Metadata
- Manage Security Certificates
- Browse and Install Third-Party Adapters
- View Preinstalled Adapters
- Allow JSON Numbers with High Precision and Scale

Inbound and Outbound Endpoints and Transport Layer Security Server Version Support

All inbound endpoints for integrations in Oracle Integration are hosted on SSL servers that can accept requests coming from clients supporting transport layer security (TLS) 1.2. This is true regardless of whether they are SOAP- or REST-enabled and regardless of the adapter used as the trigger connection. TLS version 1.3 is supported for outbound integrations.

Create a Connection

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

> **Note**
>
> You can also create a connection in the integration canvas. See Define Inbound Triggers, Outbound Invokes, and Actions.

Copyright © 2022, 2026, Oracle and/or its affiliates.

To create a connection in Oracle Integration:

1. Decide where to start:
   - Work in a project (see why working with projects is preferred).
     a. In the navigation pane, click **Projects**.
     b. Select the project name.
     c. Click **Integrations**.
     d. In the **Connections** section, click **Add** if no connections currently exist or **+** if connections already exist. The Create connection panel opens.
   - Work outside a project.
     a. In the navigation pane, click **Design**, then **Connections**.
     b. Click **Create**. The Create connection panel opens.
2. Select the adapter to use for this connection. To find the adapter, scroll through the list, or enter a partial or full name in the **Search** field.
3. Enter the information that describes this connection.

Element	Description
Name	Enter a meaningful name to help others find your connection when they begin to create their own integrations.
Identifier	Automatically displays the name in capital letters that you entered in the **Name** field. If you modify the identifier name, don't include blank spaces (for example, `SALES OPPORTUNITY`).
Role	Select the role (direction) in which to use this connection. **Note**: *Only* the roles supported by the adapter you selected are displayed for selection. Some adapters support all role combinations (trigger, invoke, or trigger and invoke). Other adapters support fewer role combinations. When you select a role, only the connection properties and security policies appropriate to that role are displayed on the Connections page. If you select an adapter that supports both invoke and trigger, but select only one of those roles, you'll get an error when you try to drag the adapter into the section you didn't select. For example, assume you configure a connection for the Oracle Service Cloud (RightNow) Adapter as only an **invoke**. Dragging the adapter to a **trigger** section in the integration produces an error.
Keywords	Enter optional keywords (tags). You can search on the connection keywords on the Connections page.
Description	Enter an optional description of the connection.

Copyright © 2022, 2026, Oracle and/or its affiliates.

Element	Description
Share with other projects	**Note**: This field only appears if you are creating a connection in a project. Select to make this connection publicly available in other projects. Connection sharing eliminates the need to create and maintain separate connections in different projects. When you configure an adapter connection in a different project, the **Use a shared connection** field is displayed at the top of the Connections page. If the connection you are configuring matches the same type and role as the publicly available connection, you can select that connection to reference (inherit) its resources. See Add and Share a Connection Across a Project.

4. Click **Create**.

   Your connection is created. You're now ready to configure the connection properties, security policies, and (for some connections) access type.

5. Follow the steps to configure a connection.
   The connection property and connection security values are specific to each adapter. Your connection may also require configuration with an access type such as a private endpoint or an agent group.

6. Test the connection.

Create a Connection with an Agent Group

After you install the on-premises connectivity 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 Monitor agents page and which are not associated with an adapter can be selected.

1. In the navigation pane, click **Design**, then **Connections**.
2. Click **Create**.

   See Create a Connection.

3. Select the adapter to configure as an invoke (target) endpoint. See About Creating Hybrid Integrations.
4. Configure the connection properties and security properties.
5. In the **Access type** section, select **Connectivity agent**, then click **Associate agent group** 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 processes the ping command on the on-premises instance when the connection is associated with a connectivity agent.
8. Click **Save**.
9. Create an integration in which you use the adapter.
10. Activate the integration. See Activate an Integration.
11. Invoke the integration.

Copyright © 2022, 2026, Oracle and/or its affiliates.

> **Note**
>
> If you receive the following error, a connection timeout has occurred. The request may be slow. In that case, process the request 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.
> ```

Configure Connection Properties and Security Properties

Enter connection and security property information for the adapter.

Configure Connection Properties

Enter connection information so your application can process requests.

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

Adapter	For Information
Adobe Sign Adapter	*Using the Adobe Sign Adapter with Oracle Integration 3*
Anthropic Adapter	*Using the Anthropic Adapter with Oracle Integration 3*
Apache Kafka Adapter	*Using the Apache Kafka Adapter with Oracle Integration 3*
ArcGIS (ESRI) Adapter	*Using the ArcGIS (ESRI) Adapter with Oracle Integration 3*
Asana Adapter	N/A
AS2 Adapter	*Using the AS2 Adapter with Oracle Integration 3*
AS4 Adapter	*Using the AS4 Adapter with Oracle Integration 3*
Amazon Simple Notification Service (SNS) Adapter	*Using the Amazon Simple Notification Service (SNS) Adapter with Oracle Integration 3*
Amazon Simple Queue Service (SQS) Adapter	*Using the Amazon Simple Queue Service (SQS) Adapter with Oracle Integration 3*
Amazon Simple Storage Service (S3) Adapter	*Using the Amazon Simple Storage Service (S3) Adapter with Oracle Integration 3*
Microsoft Entra ID Adapter	*Using the Microsoft Entra ID Adapter with Oracle Integration 3*
Azure Event Grid Adapter	*Using the Azure Event Grid Adapter with Oracle Integration 3*
Azure Service Bus Adapter	*Using the Azure Service Bus Adapter with Oracle Integration 3*
Azure Storage Adapter	*Using the Azure Storage Adapter with Oracle Integration 3*
Box Adapter	N/A

Copyright © 2022, 2026, Oracle and/or its affiliates.

Adapter	For Information
Confluent Adapter	*Using the Confluent Adapter with Oracle Integration 3*
Coupa Procurement Adapter	*Using the Coupa Procurement Adapter with Oracle Integration 3*
Coupa Sourcing Adapter	*Using the Coupa Sourcing Adapter with Oracle Integration 3*
DocuSign Adapter	*Using the DocuSign Adapter with Oracle Integration 3*
eBay Marketplace Adapter	*Using the eBay Marketplace Adapter with Oracle Integration 3*
Eventbrite Adapter	*Using the Eventbrite Adapter with Oracle Integration 3*
Facebook Adapter	*Using the Facebook Adapter with Oracle Integration 3*
FHIR Adapter	*Using the FHIR Adapter with Oracle Integration 3*
File Adapter	*Using the File Adapter with Oracle Integration 3*
FTP Adapter	*Using the FTP Adapter with Oracle Integration 3*
GCP Pub Sub Adapter	*Using the GCP Pub Sub Adapter with Oracle Integration 3*
GCP Storage Adapter	*Using the GCP Storage Adapter with Oracle Integration 3*
GitHub Adapter	N/A
GraphQL Adapter	*Using the GraphQL Adapter with Oracle Integration 3*
Google Gmail Adapter	*Using the Google Gmail Adapter with Oracle Integration 3*
Google Calendar Adapter	*Using the Google Calendar Adapter with Oracle Integration 3*
Google Tasks Adapter	*Using the Google Tasks Adapter with Oracle Integration 3*
HubSpot Adapter	*Using the HubSpot Adapter with Oracle Integration 3*
IBM DB2 Adapter	*Using the IBM DB2 Adapter with Oracle Integration 3*
IBM MQ Series JMS Adapter	*Using the IBM MQ Series JMS Adapter with Oracle Integration 3*
Jira Adapter	*Using the Jira Adapter with Oracle Integration 3*
LinkedIn Adapter	*Using the LinkedIn Adapter with Oracle Integration 3*
Magento Commerce Cloud Adapter	*Using the Magento Commerce Cloud Adapter with Oracle Integration 3*
Mailchimp Adapter	*Using the Mailchimp Adapter with Oracle Integration 3*
Marketo Adapter	*Using the Marketo Adapter with Oracle Integration 3*
Microsoft Office 365 Calendar Adapter	*Using the Microsoft Office 365 Calendar Adapter with Oracle Integration 3*
Microsoft Office 365 People Adapter	*Using the Microsoft Office 365 People Adapter with Oracle Integration 3*
Microsoft Office 365 Outlook Adapter	*Using the Microsoft Office 365 Outlook Adapter with Oracle Integration 3*

Copyright © 2022, 2026, Oracle and/or its affiliates.

Adapter	For Information
Microsoft SharePoint Adapter	*Using the Microsoft SharePoint Adapter with Oracle Integration 3*
Microsoft SQL Server Adapter	*Using the Microsoft SQL Server Adapter with Oracle Integration 3*
MLLP Adapter	*Using the MLLP Adapter with Oracle Integration 3*
Microsoft Dynamics 365 CRM Adapter	*Using the Microsoft Dynamics 365 CRM Adapter with Oracle Integration 3*
MySQL Adapter	*Using the MySQL Adapter with Oracle Integration 3*
Netezza Adapter	*Using the Netezza Adapter with Oracle Integration 3*
OData Adapter	*Using the OData Adapter with Oracle Integration 3*
OpenSearch Adapter	*Using the OpenSearch Adapter with Oracle Integration 3*
Oracle Aconex Cloud Adapter	*Using the Oracle Aconex Cloud Adapter with Oracle Integration 3*
Oracle Advanced Queuing (AQ) Adapter	*Using the Oracle Advanced Queuing (AQ) Adapter with Oracle Integration 3*
Oracle Advanced Queuing (AQ) JMS Adapter	*Using the Oracle Advanced Queuing (AQ) JMS Adapter with Oracle Integration 3*
Oracle Autonomous Data Warehouse Adapter	*Using the Oracle Autonomous Data Warehouse Adapter with Oracle Integration 3*
Oracle Autonomous Transaction Processing Adapter	*Using the Oracle Autonomous Transaction Processing Adapter with Oracle Integration 3*
Oracle Cloud Infrastructure Streaming Service Adapter	*Using the Oracle Cloud Infrastructure Streaming Service Adapter with Oracle Integration 3*
Oracle Commerce Cloud Adapter	*Using the Oracle Commerce Cloud Adapter with Oracle Integration 3*
Oracle CPQ Adapter	*Using the Oracle CPQ Adapter with Oracle Integration 3*
Oracle Database Adapter	*Using the Oracle Database Adapter with Oracle Integration 3*
Oracle Database Cloud Service Adapter	*Using the Oracle Database Cloud Service Adapter with Oracle Integration 3*
Oracle E-Business Suite Adapter	*Using the Oracle E-Business Suite Adapter with Oracle Integration 3*
Oracle Eloqua Cloud Adapter	N/A
Oracle CX Sales and B2B Service Adapter	*Using the Oracle CX Sales and B2B Service Adapter with Oracle Integration 3*
Oracle Enterprise Performance Management Cloud Adapter	*Using the Oracle Enterprise Performance Management Cloud Adapter with Oracle Integration 3*
Oracle ERP Cloud Adapter	*Using the Oracle ERP Cloud Adapter with Oracle Integration 3*
Oracle Fusion Field Service Adapter	*Using the Oracle Fusion Field Service Adapter with Oracle Integration 3*
Oracle HCM Cloud Adapter	*Using the Oracle HCM Cloud Adapter with Oracle Integration 3*
Oracle Health Inventory Management Adapter	*Using the Oracle Health Inventory Management Adapter with Oracle Integration 3*

Copyright © 2022, 2026, Oracle and/or its affiliates.

Adapter	For Information
Oracle Hospitality Adapter	*Using the Oracle Hospitality Adapter with Oracle Integration 3*
Oracle Intelligent Advisor Adapter	*Using the Oracle Intelligent Advisor Adapter with Oracle Integration 3*
Oracle Intelligent Track and Trace Adapter	*Using the Oracle Intelligent Track and Trace Adapter with Oracle Integration 3*
Oracle JD Edwards EnterpriseOne Adapter	*Using the Oracle JD Edwards EnterpriseOne Adapter with Oracle Integration 3*
Oracle Logistics Adapter	*Using the Oracle Logistics Adapter with Oracle Integration 3*
Oracle NetSuite Adapter	*Using the Oracle NetSuite Adapter with Oracle Integration 3*
Oracle NetSuite RST Adapter	*Using the Oracle NetSuite RST Adapter with Oracle Integration 3*
Oracle Primavera Cloud Adapter	*Using the Oracle Primavera Cloud Adapter with Oracle Integration 3*
Oracle Primavera Cloud Data Service Adapter	*Using the Oracle Primavera Cloud Data Service Adapter with Oracle Integration 3*
Oracle Primavera P6 EPPM Adapter	*Using the Oracle Primavera P6 EPPM Adapter with Oracle Integration 3*
Oracle Responsys Adapter	*Using the Oracle Responsys Adapter with Oracle Integration 3*
Oracle Service Cloud (RightNow) Adapter	*Using the Oracle Service Cloud (RightNow) Adapter with Oracle Integration 3*
Oracle Siebel Adapter	*Using the Oracle Siebel Adapter with Oracle Integration 3*
Oracle SOA Suite Adapter	*Using the Oracle SOA Suite Adapter with Oracle Integration 3*
Oracle Taleo Enterprise Edition Adapter	*Using the Oracle Taleo Enterprise Edition Adapter with Oracle Integration 3*
Oracle Unity Adapter	*Using the Oracle Unity Adapter with Oracle Integration*
Oracle Utilities Adapter	*Using the Oracle Utilities Adapter with Oracle Integration 3*
Oracle WebLogic JMS Adapter	*Using the Oracle WebLogic JMS Adapter with Oracle Integration 3*
PayPal Adapter	*Using the PayPal Adapter with Oracle Integration 3*
PostgreSQL Adapter	*Using the PostgreSQL Adapter with Oracle Integration 3*
Primavera Unifier Adapter	*Using the Oracle Primavera Unifier Adapter with Oracle Integration 3*
RabbitMQ Adapter	*Using the RabbitMQ Adapter with Oracle Integration 3*
REST Adapter	*Using the REST Adapter with Oracle Integration 3*
RosettaNet Adapter	*Using the RosettaNet Adapter with Oracle Integration 3*
Salesforce Adapter	*Using the Salesforce Adapter with Oracle Integration 3*
Salesforce Commerce Cloud Adapter	*Using the Salesforce Commerce Cloud Adapter with Oracle Integration 3*

Copyright © 2022, 2026, Oracle and/or its affiliates.

Adapter	For Information
Salesforce Marketing Cloud Adapter	*Using the Salesforce Marketing Cloud Adapter with Oracle Integration 3*
Salesforce REST Adapter	*Using the Salesforce REST Adapter with Oracle Integration 3*
SAP Adapter	*Using the SAP Adapter with Oracle Integration 3*
SAP Ariba Adapter	*Using the SAP Ariba Adapter with Oracle Integration 3*
SAP ASE (Sybase) Adapter	*Using the SAP ASE (Sybase) Adapter with Oracle Integration 3*
SAP Commerce Cloud (Hybris) Adapter	*Using the SAP Commerce Cloud (Hybris) Adapter with Oracle Integration 3*
SAP Concur Adapter	*Using the SAP Concur Adapter with Oracle Integration 3*
SAP SuccessFactors Adapter	*Using the SAP SuccessFactors Adapter with Oracle Integration 3*
SAP S/4HANA Cloud Adapter	*Using the SAP S/4HANA Cloud Adapter with Oracle Integration 3*
ServiceNow Adapter	*Using the ServiceNow Adapter with Oracle Integration 3*
Shopify Adapter	*Using the Shopify Adapter with Oracle Integration 3*
Shopify GraphQL Adapter	*Using the Shopify GraphQL Adapter with Oracle Integration 3*
Slack Adapter	N/A
Snowflake Adapter	*Using the Snowflake Adapter with Oracle Integration 3*
SOAP Adapter	*Using the SOAP Adapter with Oracle Integration 3*
SugarCRM Adapter	*Using the SugarCRM Adapter with Oracle Integration 3*
SurveyMonkey Adapter	*Using the SurveyMonkey Adapter with Oracle Integration 3*
Trello Adapter	*Using the Trello Adapter with Oracle Integration 3*
Twilio Adapter	*Using the Twilio Adapter with Oracle Integration 3*
Twitter Adapter	*Using the Twitter Adapter with Oracle Integration 3*
Workday Adapter	*Using the Workday Adapter with Oracle Integration 3*
Zendesk Adapter	*Using the Zendesk Adapter with Oracle Integration 3*
Zuora Adapter	*Using the Zuora Adapter with Oracle Integration 3*

You are now ready to configure connection security.

Configure Connection Security

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

1. Go to the **Security** section.

Copyright © 2022, 2026, Oracle and/or its affiliates.

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

Adapter	For Information
Adobe Sign Adapter	*Using the Adobe Sign Adapter with Oracle Integration 3*
Anthropic Adapter	*Using the Anthropic Adapter with Oracle Integration 3*
Apache Kafka Adapter	*Using the Apache Kafka Adapter with Oracle Integration 3*
ArcGIS (ESRI) Adapter	*Using the ArcGIS (ESRI) Adapter with Oracle Integration 3*
Asana Adapter	*Using the Asana Adapter with Oracle Integration 3*
AS2 Adapter	*Using the AS2 Adapter with Oracle Integration 3*
AS4 Adapter	*Using the AS4 Adapter with Oracle Integration 3*
Amazon Simple Notification Service (SNS) Adapter	*Using the Amazon Simple Notification Service (SNS) Adapter with Oracle Integration 3*
Amazon Simple Queue Service (SQS) Adapter	*Using the Amazon Simple Queue Service (SQS) Adapter with Oracle Integration 3*
Amazon Simple Storage Service (S3) Adapter	*Using the Amazon Simple Storage Service (S3) Adapter with Oracle Integration 3*
Microsoft Entra ID Adapter	*Using the Microsoft Entra ID Adapter with Oracle Integration 3*
Azure Event Grid Adapter	*Using the Azure Event Grid Adapter with Oracle Integration 3*
Azure Service Bus Adapter	*Using the Azure Service Bus Adapter with Oracle Integration 3*
Azure Storage Adapter	*Using the Azure Storage Adapter with Oracle Integration 3*
Box Adapter	*Using the Box Adapter with Oracle Integration 3*
Confluent Adapter	*Using the Confluent Adapter with Oracle Integration 3*
Coupa Procurement Adapter	*Using the Coupa Procurement Adapter with Oracle Integration 3*
Coupa Sourcing Adapter	*Using the Coupa Sourcing Adapter with Oracle Integration 3*
DocuSign Adapter	*Using the DocuSign Adapter with Oracle Integration 3*
eBay Marketplace Adapter	*Using the eBay Marketplace Adapter with Oracle Integration 3*
Eventbrite Adapter	*Using the Eventbrite Adapter with Oracle Integration 3*
Facebook Adapter	*Using the Facebook Adapter with Oracle Integration 3*
FHIR Adapter	*Using the FHIR Adapter with Oracle Integration 3*
File Adapter	NA
FTP Adapter	*Using the FTP Adapter with Oracle Integration 3*
GCP Pub Sub Adapter	*Using the GCP Pub Sub Adapter with Oracle Integration 3*
GCP Storage Adapter	*Using the GCP Storage Adapter with Oracle Integration 3*

Copyright © 2022, 2026, Oracle and/or its affiliates.

Adapter	For Information
GitHub Adapter	*Using the GitHub Adapter with Oracle Integration 3*
GraphQL Adapter	*Using the GraphQL Adapter with Oracle Integration 3*
Google Gmail Adapter	*Using the Google Gmail Adapter with Oracle Integration 3*
Google Calendar Adapter	*Using the Google Calendar Adapter with Oracle Integration 3*
Google Tasks Adapter	*Using the Google Tasks Adapter with Oracle Integration 3*
HubSpot Adapter	*Using the HubSpot Adapter with Oracle Integration 3*
IBM DB2 Adapter	*Using the IBM DB2 Adapter with Oracle Integration 3*
IBM MQ Series JMS Adapter	*Using the IBM MQ Series JMS Adapter with Oracle Integration 3*
Jira Adapter	*Using the Jira Adapter with Oracle Integration 3*
LinkedIn Adapter	*Using the LinkedIn Adapter with Oracle Integration 3*
Magento Commerce Cloud Adapter	*Using the Magento Commerce Cloud Adapter with Oracle Integration 3*
Mailchimp Adapter	*Using the Mailchimp Adapter with Oracle Integration 3*
Marketo Adapter	*Using the Marketo Adapter with Oracle Integration 3*
Microsoft Office 365 Calendar Adapter	*Using the Microsoft Office 365 Calendar Adapter with Oracle Integration 3*
Microsoft Office 365 People Adapter	*Using the Microsoft Office 365 People Adapter with Oracle Integration 3*
Microsoft Office 365 Outlook Adapter	*Using the Microsoft Office 365 Outlook Adapter with Oracle Integration 3*
Microsoft SharePoint Adapter	*Using the Microsoft SharePoint Adapter with Oracle Integration 3*
Microsoft SQL Server Adapter	*Using the Microsoft SQL Server Adapter with Oracle Integration 3*
MLLP Adapter	*Using the MLLP Adapter with Oracle Integration 3*
Microsoft Dynamics 365 CRM Adapter	*Using the Microsoft Dynamics 365 CRM Adapter with Oracle Integration 3*
MySQL Adapter	*Using the MySQL Adapter with Oracle Integration 3*
Netezza Adapter	*Using the Netezza Adapter with Oracle Integration 3*
OData Adapter	*Using the OData Adapter with Oracle Integration 3*
OpenAI Adapter	*Using the OpenAI Adapter with Orecle Integration 3*
OpenSearch Adapter	*Using the OpenSearch Adapter with Oracle Integration 3*
Oracle Aconex Cloud Adapter	*Using the Oracle Aconex Cloud Adapter with Oracle Integration 3*
Oracle Advanced Queuing (AQ) Adapter	*Using the Oracle Advanced Queuing (AQ) Adapter with Oracle Integration 3*

Copyright © 2022, 2026, Oracle and/or its affiliates.

Adapter	For Information
Oracle Advanced Queuing (AQ) JMS Adapter	*Using the Oracle Advanced Queuing (AQ) JMS Adapter with Oracle Integration 3*
Oracle Autonomous Data Warehouse Adapter	*Using the Oracle Autonomous Data Warehouse Adapter with Oracle Integration 3*
Oracle Autonomous Transaction Processing Adapter	*Using the Oracle Autonomous Transaction Processing Adapter with Oracle Integration 3*
Oracle Cloud Infrastructure Streaming Service Adapter	*Using the Oracle Cloud Infrastructure Streaming Service Adapter with Oracle Integration 3*
Oracle Commerce Cloud Adapter	*Using the Oracle Commerce Cloud Adapter with Oracle Integration 3*
Oracle CPQ Adapter	*Using the Oracle CPQ Adapter with Oracle Integration 3*
Oracle Database Adapter	*Using the Oracle Database Adapter with Oracle Integration 3*
Oracle Database Cloud Service Adapter	*Using the Oracle Database Cloud Service Adapter with Oracle Integration 3*
Oracle E-Business Suite Adapter	*Using the Oracle E-Business Suite Adapter with Oracle Integration 3*
Oracle Eloqua Cloud Adapter	*Using the Oracle Eloqua Cloud Adapter with Oracle Integration 3*
Oracle CX Sales and B2B Service Adapter	*Using the Oracle CX Sales and B2B Service Adapter with Oracle Integration 3*
Oracle Enterprise Performance Management Cloud Adapter	*Using the Oracle Enterprise Performance Management Cloud Adapter with Oracle Integration 3*
Oracle ERP Cloud Adapter	*Using the Oracle ERP Cloud Adapter with Oracle Integration 3*
Oracle Fusion Field Service Adapter	*Using the Oracle Fusion Field Service Adapter with Oracle Integration 3*
Oracle HCM Cloud Adapter	*Using the Oracle HCM Cloud Adapter with Oracle Integration 3*
Oracle Health Inventory Management Adapter	*Using the Oracle Health Inventory Management Adapter with Oracle Integration 3*
Oracle Hospitality Adapter	*Using the Oracle Hospitality Adapter with Oracle Integration 3*
Oracle Intelligent Advisor Adapter	*Using the Oracle Intelligent Advisor Adapter with Oracle Integration 3*
Oracle Intelligent Track and Trace Adapter	*Using the Oracle Intelligent Track and Trace Adapter with Oracle Integration 3*
Oracle JD Edwards EnterpriseOne Adapter	*Using the Oracle JD Edwards EnterpriseOne Adapter with Oracle Integration 3*
Oracle Logistics Adapter	*Using the Oracle Logistics Adapter with Oracle Integration 3*
Oracle NetSuite Adapter	*Using the Oracle NetSuite Adapter with Oracle Integration 3*
Oracle NetSuite RST Adapter	*Using the Oracle NetSuite RST Adapter with Oracle Integration 3*
Oracle Primavera Cloud Adapter	*Using the Oracle Primavera Cloud Adapter with Oracle Integration 3*
Oracle Primavera Cloud Data Service Adapter	*Using the Oracle Primavera Cloud Data Service Adapter with Oracle Integration 3*
Oracle Primavera P6 EPPM Adapter	*Using the Oracle Primavera P6 EPPM Adapter with Oracle Integration 3*

Copyright © 2022, 2026, Oracle and/or its affiliates.

Adapter	For Information
Oracle Responsys Adapter	*Using the Oracle Responsys Adapter with Oracle Integration 3*
Oracle Service Cloud (RightNow) Adapter	*Using the Oracle Service Cloud (RightNow) Adapter with Oracle Integration 3*
Oracle Siebel Adapter	*Using the Oracle Siebel Adapter with Oracle Integration 3*
Oracle SOA Suite Adapter	*Using the Oracle SOA Suite Adapter with Oracle Integration 3*
Oracle Taleo Enterprise Edition Adapter	*Using the Oracle Taleo Enterprise Edition Adapter with Oracle Integration 3*
Oracle Unity Adapter	*Using the Oracle Unity Adapter with Oracle Integration*
Oracle Utilities Adapter	*Using the Oracle Utilities Adapter with Oracle Integration 3*
Oracle WebLogic JMS Adapter	*Using the Oracle WebLogic JMS Adapter with Oracle Integration 3*
PayPal Adapter	*Using the PayPal Adapter with Oracle Integration 3*
PostgreSQL Adapter	*Using the PostgreSQL Adapter with Oracle Integration 3*
Primavera Unifier Adapter	*Using the Oracle Primavera Unifier Adapter with Oracle Integration 3*
RabbitMQ Adapter	*Using the RabbitMQ Adapter with Oracle Integration 3*
REST Adapter	*Using the REST Adapter with Oracle Integration 3*
RosettaNet Adapter	*Using the RosettaNet Adapter with Oracle Integration 3*
Salesforce Adapter	*Using the Salesforce Adapter with Oracle Integration 3*
Salesforce Commerce Cloud Adapter	*Using the Salesforce Commerce Cloud Adapter with Oracle Integration 3*
Salesforce Marketing Cloud Adapter	*Using the Salesforce Marketing Cloud Adapter with Oracle Integration 3*
Salesforce REST Adapter	*Using the Salesforce REST Adapter with Oracle Integration 3*
SAP Adapter	*Using the SAP Adapter with Oracle Integration 3*
SAP ASE (Sybase) Adapter	*Using the SAP ASE (Sybase) Adapter with Oracle Integration 3*
SAP Ariba Adapter	*Using the SAP Ariba Adapter with Oracle Integration 3*
SAP Commerce Cloud (Hybris) Adapter	*Using the SAP Commerce Cloud (Hybris) Adapter with Oracle Integration 3*
SAP Concur Adapter	*Using the SAP Concur Adapter with Oracle Integration 3*
SAP SuccessFactors Adapter	*Using the SAP SuccessFactors Adapter with Oracle Integration 3*
SAP S/4HANA Cloud Adapter	*Using the SAP S/4HANA Cloud Adapter with Oracle Integration 3*
ServiceNow Adapter	*Using the ServiceNow Adapter with Oracle Integration 3*
Shopify Adapter	*Using the Shopify Adapter with Oracle Integration 3*

Copyright © 2022, 2026, Oracle and/or its affiliates.

Adapter	For Information
Shopify GraphQL Adapter	*Using the Shopify GraphQL Adapter with Oracle Integration 3*
Slack Adapter	*Using the Slack Adapter with Oracle Integration 3*
Snowflake Adapter	*Using the Snowflake Adapter with Oracle Integration 3*
SOAP Adapter	*Using the SOAP Adapter with Oracle Integration 3*
SugarCRM Adapter	*Using the SugarCRM Adapter with Oracle Integration 3*
SurveyMonkey Adapter	*Using the SurveyMonkey Adapter with Oracle Integration 3*
Trello Adapter	*Using the Trello Adapter with Oracle Integration 3*
Twilio Adapter	*Using the Twilio Adapter with Oracle Integration 3*
Twitter Adapter	*Using the Twitter Adapter with Oracle Integration 3*
Workday Adapter	*Using the Workday Adapter with Oracle Integration 3*
Zendesk Adapter	*Using the Zendesk Adapter with Oracle Integration 3*
Zuora Adapter	*Using the Zuora Adapter with Oracle Integration 3*

3. Click **Save**.

   For most adapters, you are now ready to test your connection. For some adapters, you must first configure an agent group. See Create a Connection with an Agent Group.

Use OAuth 2.0 in Oracle Integration

When using Oracle Integration, many applications to which you need to connect require the use of OAuth. OAuth provides you with secure access to data in an application without having to directly share login credentials.

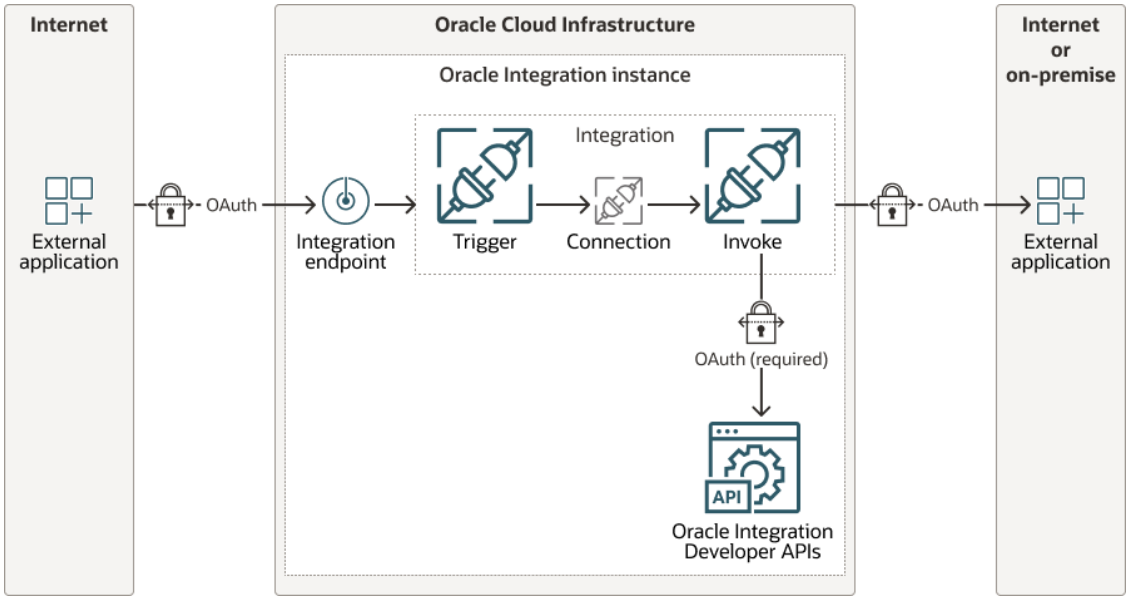
- Where Can You Use OAuth in Oracle Integration?
- Use OAuth with Inbound Calls to Oracle Integration
- Use OAuth with Outbound Calls from Oracle Integration
- You Must Use OAuth with the Oracle Integration Developer APIs
- Troubleshoot OAuth
- FAQs for the Confidential Client Application

Where Can You Use OAuth in Oracle Integration?

You can use OAuth when making inbound calls to Oracle Integration, when making outbound calls from Oracle Integration, or when using the Oracle Integration Developer APIs. OAuth is required to call the Oracle Integration Developer APIs.

The Oracle Integration Developer APIs are: REST API for Oracle Integration 3, REST API for File Server in Oracle Integration 3, OCI Process Automation REST API.

Copyright © 2022, 2026, Oracle and/or its affiliates.



OAuth with Inbound Calls to Oracle Integration

You can use OAuth to make inbound calls to Oracle Integration and trigger integrations.

For inbound calls to Oracle Integration, you use a trigger connection in an integration and the REST Adapter, SOAP Adapter, Fusion Application adapters (Oracle ERP Cloud Adapter, Oracle CX Sales and B2B Service Adapter, Oracle HCM Cloud Adapter), and any other adapters that support trigger connections and OAuth.

For configuration details, see Use OAuth with Inbound Calls to Oracle Integration.

OAuth with Outbound Calls from Oracle Integration

You can also use OAuth to connect to external systems and call external applications.

For these outbound calls, you use an invoke connection in an integration. Whether OAuth is required or not depends on the external application requirements. You can use any adapters that support invoke connections and OAuth including the REST Adapter, SOAP Adapter, Fusion Application adapters (Oracle ERP Cloud Adapter, Oracle CX Sales and B2B Service Adapter, Oracle HCM Cloud Adapter),

For configuration details, see Use OAuth with Outbound Calls from Oracle Integration.

OAuth with the Oracle Integration Developer APIs

You can also use OAuth with an invoke connection in an integration to call the Oracle Integration Developer APIs.

OAuth is required to call the Oracle Integration Developer APIs.

For details, see You Must Use OAuth with the Oracle Integration Developer APIs.

Copyright © 2022, 2026, Oracle and/or its affiliates.

Use OAuth with Inbound Calls to Oracle Integration

You make inbound calls to Oracle Integration through a trigger connection in an integration. When you use a trigger connection in an integration, you are exposing the endpoint. You can use OAuth to protect the exposed endpoints in an integration.

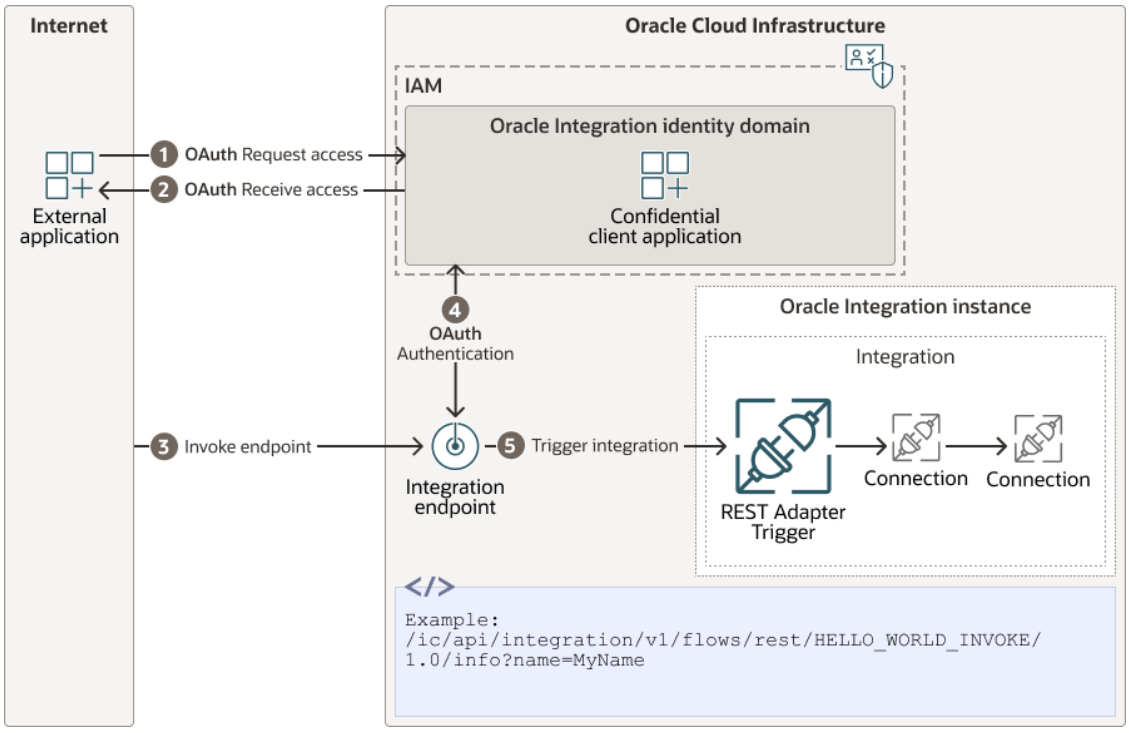
Topics

- Protect an Integration's REST Endpoint with OAuth
- Protect an Integration's SOAP Endpoint with OAuth
- Protect Application-Specific Endpoints with OAuth

Protect an Integration's REST Endpoint with OAuth

Here's a summary of what you need to configure to use OAuth with REST trigger connections in an integration.

When you use OAuth with a trigger connection, you protect the exposed endpoint. You create a confidential client application in Oracle Cloud Infrastructure Identity and Access Management (IAM) with the required roles and scopes. The external application or user requests access from the OCI IAM authorization server. Once access is received, the external application or user authenticates and accesses the integration's endpoint to trigger the integration.



Copyright © 2022, 2026, Oracle and/or its affiliates.

Configuration Summary

Is OAuth required?	Supported grant types	Required role for confidential client application
No. you can protect integration's endpoint with Basic Authorization or OAuth. See About Requests to Invoke Integrations for details.	• JWT User Assertion • Authorization code • Client credentials • Resource Owner Password Credentials (not recommended) See About OAuth 2.0 Grants for more details on each authentication type and which one to use.	Assign the confidential client application the **ServiceInvoker** role. For detailed information on each role, see What Users Can Do in the Integrations Design Section by Role.

Configuration Steps

You must be the OCI tenant and domain administrator to configure the confidential client application and assign roles.

- FAQs for the Confidential Client Application

Steps to use OAuth with the REST Adapter:

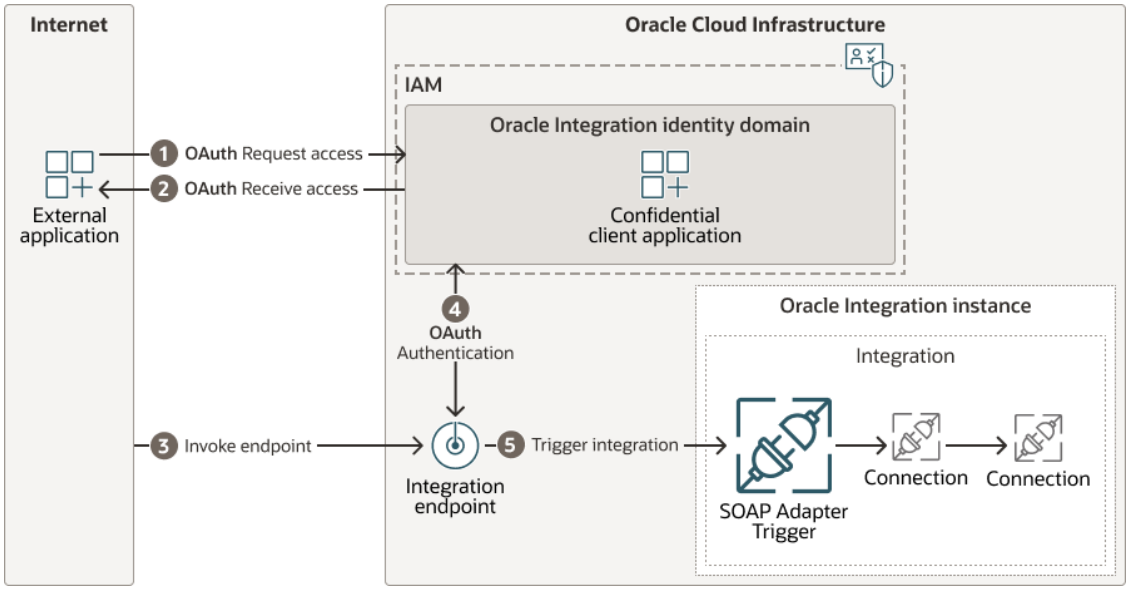
1. Create the confidential client application, assign scopes and roles, and activate it:
   a. Access the Identity Domain.
   b. Configure prerequisites for your grant type:
      - Prerequisites for Authorization Code
      - Prerequisites for Client Credentials and Resource Owner Password Credentials
      - Prerequisites for JWT User Assertion
2. Create a REST Adapter Connection.
3. Configure the REST Adapter to Expose an Integration as a REST API.

Protect an Integration's SOAP Endpoint with OAuth

Here's a summary of what you need to configure to use OAuth with SOAP trigger connections in an integration.

When you use OAuth with a trigger connection, you protect the exposed endpoint. You create a confidential client application in Oracle Cloud Infrastructure Identity and Access Management (IAM) with the required roles and scopes. The external application or user requests access from the OCI IAM authorization server. Once access is received, the external application or user authenticates and accesses the integration's endpoint to trigger the integration.

Copyright © 2022, 2026, Oracle and/or its affiliates.



Configuration Summary

Is OAuth required?	Supported grant types	Required role for confidential client application
No. you can protect integration's endpoint with Basic Authorization or OAuth. See About Requests to Invoke Integrations for details.	• JWT User Assertion • Authorization code • Client credentials • Resource Owner Password Credentials (not recommended) See About OAuth 2.0 Grants for more details on each grant type and which one to use.	Assign the confidential client application the **ServiceInvoker** role. For detailed information on each role, see What Users Can Do in the Integrations Design Section by Role.

Configuration Steps

You must be the OCI tenant and domain administrator to configure the confidential client application and assign roles.

- FAQs for the Confidential Client Application

Steps to use OAuth with the SOAP Adapter:

1. Create the confidential client application, assign scopes and roles, and activate it:
   a. Access the Identity Domain.
   b. Configure prerequisites for your grant type:
      - Prerequisites for Authorization Code
      - Prerequisites for Client Credentials and Resource Owner Password Credentials
      - Prerequisites for JWT User Assertion
2. Create a SOAP Adapter connection.
3. Add the SOAP Adapter connection to an integration.

Copyright © 2022, 2026, Oracle and/or its affiliates.

4. The steps to expose an integration as a SOAP API are similar to that of the REST Adapter. Follow the steps in Configure the REST Adapter to Expose an Integration as a REST API.

Protect Application-Specific Endpoints with OAuth

You can use OAuth to protect the exposed endpoint of any application that supports trigger connections and OAuth.

Take a look at the adapter documentation for your specific application to find out how OAuth is supported. See Configure Connection Security for links to security configuration prerequisites and steps for each adapter.

Use OAuth with Outbound Calls from Oracle Integration

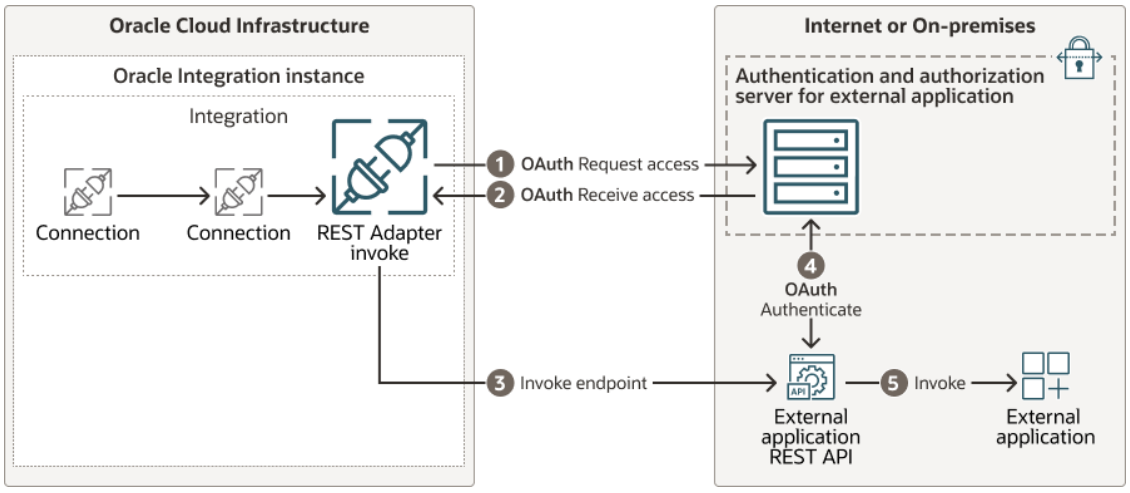
You make outbound calls from Oracle Integration through an invoke connection in an integration. You can use OAuth to securely invoke the API of an external application.

- Invoke an Application's REST Endpoint with OAuth
- Invoke an Application's SOAP Endpoint with OAuth
- Invoke Oracle Fusion Application Endpoints with OAuth
- Invoke Application-Specific Endpoints with OAuth

Invoke an Application's REST Endpoint with OAuth

When you configure an invoke connection in an integration, you can use OAuth to securely invoke the REST API of an external application. Here's a summary of what you need to configure to use OAuth with REST invoke connections in an integration.

When an integration invokes the endpoint of an external application, Oracle Integration requests access from the authorization server for the external application. Once access is granted, Oracle Integration accesses the REST endpoint, authenticates with the external authorization server, and invokes the application.



Copyright © 2022, 2026, Oracle and/or its affiliates.

Configuration Summary

Is OAuth required?	Supported grant types	Required roles
No. You can use other authentication types supported by the external application.	For supported grant types for REST Adapter invoke connections, see Authentication Types.	The role required depends on the endpoint that you are calling. If you are calling an Oracle Integration endpoint, such as an Oracle Integration Developer API or the endpoint within an integration, you will need to assign Oracle Integration roles to the client application such as ServiceInvoker, ServiceUser, or ServiceDeveloper, • **ServiceUser** is the minimum role required to call the Developer APIs. This user has access to the Oracle Integration Console as well as the APIs. Assign the ServiceUser role in most cases. • **ServiceDeveloper** is required if you're invoking the Developer APIs from within an integration or you're using the APIs to perform create operations such as creating lookups or importing an integration. • Use **ServiceInvoker** if you only need to invoke an integration's endpoints and will not be making any calls to the Oracle Integration Developer APIs. For detailed information on each role, see What Users Can Do in the Integrations Design Section by Role. If you want to use the same confidential application for calling the Oracle Integration Developer APIs and integration endpoints, use the ServiceUser or ServiceDeveloper role. If you are calling an external endpoint, the role you assign depends on the external application requirements.

Copyright © 2022, 2026, Oracle and/or its affiliates.

Configuration Steps

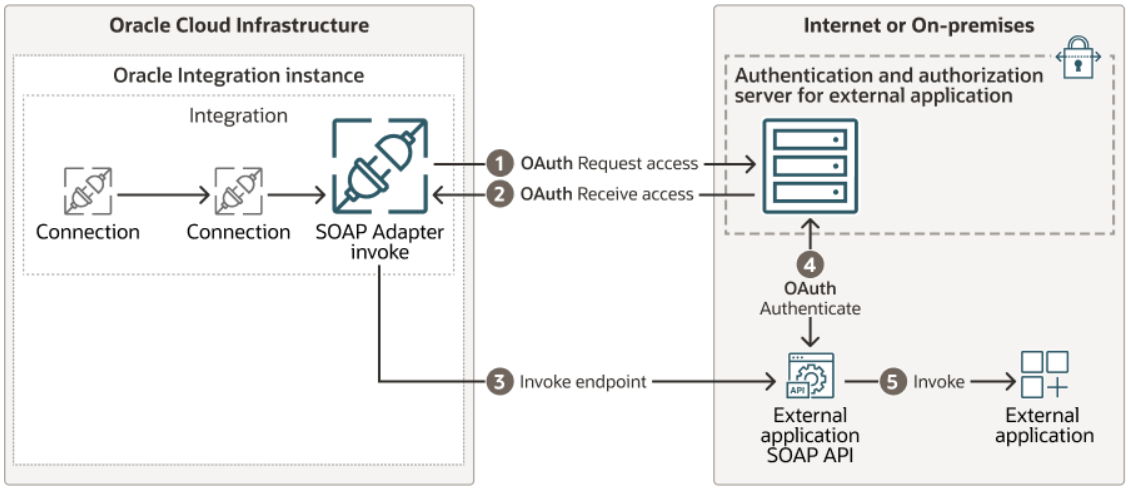
Steps to use OAuth with the REST Adapter and invoke connections:

1. If you are invoking Oracle Integration Developer APIs or the endpoint of an integration, complete this step. Otherwise, skip this step and go to step 2.
   a. Create the confidential client application, assign scopes and roles, and activate it:
      i. Access the Identity Domain.
      ii. Configure prerequisites for your grant type:
         - Prerequisites for Authorization Code
         - Prerequisites for Client Credentials and Resource Owner Password Credentials
         - Prerequisites for JWT User Assertion
2. Create a REST Adapter connection.
3. Add the REST Adapter as an invoke connection to an integration.

Invoke an Application's SOAP Endpoint with OAuth

When you configure an invoke connection in an integration, you can use OAuth to securely invoke the SOAP API of an external application. Here's a summary of what you need to configure to use OAuth with SOAP invoke connections in an integration.

When an integration invokes the endpoint of an external application, Oracle Integration requests access from the authorization server for the external application. Once access is granted, Oracle Integration accesses the SOAP endpoint, authenticates with the external authorization server, and invokes the application.



Copyright © 2022, 2026, Oracle and/or its affiliates.

Configuration Summary

Is OAuth required?	Supported grant types	Required roles
No. You can use other authentication types supported by the external application.	Supported OAuth grant types for invoke SOAP Adapter connections: • JWT User Assertion • Authorization code • Client credentials • Resource Owner Password Credentials (not recommended) See About OAuth 2.0 Grants for more details.	If you are calling an external endpoint, the role you assign depends on the external application requirements.

Configuration Steps

1. Create a SOAP Adapter Connection.
2. Add the SOAP Adapter Connection to an Integration.

Invoke Oracle Fusion Application Endpoints with OAuth

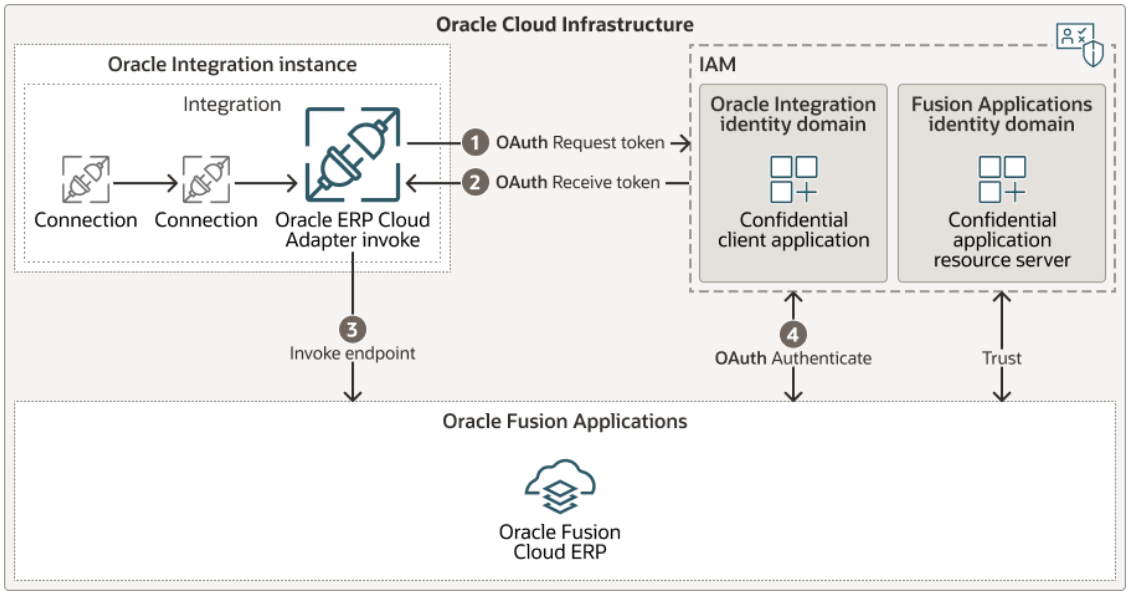
You can use OAuth to establish connections to Oracle Fusion Applications from Oracle Integration and invoke Oracle Fusion Applications endpoints. To invoke Oracle Fusion Applications, you use the corresponding adapter for the application, such as Oracle ERP Cloud Adapter, Oracle CX Sales and B2B Service Adapter, or Oracle HCM Cloud Adapter.

OAuth works similarly for the different Oracle Fusion Applications.

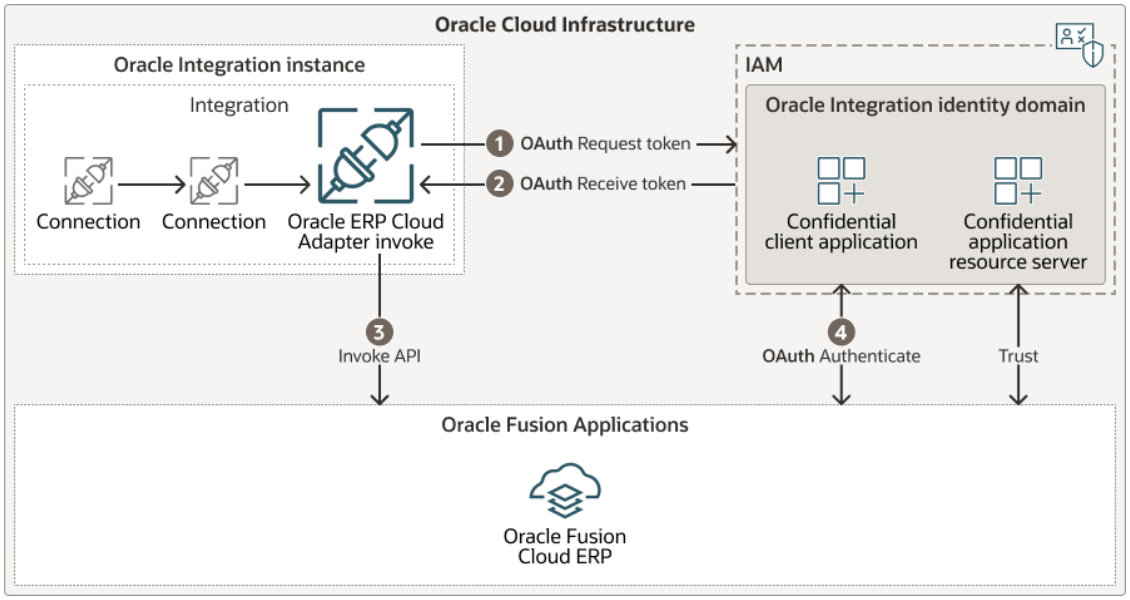
1. You create a resource application to represent the Oracle Fusion Applications resource. You can create the resource application in the Oracle Fusion Applications identity domain, or in a non-Oracle Fusion Applications identity domain such as the Oracle Integration identity domain.
2. The Oracle Fusion Applications takes the token from the specific application Oracle Integration adapter and validates it against OCI IAM.

Confidential Application Resource Server Configuration within the Fusion Applications Identity Domain

Copyright © 2022, 2026, Oracle and/or its affiliates.



Confidential Application Resource Server Configuration in a non-Fusion Application Identity Domain



Copyright © 2022, 2026, Oracle and/or its affiliates.

Configuration Summary

Is OAuth required?	Supported security policies	Required roles
No, OAuth is not required for Fusion Application connections.	Oracle ERP Cloud Adapter: • See Capabilities for supported security policies. Oracle CX Sales and B2B Service Adapter: • See Capabilities for supported security policies. Oracle HCM Cloud Adapter • See Capabilities for supported security policies.	Oracle ERP Cloud Adapter: • See Prerequisites for Creating a Connection Oracle CX Sales and B2B Service Adapter: • See Prerequisites for Creating a Connection Oracle HCM Cloud Adapter • See Prerequisites for Creating a Connection

Configuration Steps

Oracle ERP Cloud Adapter:

- See Set Up the OAuth Authorization Code Credentials Security Policy with the Oracle Fusion Applications Identity Domain
- See Set Up the OAuth Authorization Code Credentials Security Policy with a Non-Oracle Fusion Applications Identity Domain

Oracle CX Sales and B2B Service Adapter:

- See Set Up the OAuth Authorization Code Credentials Security Policy with the Oracle Fusion Applications Identity Domain
- See Set Up the OAuth Authorization Code Credentials Security Policy with a Non-Oracle Fusion Applications Identity Domain

Oracle HCM Cloud Adapter:

- See Set Up the OAuth Authorization Code Credentials Security Policy with the Oracle Fusion Applications Identity Domain
- See Set Up the OAuth Authorization Code Credentials Security Policy with a Non-Oracle Fusion Applications Identity Domain

Invoke Application-Specific Endpoints with OAuth

You can use OAuth to invoke applications that support invoke connections and OAuth.

Take a look at the adapter documentation for your specific application to find out how OAuth is supported. See Configure Connection Security for links to security configuration prerequisites and steps for each adapter.

You Must Use OAuth with the Oracle Integration Developer APIs

OAuth is required to use the Oracle Integration Developer APIs.

- What are the Oracle Integration Developer APIs

Copyright © 2022, 2026, Oracle and/or its affiliates.

- Do You Use the Runtime or Design-time URL?
- Invoke Oracle Integration Developer APIs from an External Client
- Invoke the Oracle Integration Developer APIs from within an Integration
- Call the Developer APIs with Client Credentials
- Call the Developer APIs with Authorization Code
- Call the Developer APIs with JWT User Assertion
- Call the Developer APIs with Resource Owner Password Credentials
- FAQs for the Confidential Client Application
- Troubleshoot OAuth

What are the Oracle Integration Developer APIs

OAuth is required to use the Oracle Integration Developer APIs.

The Oracle Integration Developer APIs are the default APIs available for Integrations, File Server, and Oracle Cloud Infrastructure Process Automation:

- REST API for Oracle Integration 3
- REST API for File Server in Oracle Integration 3
- REST API for Oracle Cloud Infrastructure Process Automation

Do You Use the Runtime or Design-time URL?

Which URL you use when calling APIs in Oracle Integration depends on your use case. You can find both the Runtime URL and Design-time URL in the Oracle Cloud Console.

When to Use the Runtime and Design-time URL

- Use the Runtime URL if your client is going to call both integration endpoints and Oracle Integration Developer APIs.
- Use the Design-time URL if your client is going to call only the Oracle Integration Developer APIs.

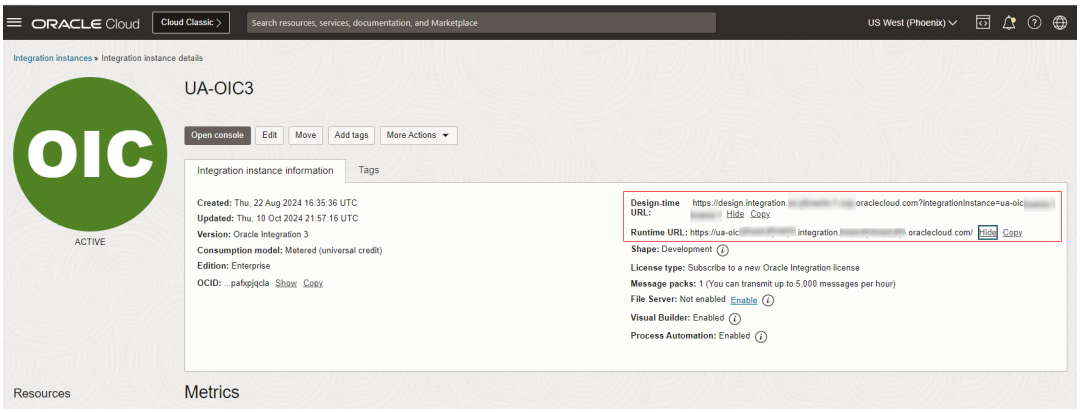
Get the Runtime and Design-time URL

You must be the OCI tenant and domain administrator to access the Oracle Cloud Console.

1. Sign in to the Oracle Cloud Console.
2. Open the navigation menu and click **Developer Services**. Under **Application Integration**, click **Integration**.

Copyright © 2022, 2026, Oracle and/or its affiliates.

3. Click a specific instance name. The Details page is displayed. You'll see the Design-time and Runtime URLs.



Get the Design-time URL from the Service Console URL

You can also find out the Design-time URL from the Service Console URL that you use to access the user interface.

1. Copy the **Service Console URL** into a browser window.
   When the login page of the integration instance displays, the URL changes starting with the word `design`. For example:

   Example:

   ```
   https://design.integration.region.ocp.oraclecloud.com/?
   integrationInstance=NameOfServiceInstance
   ```

The first part of the URL `https://design.integration.region.ocp.oraclecloud.com` is your Design-time URL and the rest of the URL `integrationInstance=NameOfServiceInstance` contains the name of the service instance.

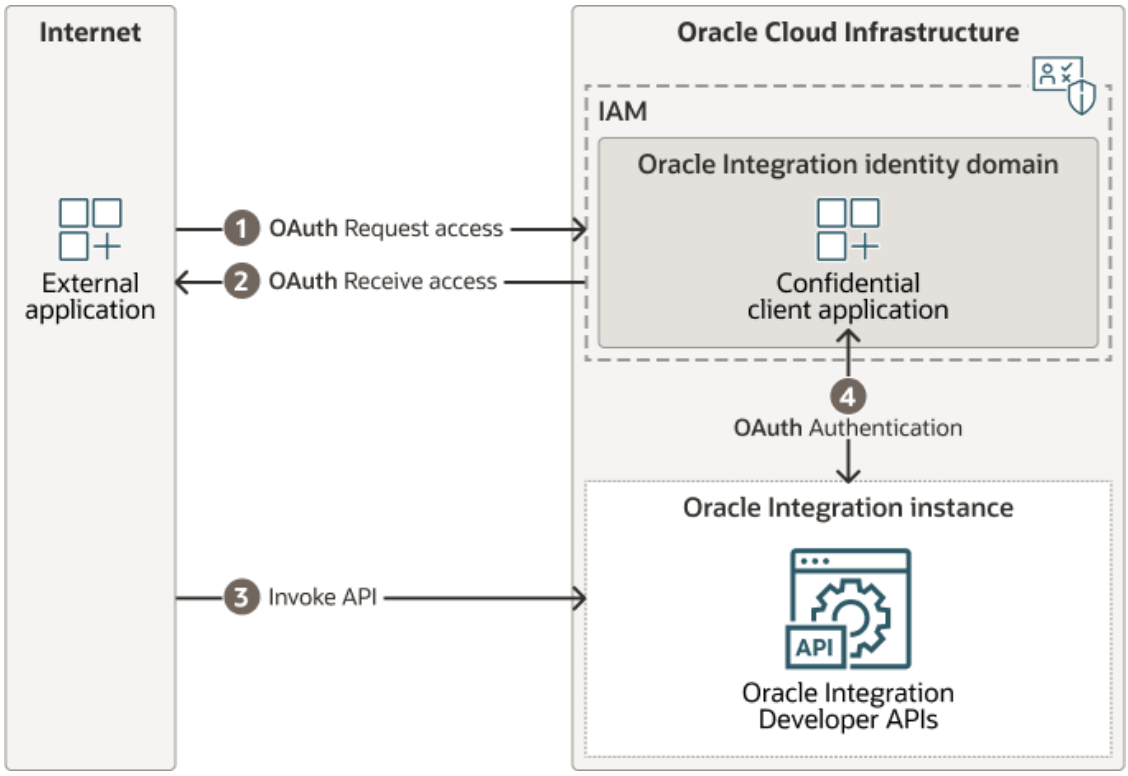
Get the Runtime URL from the Service Console URL

You can also get the Runtime URL from the Service Console URL. Replace design with the service instance name and you have the Runtime URL. For example: `https://NameOfServiceInstance.integration.region.ocp.oraclecloud.com`

Copyright © 2022, 2026, Oracle and/or its affiliates.

Invoke Oracle Integration Developer APIs from an External Client

OAuth is required to use the Oracle Integration Developer APIs. An external application or user requests access from IAM, then authenticates with IAM, and invokes the Oracle Integration Developer API.



Copyright © 2022, 2026, Oracle and/or its affiliates.

Configuration Summary

Is OAuth required?	Supported grant types	Required roles
Yes. OAuth is required for all Oracle Integration Developer APIs.	The grant types you can use are the same as that for the REST Adapter: • JWT User Assertion • Authorization code • Client credentials • Resource Owner Password Credentials (not recommended) See About OAuth 2.0 Grants for more details on each grant type and which one to use.	The role you assign your confidential client application depends on which APIs you'll be using. Assign the highest role required for your specific use case. If you're creating lookups for example, you'll need the ServiceDeveloper role. For detailed information on each role, see What Users Can Do in the Integrations Design Section by Role. • ServiceUser is the minimum role required to call the Developer APIs. This user has access to the Oracle Integration Console as well as the APIs. Assign the ServiceUser role in most cases. • ServiceDeveloper is required if you're invoking the Developer APIs from within an integration or you're using the APIs to perform create operations such as creating lookups or importing an integration. • Use ServiceInvoker if you only need to invoke an integration's endpoint and will not be making any calls to the Oracle Integration Developer APIs.

Configuration Steps

> **Note**
>
> You must be the OCI tenant and domain administrator to configure the confidential client application and assign roles.

Create the confidential client application, assign scopes and roles, and activate it:

1. Access the Identity Domain.
2. Configure prerequisites for your grant type:
   - Prerequisites for Authorization Code
   - Prerequisites for Client Credentials and Resource Owner Password Credentials
   - Prerequisites for JWT User Assertion
3. Call the Developer APIs using a client such as Postman or cURL:
   - Call the Developer APIs with Client Credentials
   - Call the Developer APIs with Authorization Code
   - Call the Developer APIs with JWT User Assertion

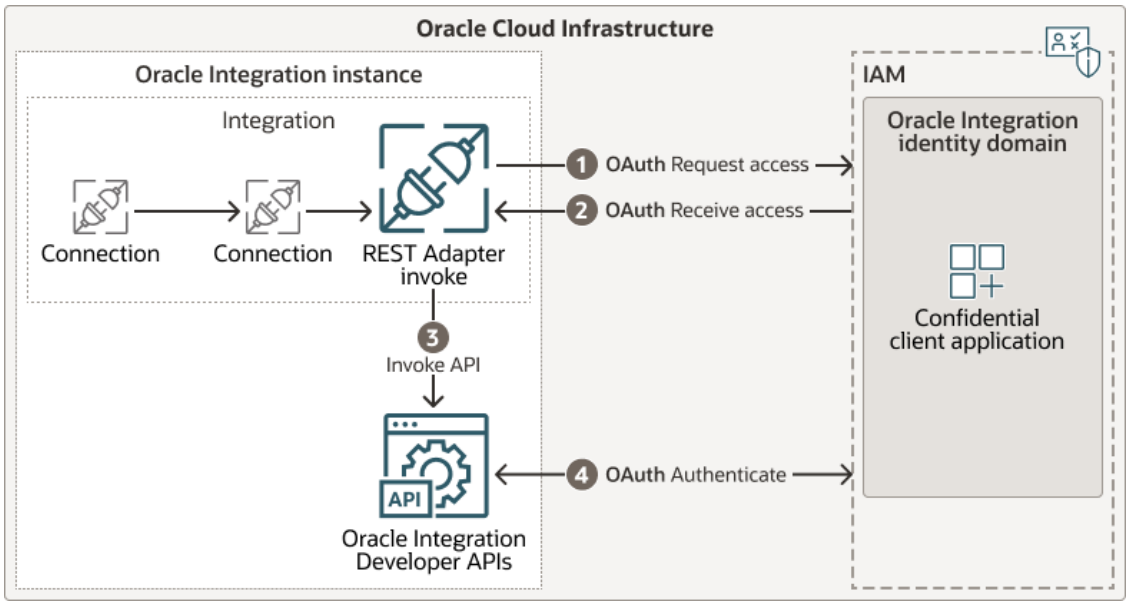
Copyright © 2022, 2026, Oracle and/or its affiliates.

- Call the Developer APIs with Resource Owner Password Credentials

Invoke the Oracle Integration Developer APIs from within an Integration

OAuth is required to use the Oracle Integration Developer APIs. Calling the Oracle Integration Developer APIs from within an integration is common.

For example, you use the scheduling Developer API to run a scheduled integration or the lookups Developer API to view lookups from within your integration.
To call the Oracle Integration Developer APIs from within an integration, you must also configure the OAuth policy in the REST Adapter invoke connection.



Copyright © 2022, 2026, Oracle and/or its affiliates.

Configuration Summary

Is OAuth required?	Supported authentication types	Required roles
Yes. OAuth is required to invoke the Oracle Integration Developer APIs.	For supported authentication types for REST Adapter invoke connections, see Authentication Types	The role you assign your confidential client application depends on which APIs you'll be using. If you're creating lookups for example, you'll need the ServiceDeveloper role. For detailed information on each role, see What Users Can Do in the Integrations Design Section by Role. • ServiceUser is the minimum role required to call the Developer APIs. This user has access to the Oracle Integration Console as well as the APIs. Assign the ServiceUser role in most cases. This user can also trigger an integration's endpoint. • ServiceDeveloper is required if you're invoking the Developer APIs from within an integration or you're using the APIs to perform create operations such as creating lookups or importing an integration.

Configuration Steps

1. Create a REST Adapter connection
2. Add the REST Adapter as an Invoke Connection to an Integration
3. Configure OAuth settings for the invoke connection within the integration.
4. Use the Design-time URL.
5. Change the invoke request parameters and add the parameter integrationInstance.
6. Pass the integrationInstance value in the mapping.
7. Trigger the integration to invoke the APIs.

Call the Developer APIs with Client Credentials

Prerequisites to Complete

Before you can call the Oracle Integration Developer APIs with Client credentials, you must create a confidential client application and perform other prerequisites.
You must be the OCI tenant and domain administrator to configure the confidential client application and assign roles.

For instructions, see:

1. Access the Identity Domain.
2. Configure prerequisites for your grant type:
   - Prerequisites for Client Credentials and Resource Owner Password Credentials

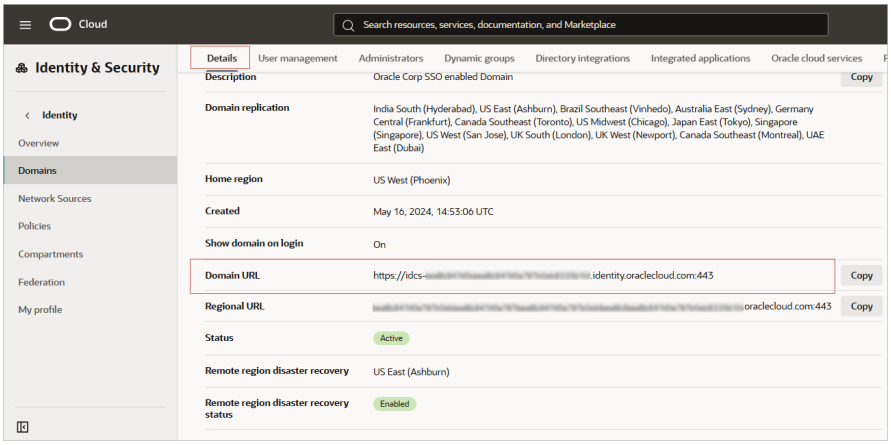
Copyright © 2022, 2026, Oracle and/or its affiliates.

Information You Need

The following is information you need to send requests through a client.

Information You Need	Where to Find It

Design-time URL or Runtime URL, depending on your use case	See Do You Use the Design-time or Runtime URL?.
	Example Design-time URL:
	`https://design-integration-region.ocp.oraclecloud.com/`
	Example Runtime URL:
	`https://myInstance-integration-region.ocp.oraclecloud.com/`

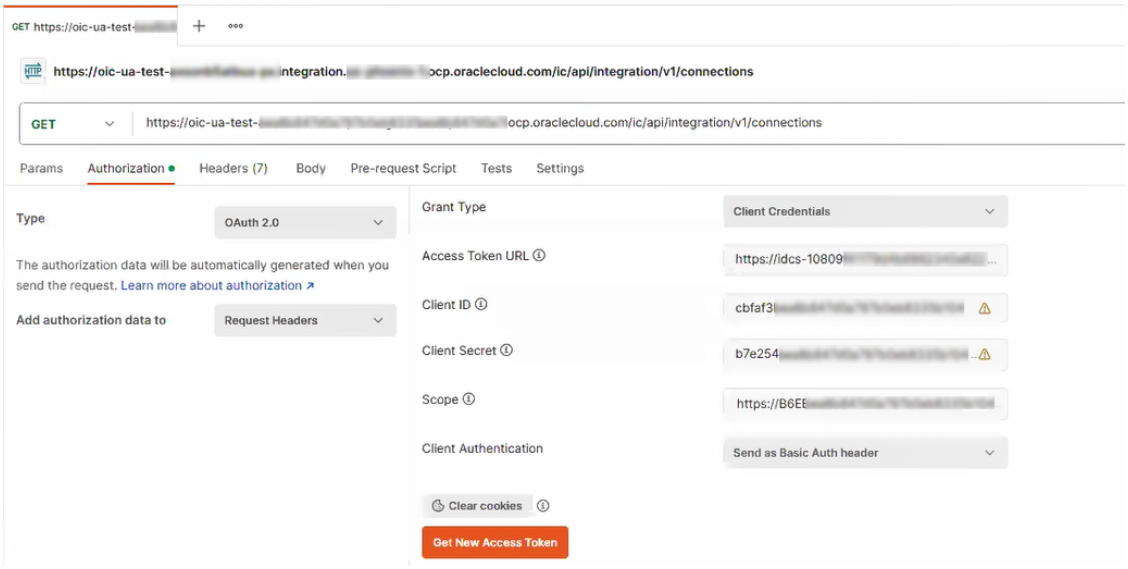
Domain Host	You can get the identity domain host from the **Domain URL** field of the domain you are in. For instructions on accessing your domain, see Access the Identity Domain. 
Client ID **Client Secret**	Add the client ID for the confidential application that you configured. To find your confidential application, access the identity domain, select **Integrated Applications**, select your application, and look for **OAuth Configuration**, then **Client ID** and **Client Secret**. For instructions, see Access the Identity Domain.

Scope	Add the same scope that you added to your confidential application. You can find the scope you added to the confidential application in your confidential application details under **Token Issuance Policy**.
	To find your confidential application, access the identity domain, select **Integrated Applications**, select your application, and look for **OAuth Configuration**, then **Client ID** and **Client Secret**. For instructions, see Access the Identity Domain.
	This scope allows users to access both integration endpoints and Oracle Integration Developer APIs:
	`https://69415C303.integration.ocp.oraclecloud.com:443urn:opc:resource:consumer::all`
	This scope allows users to only access the Oracle Integration Developer APIs:
	`https://415C303.integration.ocp.oraclecloud.com:443/ic/api/`

Copyright © 2022, 2026, Oracle and/or its affiliates.

Information You Need	Where to Find It
Integration Instance	Name of the integration instance.

Configure Postman for Client Credentials



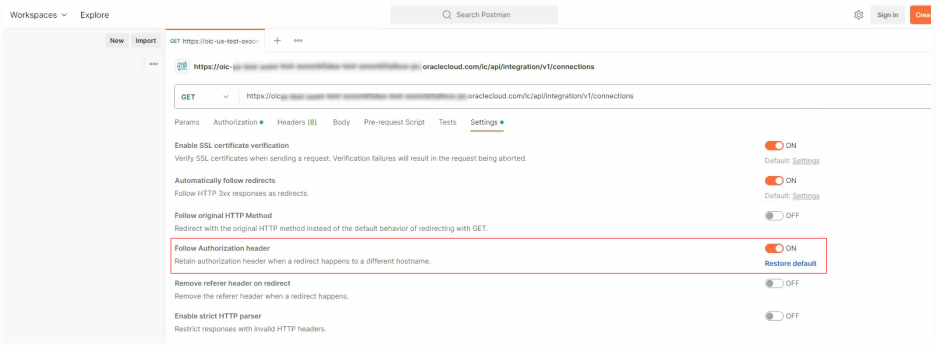
1. Use the Design-time URL or Runtime URL depending on your use case. See Do You Use the Design-time or Runtime URL?.
   Example Design-time URL with a call to the Oracle Integration Developer API for connections:

   ```
   https://design-integration-region.ocp.oraclecloud.com/ic/api/
   integration/v1/connections
   ```

   Example Runtime URL:

   ```
   https://myInstance-integration-region.ocp.oraclecloud.com/ic/api/
   integration/v1/connections
   ```


Copyright © 2022, 2026, Oracle and/or its affiliates.

> **Note**
>
> If you use the Runtime URL, you'll need to configure Postman to **Follow Authorization Header** in the request settings. This is because if you call the Oracle Integration Developer APIs with the Runtime URL, Oracle Integration redirects to the Design-time URL for the Developer APIs and the call will fail if the authorization header is missing from the request header.
>
> In cURL, use the -l option to forward the authorization header during the redirection.
>
> **To configure Postman to forward the authorization header when using the Runtime URL**:
>
> a. In the request, click **Settings** and enable **Follow authorization header**.
>
> 
>

2. Fill in the required fields.

Field	What to enter
Authorization type	`OAuth 2.0`
Grant Type	`Client credentials`
Access Token URL	Use the identity domain host you identified from your Domain URL. Example: `https://<identity_domain_host>/oauth2/v1/token`
Client ID	Add the client ID for the confidential application that you configured.
Client Secret	Add the client secret for the confidential application that you configured.
Scope	Add the same scope that you added to your confidential application.
Client Authentication	You can choose anything here, does not apply.

3. Click **Get New Access Token**.
4. Click **Use Token**.
   Your token is attached to your request in the Header section.
5. Click **Send** to make the call to the API. You should get a list of connections.

cURL command for Client Credentials

1. Get the access token to be able to make requests with the client credentials. For example:

```
curl -i -H 'Authorization: Basic OGQyM...ZDA0Mjcz' -H 'Content-Type:
application/x-www-form-urlencoded;charset=UTF-8' --request POST https://
<identity_domain_host>/oauth2/v1/token -d
```


Copyright © 2022, 2026, Oracle and/or its affiliates.

```
'grant_type=client_credentials&scope=https://<Resource APP
Audience>urn:opc:resource:consumer::all'
```

Syntax:

```
curl -i -H 'Authorization: Basic <base64Encoded clientid:secret>' -H
'Content-Type: application/x-www-form-urlencoded;charset=UTF-8' --request
POST https://<identity_domain_host>.identity.oraclecloud.com/oauth2/v1/
token -d 'grant_type=client_credentials&scope=<app scope>'
```

where:

- `<identity_domain_host>` is the host name in the **Domain URL** field of the domain you are in.
- `<base64-clientid-secret>` - Base 64 encode clientId:ClientSecret
- `<app scope>` - Add the same scope that you added to your confidential application.

2. Capture the `access_token` from the response.

```
{
    "access_token": "eyJ4NXQjG...dfsdfsFgets2ed",
    "token_type": "Bearer",
    "expires_in": 3600
}
```

3. Use the `access_token` in the authorization header to invoke the Oracle Integration Developer APIs.

```
curl --location --request GET 'https://<OIC_host>/ic/api/integration/v1/
connections' \
--header 'Authorization: Bearer eyJ4NXQjG...dfsdfsFgets2ed'
```

- Where `<oic_host>` is the Design-time URL host name or Runtime URL host name. See Do You Use the Design-time or Runtime URL?.

> **Note**
>
> If you use the Runtime URL host name, you'll need to use the `-l` option to forward the authorization header because you are automatically redirected to the Design-time host.

Call the Developer APIs with Authorization Code

Prerequisites to Complete

Before you can call the Oracle Integration Developer APIs with Authorization code, you must create a confidential client application and perform other prerequisites.
You must be the OCI tenant and domain administrator to configure the confidential client application and assign roles.

For instructions, see:

Copyright © 2022, 2026, Oracle and/or its affiliates.

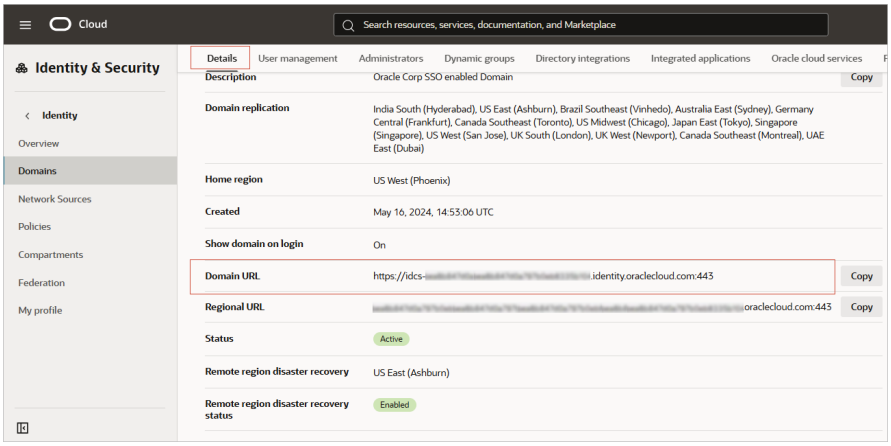
1. Access the Identity Domain.
2. Configure prerequisites for your grant type:
   - Prerequisites for Authorization Code

Information You Need

The following is information you need to send requests through a client.

Information You Need	Where to Find It

Design-time URL or Runtime URL, depending on your use case	See Do You Use the Design-time or Runtime URL?.
	Example Design-time URL:
	`https://design-integration-region.ocp.oraclecloud.com/`
	Example Runtime URL:
	`https://myInstance-integration-region.ocp.oraclecloud.com/`

Domain Host	You can get the identity domain host from the **Domain URL** field of the domain you are in. For instructions on accessing your domain, see Access the Identity Domain.. 
Client ID **Client Secret**	Add the client ID for the confidential application that you configured. To find your confidential application, access the identity domain, select **Integrated Applications**, select your application, and look for **OAuth Configuration**, then **Client ID** and **Client Secret**. For instructions, see Access the Identity Domain.

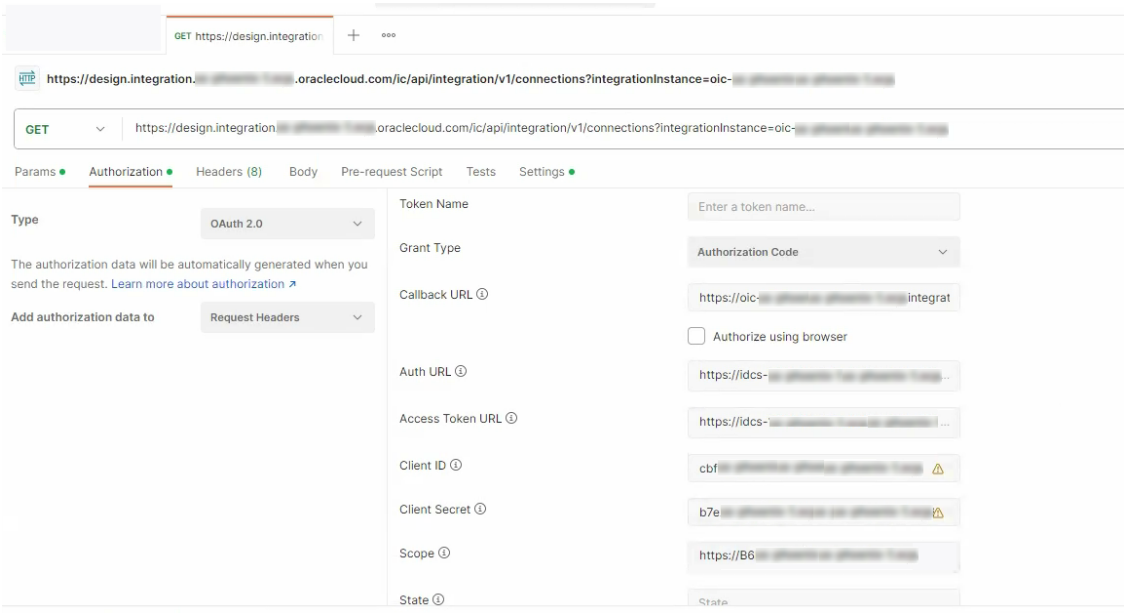
Copyright © 2022, 2026, Oracle and/or its affiliates.

Information You Need	Where to Find It

Scope	Add the same scope that you added to your confidential application. You can find the scope you added to the confidential application in your confidential application details under **Token Issuance Policy**.
	To find your confidential application, access the identity domain, select **Integrated Applications**, select your application, and look for **OAuth Configuration**, then **Client ID** and **Client Secret**. For instructions, see Access the Identity Domain.
	This scope allows users to access both integration endpoints and Oracle Integration Developer APIs:
	`https://69415C303.integration.ocp.oraclecloud.com:443urn:opc:resource:consumer::all`
	This scope allows users to only access the Oracle Integration Developer APIs:
	`https://415C303.integration.ocp.oraclecloud.com:443/ic/api/`

Integration Instance	Name of the integration instance.
Redirect URL	You can find the redirect URL in the **Authorization** area of the confidential client.

Configure Postman for Authorization Code



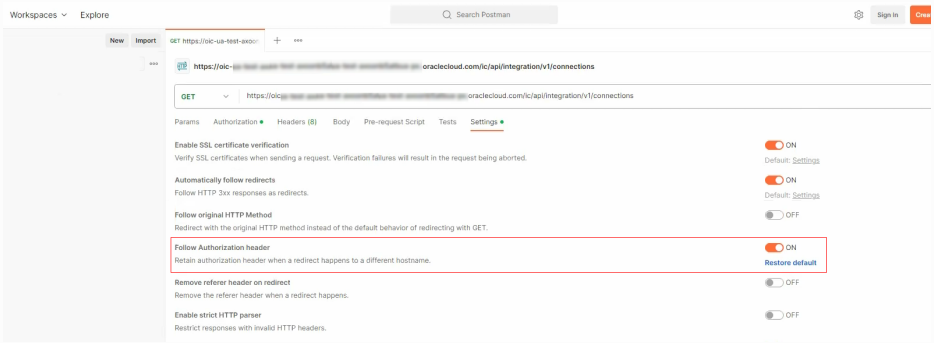
1. Use the Design-time URL or Runtime URL depending on your use case. See Do You Use the Design-time or Runtime URL?.
   Example Design-time URL with a call to the Oracle Integration Developer API for connections:

   ```
   https://design-integration-region.oraclecloud.com/ic/api/integration/v1/connections
   ```


Copyright © 2022, 2026, Oracle and/or its affiliates.

Example Runtime URL:

```
https://myInstance-integration-region.oraclecloud.com/ic/api/
integration/v1/connections
```

> **Note**
>
> If you use the Runtime URL, you'll need to configure Postman to **Follow Authorization Header** in the request settings. This is because if you call the Oracle Integration Developer APIs with the Runtime URL, Oracle Integration redirects to the Design-time URL for the Developer APIs and the call will fail if the authorization header is missing from the request header.
>
> In cURL, use the `-l` option to forward the authorization header during the redirection.
>
> **To configure Postman to forward the authorization header when using the Runtime URL**:
>
> a. In the request, click **Settings** and enable **Follow authorization header**.
>
> 
>

2. Fill in the required fields.

Field	What to enter
Authorization type	`OAuth 2.0`
Grant Type	`Authorization Code`
Callback URL	This is the Redirect URL you configured in the confidential client application. You can find the Redirect URL in the confidential client application under the **Authorization** section.
Auth URL	Use the identity domain host you identified from your domain URL to create the Auth URL. For example: `https://<identity_domain_host>/oauth2/v1/authorize`
Access Token URL	Use the identity domain host you identified from your Domain URL. For example: `https://<identity_domain_host>/oauth2/v1/token`
Client ID	Add the client ID for the confidential application that you configured.
Client Secret	Add the client secret for the confidential application that you configured.
Scope	Add the same scope that you added to your confidential application.
Client Authentication	You can choose anything here, does not apply.

3. Click **Get New Access Token**.
4. Click **Use Token**.

Copyright © 2022, 2026, Oracle and/or its affiliates.

Your token is attached to your request in the Header section.

5. Click **Send** to make the call to the API. You should get a list of connections.

cURL command for Authorization Code

1. From your browser, request an authorization code. For example:

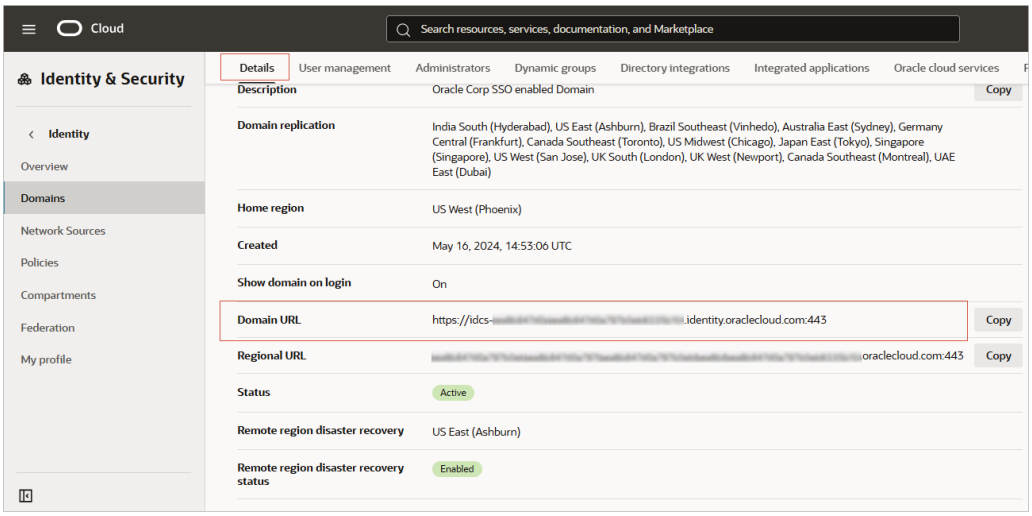
```
GET
      https://<identity_domain_host>/oauth2/v1/authorize?
client_id=<clientID>&response_type=code&redirect_uri=https://
app.getpostman.com/oauth2/callback&scope=https://
<Resource_APP_Audience>urn:opc:resource:consumer::all%20offline_access&nonc
e=121&state=12345544
```

Syntax:

```
GET
      https://<identity_domain_host>.identity.oraclecloud.com/oauth2/v1/
authorize?client_id=<client-id>&response_type=code&redirect_uri=<client-
redirect-uri>&scope=<app_scope>%20offline_access&nonce=<nonce-
value>&state=<unique_value>
```

where:

- `<identity_domain_host>` is the host name in the Domain URL field of the domain you are in.



- `<client-id>` - ID of Client application generated
- `<client-redirect-uri>` - Redirect URI, in client application. The **Redirect URL** needs to be accessed when you click on provide consent on the connection. This is the redirect URL of the Oracle Integration host. Format: `https://<Service Console URL>/icsapis/agent/oauth/callback`. This is the standard callback URL for the Oracle Integration instance.
- `<app scope>` - Add the same scope that you added to your confidential application.

Copyright © 2022, 2026, Oracle and/or its affiliates.

This scope allows users to access both integration endpoints and Oracle Integration Developer APIs:

```
https://
69415C303.integration.ocp.oraclecloud.com:443urn:opc:resource:consumer::
all
```

This scope allows users to only access the Oracle Integration Developer APIs:

```
https://415C303.integration.ocp.oraclecloud.com:443/ic/api/
```

- `nonce` - Optional, unique value to mitigate replay attacks
- `state` - Recommended, Opaque to IAM. Value used to maintain state between the request and the callback

2. If the user is not already logged in, you are challenged to authenticate your user credentials.
After authentication is successful, the client URL is redirected with the authorization code and state added to the URL.

```
##Response URL
https://<redirect_URL>?code=<code_value>=&state=<state_value>

###Client should validate state received is same as one sent in request.
```

3. Capture the `code value` from the above response and make the request to get the access token. For example:

```
curl -i -H 'Authorization: Basic MDMx..NGY1' -H 'Content-Type:
application/x-www-form-urlencoded;charset=UTF-8' --request POST https://
<identity_domain_host>/oauth2/v1/token -d
'grant_type=authorization_code&code=AQAg...3jKM4Gc=&redirect_uri=https://
app.getpostman.com/oauth2/callback
```

Syntax:

```
curl -i -H 'Authorization: Basic <base64-clientid-secret>' -H 'Content-
Type: application/x-www-form-urlencoded;charset=UTF-8' --request POST
https://<identity_domain_host>.identity.oraclecloud.com/oauth2/v1/token -d
'grant_type=authorization_code&code=<authz-code>&redirect_uri=<client-
redirect-uri>
```

where:

- `<base64-clientid-secret>` - BAse 64 encode clientId:ClientSecret
- `<authz-code>` - code value received as response on redirect.
- `<client-redirect-uri>` - Redirect URI, in client application.

4. Capture the `access_token` and `refresh_token` from the response.

```
{
    "access_token": "eyJ4NXQjG...dfsdfsFgets2ed",
    "token_type": "Bearer",
```


Copyright © 2022, 2026, Oracle and/or its affiliates.

```
    "expires_in": 3600,
    "refresh_token": "AQAgY2MzNjVlOTVhOTRh...vM5S0MkrFSpzc="
}
```

5. Use the `access_token` in the authorization header to invoke the Oracle IntegrationDeveloper API.

```
curl --location --request GET 'https://<OIC_host>/ic/api/integration/v1/
connections' \
--header 'Authorization: Bearer eyJ4NXQjG...dfsdfsFgets2ed'
```

   - Where `<oic_host>` is the Design-time URL host name or Runtime URL host name. See Do You Use the Design-time or Runtime URL?.

> **Note**
>
> If you use the Runtime URL host name, you'll need to use the `-l` option to forward the authorization header because you are automatically redirected to the Design-time host.

Update the Access Token

To update the access token, use the refresh token and make the request.

1. Capture the `access_token` and `refresh_token` from a response for further use. For example:

```
curl -i -H 'Authorization: Basic OGQyM...ZDA0Mjcz' -H 'Content-Type:
application/x-www-form-urlencoded;charset=UTF-8' --request POST https://
<Identity_Domain_Host>.identity.oraclecloud.com/oauth2/v1/token  -d
'grant_type=refresh_token&refresh_token=AQAgY2MzNjVlOTVhOTRh...vM5S0MkrFSpz
c='
```

   Syntax:

```
curl -i -H 'Authorization: Basic <base64-clientid-secret>' -H 'Content-
Type: application/x-www-form-urlencoded;charset=UTF-8' --request POST
https://<Identity_Domain_Host>.identity.oraclecloud.com/oauth2/v1/token  -
d 'grant_type=refresh_token&refresh_token=<refresh_token>'
```

Call the Developer APIs with JWT User Assertion

Prerequisites to Complete

Before you can call the Oracle Integration Developer APIs with JWT User Assertion, you must create a confidential client application and perform other prerequisites.
You must be the OCI tenant and domain administrator to configure the confidential client application and assign roles.

For instructions, see:

1. Access the Identity Domain.

Copyright © 2022, 2026, Oracle and/or its affiliates.

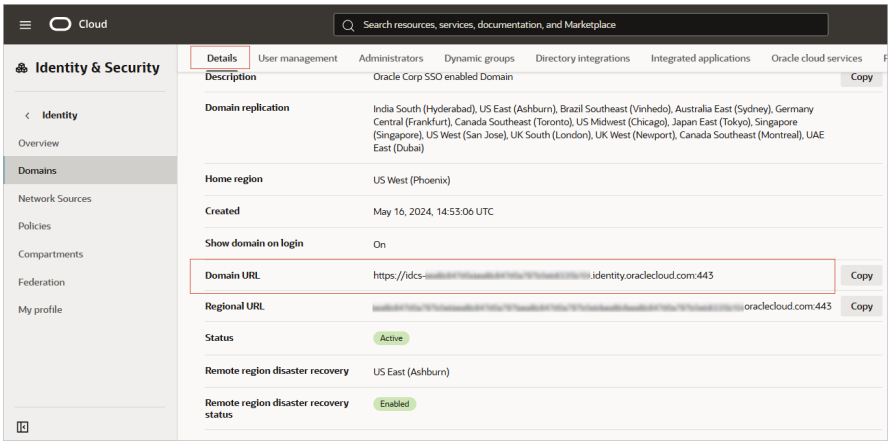
2. Configure prerequisites for your grant type:
   - Prerequisites for JWT User Assertion

Information You Need

The following is information you need to send requests through a client.

Information You Need	Where to Find It

Design-time URL or Runtime URL, depending on your use case	See Do You Use the Design-time or Runtime URL?.
	Example Design-time URL:
	`https://design-integration-region.ocp.oraclecloud.com/`
	Example Runtime URL:
	`https://myInstance-integration-region.ocp.oraclecloud.com/`

Domain Host	You can get the identity domain host from the **Domain URL** field of the domain you are in. For instructions on accessing your domain, see Access the Identity Domain. 
Client ID **Client Secret**	Add the client ID for the confidential application that you configured. To find your confidential application, access the identity domain, select **Integrated Applications**, select your application, and look for **OAuth Configuration**, then **Client ID** and **Client Secret**. For instructions, see Access the Identity Domain.

Copyright © 2022, 2026, Oracle and/or its affiliates.

Information You Need	Where to Find It

Scope	Add the same scope that you added to your confidential application. You can find the scope you added to the confidential application in your confidential application details under **Token Issuance Policy**.
	To find your confidential application, access the identity domain, select **Integrated Applications**, select your application, and look for **OAuth Configuration**, then **Client ID** and **Client Secret**. For instructions, see Access the Identity Domain.
	This scope allows users to access both integration endpoints and Oracle Integration Developer APIs:
	`https://69415C303.integration.ocp.oraclecloud.com:443urn:opc:resource:consumer::all`
	This scope allows users to only access the Oracle Integration Developer APIs:
	`https://415C303.integration.ocp.oraclecloud.com:443/ic/api/`

Integration Instance	Name of the integration instance.

Configure Postman for JWT User Assertion

JWT User Assertion is not directly supported with Postman. You need to define scripts to generate the assertion. Read the blog Demystifying OAuth Using the JWT User Assertion in OIC for more details and sample scripts.

cURL command for JWT User Assertion

1. Once you generate the JWT user assertion, generate the access token.
   Syntax:

   ```
   curl -i -H 'Authorization: Basic <base64Encoded clientid:secret>' -H
   'Content-Type: application/x-www-form-urlencoded;charset=UTF-8' --request
   POST https://<Identity_Domain_Service_Instance>.identity.oraclecloud.com/
   oauth2/v1/token -d 'grant_type=urn%3Aietf%3Aparams%3Aoauth%3Agrant-
   type%3Ajwt-bearer&assertion=<user assertion>&scope=<app_scope>'
   ```

   Where:

   - `grant_type` - urn:ietf:params:oauth:grant-type:jwt-bearer
   - `<base64-clientid-secret>` - Base 64 encode clientId:ClientSecret
   - `<user assertion>` - User assertion generated
   - `<app scope>` - Scope added while creating application in client configuration section (Ends with urn:opc:resource:consumer::all)
2. Capture the `access_token` from the response.

   ```
   {
       "access_token": "eyJ4NXQjG...dfsdfsFgets2ed",
   ```


Copyright © 2022, 2026, Oracle and/or its affiliates.

```
    "token_type": "Bearer",
    "expires_in": 3600
}
```

3. Use an `access_token` in the authorization header to invoke the Oracle Integration Developer APIs.

```
curl --location --request GET 'https://OIC host/ic/api/integration/v1/
connections' \
--header 'Authorization: Bearer eyJ4NXQjG...dfsdfsFgets2ed'
```

- Where <oic_host> is the Design-time URL host name or Runtime URL host name. See Do You Use the Design-time or Runtime URL?.

> **Note**
>
> If you use the Runtime URL host name, you'll need to use the `-l` option to forward the authorization header because you are automatically redirected to the Design-time host.

Call the Developer APIs with Resource Owner Password Credentials

Prerequisites to Complete

Before you can call the Oracle Integration Developer APIs with Resource Owner Password Credentials, you must create a confidential client application and perform other prerequisites. For instructions, see:

1. Access the Identity Domain.
2. Configure prerequisites for your grant type: Prerequisites for Client Credentials and Resource Owner Password Credentials.

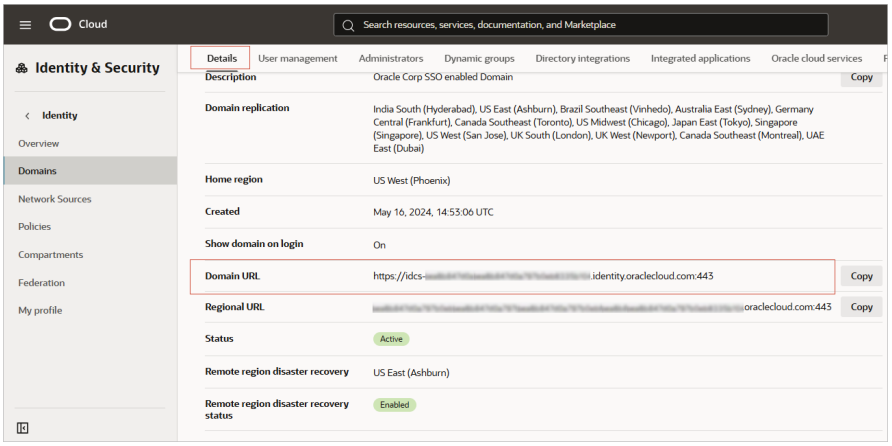
Information You Need

The following is information you need to send requests through a client.

Information You Need	Where to Find It

Design-time URL or Runtime URL, depending on your use case	See Do You Use the Design-time or Runtime URL?.
	Example Design-time URL:
	`https://design-integration-region.ocp.oraclecloud.com/`
	
	Example Runtime URL:
	`https://myInstance-integration-region.ocp.oraclecloud.com/`

Copyright © 2022, 2026, Oracle and/or its affiliates.

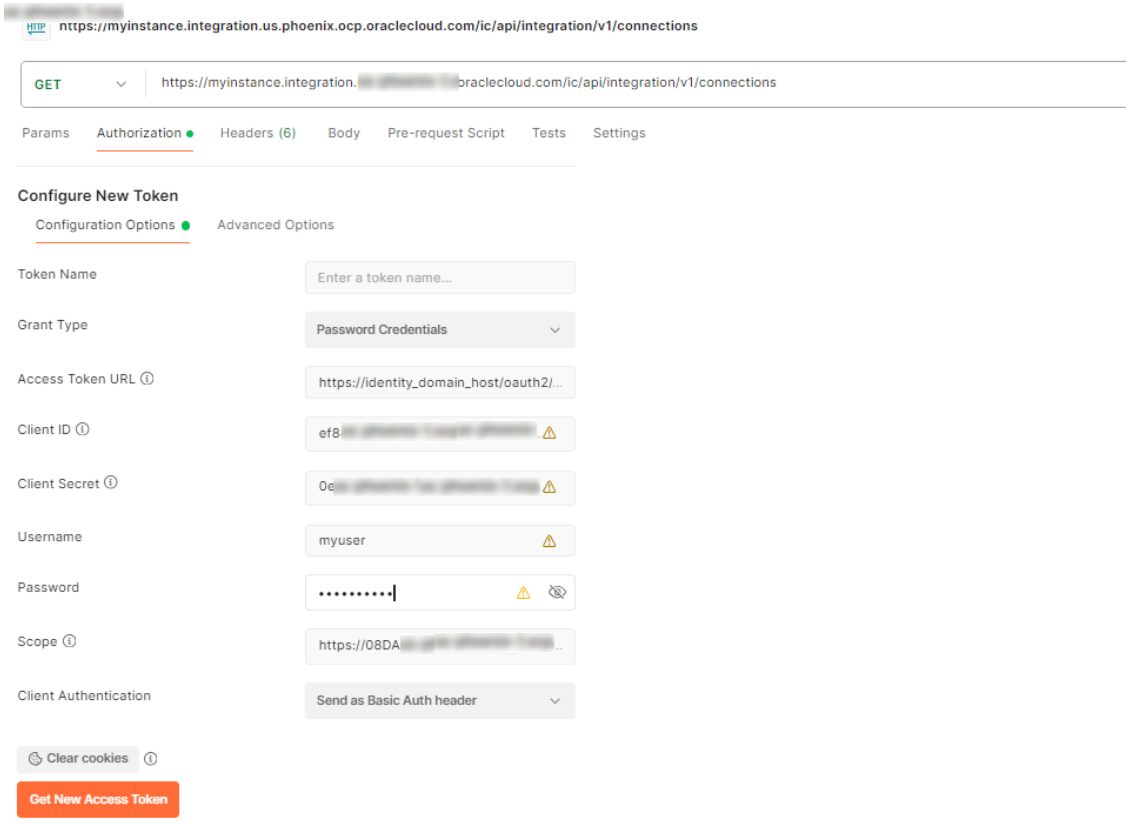
Information You Need	Where to Find It
Domain Host	You can get the identity domain host from the **Domain URL** field of the domain you are in. For instructions on accessing your domain, see Access the Identity Domain. 
Client ID **Client Secret**	Add the client ID for the confidential application that you configured. To find your confidential application, access the identity domain, select **Integrated Applications**, select your application, and look for **OAuth Configuration**, then **Client ID** and **Client Secret**. For instructions, see Access the Identity Domain.

Scope	Add the same scope that you added to your confidential application. You can find the scope you added to the confidential application in your confidential application details under **Token Issuance Policy**.
	To find your confidential application, access the identity domain, select **Integrated Applications**, select your application, and look for **OAuth Configuration**, then **Client ID** and **Client Secret**. For instructions, see Access the Identity Domain.
	This scope allows users to access both integration endpoints and Oracle Integration Developer APIs:
	`https://69415C303.integration.ocp.oraclecloud.com:443urn:opc:resource:consumer::all`
	This scope allows users to only access the Oracle Integration Developer APIs:
	`https://415C303.integration.ocp.oraclecloud.com:443/ic/api/`

Integration Instance	Name of the integration instance.

Copyright © 2022, 2026, Oracle and/or its affiliates.

Configure Postman for Resource Owner Password Credentials



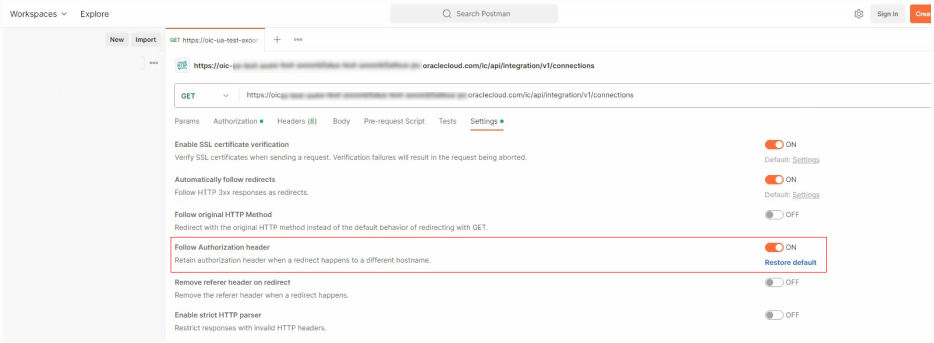
1. Use the Design-time URL or Runtime URL depending on your use case. See Do You Use the Design-time or Runtime URL?.
   Example Design-time URL with a call to the Oracle Integration Developer API for connections:

   ```
   https://design-integration-region.oraclecloud.com/ic/api/integration/v1/
   connections
   ```

   Example Runtime URL:

   ```
   https://myInstance-integration-region.oraclecloud.com/ic/api/
   integration/v1/connections
   ```


Copyright © 2022, 2026, Oracle and/or its affiliates.

> **Note**
>
> If you use the Runtime URL, you'll need to configure Postman to **Follow Authorization Header** in the request settings. This is because if you call the Oracle Integration Developer APIs with the Runtime URL, Oracle Integration redirects to the Design-time URL for the Developer APIs and the call will fail if the authorization header is missing from the request header.
>
> In cURL, use the `-l` option to forward the authorization header during the redirection.
>
> **To configure Postman to forward the authorization header when using the Runtime URL**:
>
> a. In the request, click **Settings** and enable **Follow authorization header**.
>
> 
>

2. In the Authorization tab, fill in the required fields.

Field	What to enter
Authorization type	`OAuth 2.0`
Grant Type	`Password credentials`
Access Token URL	Use the identity domain host you identified from your Domain URL. Example: `https://<identity_domain_host>/oauth2/v1/token`
Client ID	Add the client ID for the confidential application that you configured.
Client Secret	Add the client secret for the confidential application that you configured.
User Name and Password	Provide a valid user name and password. This will generate the token with the resource owner password credentials.
Scope	Add the same scope that you added to your confidential application.
Client Authentication	You can choose anything here, does not apply.

3. Click **Get New Access Token**.
4. You are prompted to enter a user name and password.
   This user must have been created in the IAM domain. You cannot use a single sign-on user name and password as this is a different authentication method. If you have single sign-on for your instance, you will need to copy the same users in IDCS and maintain a separate password for local IAM access.
5. Click **Use Token**.
   Your token is attached to your request in the Header section.
6. Click **Send** to make the call to the API. You should get a list of connections.

Copyright © 2022, 2026, Oracle and/or its affiliates.

cURL command for Resource Owner Password Credentials

1. To fetch the access client, make a request with the user name and password in the payload. For example:

   ```
   curl -i -H 'Authorization: Basic OGQyM...ZDA0Mjcz' -H 'Content-Type:
   application/x-www-form-urlencoded;charset=UTF-8' --request POST https://
   <identity_domain_host>/oauth2/v1/token -d
   'grant_type=password&username=sampleUser&password=SamplePassword&scope=http
   s://<Resource_APP_Audience>urn:opc:resource:consumer::all%20offline_access'
   ```

   Syntax:

   ```
   curl -i -H 'Authorization: Basic <base64Encoded_clientid:secret>' -H
   'Content-Type: application/x-www-form-urlencoded;charset=UTF-8' --request
   POST https://<Identity_Domain_Service_Instance>.identity.oraclecloud.com/
   oauth2/v1/token -d 'grant_type=password&username=<user-
   name>&password=<password>&scope=<App_Scope>%20offline_access'
   ```

   Where:

   - `<base64-clientid-secret>` - Base 64 encode clientId:ClientSecret
   - `<username>` - user for token needs to be issued (must be in serviceinvoker role).
   - `<password>`- password for above user
   - `<app_scope>` - Scope added while creating application in client configuration section (Ends with urn:opc:resource:consumer::all)

2. Capture the `access_token` and `refresh_token` from the response.

   ```
   {
       "access_token": "eyJ4NXQjG...dfsdfsFgets2ed",
       "token_type": "Bearer",
       "expires_in": 3600,
       "refresh_token": "AQAgY2MzNjVlOTVhOTRh...vM5S0MkrFSpzc="
   }
   ```

3. Use the `access_token` in the authorization header to invoke the Oracle Integration trigger endpoint.

   ```
   curl --location --request GET 'https://OIC host/OIC endpoint' \
   --header 'Authorization: Bearer eyJ4NXQjG...dfsdfsFgets2ed'
   ```

4. To update the access token, use the refresh token and make a request.
5. Capture the `access_token` and `refresh_token` from the response for further use. For example:

   ```
   curl -i -H 'Authorization: Basic OGQyM...ZDA0Mjcz' -H 'Content-Type:
   application/x-www-form-urlencoded;charset=UTF-8' --request POST https://
   <Identity_Domain_Service_Instance>.identity.oraclecloud.com/oauth2/v1/
   token  -d
   ```


Copyright © 2022, 2026, Oracle and/or its affiliates.

```
'grant_type=refresh_token&refresh_token=AQAgY2MzNjVlOTVhOTRh...vM5S0MkrFSpz
c='
```

Syntax:

```
curl -i -H 'Authorization: Basic <base64-clientid-secret>' -H 'Content-
Type: application/x-www-form-urlencoded;charset=UTF-8' --request POST
https://<Identity_Domain_Service_Instance>.identity.oraclecloud.com/
oauth2/v1/token  -d
'grant_type=refresh_token&refresh_token=<refresh_token>'
```

FAQs for the Confidential Client Application

A prerequisite to use OAuth with trigger connections and the Oracle Integration Developer APIs is that you create a confidential client application in Oracle Cloud Infrastructure Identity and Access Management (OCI IAM), configure the authorization grant, assign scopes, assign roles, and activate the confidential client application. You must be the OCI tenant and domain administrator to configure the confidential application.

When do you need a confidential client application?

You need to configure a confidential client application when you use OAuth with a trigger connection, or when you use the Oracle Integration Developer APIs. The confidential client application acts as the configuration to enable OAuth on associated integration applications.

What are the steps to configure the confidential client application and use OAuth in Oracle Integration?

The steps that you need to follow depend on your use case. To know more about where you can use OAuth in Oracle Integration and a summary of configuration steps for each use case, see:

- Use OAuth with Inbound Calls to Oracle Integration
- Use OAuth with Outbound Calls from Oracle Integration
- You Must Use OAuth with the Oracle Integration Developer APIs

You must be the OCI tenant and domain administrator to configure the confidential application

You must be the OCI tenant and domain administrator to configure the confidential application, assign scopes and roles, and activate it.

How many confidential client applications do I need?

How many confidential applications you configure depends on your use case.

- In general, you need one confidential application per Oracle Integration instance. You can configure one confidential client application to handle one or more OAuth authorization grant types.
- If you want to isolate the configuration users use, configure different confidential client applications per OAuth authorization grant type.
- If you have multiple identity domains, you need to configure one confidential client application per domain, because you can only access the confidential client application within a domain.

Copyright © 2022, 2026, Oracle and/or its affiliates.

Which OAuth authorization grant type do I use?

In general, these are the supported grant types:

- JWT User Assertion
- Client credentials
- Authorization code
- Resource Owner Password Credentials (not recommended)

Which one you configure in your confidential application depends on which ones your client supports.

- If the client is programmatic or SDK-based, use Client credentials or JWT User Assertion.
- If the client is browser-based and requires user interaction, use Authorization code.

For additional information on OAuth authorization grant types, see About OAuth 2.0 Grants

What are the scopes and which ones are required?

Scopes limit what have access to in the Oracle Integration instance.

There are two Oracle Integration instance scopes that you add to the confidential application:

It's recommended to assign both scopes to the confidential application.

This scope allows users to access both integration endpoints and Oracle Integration Developer APIs:

`https://<id>.host.oraclecloud.com:443urn:opc:resource:consumer::all`

This scope allows users to only access the Oracle Integration Developer APIs:

`https://<id>.host.oraclecloud.com:443/ic/api/`

Troubleshoot OAuth

Go through the following troubleshooting. You can resolve most issues by checking your configuration in these areas.

- Token Refresh Error: How do I Check the Configuration?
- Check Scopes Match Primary Audience in Your Oracle Integration Instance
- Unauthorized Errors: Check Roles Assigned to Your Confidential Application

Token Refresh Error: How do I Check the Configuration?

Check your configuration in the Oracle Cloud Infrastructure under **Oracle Cloud Services** and **Integrated Applications**. Both configurations are required for OAuth to work.

Step 1: Identify Your Oracle Integration Instance Information

Step 2: Check Token Refresh Settings in Domains

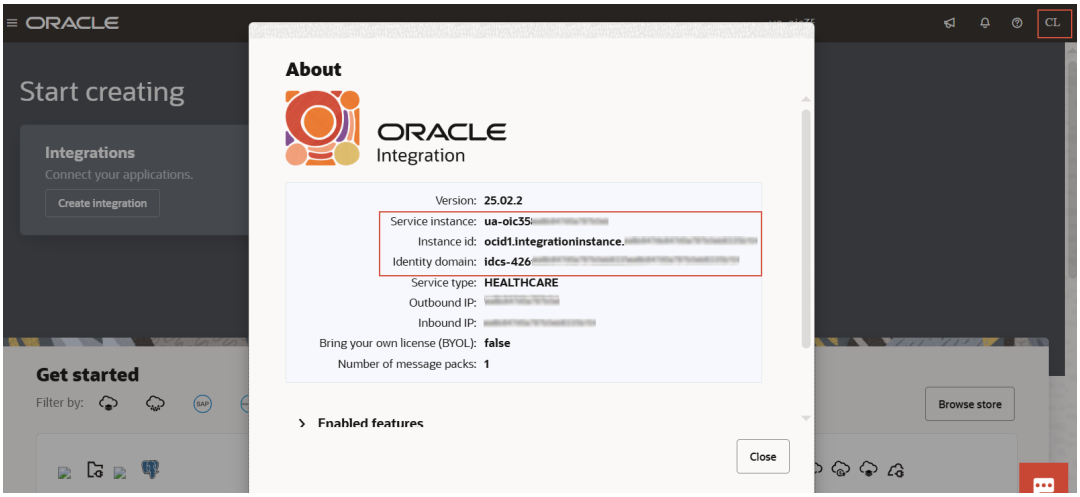
Step 3: Check Token Refresh and Scope Settings in Integrated Applications

Copyright © 2022, 2026, Oracle and/or its affiliates.

Step 1: Identify Your Oracle Integration Instance Information

Identify information about your Oracle Integration such as service name and domain to help you in troubleshooting.

1. Log into Oracle Integration console.
2. In the upper-right corner, click the avatar for your name, and then select **About**. You'll see information about your instance.



3. Copy your **Service instance** name, **Instance id**, and **Identity domain** values to a text file so that you can compare with what you have in your configuration.

Go to the next step Step 2: Check Token Refresh Settings in Domains

Step 2: Check Token Refresh Settings in Domains

How does token refresh work?

You have an access token and a refresh token. Both tokens have an expiry time.

In order for access tokens to be automatically refreshed, you need to enable **Allow token refresh** for your service instance in **Oracle Cloud Services**. When **Allow token refresh** is enabled, whenever an integration is running and the access token expires, the refresh token is used to automatically get a new access token. If an integration runs before the access token expires, the access token and refresh token are automatically refreshed. The token expiry time starts again at the maximum amount every time the tokens are refreshed.

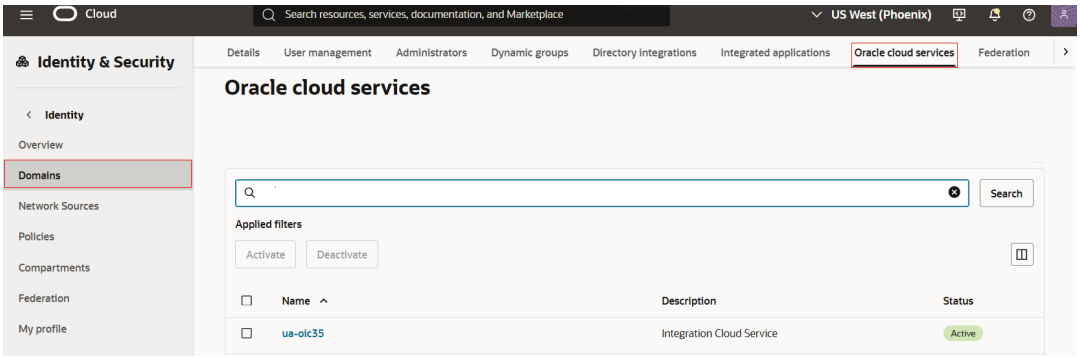
If an integration does not run for a while, it's possible for the access token and refresh token to both expire. If that happens, you'll need to manually get a new access token.

Check Token Refresh Settings

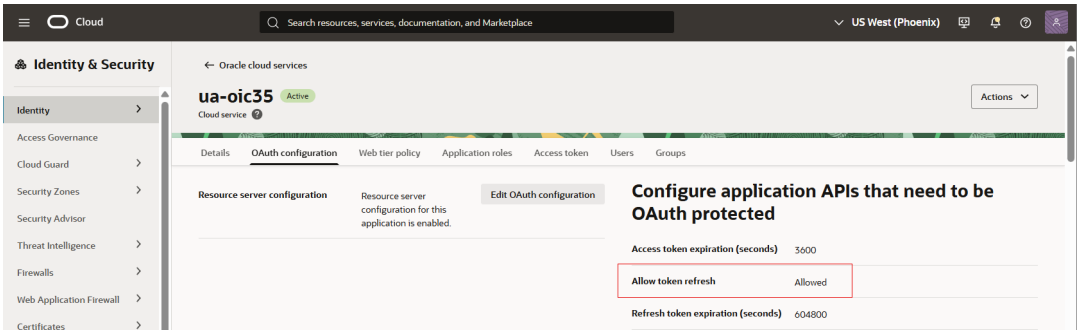
1. Log in to the Oracle Cloud Console with your identity domain administrator credentials and select **Identity & Security**, then **Domains**.
2. Select your compartment, then your identity domain.
3. In the navigation bar, select **Oracle Cloud Services**.
4. Find and select your integration instance.

Copyright © 2022, 2026, Oracle and/or its affiliates.

> **Tip**
>
> Make sure the name is the same as the Service instance name you got from the Oracle Integration Console.



5. In your Oracle Integration instance details, select **OAuth Configuration**, and make sure **Allow token refresh: allowed**.



If you don't see **Allow token refresh: allowed**, select **Edit OAuth configuration** and make sure the setting is checked.

6. Configure token refresh expiry (optional).

   Select **Edit OAuth Configuration**. You can configure access token and refresh token expiry times for your Oracle Integration instance.

   You might want to have different refresh expiry settings for development and production instances.

   To determine how to set the duration of your refresh expiry time:

   - Development environments: Determine what is the longest amount of time that you will not be running integrations. For example, do you run tests every two months? If that's the case, set your refresh token expiry time to 90 days.
   - Production environments: Determine what is the longest time interval that you think something is not going to run. For example, if you have a scheduled integration that runs every two weeks, then set your refresh token expiry to be one month.

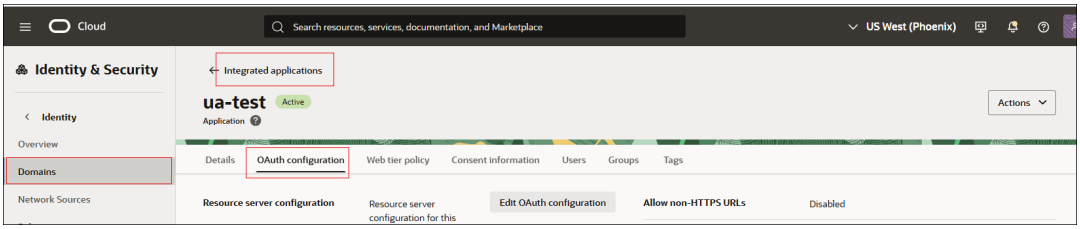
Go to the next step Step 3: Check Token Refresh and Scope Settings in Integrated Applications

Copyright © 2022, 2026, Oracle and/or its affiliates.

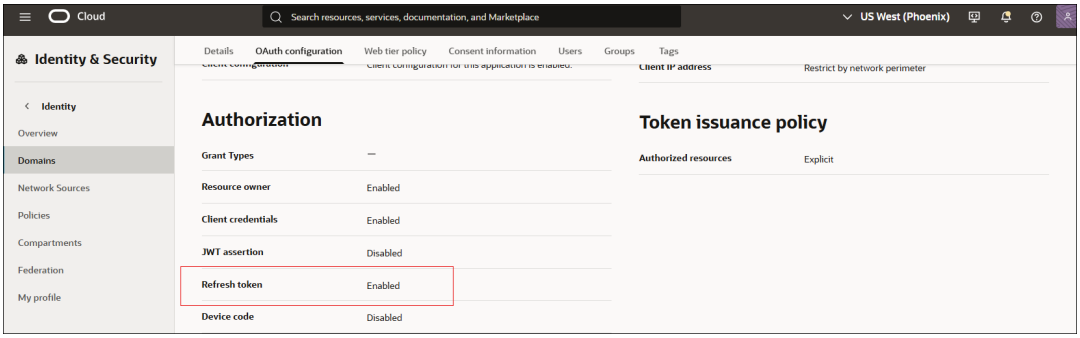
Step 3: Check Token Refresh and Scope Settings in Integrated Applications

Check that Refresh token is enabled in your confidential application and that the scopes assigned to your confidential application match your Oracle Integration instance.

1. Log in to the Oracle Cloud Console with your identity domain administrator credentials and select **Identity & Security**, then **Domains**.
2. Select your compartment, then your identity domain.
3. Select **Integrated Applications**.
4. Find and select the confidential application you created.
5. Select **OAuth Configuration**.

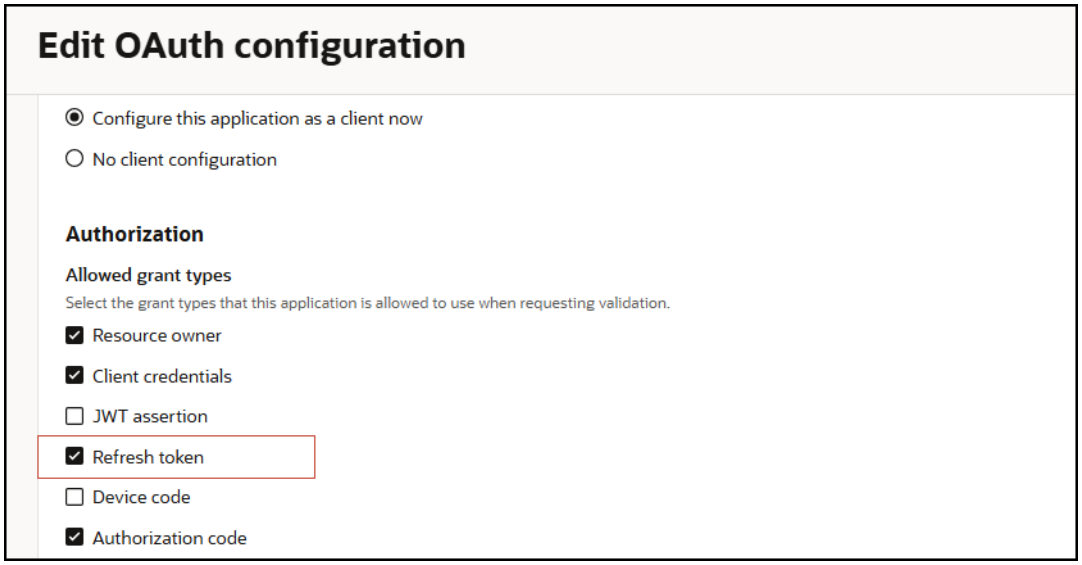


6. In the **Authorization** section of your confidential application details, make sure **Refresh token** is set to **Enabled**.



If **Refresh token** isn't enabled, under **OAuth Configuration**, select **Edit OAuth configuration** and make sure **Refresh token** is checked.

Copyright © 2022, 2026, Oracle and/or its affiliates.

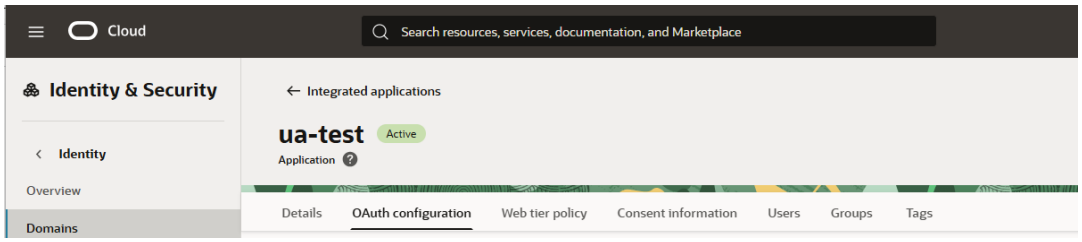


7. Select **Save Changes**.

Check Scopes Match Primary Audience in Your Oracle Integration Instance

Make sure the scopes assigned to your confidential application match the primary audience value in your Oracle Integration instance. This makes sure your confidential application is associated with the correct Oracle Integration instance.

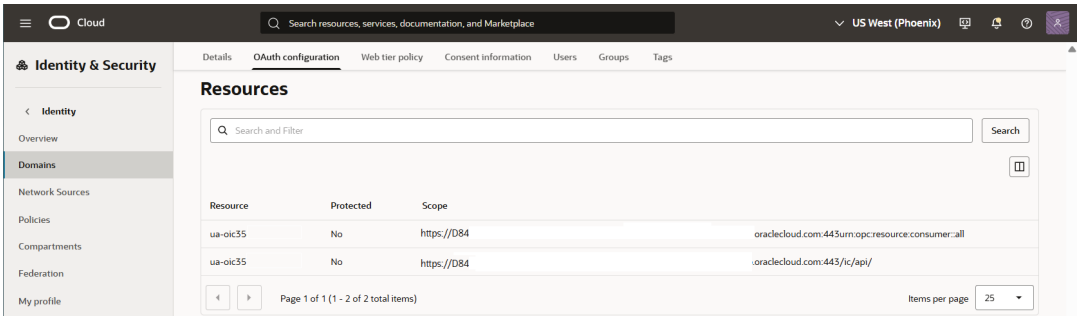
1. Log in to the Oracle Cloud Console with your identity domain administrator credentials and select **Identity & Security**, then **Domains**.
2. Select your compartment, then your identity domain.
3. Select **Integrated Applications**.
4. Find and select the confidential application you created.
5. Select **OAuth Configuration**.



6. Under **Resources**, check the scopes assigned to your confidential application match your Oracle Integration instance.

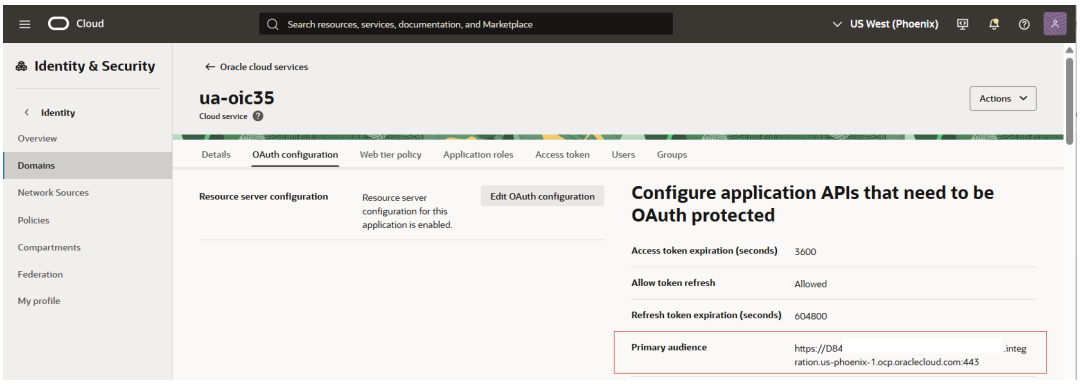
   Scopes in your confidential application:

Copyright © 2022, 2026, Oracle and/or its affiliates.



- Check the **Resource** column in your confidential application matches your instance name in **Identity & Security**, then **Domains**, then **Oracle Cloud Services** .
- Check that the **Scope** column in your confidential application including the value immediately after the https:// matches the value in the **Primary audience** for your instance in **Identity & Security**, then **Domains**, then **Oracle Cloud Services**, **Configure application APIs that need to be OAuth protected** .

Scopes in your Oracle Integration instance:



Unauthorized Errors: Check Roles Assigned to Your Confidential Application

Make sure your confidential application has the correct roles assigned.

If you are making inbound calls to Oracle Integration to trigger an integration:

- Assign the confidential client application the **ServiceInvoker** role.

If you are calling the Oracle Integration Developer APIs:

- The role you assign your confidential client application depends on which APIs you'll be using. If you're creating lookups for example, you'll need the ServiceDeveloper role.
  - **ServiceUser** is the minimum role required to call the Developer APIs. This user has access to the Oracle Integration Console as well as the APIs. Assign the ServiceUser role in most cases. This user can also trigger an integration's endpoint.
  - **ServiceDeveloper** is required if you're invoking the Developer APIs from within an integration, and if you're using the APIs to perform create operations such as creating lookups or importing an integration.

For detailed information on each role, see What Users Can Do in the Integrations Design Section by Role.

Copyright © 2022, 2026, Oracle and/or its affiliates.

- Make sure your client application has the appropriate role. See Validate the Oracle Integration application and user roles.

Test the Connection

Test your connection to ensure that it's configured successfully.

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

If Your Connection...	Then...
Doesn't use a WSDL	The test starts automatically and validates the inputs you provided for the connection.
Uses a WSDL	A dialog prompts you to select the type of connection testing to perform: • **Validate and Test**: Performs a full validation of the WSDL, including processing of the imported schemas and WSDLs. Complete validation can take several minutes depending on the number of imported schemas and WSDLs. No requests are sent to the operations exposed in the WSDL. • **Test**: Connects to the WSDL URL and performs a syntax check on the WSDL. No requests are sent to the operations exposed in the WSDL.

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

Edit a Connection

You can edit connection settings after creating a new connection.

1. Decide where to start:
   - Work in a project.
     a. In the navigation pane, click **Projects**.
     b. Select the project name.
     c. Click **Integrations**.
     d. In the **Connections** section, hover over the connection to edit.
   - Work outside a project.
     a. In the navigation pane, click **Design**, then **Connections**.
     b. Hover over the connection to edit.
2. Click **Actions**, then select **Edit**.

Copyright © 2022, 2026, Oracle and/or its affiliates.

> **Note**
>
> You can also edit a connection from the integration canvas. See Define Inbound Triggers, Outbound Invokes, and Actions.

3. On the Connections page, make any necessary edits. See Configure Connection Properties and Security Properties and Create a Connection with an Agent Group. The connection property and connection security values are specific to each adapter.

   If you edit a connection currently used by an active integration, you can click **Save & Reactivate** to apply the connection changes to active integrations. Schedule integrations with an active schedule are excluded. Any queued or in-progress schedule integration instances are aborted.

4. Test the connection. See Test the Connection.

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. Decide where to start:
   - Work in a project.
     a. In the navigation pane, click **Projects**.
     b. Select the project name.
     c. Click **Integrations**.
     d. In the **Connections** section, hover over the connection to clone.
   - Work outside a project.
     a. In the navigation pane, click **Design**, then **Connections**.
     b. Hover over the connection to clone.
2. Click **Actions**, then select **Clone**.

   The Clone connection dialog is displayed.
3. Enter any necessary connection information.
4. Click **Clone**.

   Cloning a connection does not copy the credentials.
5. Hover your cursor over the cloned connection, then click **Edit** to configure the credentials of your cloned connection.

See Edit a Connection.

Delete a Connection

You can delete a connection that is no longer needed. A connection currently used in an active integration cannot be deleted. Deleting the connection would invalidate the integration. Active integrations would stop running. Remove the connection from the integration and then delete the connection.

1. Decide where to start:

Copyright © 2022, 2026, Oracle and/or its affiliates.

- Work in a project.
  a. In the navigation pane, click **Projects**.
  b. Select the project name.
  c. Click **Integrations**.
  d. In the **Connections** section, hover over the connection to delete.
- Work outside a project.
  a. In the navigation pane, click **Design**, then **Connections**.
  b. Hover over the connection to delete.

2. Click **Actions**, then select **Delete**.

   If the connection is currently used in an active integration, it cannot be deleted.
3. Click **Delete** to confirm deletion.

Unlock a Connection

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

Unlock the connection in either of two ways:

1. Log in again as the same user who was editing the connection when the browser failed.

   For example, if you were logged in as `oicdeveloper` when the browser failed, log in again as `oicdeveloper`.
2. Log out.
   This action unlocks the connection.

or

1. Log in as a user with the Administrator role.
2. Decide where to start:
   - Work in a project.
     a. In the navigation pane, click **Projects**.
     b. Select the project name.
     c. Click **Integrations**.
     d. In the **Connections** section, hover over the connection to unlock.
   - Work outside a project.
     a. In the navigation pane, click **Design**, then **Connections**.
     b. Hover over the connection to unlock.
3. Click **Actions**, then select **Unlock**.
4. Click **Unlock** when prompted to confirm.

Copyright © 2022, 2026, Oracle and/or its affiliates.

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.

> **Note**
>
> The **Refresh Metadata** menu option is only available with adapters that have implemented metadata caching.

1. Decide where to start:
   - Work in a project (see why working with projects is preferred).
     a. In the navigation pane, click **Projects**.
     b. Select the project name.
     c. Click **Integrations**.
     d. In the **Connections** section, hover over the adapter connection to refresh.
   - Work outside a project.
     a. In the navigation pane, click **Design**, then **Connections**.
     b. Hover over the adapter connection to refresh.
2. Click **Actions**, then select **Refresh metadata**.

   If successful, the following message is displayed.

   ```
   Metadata refresh for connection connection_name has been initiated
   successfully.
   ```

Manage Security Certificates

You can manage security certificates in Oracle Integration.

Topics:

- Upload a Certificate to Connect with External Services
- Update or Delete a Certificate

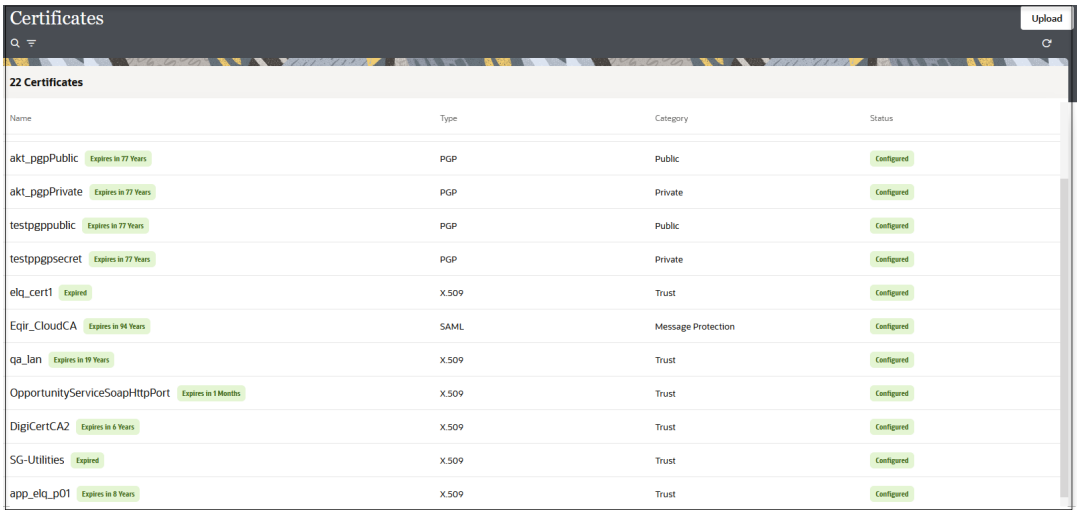
Copyright © 2022, 2026, Oracle and/or its affiliates.

Upload a Certificate to Connect with External Services

Certificates allow Oracle Integration to connect with external services. If the external service/endpoint needs a specific certificate, request the certificate and then import it into Oracle Integration.

If you make an SSL connection in which the root certificate does not exist in Oracle Integration, an exception error 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.

1. Sign in to Oracle Integration.
2. In the navigation pane, click **Settings**, then **Certificates**.
   All certificates currently uploaded to the trust store are displayed on the Certificates page.
3. Click **Filter** to filter by certificate expiration date, status, and type. Certificates installed by the system cannot be deleted.



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

Digital Signature

The digital signature security type is typically used with adapters created with the Rapid Adapter Builder. See Learn About the Rapid Adapter Builder in Oracle Integration in *Using the Rapid Adapter Builder with Oracle Integration 3*.

Copyright © 2022, 2026, Oracle and/or its affiliates.

1. Click **Browse** to select the digital certificate. The certificate must be an X509Certificate. This certificate provides inbound RSA signature validation. See RSA Signature Validation in *Using the Rapid Adapter Builder with Oracle Integration 3*.
2. Click **Upload**.

X.509 (SSL transport)

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

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

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

SAML (Authentication & Authorization)

1. Note that **Message Protection** is automatically selected as the only available certificate category and cannot be deselected. Use this option to upload a keystore certificate with SAML token support. Create, read, update, and delete (CRUD) operations are supported with this type of certificate.
2. Click **Browse**, then select the certificate file (`.cer` or `.crt`) to upload.
3. Click **Upload**.

PGP (Encryption & Decryption)

1. Select a certificate category. Pretty Good Privacy (PGP) provides cryptographic privacy and authentication for communication. PGP is used for signing, encrypting, and decrypting files. You can select the private key to use for encryption or decryption when configuring the stage file action.
   See olink:ICSUG-GUID-4357B1C8-0FBB-482F-9ED4-041ADE1B857F.
   a. **Private**: Uses a private key of the target location to decrypt the file.
      i. Click **Browse**, then select the PGP file to upload.
      ii. Enter the PGP private key password.
   b. **Public**: Uses a public key of the target location to encrypt the file.
      i. Click **Browse**, then select the PGP file to upload.
      ii. In the **ASCII-Armor Encryption Format** field, select **Yes** or **No**.
         - **Yes** shows the format of the encrypted message in ASCII armor. ASCII armor is a binary-to-textual encoding converter. ASCII armor formats encrypted

Copyright © 2022, 2026, Oracle and/or its affiliates.

messaging in ASCII. This enables messages to be sent in a standard messaging format. This selection impacts the visibility of message content.

- **No** causes the message to be sent in binary format.

iii. From the **Cipher Algorithm** list, select the algorithm to use. Symmetric-key algorithms for cryptography use the same cryptographic keys for both encryption of plain text and decryption of cipher text. The following supported cipher algorithms are FIPS-compliant:

- AES128
- AES192
- AES256
- TDES

c. Click **Upload**.

Signing key

A signing key is a secret key used to establish trust between applications. Signing keys are used to sign ID tokens, access tokens, SAML assertions, and more. Using a private signing key, the token is digitally signed and the server verifies the authenticity of the token by using a public signing key. You must upload a signing key to use the OAuth Client Credentials using JWT Client Assertion and OAuth using JWT User Assertion security policies in REST Adapter invoke connections. Only PKCS1- and PKCS8-formatted files are supported.

1. Select **Public** or **Private**.
2. Click **Browse** to upload a key file.
   If you selected **Private**, and the private key is encrypted, a field for entering the private signing key password is displayed after key upload is complete.
3. Enter the private signing key password. If the private signing key is not encrypted, you are not required to enter a password.
4. Click **Upload**.

Update or Delete a Certificate

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

1. In the navigation pane, click **Settings**, then **Certificates**.
2. Use search or filter the list to find the certificate you want to update or delete.
3. Hover over the certificate name, and click **Actions** ⋮.
4. To update the certificate, select **Update**.
   a. Update the certificate as required, such as uploading a new certificate. See Upload a Certificate to Connect with External Services.

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

   b. Click **Update**.

Copyright © 2022, 2026, Oracle and/or its affiliates.

5. To delete the certificate, select **Delete**, and then click **Delete** when prompted to confirm.

Browse and Install Third-Party Adapters

You can browse the Integration Store and install adapters created by third parties with the Rapid Adapter Builder.

The Rapid Adapter Builder transforms the adapter experience in Oracle Integration by allowing you to build an adapter for any application that exposes REST APIs. See Learn About the Rapid Adapter Builder in Oracle Integration in *Using the Rapid Adapter Builder with Oracle Integration 3*.

1. In the navigation pane, click **Home**.
2. Click **Browse store**.
3. Display the third-party adapters:
   a. Click **Adapter** in the header bar.

      or

   b. Select **Adapter** from the **Type** list in the **Filter**, and click **Apply**.

   Adapters created by third-parties are displayed.
4. Browse through the adapters.
5. Click **Get** to install an adapter for the first time or **Get latest** to update an existing installation of this adapter to the latest version.

   The adapter is installed in your instance and available for selection and configuration in the Create connection panel both inside and outside of a project. See Create a Connection.
6. In the navigation pane, click **Design**, then **Adapters**.

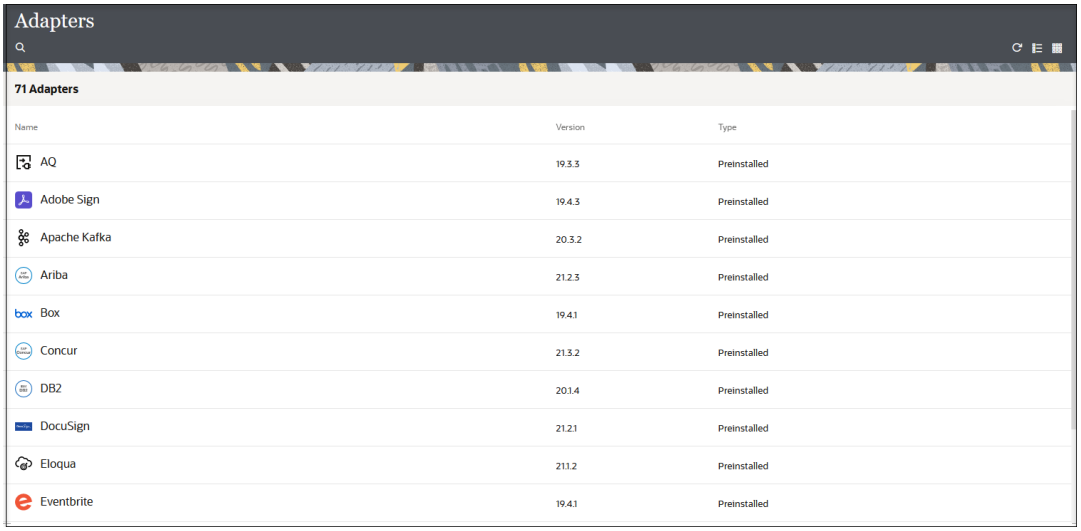
   The adapter you installed is identified in the **Type** column as **Third party**.

View Preinstalled Adapters

You can view the adapters preinstalled with Oracle Integration. You can also create a connection for a selected adapter on this page.

1. In the navigation pane, click **Design**, then **Adapters**.

Copyright © 2022, 2026, Oracle and/or its affiliates.



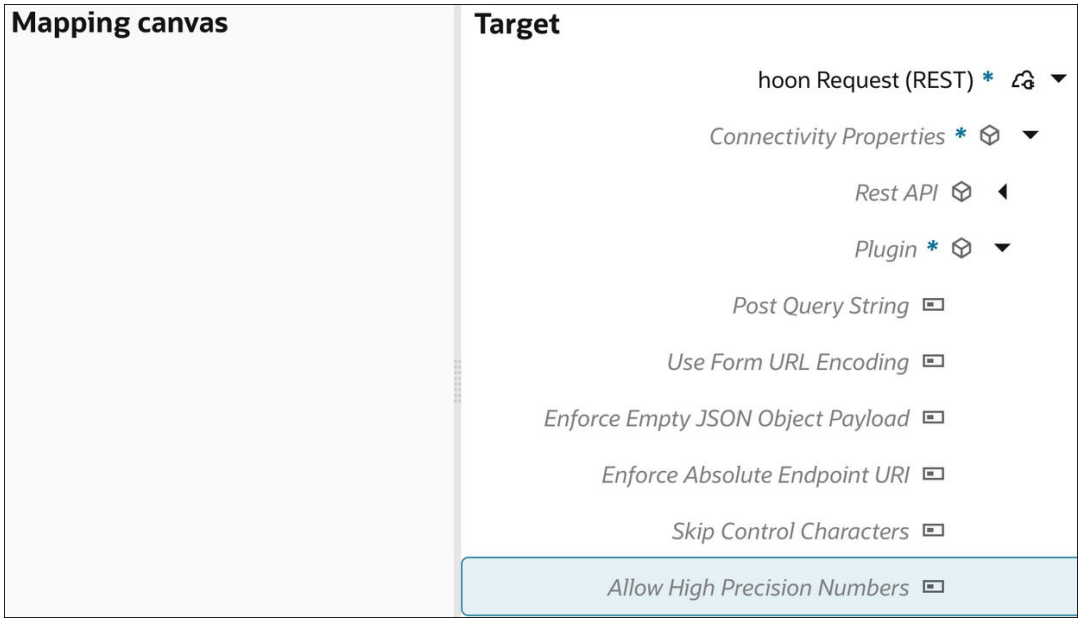
2. Click the adapter name or hover over a row and click **View** to display information about the adapter.
3. To create an adapter connection, hover over its name, then click **Actions** and select **Create Connection**. The Create connection panel is displayed for that adapter.

Allow JSON Numbers with High Precision and Scale

For JSON numbers with high precision and scale, Oracle Integration automatically converts the number to four decimal places. To prevent this conversion, you have several options based on the adapter you are using.

- For the REST Adapter, you must set the **Allow High Precision Numbers** connectivity property in the mapper to allow JSON numbers with high precision and scale.
  The default value for this connectivity property is false. Oracle Integration internally restricts the precision scale to four (decimal places) by default. Set this property to true for a higher precision scale. Use this property value carefully because undefined precision does not mean infinite precision.

Copyright © 2022, 2026, Oracle and/or its affiliates.



- For all other adapters that supply a schema for response processing, the following schema annotation is included.

  ```
  <xs:schema xmlns:nxsd="http://xmlns.oracle.com/pcbpel/nxsd"
  nxsd:version="JSON" nxsd:allowHighPrecisionNumbers="true">
  ```

  If you receive a REST response with more than 10 decimal places, Oracle Integration converts it to XML, but does not round off the values.

  If you do not use this schema annotation, the value is rounded off at four decimal places.

Copyright © 2022, 2026, Oracle and/or its affiliates.

7

Create Application Integrations

You can create business object- or business event-based, application integrations. Application integrations can be request/reply or fire-and-forget types. Application integrations use Oracle BPEL Process Manager capabilities. Oracle BPEL Process Manager enables you to define how a business process that involves web services runs. BPEL messages invoke remote services and orchestrate the running of processes. When designing integrations, you can create multiple routing expressions.

> **Note**
>
> Ensure that you optimize integration design. If an application 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 and Design Best Practices.

Topics:

- Create an Integration
- Design an Application Integration
- Define Inbound Triggers, Outbound Invokes, and Actions
- Add Suggestions, Invokes, and Actions Through an Inline Menu
- Cut and Paste Elements in the Integration Canvas
- Expand and Collapse Actions in the Integration Canvas
- Create Global Variables
- Define Ad-Hoc Maps
- Import a Map File into an Integration
- Add Actions to an Integration
- Add Global Fault Handling to Integrations
- 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 in Messages
- Display Errors and Warnings in an Integration

Create an Integration

Creating an integration includes defining the trigger and invoke application connections, and defining how data is mapped between the two applications. The following procedure provides general instructions for creating an integration, with links to more detailed information for

Copyright © 2022, 2026, Oracle and/or its affiliates.

certain steps. When 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 Manage Lookups for instructions.

To create an integration:

1. Decide where to start:
   - Work in a project (see why working with projects is preferred).
     a. In the navigation pane, click **Projects**.
     b. Select the project name.
     c. Click **Integrations** .
     d. In the **Integrations** section, click **Add** if no integrations currently exist or **+** if integrations already exist.
        The Add integration panel opens.
     e. Click **Create**.
   - Work outside a project.
     a. In the navigation pane, click **Design**, then **Integrations**.
     b. On the Integrations page, click **Create**.
        The Create integration panel appears.
2. Select the type of integration pattern applicable to your business needs. See Understand Integration Styles.
3. Enter the following information:

Field	Description
Name	Provide a meaningful name so that others can understand the integration. You can include English alphabetic characters, numbers, underscores, and dashes in the identifier.
Identifier	Accept the default identifier value. The identifier is the same as the integration name you provided, but in upper case.

Copyright © 2022, 2026, Oracle and/or its affiliates.

Field	Description
Version	Accept the default version number of `01.00.0000`. Or, if you want to change the version number, enter the version using numbers only in this format: `xx.xx.xxxx`. Integrations are uniquely identified by an identifier and version. Note the version format of `xx.yy.zzzz`, where `xx` is the major version and `yy.zzzz` is the minor version. Integrations having the same identifier, but a different major version, can be active at the same time. For example, `INT-A/1.00.0000` and `INT-A/2.00.0000` can be active at the same time. When activating an integration while another integration of the same identifier and same major version is already active, the currently activated integration is deactivated prior to activating the selected integration. For example, if two integrations have the following integration states: • `INT-A/2.00.0000` - Not active • `INT-A/2.10.0000` - Not active Integration `INT-A/2.00.0000` is then activated. • `INT-A/2.00.0000` is now active. • `INT-A/2.10.0000` is not active. Integration `2.10.0000` is then activated. • `INT-A/2.00.0000` is now not active. • `INT-A/2.10.0000` is now active.
Documentation URL	Optionally enter a URL to a design document that provides the details of the integration. The valid URL format is: `http(s)://hostname:port/documentation_path`
Keywords	Enter keywords (tags) to identify the integration. You can search for integrations on the Integrations page using keywords.
Package **Note**: This field only appears if you are creating a globally available integration (that is, outside a project).	Enter a new or existing package name in which to place your integration. When you enter the initial letters of an existing package, it is displayed for selection. See Manage Packages and About Packages.
Description	Provide a meaningful description so that others can understand the integration.
Available to other projects **Note**: This field only appears if you are creating an integration in a project.	Selecting this check box enables this project integration to be called by a parent integration in a different project. If you don't select this check box, this integration can only be called by a parent integration in the same project. See Invoke a Child Integration from a Parent Integration.

4. Click **Create**.

Copyright © 2022, 2026, Oracle and/or its affiliates.

5. See the following sections based on your selection.

If You Selected...	See...
Application	Design an Application Integration
Schedule	Create a Schedule Integration
Event	Subscribe to Events in an Integration

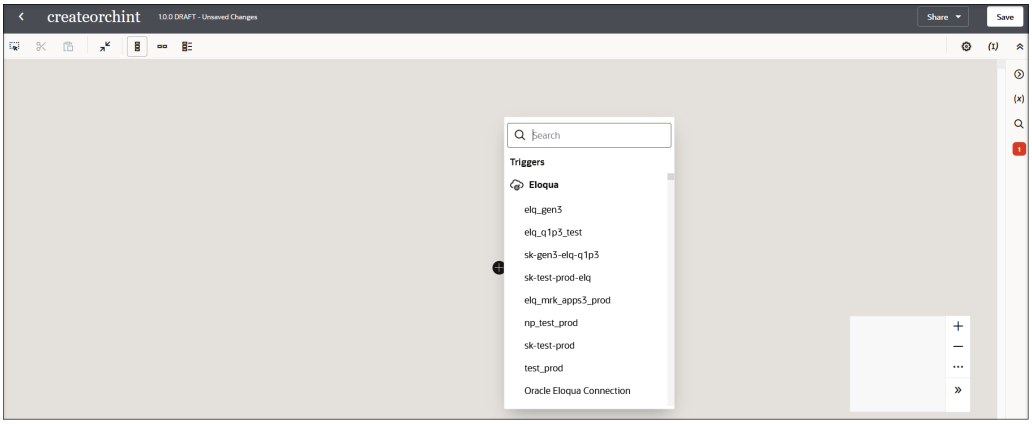
Design an Application Integration

This section describes how to use the integration canvas to design an application integration. It also provides an overview of the integration canvas in which you design your integration.

1. Follow the steps in Create an Integration to create an application integration.

   An empty integration canvas with the following sections is displayed:

   - The empty integration canvas is identified by a **+** sign and a **Triggers** menu of available trigger adapter connections in the middle of the canvas. The **Triggers** menu is automatically displayed the first time. After that, you need to click the **+** sign to invoke an inline menu for designing your integration. Click an adapter to display the number of configured adapter connections available for adding to an integration. This invokes the Adapter Endpoint Configuration Wizard to create an inbound connection in the integration.

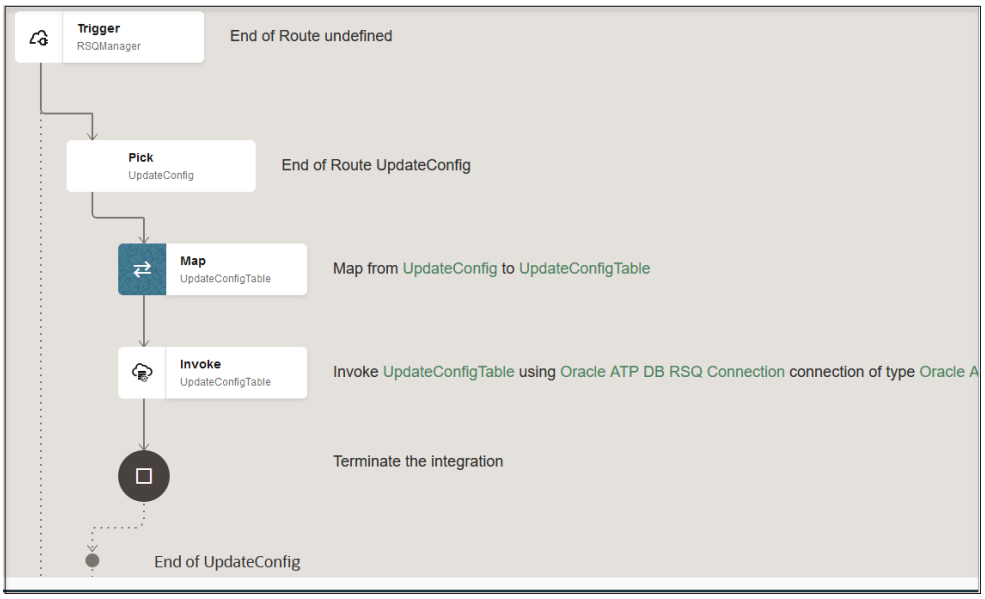


   - You can also select **Triggers** in the right pane to display the same menu of available trigger adapter connections.
   - Click the following items to design the integration:
     - **Selection Mode**: Select elements in the canvas for cutting and pasting.
     - **Cut**: Cut actions and invokes to paste elsewhere in the integration.
     - **Paste**: Paste actions and invokes to other sections of the integration.
     - **Collapse All Collections**: Collapse logic actions such as a scope, for each, switch, while, and others that include child actions.

Copyright © 2022, 2026, Oracle and/or its affiliates.

- **Refresh Endpoints**
  : Click to refresh endpoints in the integration. This icon appears if your integration includes connections that can be regenerated, See Refresh Endpoints for Integrations.

- **Vertical** : Display a vertical view of the integration.
- **Horizontal** : Display a horizontal view of the integration.
- **Outline** : Display the integration vertically with child actions indented. Details about each action in the integration are displayed to the right. You can edit the integration by selecting the **+** sign to display an inline menu for adding invokes and actions. You can also edit actions or add routes to switches by clicking the appropriate action.

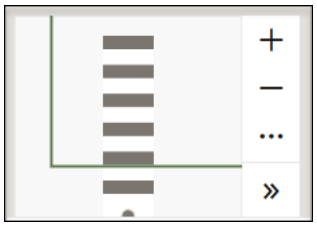


- /: Start and end integration testing from within the integration canvas. These icons only appear for REST Adapter connection trigger-based integrations and scheduled integrations. See Test Integrations from Inside the Integration Canvas.
- **Share** drop-down list: Export the integration or print the design of the integration.
- **Primary Info** : Display a panel showing integration creation properties (identifier, name, version, package, documentation URL, keywords, smart tags, and description), a button to delete the integration. and a report that shows details about the actions and invokes used in the integration. You can change the name, package, documentation URL, and description in this panel.
- **Business Identifiers** : Create business identifiers for the integration. At least one identifier is required. Business identifiers enable you to track integration instances during runtime. See Track Integration Instances.
- **Minimize** and **Maximize** : Minimize to show the options above the integration canvas and maximize to hide the options above the integration canvas.

Copyright © 2022, 2026, Oracle and/or its affiliates.

- **View Global Fault Handler** : View the global fault action and re-throw fault action. You can add and delete re-throw faults and other actions, as necessary.
- **Triggers** : Create inbound triggers for the integration. This icon does not appear after you create the trigger for your integration.
- **Invokes** : Create outbound invokes for the integration.
- **Actions** : Display a pane of action that you can add to the integration. See Define Inbound Triggers, Outbound Invokes, and Actions.
- **Global Variables** : Add a global variable to the integration that can be selected for use in actions.
- **Search** : Search for triggers, invokes, and actions in the integration. You can select the element in the drop-down list or the structure.

At the bottom right of the integration canvas is a small box with icons for adjusting the integration view, including zooming in, zooming out, expanding the integration to 100%, fitting the integration into the canvas, and minimizing/maximizing the box.



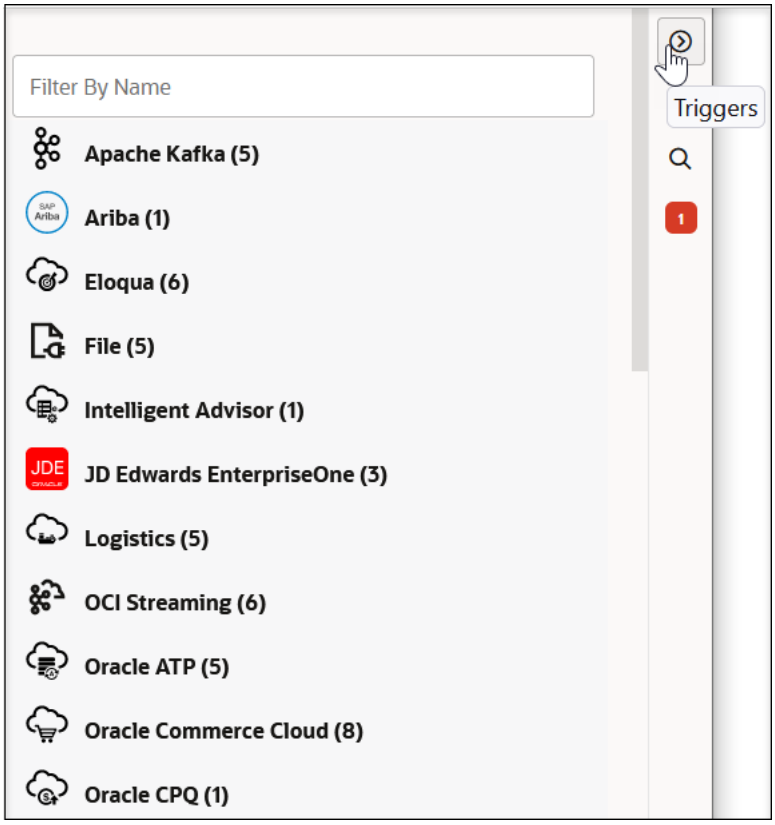
Define Inbound Triggers, Outbound Invokes, and Actions

You define a single inbound trigger and one or more outbound invokes in your integration. You can also add actions to the integration.

1. To add a trigger connection to the integration canvas, click **+** in the integration canvas or click **Triggers** on the side of the canvas. Both options show a list of available trigger connections. This icon disappears once you define the trigger.

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

Copyright © 2022, 2026, Oracle and/or its affiliates.



The Create connection panel is displayed for creating 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. If you edit a connection so that it is now *not* 100% complete, then save and close the Connections page, a message is displayed in the banner indicating that it is not 100% complete and cannot be displayed in the panel. For the connection to be displayed in the panel again, you must return to the Connections page and configure the connection to be 100% complete.

2. Add the configured adapter by clicking **+** in the integration canvas and selecting it from the list.

   The Adapter Endpoint Configuration Wizard is displayed.

3. Complete the pages of the wizard to configure the selected adapter. See Understand Trigger and Invoke Connections.

   When complete, a configured trigger is displayed in the canvas. An unconfigured mapper icon is displayed immediately after the trigger. Because this trigger was configured to send a response in this example, a return icon is displayed in the integration canvas. You *cannot* delete a trigger in an integration; no delete option is available when you click the trigger and select **Actions** ⋮.

> **Note**
>
> On the side of the canvas, the **Triggers** icon is replaced by **Invokes** icon, which enables you to add multiple outbound invoke connections to the integration.

Copyright © 2022, 2026, Oracle and/or its affiliates.

An **Actions**  icon is displayed after **Invokes**  . When expanded, this section shows the following actions that you can add to the integration:

Actions:

- **Assign**: Assign variables and values using the Expression Builder.
- **B2B**: Translate a message to or from the Electronic Data Interchange (EDI) format in the integration. This action converts an EDI document into an Oracle Integration message, or vice verse.
- **Data stitch**: Incrementally build a message payload from one or more existing payloads. The stitch action provides an editor for assigning values to variables.
- **Healthcare**: Exchange messages and events between healthcare applications.
- **Logger**: Log messages to the activity stream and logs by adding the logger action at any point in the integration. You can create a log message that is a static message or variable-populated message in the Expression Builder.
- **Map**: Add mappings to the integration, when needed. Mappings are also automatically added when you add triggers, invokes, and some actions to an integration. In most cases, the messages you transfer between applications in an integration have different data structures. Mappings enable you to define how data is transferred, or mapped, between applications.
- **Note**: 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.
- **Notification**: Send a notification email to relevant users at specific points in the processing 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.
- **Stage File**: Process files in integrations. The stage file action can process each file downloaded by the FTP Adapter. The stage file action can read (and remove any trailer), write, zip, and unzip files in a stage integration known to Oracle Integration.
- **Wait**: Delay the processing of an integration for a specified period of time. A typical use case for this action is to invoke a specific operation at a certain time.

Error Handling:

- **Re-throw Fault**: Send failed messages to the error hospital for further analysis. If the integration contains a defined global fault, the error captured by the re-throw fault action is sent through the global fault and onto the error hospital for analysis. If no global fault is defined, the fault is sent.
- **Throw New Fault**: Generate an error and configure a code, reason, details, and skip condition for that error at a certain point during the processing of an integration.

OCI AI Services:

- **OCI Document Understanding**: Use machine learning and artificial intelligence to process and extract information from documents.
- **OCI Generative AI**: Ask questions and receive responses.
- **OCI Generative AI Agents RAG**: Query diverse enterprise knowledge bases and receive context-specific answers based on the current data.
- **OCI Language**: Perform in-depth text analysis and machine translation to process unstructured text.

Copyright © 2022, 2026, Oracle and/or its affiliates.

- **OCI Speech**: Transcribe natural language speech to text.
- **OCI Vision**: Perform deep-learning–based image analysis to automatically extract textual or visual information from images.

Logic:

- **For each**: Loop over a repeating element and process one or more actions within the scope of the for-each action. The number of looping iterations is based on a user-selected repeating element.
- **Parallel**: Process tasks in parallel to improve integration performance and response times. A parallel action allows the path of an integration to be split into multiple branches.
- **Scope**: Manage a group of actions. The scope action is a collection of child actions and invokes that can have their own fault handlers. The scope action provides the behavior context for the child actions. The actions defined in the parent scope have local visibility inside the scope.
- **Switch**: Define branches to add routing expressions in the integration. You define one or more branches. An otherwise branch is taken if the routing expressions for the initial branches do not resolve to true.
- **While**: Loop over actions or invoke connections while a specific condition is met. You define the condition for the while loop in the Expression Builder.

Call:

- **Decision**: Invoke an activated decision model in an integration. See Design Decision Models in *Using Decisions in Oracle Integration 3*.
- **File server**: Obtain a list of files, write or move a file to File Server, download and read a file from anywhere, and delete a file on File Server.
- **Integration**: Invoke another integration from an integration. When you drag this action into the integration, you are prompted to select an active integration to invoke.
- **Javascript**: Add JavaScript functions to the integration. This enables you to call JavaScript functions from an integration.
- **OCI Function**: Invoke native Oracle Cloud Infrastructure functions.
- **OCI Object storage**: Invoke Oracle Cloud Infrastructure Object Storage.
- **Process**: Invoke a process from an integration. When you drag this action into the integration, you are prompted to select a process application and a process to invoke.
- **Publish event**: Publish an event in an integration to which subscription integrations can subscribe. See About Publishing and Subscribing to Events.
- **Robot flow**: Add a robot for the integration to call. See Design an Integration That Calls a Robot in *Using Robots in Oracle Integration 3*.

End:

- **Callback**: 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.
- **Fault return**: Return a fault. If the trigger is synchronous with at least one fault, the fault return action is available. If a trigger defines multiple faults, you are prompted to select a fault. If the trigger is asynchronous with at least one fault, the fault return action is available. Once the fault is added, a mapper with the type of fault definition is also added. This action is valid only if the trigger has at least one fault.

Copyright © 2022, 2026, Oracle and/or its affiliates.

- **Return**: Return an immediate response.
- **Stop**: Terminate the integration. No response message is returned to the trigger.

4. On the 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. See Step 1 for information.
5. Click the adapter type (for example, a REST Adapter) to display the configured connections for that adapter.
6. Drag the specific configured adapter connection to the **+** sign in the integration canvas.
7. Drop the adapter.

   This invokes the Adapter Endpoint Configuration Wizard.
8. Complete the pages of the wizard to configure the invoke connection.

   When complete, a configured invoke connection is displayed in the integration 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 integration except for the system-generated reply or stop element 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 maps and other actions being removed.

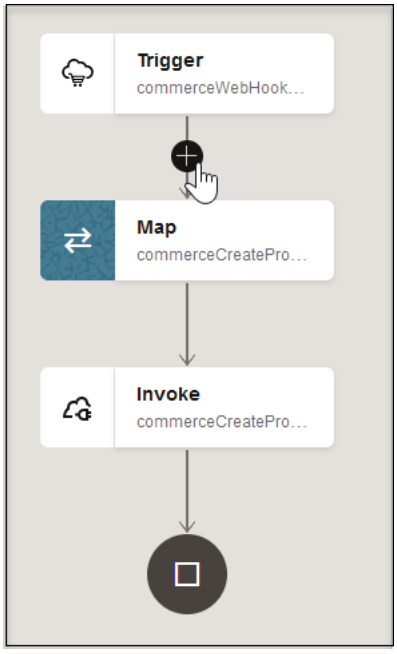
   You can place your cursor anywhere in the canvas to move the integration, when needed. You can also move the integration from within the large box in the upper right corner.

Add Suggestions, Invokes, and Actions Through an Inline Menu

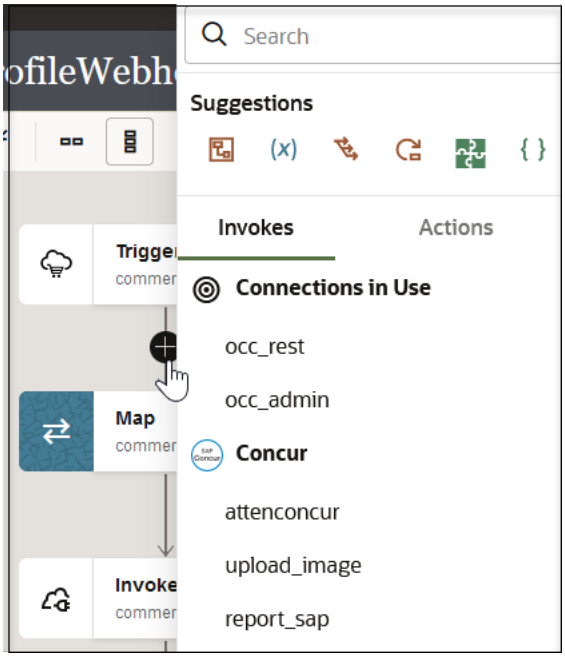
As an alternative to dragging suggestions, invoke connections, and actions from the right navigation pane, you can hover your cursor over the **+** sign that is displayed between the elements in your integration. When you click the **+** sign, a list of actions and invokes available for adding to your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

1. Hover over the lines between the elements to display a **+** sign.
2. Click the **+** sign.

Copyright © 2022, 2026, Oracle and/or its affiliates.



A menu with suggestions, invokes, and actions is displayed.



Element	Description
Suggestions	Displays actions that are commonly placed between the two elements where you clicked.
Invokes	Displays adapter invoke connections available to add. The option is only displayed when a connection can be added after the previous element.

Copyright © 2022, 2026, Oracle and/or its affiliates.

Element	Description
Actions	Displays actions that are allowed to be placed between the two elements where you clicked.

3. Click the **Invokes** tab or **Actions** tab.
4. Select the appropriate invoke or action to add to your integration or begin entering the action type (for example, a while action) or invoke type (for example, a SOAP Adapter) in the **Search** field 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.

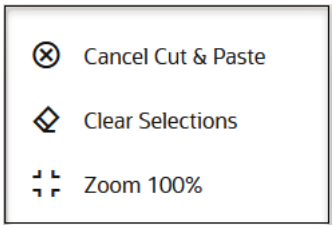
Cut and Paste Elements in the Integration Canvas

You can cut and paste elements (actions and invokes) in the integration canvas. This enables you to move elements to other places in your integration and eliminates the need to recreate them. The cut and paste feature replaces the reposition feature in Oracle Integration Generation 2.

- Guidelines
- Options for Cutting and Pasting Elements
- Example: Cut and Paste a Single Element
- Example: Cut and Paste Multiple Elements

Guidelines

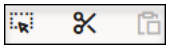
- You can cut and paste individual elements or multiple contiguous elements.
- You can cut and paste more logically complex actions such as a scope action and all the elements within its container or branching actions such as a switch action that includes multiple branching actions such as for-each actions inside it.
- You cannot cut a trigger, either individually or as part of a group of selected elements.
- You cannot cut an end node.
- You cannot cut branches from some actions, such as a routing branch in a switch action.
- If you cut an element and then decide you don't want to paste it, press the **Esc** key or right-click anywhere in the integration canvas and select **Cancel Cut & Paste**. This cancels the cut action.



Options for Cutting and Pasting Elements

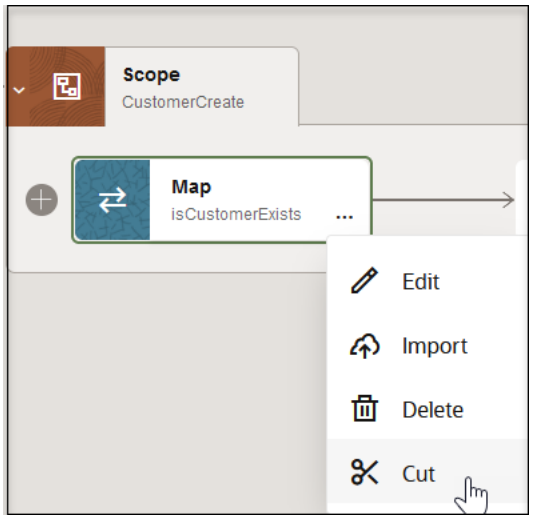
You can cut and paste elements in the integration canvas with toolbar options above the integration canvas or with inline menus.

Copyright © 2022, 2026, Oracle and/or its affiliates.



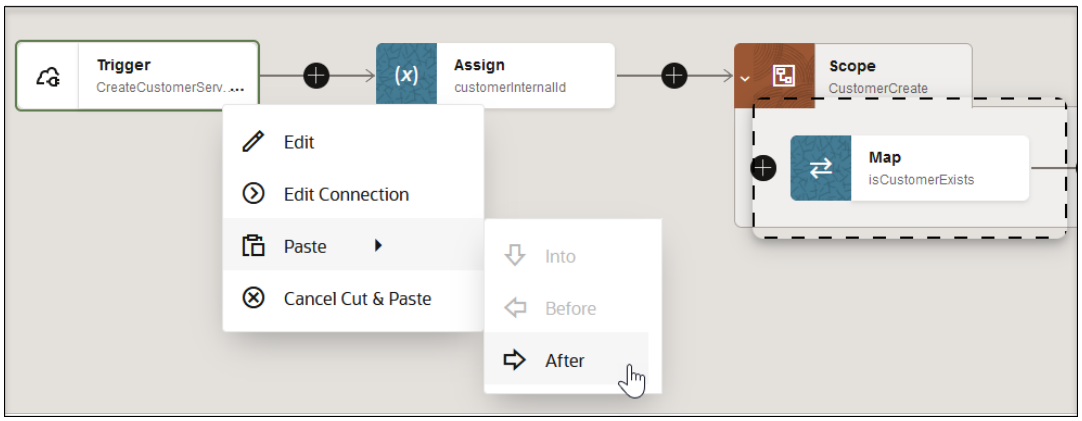
You can cut elements in multiple ways:

- Select a single element or multiple elements (with the **Selection Mode** option) in the integration canvas and click in the toolbar.
- Select a single element or multiple elements in the integration canvas, then click **Actions** and select **Cut** for the selected element or elements.



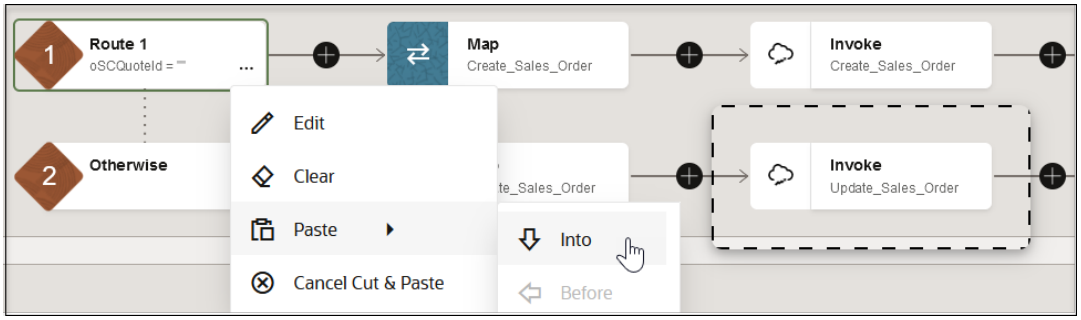
You can paste elements in multiple ways:

- Select the element either before or after the location at which you want to paste it, then click **Actions** and select **Paste**, and either **Before** or **After** (depending on which element you selected).

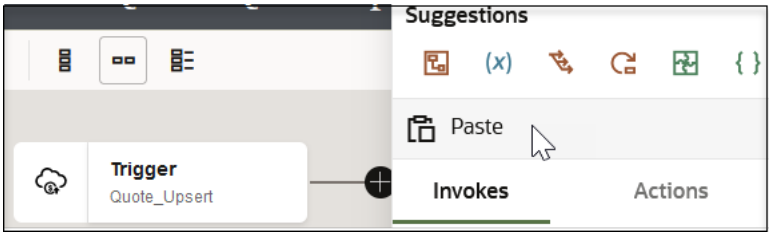


- Paste inside an action such as a scope or a branch of a switch action by clicking **Actions** and selecting **Paste**, and then **Into**.

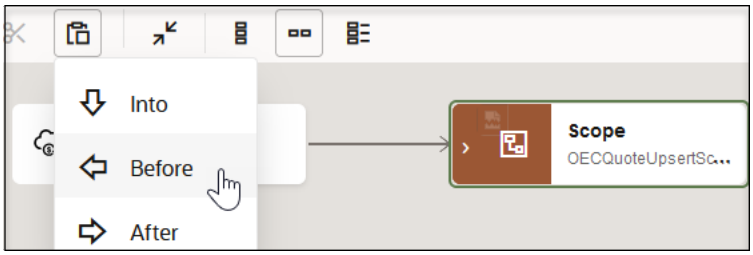
Copyright © 2022, 2026, Oracle and/or its affiliates.



- Click **Insert** + in the location at which to move the element, then select **Paste**.



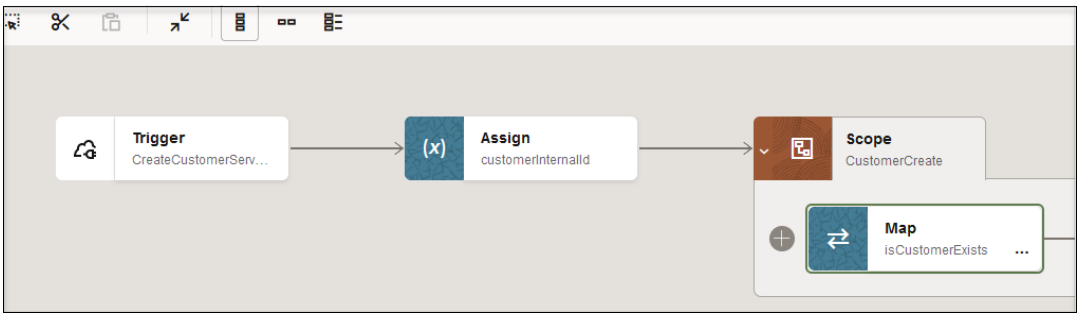
- Click **Paste** above the toolbar to paste the selected element in the appropriate location. The selections you make in the integration canvas determine when the toolbar options are enabled.



Example: Cut and Paste a Single Element

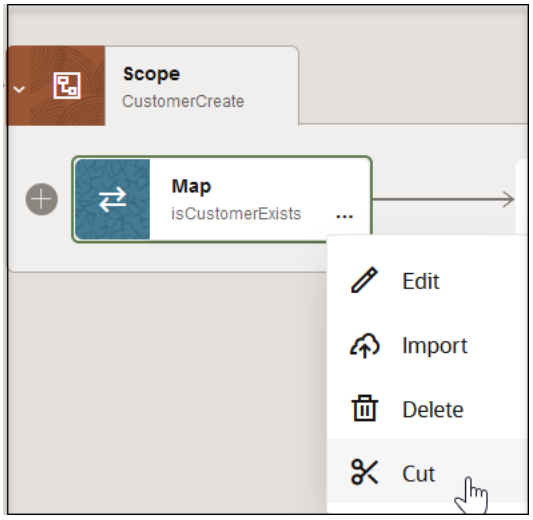
This example shows one way to cut and paste a single element. Other cut and paste methods are described in Options for Cutting and Pasting Elements.

1. Click the element to move. For this example, a map action is selected to move. This action enables **Cut** in the toolbar.



Copyright © 2022, 2026, Oracle and/or its affiliates.

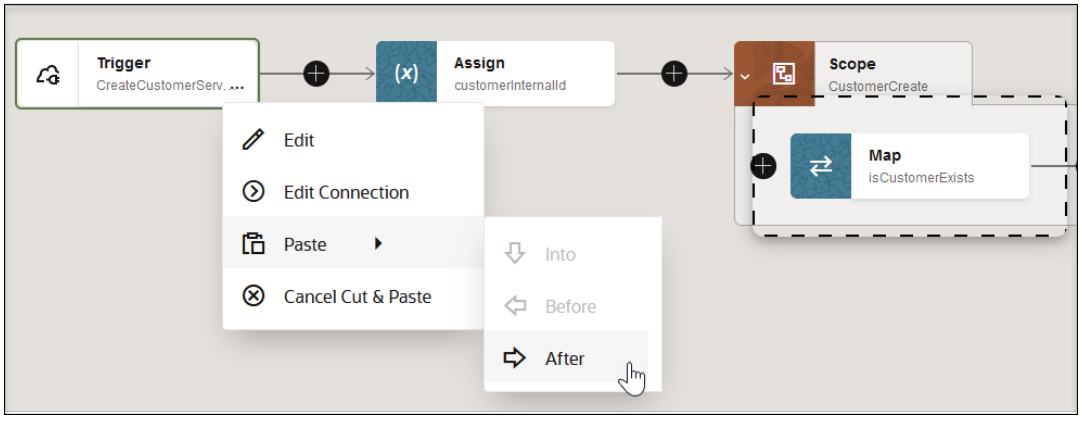
2. In the selected action, click **Actions** ⋮ and select **Cut**.



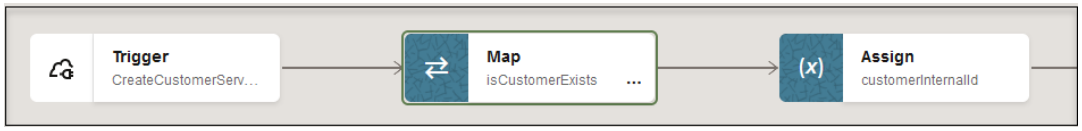
An animated dashed box is displayed around the action to cut. You cannot click inside this box and perform any actions. You can also expand the number of elements to cut by dragging the dashed box around other elements.

3. Select the element next to where you want to paste the element.

4. Click **Actions** ⋮ and select **Paste**, then **After**.
For this example, the trigger element is selected. As an alternative, you can also select the assign action and paste the map action before it. The menu context changes based on where you select to paste the element.



The map element is pasted to its new location after the trigger and before the assign action.

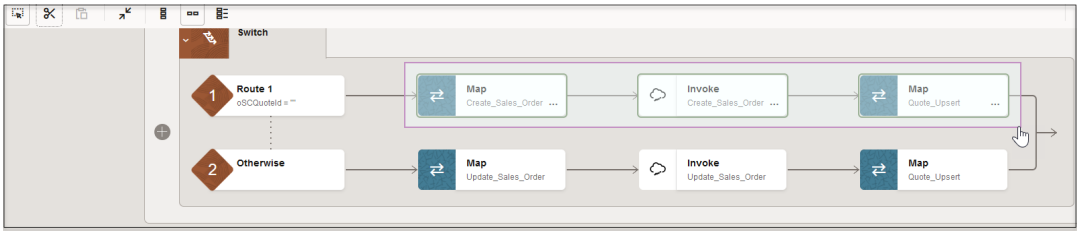


Copyright © 2022, 2026, Oracle and/or its affiliates.

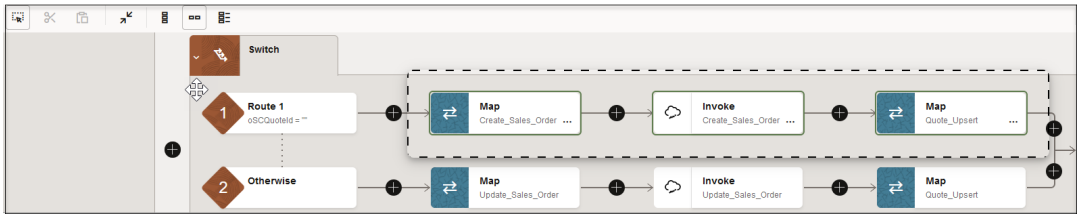
Example: Cut and Paste Multiple Elements

This example shows one way to use cut and paste multiple elements. Other cut and paste methods are described in Options for Cutting and Pasting Elements.

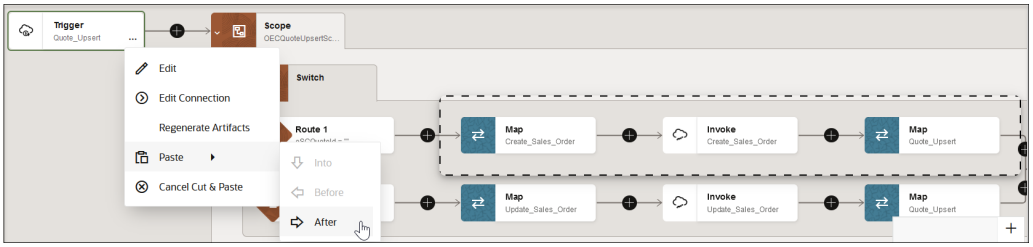
1. Click **Selection Mode** in the toolbar.
2. Drag your cursor around the elements to cut. For this example, a map, an invoke, and another map are selected.



3. Click **Cut** in the toolbar.
An animated dashed box is displayed around the elements to cut. You cannot click inside this box and perform any actions.

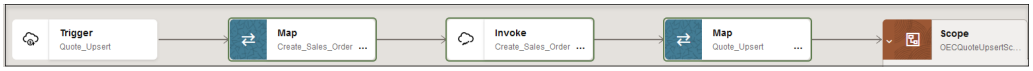


4. Select the element next to where you want to paste the element. For this example, the trigger is selected.
5. Paste the selected elements after the trigger in either of two ways:
   a. Click **Actions** and select **Paste**, then **After**.
   or
   b. Click **Paste** in the toolbar, then select **After**.



The map, invoke, and map are pasted to their new location after the trigger and before the scope action.

Copyright © 2022, 2026, Oracle and/or its affiliates.



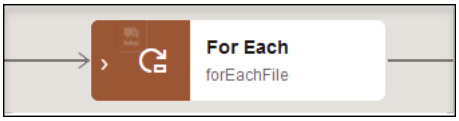
Expand and Collapse Actions in the Integration Canvas

You can expand and collapse actions that include other actions and invokes in the integration canvas. This includes logic actions such as a scope, for-each, while, and switch; stage file actions using the read file in segments operation; and parent integrations that invoke child integrations.

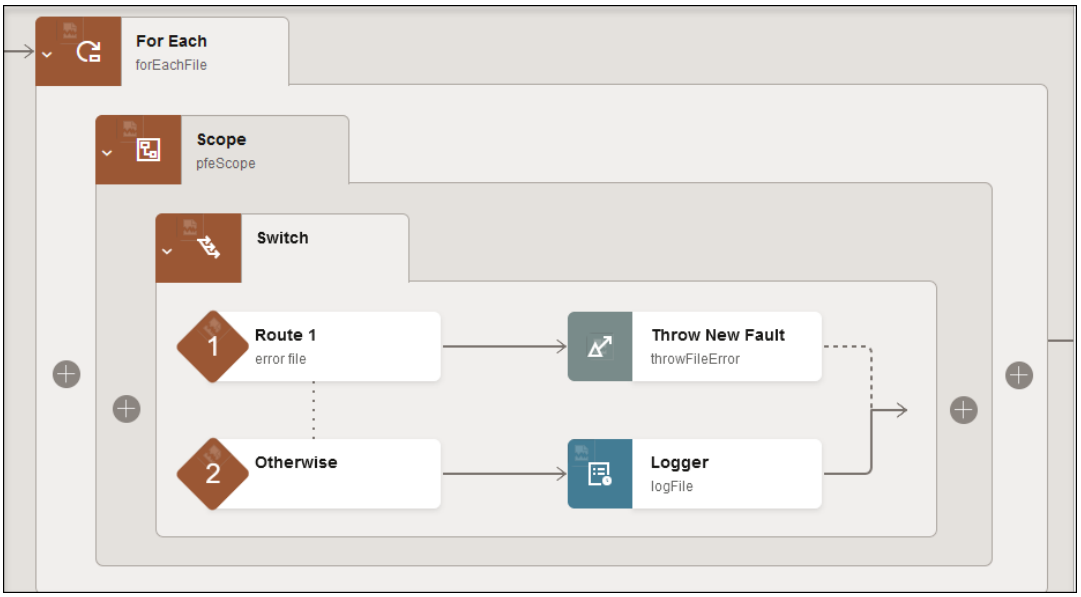
The following methods enable you to toggle back and forth between expanded and collapsed actions:

- Double-click an action.
- Click the **expand** (**v**) and **collapse** (**>**) icons on an action.
- Click **Actions** ⋮ and select **Expand** or **Collapse**.

1. Expand the action (for this example, a for-each action) through one of the methods described above.



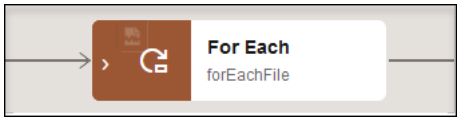
The action is expanded to show any invokes and actions inside. In this example, the for-each action includes a switch action, which is also automatically expanded.



2. Collapse the action through one of the methods described above.

   The action and all of its contents are collapsed.

Copyright © 2022, 2026, Oracle and/or its affiliates.

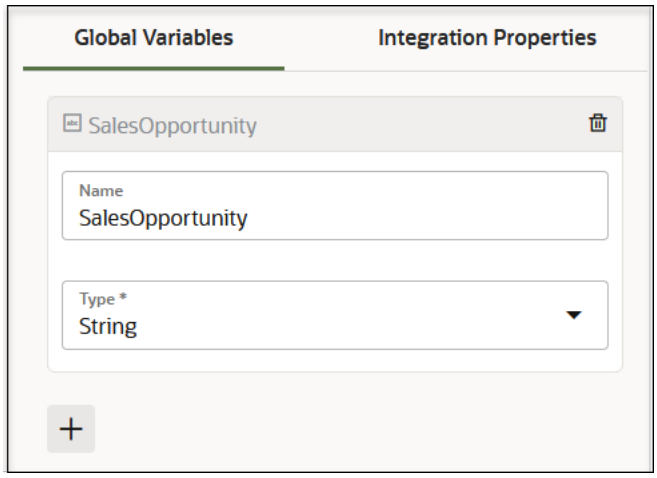


Create Global Variables

You can create complex and simple type global variables that are available for use throughout an integration (for example, when building an expression in an assign action).

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

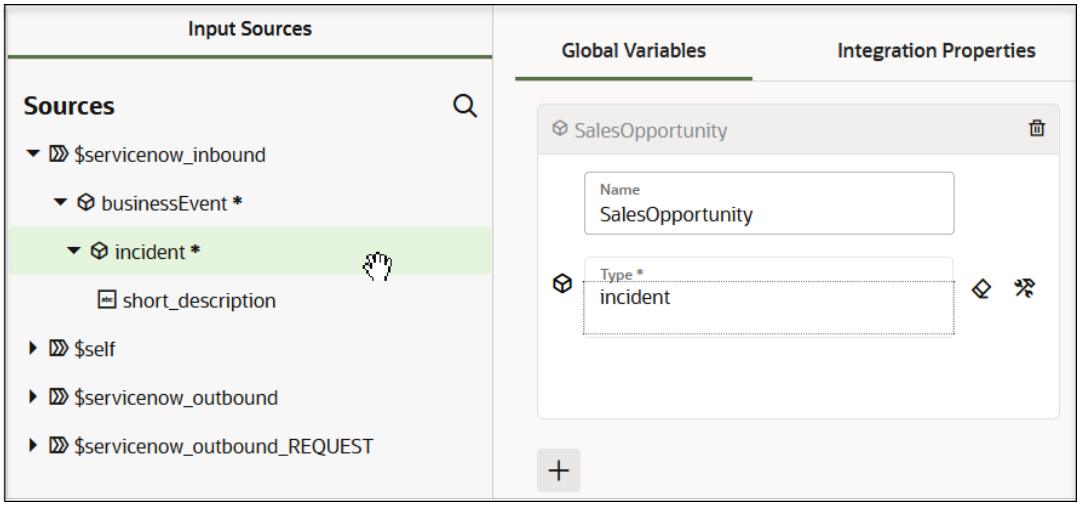
1. On the right side of the integration canvas, click **Global Variables** (x).
2. Click **Add Variable** if no global variables currently exist. Otherwise, click **+**.
3. Enter a name and select the data type from the drop-down list. The **Object** data type is considered a complex type.



You can also use integration properties as global variables. Integration properties are user-defined properties that can be used as read-only data by actions in an integration (for example, by notifications and logs). Integration properties are external to the integration. For example, an email address used by a notification action in an integration can be externalized as a sendTo integration property. This enables the property to be configured without the need to edit the integration or pass the value at runtime.

4. If you add a global variable that is an **Object** data type, the panel is expanded on the left to display a **Sources** tree with all the elements from the request object.
5. Expand the tree and drag an element (root/child) to the **Type** field. This marks the current global variable type as a selected node type. The **Type** list selection changes from **Object** to the node name that you selected.

Copyright © 2022, 2026, Oracle and/or its affiliates.



Note the following complex data type design rules:

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

6. Click **+** to add more global variables. You can add a maximum of 20 global variables.
7. Click **Save** when complete.
8. If you want to edit a global variable, click **Delete** to delete the variable, then recreate the variable. You cannot edit global variables.

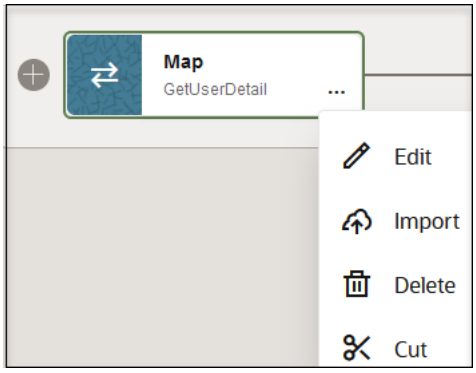
   The global variables you created are now available for selection in other parts of the integration canvas, such as in an assign action.

Define Ad-Hoc Maps

You can add ad-hoc maps to an integration, when needed. You can also delete the maps that are automatically created when you add a trigger or invoke to the integration, if they are not needed.

1. Delete any unneeded mappers by clicking **Actions** ..., then selecting **Delete**.

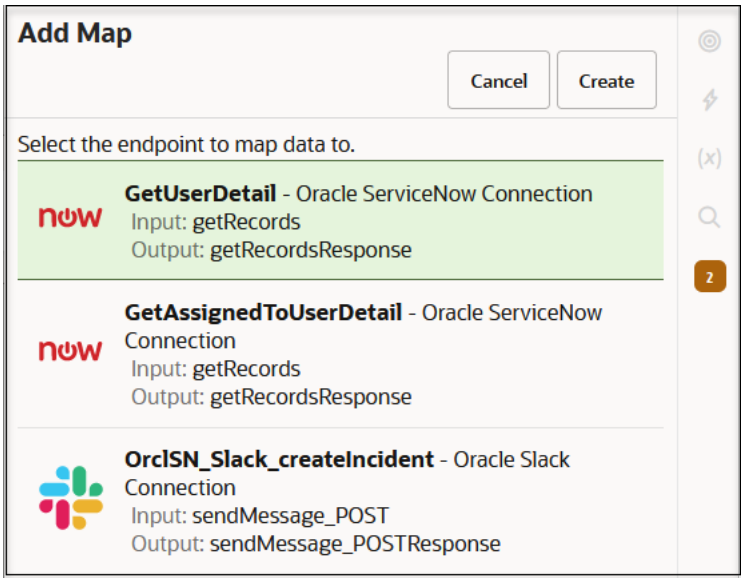
Copyright © 2022, 2026, Oracle and/or its affiliates.



2. Add a map action to an integration in either of the following ways:

   - On the side of the canvas, click **Actions** and drag the **Map** action to the appropriate location.
   - Click + at the location where you want to add the map action, then select **Map**.

   The Add Map dialog is displayed.



3. Select the endpoint to which to map.

   The mapper is displayed.

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

Copyright © 2022, 2026, Oracle and/or its affiliates.

Import a Map File into an Integration

You can import an XSL map file that was previously exported from the *same* integration. This action overwrites the existing mapping file.

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

1. Click the map action into which you want to import the file.
2. Click **Actions** ⋮, then select **Import**.
3. 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 that includes the map file into Oracle Integration.

Add Actions to an Integration

You add actions as part of integration design. Actions perform the activities in an integration. Actions are categorized as follows:

Topics:

- Actions Category
- OCI AI Services Category
- Logic Category
- Call Category
- Receive Requests for Multiple Resources in a Single REST Adapter Trigger with a Pick Action

Actions Category

The Actions category enables you to add assign, B2B, data stitch, healthcare, logger, map, note, notification, stage file, and wait actions to an integration.

Topics:

- Assign Values to Scalar Variables in an Assign Action
- Translate an EDI Document with the B2B Action
- Build Complex Assignment Statements with a Data Stitch Action
- Convert HL7 Messages with a Healthcare Action
- Log Messages with a Logger Action
- Add Placeholder Notes with a Note Action
- Send Notification Emails During Stages of the Integration with a Notification Action
- Process Files in Schedule Integrations with a Stage File Action
- Delay Integration Processing for a Specified Time Period with a Wait Action

Copyright © 2022, 2026, Oracle and/or its affiliates.

Assign Values to Scalar Variables in an Assign Action

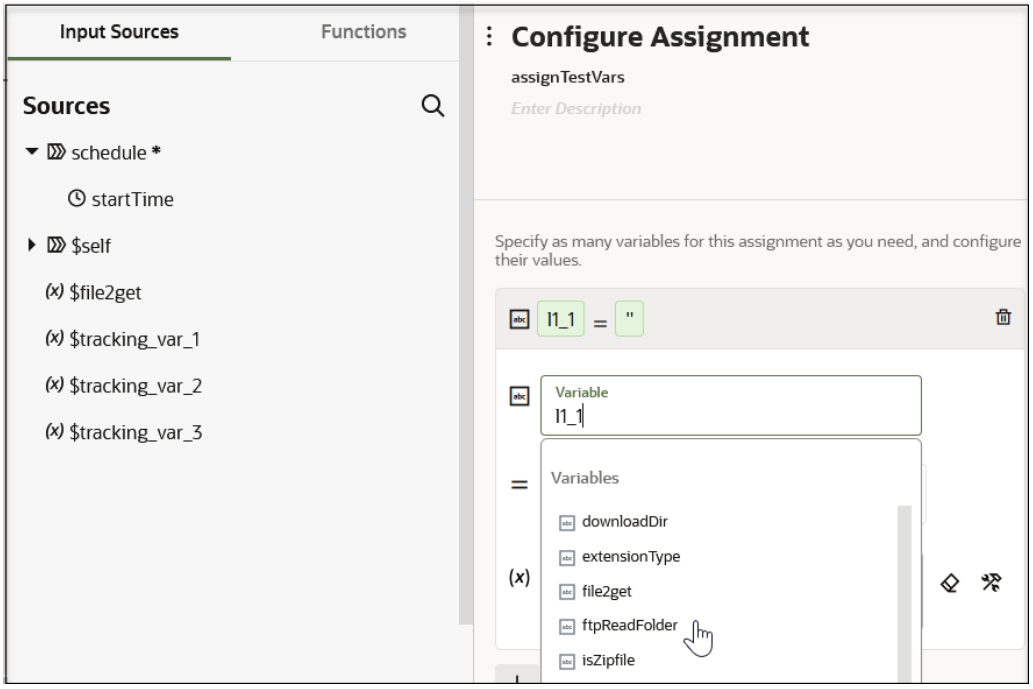
You can assign values to scalar variables in integrations with an assign action.

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

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

   The Configure Assign panel opens.
2. Click **Edit** under **Configure Assign** to enter a name and optional description for the assign action.
3. Click the **+** sign, then select a data type:
   - **String**: Create a string data type.
   - **Fault**: Catch and aggregate faults in parallel processing actions in schedule integrations. This operation is only supported in schedule integrations.
4. If you select **String**:
   a. Enter a string variable name or select an existing variable from the list.

Copyright © 2022, 2026, Oracle and/or its affiliates.



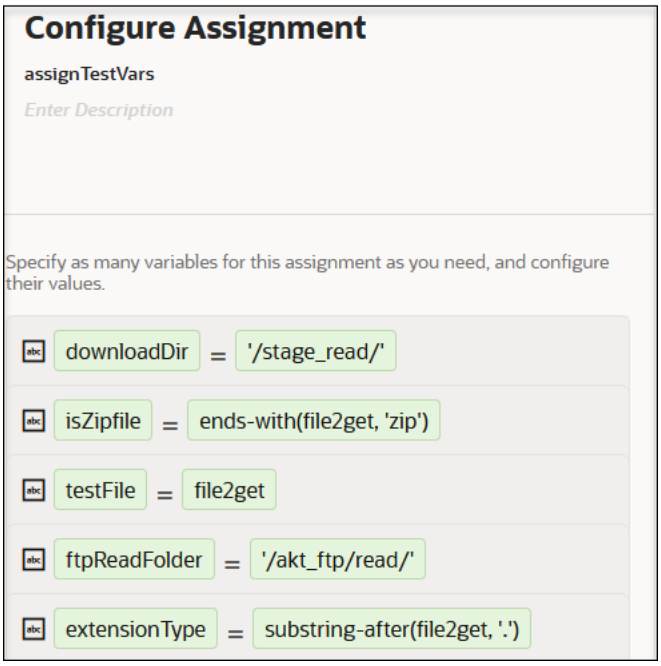
The **Operations** field defaults to an equal sign (**=**) and cannot be changed.

b. Enter a value between the single quotes.

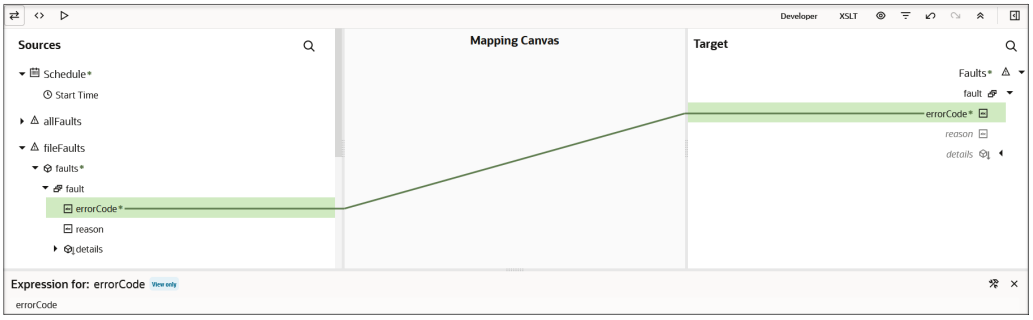


c. Add more string variables, when needed.

Copyright © 2022, 2026, Oracle and/or its affiliates.



5. If you select **Fault**:
   a. Enter a fault variable name or select an existing fault from the list.
   b. Select an operation from the **Operation** list.
      - **Append**: Adds a new fault to the end of the fault list.
      - **Clear**: Removes all entries in the fault list.
   c. If you selected **Append**, click **Edit** to select a fault from the mapper in the **Value** column. This option is not available with the **Clear** operation.



6. Add more fault variables, as needed.
7. Click **Save**.

Variable assignments can be of great complexity. You can use assignments in switch actions and maps. For example, the upper branch of a switch action is taken (if **$FetchContactAssign = "failed"**), which processes a **contactCreateProcessing** scope. Otherwise, a **contactUpdateSkip** scope is processed.

You can also configure the primary tracking variable and both custom field tracking variables (update and access values). You can map tracking variables to output variables or create complex expressions for an assign or switch activity.

Copyright © 2022, 2026, Oracle and/or its affiliates.

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

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

- Retrieve the Name and IDs from Account Records with the queryMore Operation in *Using the Salesforce Adapter with Oracle Integration 3*
- Design a Basic Asynchronous CRUD Operation in *Using the Oracle NetSuite Adapter with Oracle Integration 3*
- Use the Extract API in a Schedule Integration in *Using the SAP Concur Adapter with Oracle Integration 3*
- Use the Pagination Concept in an Integration in *Using the Jira Adapter with Oracle Integration 3*

Translate an EDI Document with the B2B Action

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

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

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

   The Configure B2B Action Wizard is displayed.
2. Enter a name and optional description.
3. Select how you want to use the B2B action, then click **Continue**.
   - **B2B Trading Partner mode**: Select to include B2B trading partner profiles and message persistence in your integration.
   - **Standalone mode**: Select to use the B2B action independently in your integration *without* a B2B trading partner profile or message persistence.
4. If you select **B2B Trading Partner mode**, follow these wizard steps:
   a. Select the B2B message direction for this B2B action:
      - **Inbound**: The B2B message is sent from the trading partner to the host trading partner (partner where Oracle Integration is installed).

Copyright © 2022, 2026, Oracle and/or its affiliates.

- **Outbound**: The B2B message is sent from the host trading partner (partner where Oracle Integration is installed) to the trading partner.

b. Select the operation for the B2B action to perform based on the direction.

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

c. If you select either of the following, the Select Data Format page is displayed.

- **Inbound** message direction and **Fetch Message** operation
- **Outbound** message direction and **Translate** operation

i. Select the document definition for the B2B action to handle, and click **Search** to refresh the page.

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

iii. Click **>**.

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

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

a. Specify the message translation and document format details.

Element	Description
Select the direction in which to translate the message	• **Inbound EDI message to Oracle Integration message**: When an integration receives an EDI document from a business partner, it is considered an inbound document (an EDI document is translated to XML). • **Oracle Integration message to outbound EDI message**: When an integration sends an EDI document to a business partner, it is considered an outbound document (an EDI document is generated from XML).

Copyright © 2022, 2026, Oracle and/or its affiliates.

Element	Description
Document Standard	Select the document standard to follow when exchanging business documents between partners. • **Custom** • **Delimited** • **EDIFACT** • **FixedLength** • **OAG** • **X12** • **X12HIPAA**
Document Version	Select the version of the document standard to use.
Document Type	Select the document type (for example, purchase order, invoice, shipping notice, or others). The document types available for selection are based on the document version you selected.
Document Definition	Select the document definition that you created on the Documents page. For **EDIFACT**, **X12**, or **X12HIPAA**, you can also select **Standard**. See Create Custom Document Definitions in *Using B2B for Oracle Integration 3*.
EDI Character Encoding	Select the character encoding that the inbound EDI document is expected to use.
Perform validations on input data	• **Yes**: Validates the structure and data of an inbound EDI message. Enabling message validation has an impact on performance. If errors are found, translation does not succeed. • **No**: Errors are ignored during translation and the message is passed through in its current format. You can override this setting at runtime with the **Validate** request element in the mapper.

b. Click **Continue**.

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

Element	Description
Run a test translation with sample data (optional)	Select or drag and drop a sample file. To test inbound EDI message translation, upload an EDI document. To test outbound EDI message generation, upload an XML document.
Translate	Click to translate your sample data. Output is displayed in the **Output of translation** field. Any errors are displayed in the **Error in translation** field.

d. Click **Continue**.

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

Build Complex Assignment Statements with a Data Stitch Action

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

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

Copyright © 2022, 2026, Oracle and/or its affiliates.

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

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

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

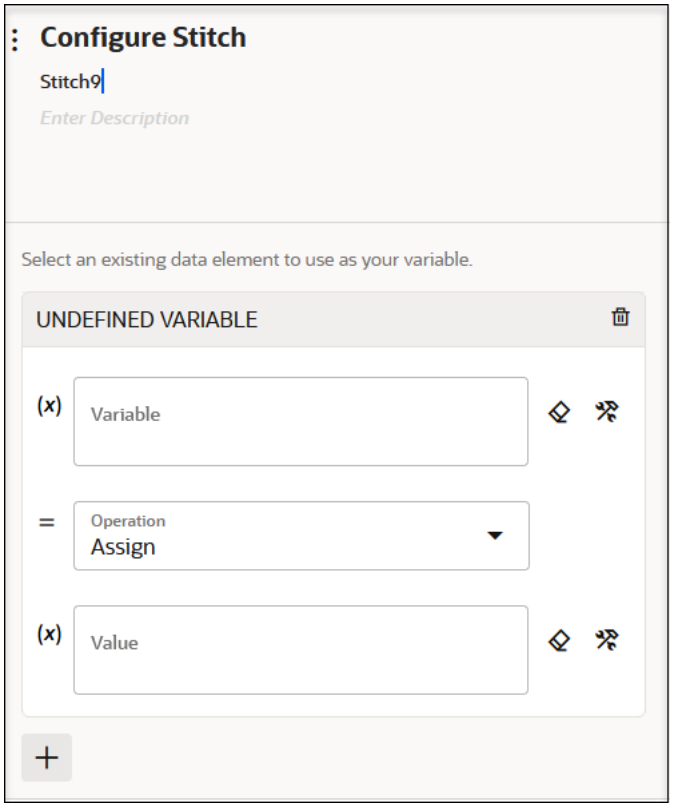
- Append: Appends at the end of the repeating/unbounded target element, the selected element, or the value. For example, you have an existing purchase order payload containing five lines and want to add a sixth line. The data stitch action enables you to append a sixth line to the existing array of lines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

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

   - On the side of the canvas, click **Actions** and drag the **Data stitch** action to the appropriate location.
   - Click + at the location where you want to add the data stitch action, then select **Data stitch**.

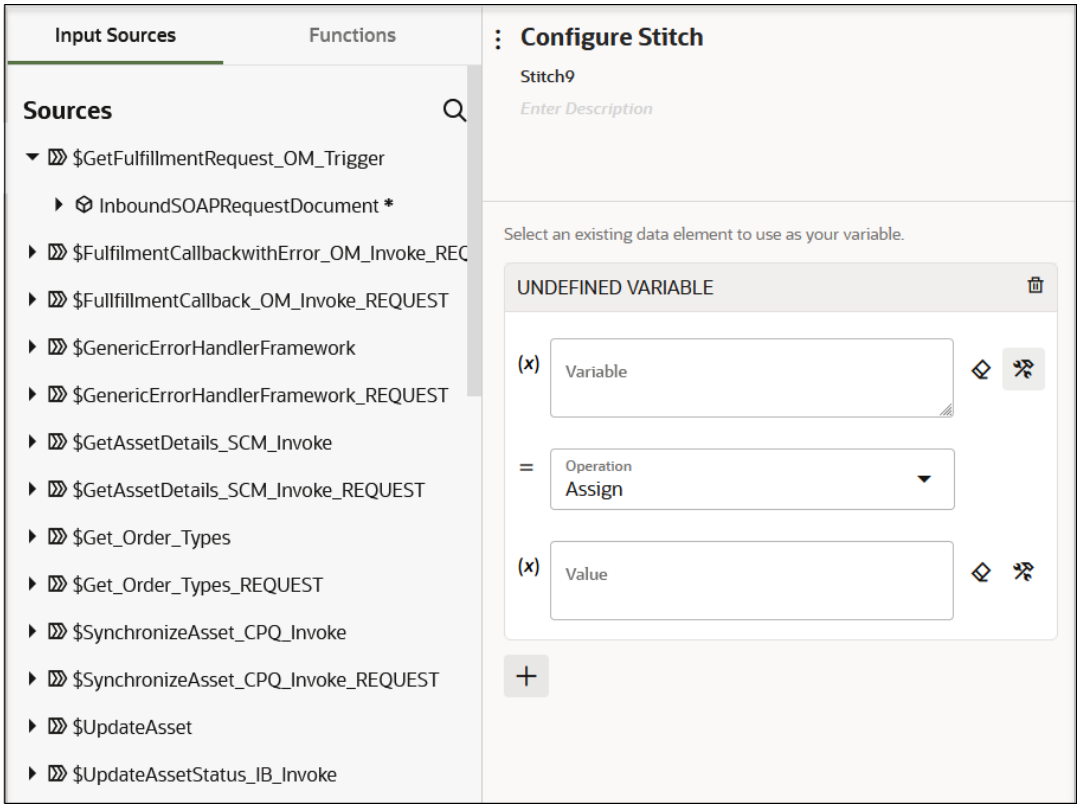
   The Configure Stitch panel opens.

Copyright © 2022, 2026, Oracle and/or its affiliates.



2. Click **Edit** under **Configure Stitch** to enter a name and optional description for the data stitch action.
3. Click the **Variable** field and manually enter a variable name, select a variable from the drop-down list, or click **Switch to Developer View** to open the **Sources** tree for dragging a variable name.

Copyright © 2022, 2026, Oracle and/or its affiliates.

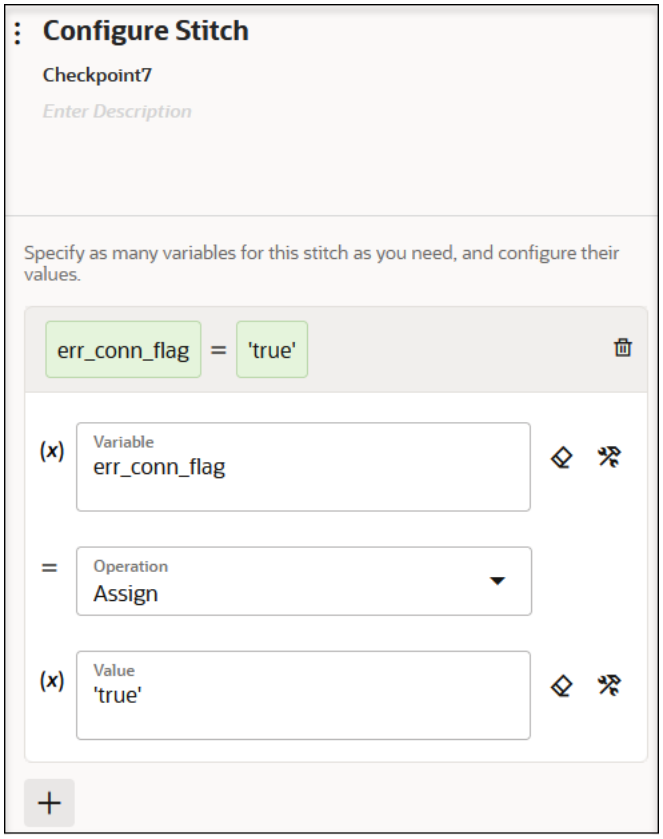


Global variables and business identifier tracking variables are displayed for selection. A data element is a subcomponent of a variable. A complex variable conforms to a series of data elements.

4. From the **Operation** list, select the operation to perform on the variable:
   - **Append**
   - **Assign**
   - **Remove**
5. Click the **Value** field and specify a value in any of the following ways:
   - Entering manually.
   - Selecting from the list.
   - Clicking **Switch to Developer View** to open the **Sources** tree.

   If you selected the **Remove** operation, the **Value** field is not displayed.
6. Click **+** if you want to create additional complex assignment statements.
7. When complete, click **Save** to save your statements and close the Configure Stitch panel. For this example, the following statement was created.

Copyright © 2022, 2026, Oracle and/or its affiliates.



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

Convert HL7 Messages with a Healthcare Action

Convert a message to or from HL7 format with a healthcare action in an integration. The healthcare action converts native inbound HL7 messages into Oracle Integration XML payloads and generates native HL7 messages from Oracle Integration XML payloads to send to external applications that support HL7. XML-formatted CDA messages can also be passed through the healthcare action.

> **Note**
>
> If your Oracle Integration instance does not include the Healthcare edition, you cannot drag the healthcare action into an integration. See Create an Oracle Integration Instance in *Provisioning and Administering Oracle Integration 3*.

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

Copyright © 2022, 2026, Oracle and/or its affiliates.

The Configure healthcare action call wizard is displayed.

2. Enter the following details:

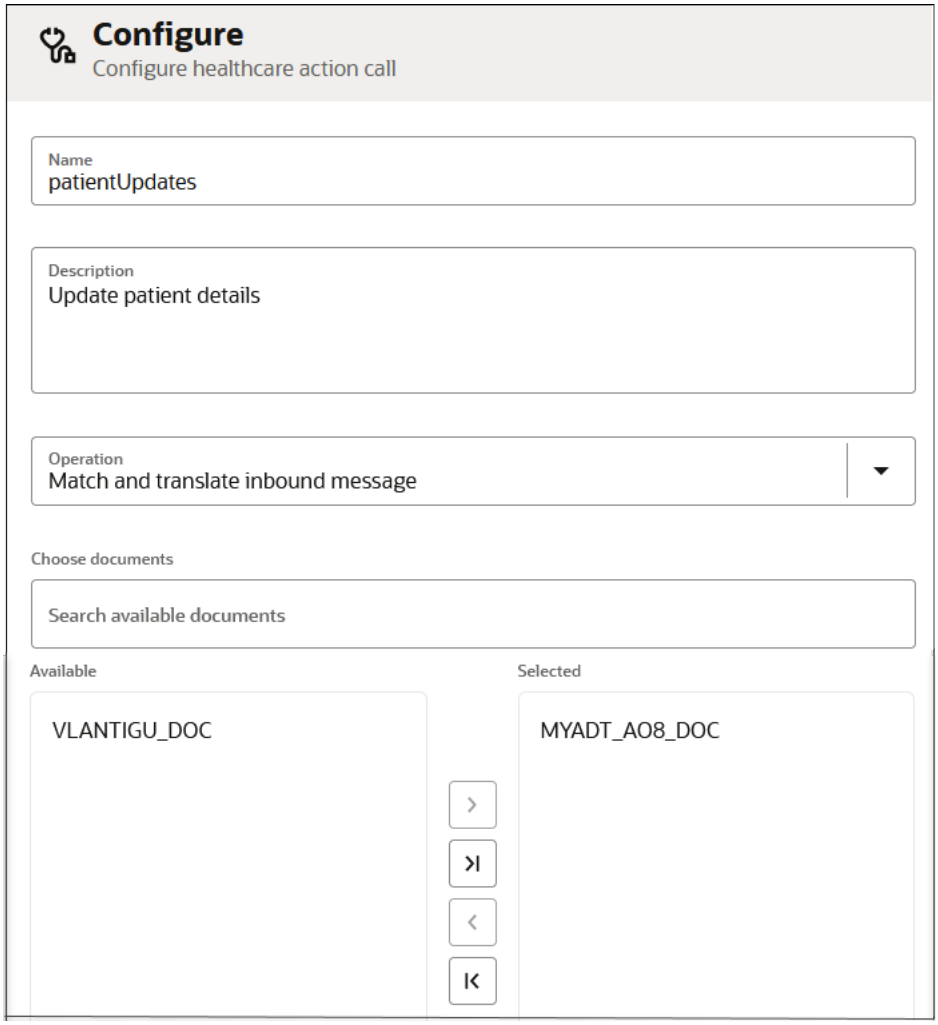
Element	Description
Name	Enter a name.
Description	Enter an optional descriptions for what this action does.
Operation	Select an operation to perform: • **Convert message reference to document**: Use this operation to identify the incoming message type and version. This operation returns a reference to a translated message and metadata needed to route the integration to an appropriate child integration to process the message. When you call the operation, you need to specify a list of HL7 or CDA documents with which you want to match. • **Match and translate inbound message**: Use this operation to convert inbound HL7-formatted messages into XML format or pass through XML-formatted CDA documents. Specify the HL7 message or CDA document you want this action to receive. These are the documents that you configured during design time. This operation takes the following: – A raw HL7 message off the wire, parses it, and matches it with the document definition you select in the **Choose documents** section. The message is then converted to an Oracle Integration XML payload. – An XML-formatted CDA document and passes it through the healthcare action. • **Translate to outbound message**: Use this operation to convert an XML payload message to an outbound HL7 message or pass through an XML-formatted CDA document. • **Split payload into chunks**: Use this operation if your inbound HL7 message includes a large payload and you need to debatch (split) the payload into manageable chunks. See Handle Inbound Debatching of Large HL7 Message Payloads.
Ignore non-modeled segments	Select this checkbox for incoming HL7 messages that contain unexpected custom segments to be successfully processed. With this option selected, unexpected custom segments not found in the schema are ignored completely, no errors are raised, and a response is successfully generated at runtime.
Choose documents This field is displayed if you select the **Match and translate inbound message** operation.	Select *all* the HL7 message definitions that you expect to receive on the port number you configured for your MLLP Adapter connection. You can also select CDA documents. This integration receives and routes all messages sent to that port number. Documents that are not selected are ignored. If you created the integration in a project, only documents available in that project are visible for selection.
Document This field is displayed if you select the **Convert message reference to document** or **Translate to outbound message** operation.	Select the document to use. If you created the integration in a project, only documents available in that project are visible for selection.
Split message count This field is displayed if you select the **Split payload into chunks** operation.	Enter the number of messages into which to split the payload. A value between 10 and 10000 messages is supported. The default size is 1000 chunks.

Copyright © 2022, 2026, Oracle and/or its affiliates.

Element	Description
Split batch size This field is displayed if you select the **Split payload into chunks** operation.	Enter the size for each message. A value between 1 and 100 MB is supported. The default is 1 MB.

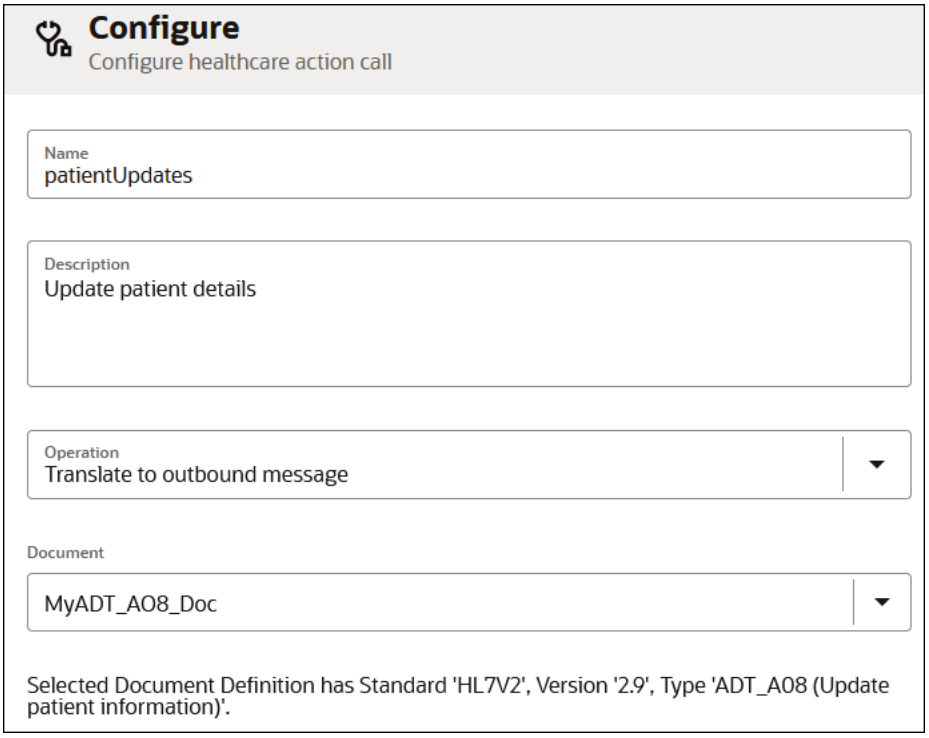
For example, if you select an ADT_A08 HL7 message:

- If you select **Match and translate inbound message**, the selected message document ADT_A08 message document is converted from HL7 format to an Oracle Integration XML-formatted message.

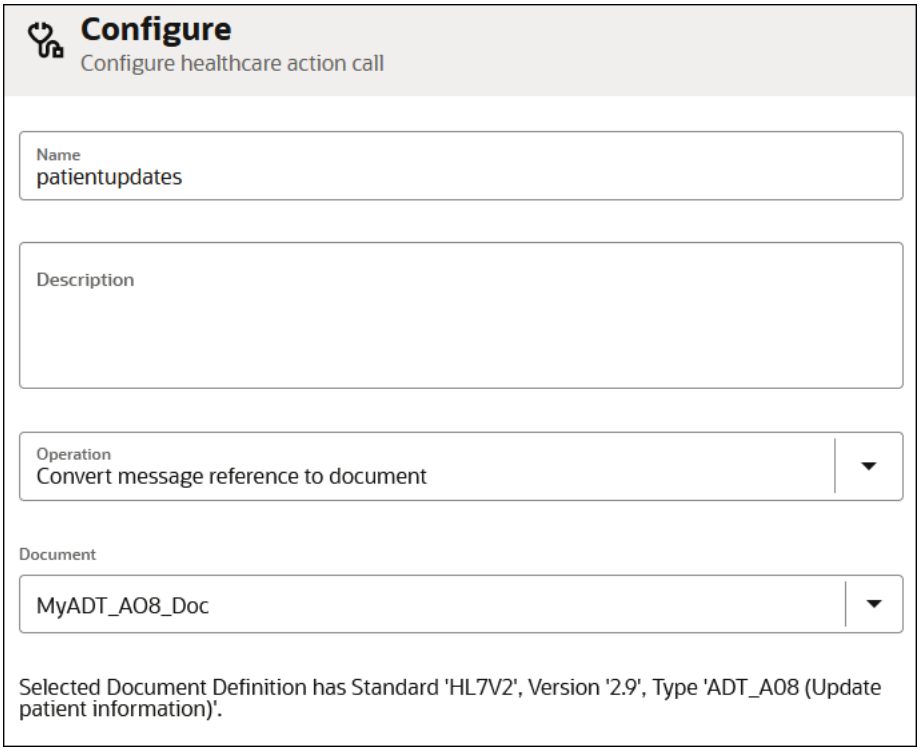


- If you select **Translate to outbound message**, the selected ADT_A08 message is converted from an Oracle Integration XML message to an HL7-formatted message for delivery to an external application that supports HL7.

Copyright © 2022, 2026, Oracle and/or its affiliates.

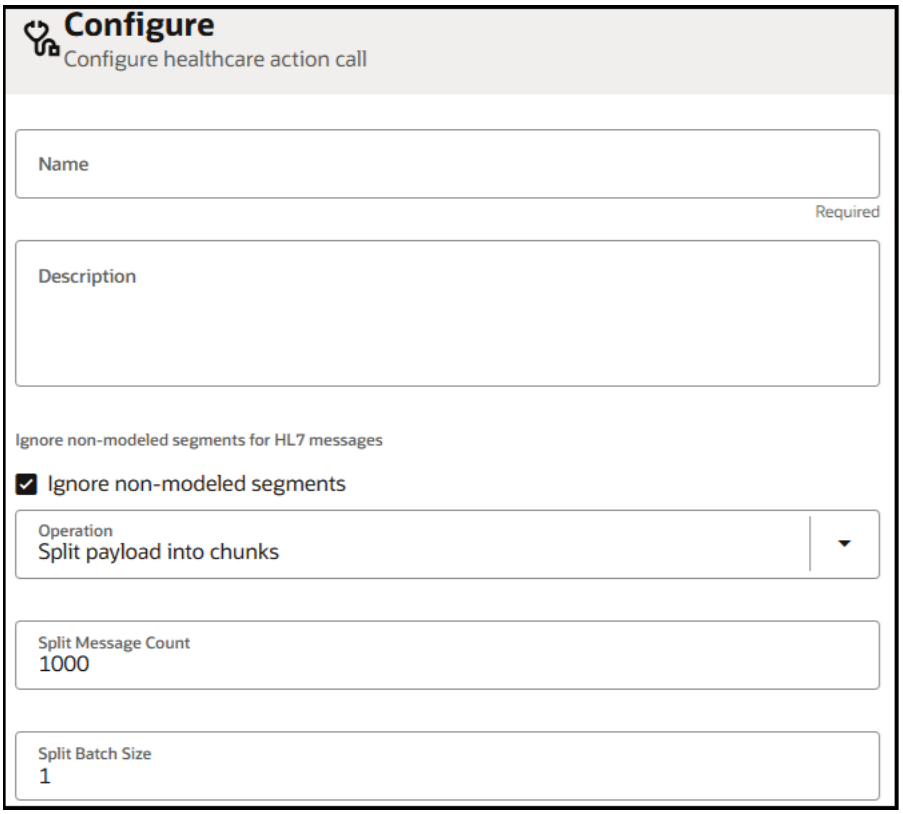


- If you select **Convert message reference to document**, the message reference is converted into a mappable payload. This operation returns the contents of the HL7 message as a variable in your integration that is ready to use in the business logic and mapper of your integration. The variable appears on the source side of the mapper.



Copyright © 2022, 2026, Oracle and/or its affiliates.

- If you select **Split payload into chunks**, the payload of the inbound HL7 message is debatched (split) into manageable chunks.



3. Click **Finish**.

Examples of using all three operations in a use case are provided. See Design Time and Runtime Overview of an Oracle Integration for Healthcare Use Case.

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.

Create a Logger Action

To create a logger action:

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

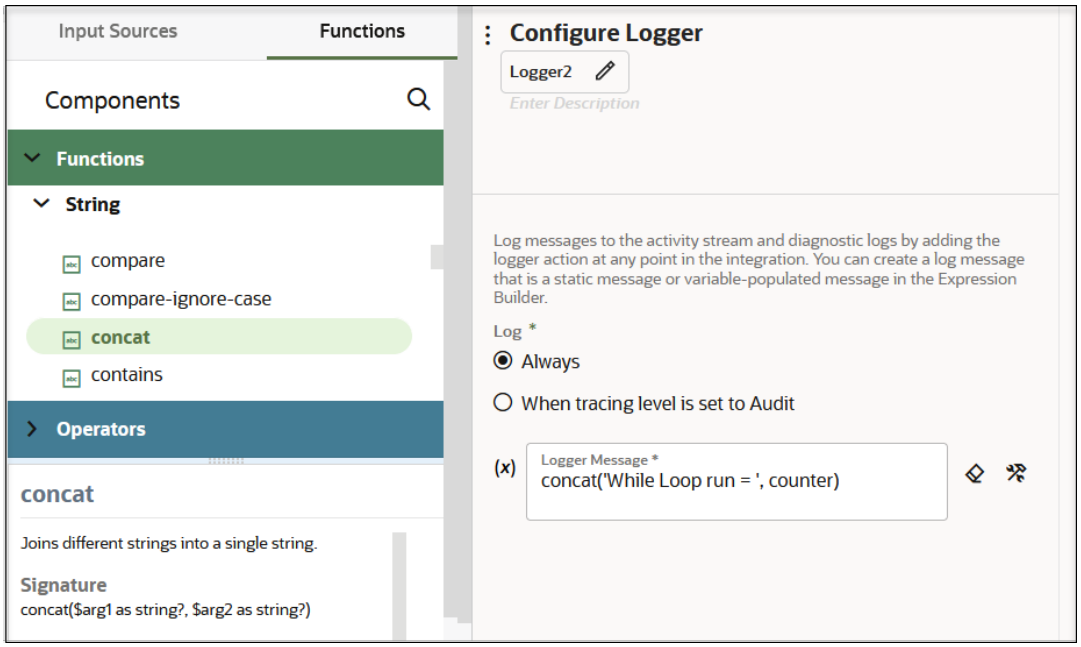
   - On the side of the canvas, click **Actions** and drag the **Log** action to the appropriate location.
   - Click + at the location where you want to add the log action, then select **Log**.

   The Configure Logger panel opens.

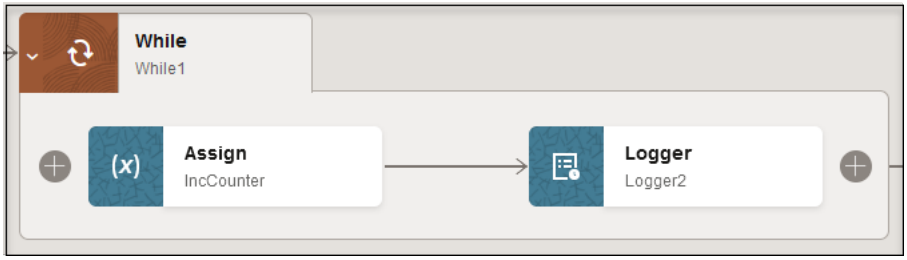
2. Click **Edit** under **Configure Logger** to enter a name and optional description for the log action.

Copyright © 2022, 2026, Oracle and/or its affiliates.

3. Select whether to always log this message or log it only when tracing is set to audit. You can enable tracing when you activate an integration. See Activate an Integration.
4. Specify a static or variable-populated message to be displayed in the activity stream and diagnostic logs. Drag elements from the **Sources** tree to create a message.



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



Track 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 **Observability**, then **Instances**.
2. Click the integration instance.
3. Scroll through the activity stream.
4. Select **Download** to download the logs.
5. Open the zip file and view the log messages you created.

Copyright © 2022, 2026, Oracle and/or its affiliates.

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.

1. Add a note action to an integration in either of the following ways:
   - On the side of the canvas, click **Actions** and drag the **Note** action to the appropriate location.
   - Click + at the location where you want to add the note action, then select **Note**.
   - Click **Edit** under **Configure Note** to enter a name and optional description for the note action.
2. Enter your notes, then click **Save**. You can add up to 256 characters.
3. Hover over the icon to display the note text.

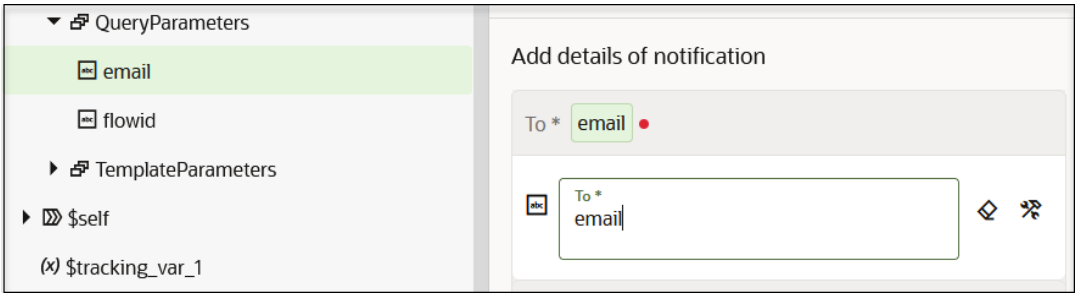
Send Notification Emails During Stages of the Integration with a Notification Action

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

1. Add a notification action to an integration in either of the following ways:
   - On the side of the canvas, click **Actions** and drag the **Notification** action to the appropriate location.
   - Click + at the location where you want to add the notification action, then select **Notification**.
2. Click **Edit** under **Configure Notification** to enter a name and optional description for the notification action.
3. Click the **To** field.

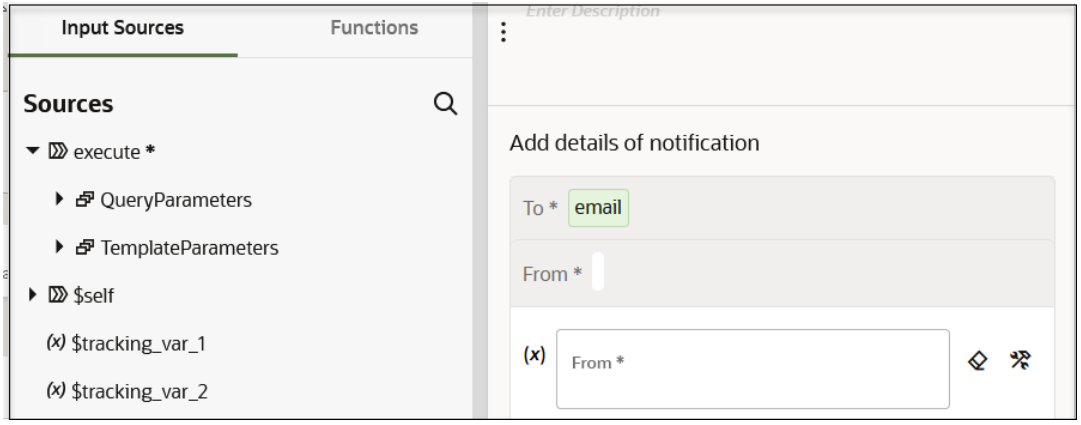
   This opens the **Sources** tree for defining an expression and a second **To** field for editing immediately after the first field.
4. Manually enter an email address, select from the drop-down list, or drag an element from the **Sources** tree to the **To** field:

Copyright © 2022, 2026, Oracle and/or its affiliates.



5. Click the **From** field.

   This opens a second **From** field for editing immediately after the first field.



6. Manually enter an email address, select from the drop-down list, or drag an element from the **Sources** tree. For this example, an **email** element is dragged from the **Sources** tree.

   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 by selecting **Settings**, and then **Notifications**.

   > **Note**
   >
   > If you send emails with the notification action and the **From** address is not one of the approved senders configured on the Notifications page, the **From** address defaults to the approved sender address of `no-reply@mail.integration.`*`region`*`.ocp.oraclecloud.com`. For example, if your region is Ashburn, the address is `no-reply@mail.integration.us-ashburn-1.ocp.oraclecloud.com`. See Configure Notification Emails.

   See Send System Status Reports with Notification Emails.

7. Click the **Subject** field.

   This opens a second **Subject** field for editing immediately after the first field.

8. Manually enter a subject in plain text, select from the drop-down list, or build an expression from the **Source** or **Functions** tree. To drag a function, ensure that you first select **Switch to Developer View** .

Copyright © 2022, 2026, Oracle and/or its affiliates.

For this example, a **concat** function is selected for use from the **Functions** section.

```
concat ( 'Hello', name )
```

9. Click the **Body** field.

   This opens a second **Body** field for editing immediately after the first field.

10. Enter a message using plain text and 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 after this field. After creating parameters, enter them inside { } brackets.

```
<html>
<body>
<p>Hello <strong><span style="color: #2925BF;">{name},</span></strong></p>
<p><span style="color: #298022;"><strong>Welcome to Oracle Integration!</
strong></span></p>
</body>
</html>
```

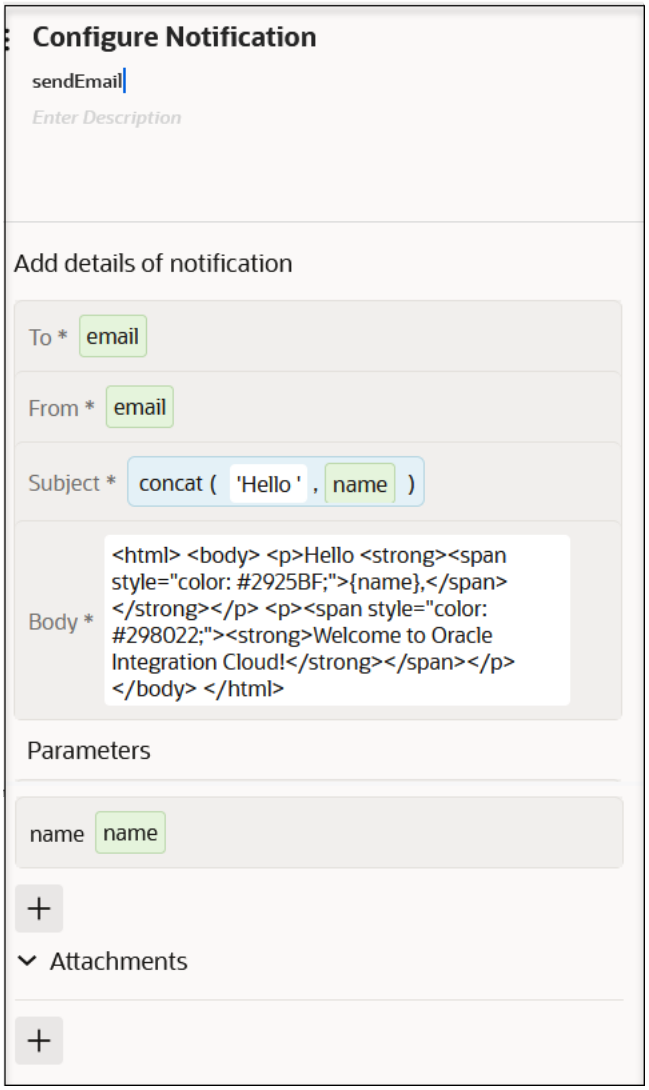
11. Click **+** to add parameters.
12. Enter a parameter name and description or build an expression.
13. Click **+** to add attachments.
14. Select the attachments to add.
15. In the **Attachments** section, click **+** to open a page to select an attachment file.

    You can edit or delete attachments once added. Examples of attachments may include file reference attachments that are available for selection:

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

16. When complete, the Configure Notification panel looks as follows for this example:

Copyright © 2022, 2026, Oracle and/or its affiliates.



17. Click **Save**.

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

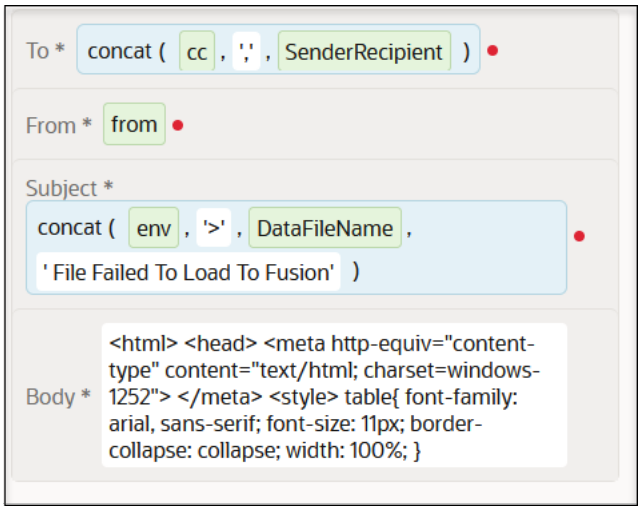
Copyright © 2022, 2026, Oracle and/or its affiliates.

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

Set the content-type in a Notification Action

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

Code the `content-type` as follows:



Process Files in Schedule Integrations with a Stage File Action

You can use the stage file action to process files or file references in schedule 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, encrypt, decrypt, and list files in a staged location known to Oracle Integration.

The stage file action can also read (and remove any trailer) and unzip referenced files in a staged location. The stage file action is similar in functionality to adapters. However, unlike adapters, you do not need to create a connection to use the stage file action. The stage file action has no credential properties and security policies. The stage file action also differs from the File Adapter and FTP Adapter in that it provides the ability to define a file format for read and write operations. For the stage file action to process or act upon files and attachments, they must be available in Oracle Integration. Make the files available in Oracle Integration

Copyright © 2022, 2026, Oracle and/or its affiliates.

either using the download operation in the FTP Adapter or consuming the SOAP/REST APIs that return multipart or MTOM attachments. Oracle Integration flows exposed as REST endpoints using the REST Adapter and exposing an interface to accept multipart attachments also automatically stage files in Oracle Integration when the requests are posted to the endpoint.

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

- Configure an FTP Adapter with the following settings:
  - **Download File** operation
  - **Unzip the File** option
  - Input directory and download directory path
- Because the ZIP file may contain multiple files, configure a for-each action after the FTP Adapter to iterate over repeated elements.
- To read each file from the input directory, configure a stage file action after 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.

Copyright © 2022, 2026, Oracle and/or its affiliates.

Create 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 in the Stage File panel.

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

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

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

Open the Stage File Panel

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

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

Copyright © 2022, 2026, Oracle and/or its affiliates.

The Configure Operation page is displayed.

- Configure Stage File Operations
- Define the Schema File
- Define the Schema Format
- Review the Stage File Action in the Integration Canvas

Configure Stage File Operations

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

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

Read Entire File

Table 7-1 - Read Entire File

Property	Description
Configure File Reference	• **Yes**: Select to process an upstream operation that provides a file reference. Once selected, you specify the file reference. • **No**: Select to process a file name.
Specify the File Reference (Appears if you select **Yes** for the **Configure File Reference**.)	Click the field and specify a file reference in any of the following ways: • Entering manually. • Selecting from the list. • Clicking **Switch to Developer View** to open the **Sources** tree.

Copyright © 2022, 2026, Oracle and/or its affiliates.

Table 7-1 (Cont.) - Read Entire File

Property	Description
Specify the File Name (Appears if you select **No** for the **Configure File Reference**.)	Click the field and specify a file name (for example, `/compress:schedule/compress:start Time`) in any of the following ways: • Entering manually. • Selecting from the list. • Clicking **Switch to Developer View**  to open the **Sources** tree. **Note**: The file size must be less than 10 MB. For files greater than 10 MB, use the **Read File in Segments** operation.
Specify the Directory to read from	Click the field and specify a directory from which to read files (for example, `/compress:schedule/compress:start Time`) in any of the following ways: • Entering manually. • Selecting from the list. • Clicking **Switch to Developer View**  to open the **Sources** tree.
Remove Trailer	Select to not remove the trailer, to remove the last row, or to remove the last *n* rows.
Decrypt Content	Click the check box, then select the private key of the target location to use to decrypt the file.
Verify If you select **Decrypt Content**, the **Verify** check box is enabled and the name changes to **Verify Content**.	Select the key to verify the content during decryption. a. Select **Verify Content**. b. Select the key to use from the list. You upload the PGP keys to use on the Certificates page. See Upload a Certificate to Connect with External Services.

Read File in Segments

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

You can also read large XML files containing repeating elements and multiple namespaces. A use case is provided.

See Read Large XML Files Containing Multiple Namespaces.

Copyright © 2022, 2026, Oracle and/or its affiliates.

Note

- The recommended segment size in Oracle Integration 3 is 2000 records per segment (configurable). When an Oracle Integration Generation 2 integration (with a fixed segment size of 200) is opened for the first time in Oracle Integration 3, the segment size defaults to 2000. Modify this value as needed based on your requirements.
- If you select the **Read File in Segments** operation, you cannot specify an opaque or JSON schema. If you do, you receive the following runtime error:

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

Table 7-2 - Read File in Segments

Property	Description
Configure File Reference	• **Yes**: Select to process an upstream operation that provides a file reference. • **No**: Select to process a file name.
Specify the File Reference (Appears if you select **Yes** for the **Configure File Reference**.)	Click the field and specify the file reference in any of the following ways: • Entering manually. • Selecting from the list. • Clicking **Switch to Developer View**  to open the **Sources** tree.
Specify the File Name (Appears if you select **No** for the **Configure File Reference**.)	Click the field and specify a file name in any of the following ways: • Entering manually. • Selecting from the list. • Clicking **Switch to Developer View**  to open the **Sources** tree. 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.

Copyright © 2022, 2026, Oracle and/or its affiliates.

Table 7-2 (Cont.) - Read File in Segments

Property	Description
Specify the Directory to read from	Click the field and specify a directory from which to read files in any of the following ways: • Entering manually. • Selecting from the list. • Clicking **Switch to Developer View**  to open the **Sources** tree.
Segment Size	Enter the segment size. The segment size defaults to 200 records per segment, but can be changed. The minimum value is 200 records and the maximum value is 2000 records. 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.
Sequential Processing	Select to read the segments sequentially.
Remove Trailer	Select to not remove the trailer, to remove the last row, or to remove the last *n* rows.

Write File

> **Note**
>
> - When writing to a stage file, there is no built-in support for headers. You must create headers by explicitly writing them to the file.
> - Multiple stage file action write operations appending data to the same stage file within a loop results in the data being overwritten. This results in data loss.

Table 7-3 - Write File

Property	Description
Specify the File Name	Click the field and specify a file name in any of the following ways: • Entering manually. • Selecting from the list. • Clicking **Switch to Developer View**  to open the **Sources** tree.
Specify the Output Directory	Click the field and specify an output directory in any of the following ways: • Entering manually. • Selecting from the list. • Clicking **Switch to Developer View**  to open the **Sources** tree.

Copyright © 2022, 2026, Oracle and/or its affiliates.

Table 7-3 (Cont.) - Write File

Property	Description
Append	Optionally select to append records to the existing file. **Note**: If you append JSON or XML content, the final file is invalid XML or JSON.
Get file attributes value on mapper	Select this checkbox *only* if you want to calculate the **size** and **checksum** attribute values for your file in the mapper response. If selected, both attributes are enabled in the response mapper underneath **ICSFile**. By default, the **size** and **checksum** attributes are disabled in the mapper response because calculating the values for both attributes is performance-intensive.
Encrypt	Click the check box, then select the public key of the target location to use to encrypt the file.
Sign If you select **Encrypt**, the **Sign** check box is enabled and the name changes to **Sign Content**.	Select the private PGP key to sign the file. a. Select **Sign Content**. b. Select the PGP key to use from the list. You upload the PGP keys to use on the Certificates page. See Upload a Certificate to Connect with External Services.

Zip Files

Table 7-4 - Zip Files

Property	Description
Specify the File Name	Click the field and specify a file name in any of the following ways: • Entering manually. • Selecting from the list. • Clicking **Switch to Developer View** to open the **Sources** tree.
Specify the Directory to zip	Click the field and specify a directory to zip up in any of the following ways: • Entering manually. • Selecting from the list. • Clicking **Switch to Developer View** to open the **Sources** tree.
Specify the Output Directory	Click the field and specify an output directory in which for the ZIP file in any of the following ways: • Entering manually. • Selecting from the list. • Clicking **Switch to Developer View** to open the **Sources** tree.

Copyright © 2022, 2026, Oracle and/or its affiliates.

Table 7-4 (Cont.) - Zip Files

Property	Description
Get file attributes value on mapper	Select this checkbox *only* if you want to calculate the **size** and **checksum** attribute values for your file in the mapper response. If selected, both attributes are enabled in the response mapper underneath **ICSFile**. By default, the **size** and **checksum** attributes are disabled in the mapper response because calculating the values for both attributes is performance-intensive.

Unzip File

Table 7-5 - Unzip File

Property	Description
Configure File Reference	• **Yes**: Select to process an upstream operation that provides a ZIP file reference. Once selected, you specify the file reference and the directory in which to unzip the file. • **No**: Select to process a ZIP file.
Specify the File Reference (Appears if you select **Yes** for the **Configure File Reference**.)	Click the field and specify a file reference in any of the following ways: • Entering manually. • Selecting from the list. • Clicking **Switch to Developer View**  to open the **Sources** tree.
Specify the Zip File Name (Appears if you select **No** for the **Configure File Reference**.)	Click the field and specify a ZIP file name to read in any of the following ways: • Entering manually. • Selecting from the list. • Clicking **Switch to Developer View**  to open the **Sources** tree.
Specify the Zip File Directory	Click the field and specify a directory in which to unzip the file in any of the following ways: • Entering manually. • Selecting from the list. • Clicking **Switch to Developer View**  to open the **Sources** tree.
Specify the Directory to Unzip	Click the field and specify the directory to unzip in any of the following ways: • Entering manually. • Selecting from the list. • Clicking **Switch to Developer View**  to open the **Sources** tree.

Copyright © 2022, 2026, Oracle and/or its affiliates.

Table 7-5 (Cont.) - Unzip File

Property	Description
Get file attributes value on mapper	Select this checkbox *only* if you want to calculate the **size** and **checksum** attribute values for your file in the mapper response. If selected, both attributes are enabled in the response mapper underneath **ICSFile**. By default, the **size** and **checksum** attributes are disabled in the mapper response because calculating the values for both attributes is performance-intensive.

Encrypt File

Table 7-6 - Encrypt File

Property	Description
Specify the File Reference	Click the field and specify a file reference in any of the following ways: • Entering manually. • Selecting from the list. • Clicking **Switch to Developer View**  to open the **Sources** tree.
Specify the File Name	Click the field and specify a file name in any of the following ways: • Entering manually. • Selecting from the list. • Clicking **Switch to Developer View**  to open the **Sources** tree.
Specify the Output Directory	Click the field and specify an output directory in any of the following ways: • Entering manually. • Selecting from the list. • Clicking **Switch to Developer View**  to open the **Sources** tree.
Get file attributes value on mapper	Select this checkbox *only* if you want to calculate the **size** and **checksum** attribute values for your file in the mapper response. If selected, both attributes are enabled in the response mapper underneath **ICSFile**. By default, the **size** and **checksum** attributes are disabled in the mapper response because calculating the values for both attributes is performance-intensive.
Specify PGP Key to encrypt file	Select the private key of the target location to use to encrypt the file. You can encrypt a file up to 1 GB in size. When using the mapper, note that encrypt is visible as an element for mapping.

Copyright © 2022, 2026, Oracle and/or its affiliates.

Table 7-6 (Cont.) - Encrypt File

Property	Description
Sign Content	Select the public PGP key to sign the file. **a.** Select the check box. **b.** Select the PGP key to use to sign the file during encryption. You upload the PGP keys to use on the Certificates page. See Upload a Certificate to Connect with External Services.

Decrypt File

Table 7-7 - Decrypt File

Property	Description
Specify the File Reference	Click the field and specify a file reference in any of the following ways: • Entering manually. • Selecting from the list. • Clicking **Switch to Developer View**  to open the **Sources** tree.
Specify the File Name	Click the field and specify a file name in any of the following ways: • Entering manually. • Selecting from the list. • Clicking **Switch to Developer View**  to open the **Sources** tree.
Specify the Output Directory	Click the field and specify an output directory in any of the following ways: • Entering manually. • Selecting from the list. • Clicking **Switch to Developer View**  to open the **Sources** tree.
Get file attributes value on mapper	Select this checkbox *only* if you want to calculate the **size** and **checksum** attribute values for your file in the mapper response. If selected, both attributes are enabled in the response mapper underneath **ICSFile**. By default, the **size** and **checksum** attributes are disabled in the mapper response because calculating the values for both attributes is performance-intensive.
Specify PGP Key to decrypt file	Select the private key of the target location to use to decrypt the file. You can decrypt a file up to 1 GB in size. When using the mapper, note that decrypt is visible as an element for mapping.

Copyright © 2022, 2026, Oracle and/or its affiliates.

Table 7-7 (Cont.) - Decrypt File

Property	Description
Verify Content	Select the PGP key to verify the file. **a.** Select the check box. **b.** Select the PGP key to use to verify the file during decryption. You upload the PGP keys to use on the Certificates page. See Upload a Certificate to Connect with External Services.

List File

Table 7-8 - List File

Property	Description
Specify the Directory to List Files from	Click the field and specify a directory from which to list files in any of the following ways: • Entering manually. • Selecting from the list. • Clicking **Switch to Developer View**  to open the **Sources** tree.
Specify the File Pattern to use	Specify the pattern of the file name to transfer to the output directory. Click the **?** icon for the list of supported patterns.
List Files Recursively	Select to list the files recursively.
Get file attributes value on mapper	Select this checkbox *only* if you want to calculate the **size** and **checksum** attribute values for your file in the mapper response. If selected, both attributes are enabled in the response mapper underneath **ICSFile**. By default, the **size** and **checksum** attributes are disabled in the mapper response because calculating the values for both attributes is performance-intensive.

2. When complete, click **Continue** .

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

Define the Schema File

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

1. Provide the following details.

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

Copyright © 2022, 2026, Oracle and/or its affiliates.

Property	Description
Which one the following choices would be used to describe the structure of the file contents?	Select an option: • **Sample delimited document (erg. CSV)**: Select to create a new schema from a CSV file. On a subsequent page of this panel, you are prompted to select the CSV file from which to create the schema. • **XML schema (XSD) document**: Select to create a new schema from an XML schema (XSD) file. You can upload a ZIP file with nested XML schema files. • **Sample XML document (Single or No Name Space)**: Select to create a new schema from a sample XML file with a single or no name space. • **Sample JSON document**: Select to create a new schema from a JSON file. This option only appears if you selected the **Read Entire File** or **Write File** operation.

2. When complete, click **Continue**.

Define 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 based on the schema option you selected:

> **Note**
>
> Oracle does not certify schemas that were manually updated with undocumented annotations. Use discretion if specifying these schemas. For example, runtime errors can occur if you use Oracle SOA Suite schemas that are annotated with constructs that are undocumented and not certified in Oracle Integration 3.

- **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 7-9 - Sample Delimited Document (e.g. CSV)

Element	Description
• **Select the Delimited Data File**	Drag and drop 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.
• **Enter the Record Name**	Enter the record name. This becomes the parent element in the created schema file for the record names selected as column headers from the CSV file.

Copyright © 2022, 2026, Oracle and/or its affiliates.

Table 7-9 (Cont.) - Sample Delimited Document (e.g. CSV)

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

Table 7-10 - XML Schema (XSD) Document

Element	Description
• **Select a New File**	Drag and drop an existing XML schema file or schema archive file from the file system.
• **Selected File Name**	Displays the selected schema file name.
• **Select the Element Name**	Select the schema element. This field is displayed after the XML schema file is selected. The element name is treated as the root element in the uploaded schema file.

Copyright © 2022, 2026, Oracle and/or its affiliates.

Table 7-10 (Cont.) - XML Schema (XSD) Document

Element	Description
• **Select Repeating Batch Element**	Click the field and specify the repeating node in the schema to support payload chunking in any of the following ways: • Entering manually. • Selecting from the list. • Clicking **Switch to Developer View** to open the **Sources** tree.

Table 7-11 - Sample XML Document (Single or No NameSpace)

Element	Description
• **Select a New File**	Drag and drop a sample XML document from the file system. The file must contain no namespace or a single namespace.
• **Selected File Name**	Displays the selected schema file name.
• **Select the Schema Element**	Select the schema element. This field is displayed after the sample XML file is selected. The element name is treated as the root element in the uploaded schema file.
• **Select Repeating Batch Element**	Click the field and specify the repeating node in the schema to support payload chunking in any of the following ways: • Entering manually. • Selecting from the list. • Clicking **Switch to Developer View** to open the **Sources** tree.

Table 7-12 - Sample JSON document

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

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

Review 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.

Copyright © 2022, 2026, Oracle and/or its affiliates.

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).

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

- Create an Integration to Import and Process Bulk Files in *Using the FTP Adapter with Oracle Integration 3*
- Process RaaS Reports that Support the Delivery of Data in CSV Format in *Using the Workday Adapter with Oracle Integration 3*
- Produce Messages to an Apache Kafka Topic in *Using the Apache Kafka Adapter with Oracle Integration 3*
- Use Bulk Response Operations in an Integration in *Using the Salesforce Adapter with Oracle Integration 3*

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

The wait action enables you to delay the processing of an integration for a specified period of time. Use this action in schedule integrations, asynchronous integrations, synchronous integrations, and fire-and-forget integrations. A typical use for this action is to invoke a specific operation at a certain time. You can specify literal values or build XPath expressions that you can configure with property values during runtime.

Create a Wait Action

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

   - On the side of the canvas, click **Actions** and drag the **Wait** action to the appropriate location.

Copyright © 2022, 2026, Oracle and/or its affiliates.

- Click + at the location where you want to add the wait action, then select **Wait**.

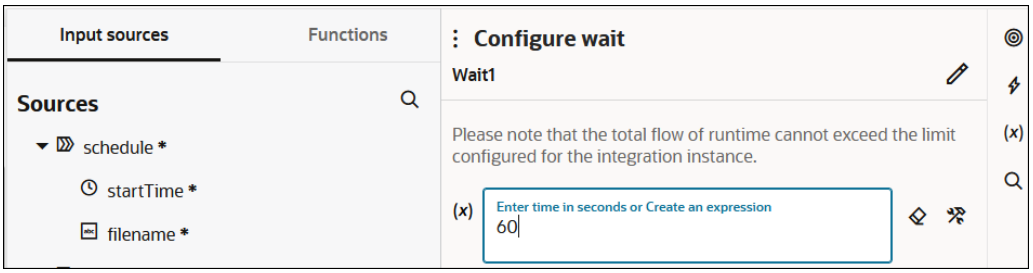
The Configure wait panel opens.

2. Click **Edit** under **Configure wait** to enter a name and optional description for the wait action.

   Configure the time to wait before processing the integration in either of two ways:

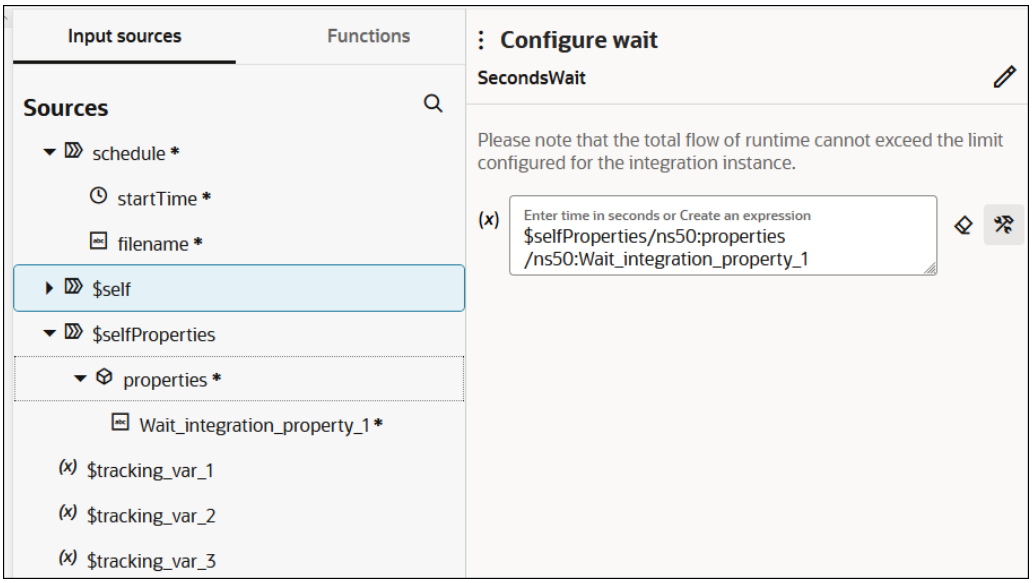
3. If you want to use literal (hard coded) values:

   a. Enter the number of seconds to wait before processing the integration. Literal values can only be in seconds, with no upper limit. The wait time cannot exceed the total running time of the integration.



4. If you want to enter an XPath expression:
   This option enables you to update a wait action parameter with an integration property value after the integration is activated. This update does not require a re-activation of the integration.

   a. Drag an element from the **Input sources** section.



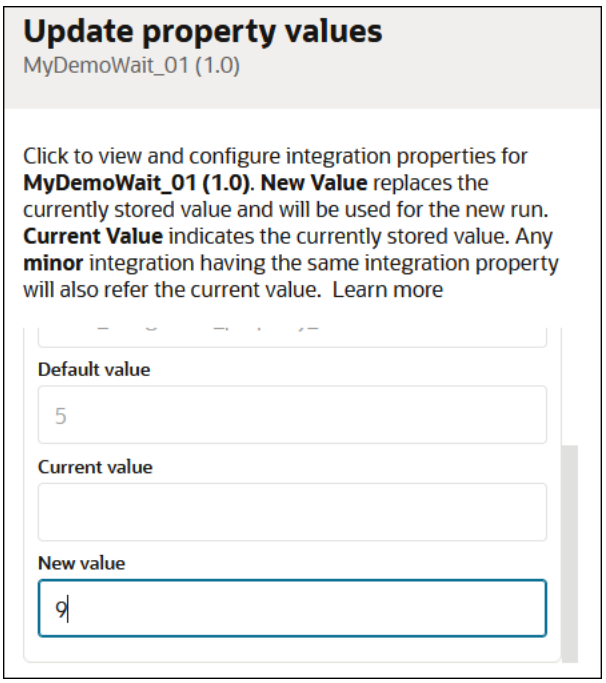
   b. Complete design of your integration and exit the integration canvas.

   c. Activate the integration.

   d. Hover your cursor over the integration, and select **Actions** ..., then select **Update property values**.

Copyright © 2022, 2026, Oracle and/or its affiliates.

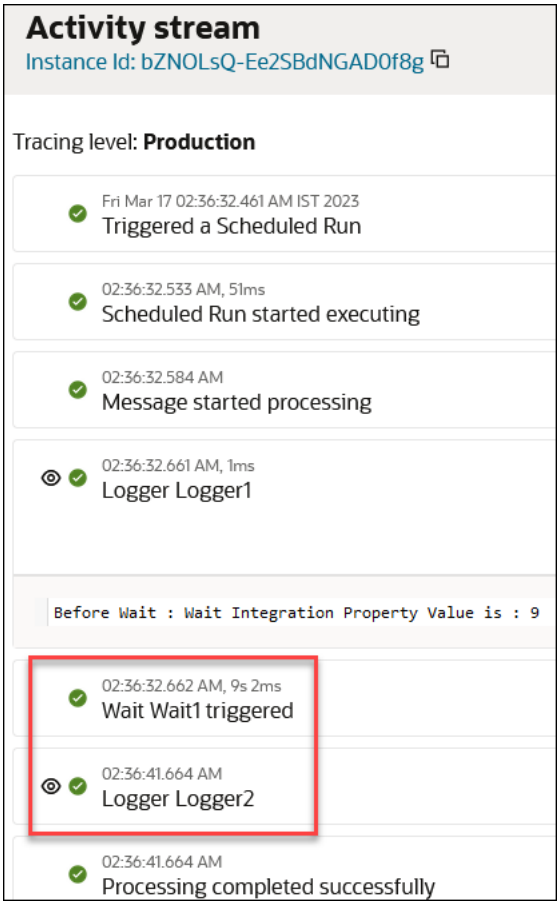
e. Enter a value to use at runtime in the **New value** field (for this example, 9 is entered). This value overrides the value provided in the **Default value** field.



f. Run the integration and view the activity stream.

g. Click **View** for the action in front of the wait action (for this example, a logger action) and note that the integration property value is **9** and the wait action was initiated at **02:36:32** and the next action (for this example, a second logger action) was initiated at **02:36:41** (nine seconds later).

Copyright © 2022, 2026, Oracle and/or its affiliates.



Track the Status of a Wait Action During Runtime

During runtime, you can track the status of the wait action on the Tracking page through the tracking diagram and activity stream for an activated integration.

1. In the navigation pane, click **Observability**, then **Instances**.
2. Click the business identifier value of the integration to track.

   The integration flow (including any wait actions) is displayed.

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

- Use the Extract API in a Schedule Integration in *Using the SAP Concur Adapter with Oracle Integration 3*
- Process RaaS Reports that Support the Delivery of Data in CSV Format in *Using the Workday Adapter with Oracle Integration 3*
- Export Candidate Data from Oracle Talent Acquisition Cloud (Taleo EE) in *Using the Oracle Taleo Enterprise Edition Adapter with Oracle Integration 3*

Copyright © 2022, 2026, Oracle and/or its affiliates.

Error Handling Category

The actions category enables you to add re-throw fault and throw new fault actions to an integration.

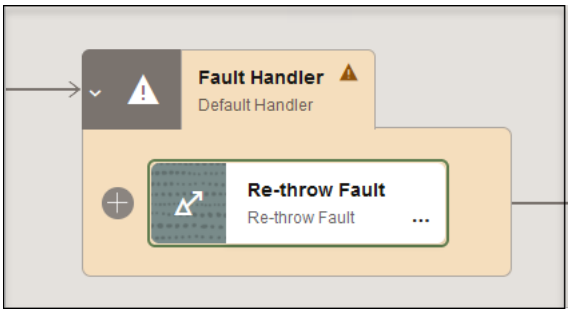
Topics:

- Catch Faults with a Re-throw Fault Action
- Throw Faults with a Throw New Fault Action

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.

1. Create an integration that includes a scope action. See Manage a Group of Actions and Fault Handlers with a Scope Action.
2. Double-click the scope.
3. Select **Actions** ..., then **Fault Handlers**, and then **Default Handler**.
   The scope action is now in fault design mode.
4. Add a re-throw fault action to an integration in either of the following ways:
   - On the right side of the canvas, click **Actions** and drag the **Re-throw Fault** action to the **+** sign.
   - Click + at the location where you want to add the re-throw fault action, then select **Re-throw Fault**.
     This creates a re-throw fault action in the fault handling section of 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. Double-click the scope action to return to design mode.

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

Copyright © 2022, 2026, Oracle and/or its affiliates.

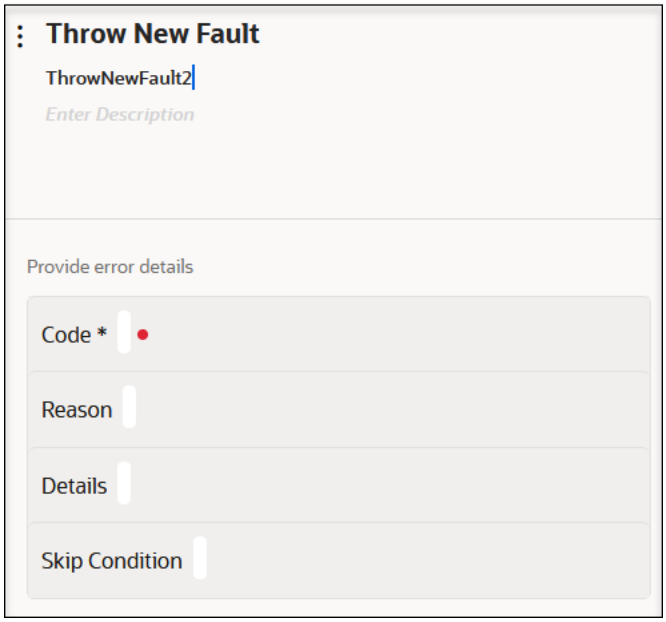
6. Design your scope action.
7. To return to fault mode, double-click the scope.
8. Select **Actions** ⋯, then **Fault Handlers**, and then **Default Handler**.

Throw Faults with a Throw New Fault Action

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

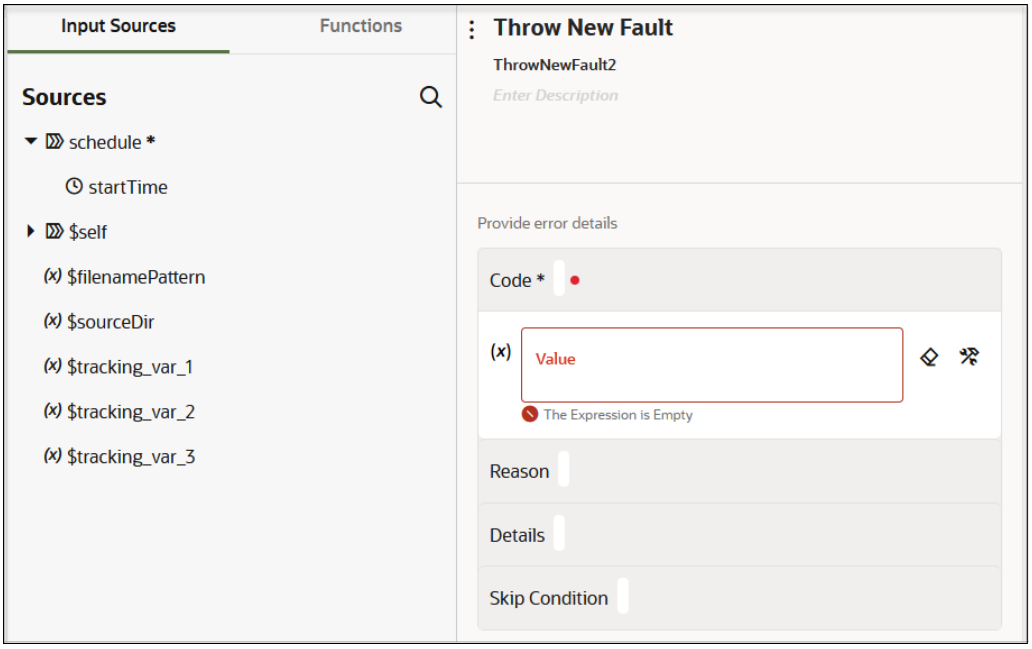
1. Add a throw new fault action to an integration in either of the following ways:
   - On the side of the canvas, click **Actions** and drag the **Throw New Fault** action to the appropriate location.
   - Click + at the location where you want to add the throw new fault action, then select **Throw New Fault**.
2. Click **Edit** under **Throw New Fault** to enter a name and optional description for the throw new fault action.

   The Throw New Fault panel is displayed.



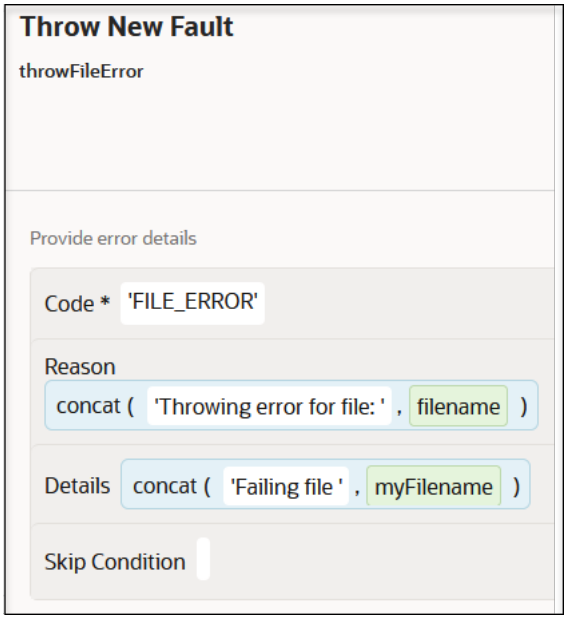
3. Click a field to open the **Sources** panel and specify values in any of the following ways:
   - Enter manually.
   - Select from the list.
   - Click **Switch to Developer View** to drag a value from the **Sources** panel.

Copyright © 2022, 2026, Oracle and/or its affiliates.



4. Complete the fields:

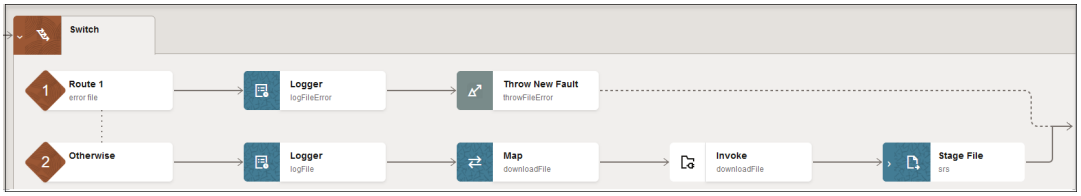
Element	Description
Code	Create a code value. This is a mandatory field.
Reason	Define a reason for the error.
Details	Define additional error details.
Skip Condition	Define a condition to prevent the fault from being thrown.



5. Click **Save**

Copyright © 2022, 2026, Oracle and/or its affiliates.

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 processing 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 processing 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.

OCI AI Services Category

The OCI AI Services category enables you to add OCI document understanding, OCI vision, OCI generative AI, OCI language, OCI generative AI agents RAG, and OCI speech actions to an integration.

> **Note**
>
> OCI AI service actions are available only in the OCI commercial realm (realm key: OC1). They are not available in OCI government realms. See AI Feature Availability in *Provisioning and Administering Oracle Integration 3*.

Topics:

- Prerequisites
- Extract Document Information with a Document Understanding Action
- Analyze and Extract Information from Images with a Vision Action
- Generate, Embed, or Rerank Text with a Generative AI Action
- Perform Text Analysis and Translation with a Language Action
- Ask Questions and Receive Answers Based on Current Enterprise Data with a Generative AI Agents RAG Action
- Transcribe Speech to Text and Synthesize Text to Speech with a Speech Action

Prerequisites

To use the Oracle Cloud Infrastructure AI services such as OCI Document Understanding, OCI Generative AI, OCI Generative AI Agents RAG, OCI Language, OCI Speech, and OCI Vision in an integration, you must satisfy the following prerequisites in the Oracle Cloud Console.

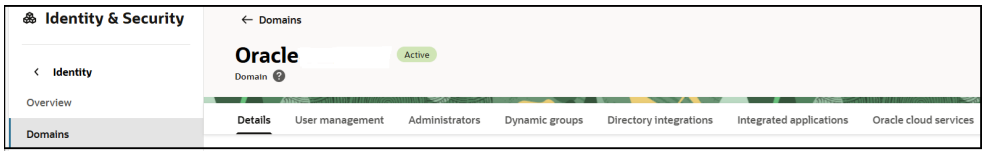
- Ensure that your cloud tenancy uses identity domains.

Copyright © 2022, 2026, Oracle and/or its affiliates.

- Create a dynamic group and a policy to grant access to the specific OCI AI service that you want to use (such as OCI Document Understanding, OCI Generative AI, OCI Generative AI Agents RAG, OCI Language, OCI Speech, or OCI Vision).

  Create the required dynamic group and assign a policy to that group to allow your Oracle Integration instance to access the specific OCI AI service that you want to use. The policy defines the permissions for the dynamic group and determines which operations the dynamic group can perform on the OCI AI service.

  1. Log in to the Oracle Cloud Console.
  2. Obtain the client ID of the OAuth application for the Oracle Integration instance.
     a. In the upper right corner, select **Profile**, then click the identity domain.
     b. In the menu bar, click **Oracle Cloud Services**.

     The Oracle Cloud Services page for your domain appears.
     c. In the **Name** column, click your service instance.
     d. Click **OAuth configuration**.
     e. Scroll down to the **General Information** section and copy the client ID value to use to create your dynamic group.
     f. Scroll to the top and click **Oracle cloud services** to return to the main page for the identity domain.

  3. In the menu bar, click **Dynamic groups**.
  4. Click **Create dynamic group**.
  5. Enter the following details:
     a. In the **Name** and **Description** fields, enter values. These fields are required.
     b. In the **Matching rules** section, enter the required rule. The resource ID you specify must match the client ID of the OAuth application of your Oracle Integration instance. Ensure that you enclose the value in single quotes. For example:

     ```
     resource.id = 'client_ID'
     ```

     c. Click **Create**.
  6. In the left navigation pane, click **Policies**.
  7. Click **Create Policy**.
  8. Select the compartment in which to create the policy.
  9. Complete the following details and click **Create**.
     a. In the **Name** and **Description** fields, enter values. These fields are required.

Copyright © 2022, 2026, Oracle and/or its affiliates.

b. In the **Policy Builder** section, build the required policy for the dynamic group:

OCI AI service	Policy
OCI Document Understanding	`allow dynamic-group dynamic_group to manage ai-service-document-family in compartment compartment_name`
OCI Generative AI	`allow dynamic-group <dynamic-group_name> to manage generative-ai-family in compartment <compartment-name>`

OCI Generative AI Agents RAG	`allow dynamic-group <dynamic-group_name> to manage object-family in compartment <compartment-name>`
	
	`allow dynamic-group <dynamic-group_name> to manage genai-agent-family in tenancy`

OCI Language	`allow dynamic-group <dynamic-group_name> to manage ai-service-language-family in compartment <compartment-name>` Also, see Language Policies, Custom Model Policies, and Async Job Policies.
OCI Speech	`allow dynamic-group <dynamic-group_name> to manage ai-service-speech-family in compartment <compartment-name>`
OCI Vision	`allow dynamic-group dynamic_group to manage ai-service-vision-family in compartment compartment_name`

Where:

- *dynamic_group* or *dynamic-group_name*: Is the dynamic group name you specified in Step 6.
- *compartment_name* or *compartment-name*: Is the compartment in which your Oracle Integration instance is located.

This enables the Oracle Integration instance associated with the dynamic group to call the specific OCI AI service that you want to use in this particular compartment.

Copyright © 2022, 2026, Oracle and/or its affiliates.

See About Document Understanding Policies, About Vision Policies, and About Speech Policies.

Extract Document Information with a Document Understanding Action

You can use AI to extract information from invoices, receipts, passports, drivers licenses, healthcare IDs, and custom documents with a document understanding action. You can also extract information from tables and plain text files, or classify documents. You can also create a processor job, get processor job status, and cancel a processor job.

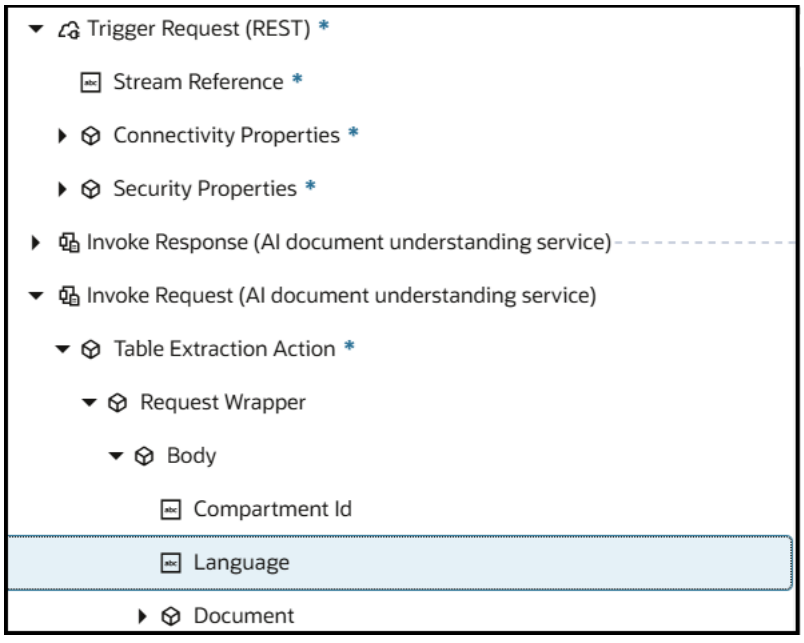
- Capabilities
- Prerequisites
- Invoke Oracle Cloud Infrastructure Document Understanding from an Integration

You can optionally store and access the invoices, receipts, passports, drivers licenses, and healthcare IDs from which to exact data in Oracle Cloud Infrastructure Object Storage. See Object Storage.

Capabilities

Oracle Cloud Infrastructure Document Understanding is an AI service that enables you to extract text, tables, and other key data from PDF, PNG, and JPEG files through APIs and command line interface tools. Oracle Cloud Infrastructure Document Understanding automates repetitive business processing tasks with prebuilt AI models and customizes document extraction to satisfy your industry-specific needs. Oracle Integration supports using Oracle Cloud Infrastructure Document Understanding in an integration with the document understanding action.

Support is also provided in the mapper for extracting data in multiple languages.



See AI Document Understanding.

Copyright © 2022, 2026, Oracle and/or its affiliates.

Prerequisites

See Prerequisites for information about the prerequisites you must satisfy in the Oracle Cloud Console.

Invoke Oracle Cloud Infrastructure Document Understanding from an Integration

1. Add an **OCI Document Understanding** action to an integration in either of the following ways:
   - On the side of the canvas, click **Actions** and drag the **OCI Document Understanding** action to the appropriate location.
   - Click + at the location where you want to add the document understanding action, then select **OCI Document Understanding**.
2. Enter a name and optional description.
3. From the **Select Categories** list, select the task you want to perform.
   - Pre-trained Models
   - Custom models
   - Processor Job

Pre-trained Models

Follow these steps to configure a pretrained model action for use.

1. From the **Action** list, select the action to perform, then click **Continue**.

Action	Description	See...
Analyze document (with confidence score)	Extracts an element from a document and provides a confidence score on its certainty. A confidence score is a numerical value between 0 and 1 indicating how certain the model is about the accuracy of the extracted piece of information, such as a text field, key-value pair, or element within a document. For example, extracting the date from an invoice may yield a confidence score of 0.86, meaning the model is 86% confident that the extracted value is correct. Confidence scores help you identify which elements were extracted with high certainty, and which elements may require human review.	

Copyright © 2022, 2026, Oracle and/or its affiliates.

Action	Description	See...
Analyze document	Extracts information from a document. This action accepts an inline document in base64 format or retrieves the document from object storage, performs the extraction, and returns an immediate response.	Step 2 to configure this selection in the wizard.
Table extraction	Accepts a text file that contains tables and extracts the elements in tabular format. For example, if a receipt in PDF format contains a table that includes the taxes and total amount, the table is identified and the table structure is extracted. See Table Extraction.	
Text extraction	Identifies the plain text in a file and extracts and returns the results in words and lines. With this action, you don't select a specific document type such as an invoice or driver's license. The plain text document you provide is processed. This action accepts an inline document in base64 format or retrieves the document from object storage and returns the words and lines.	See Step 3 to configure this selection in the wizard.
Document classification	Classifies and returns the type of document. For example, if you send a pay slip, check, or bank statement it returns the pay slip, check, or bank statement, respectively as the document type. This action accepts an inline document in base64 format or retrieves the document from object storage and returns the document type in an immediate response. See Document Classification.	Step 3 to configure this selection in the wizard.

2. If you selected the **Analyze document** or **Analyze document (with confidence score)** action:
   a. From the **Compartment** list, select the Oracle Cloud Infrastructure compartment in which Oracle Integration is installed.
   b. From the **Document type** list, select an action to perform, then click **Continue**.

Copyright © 2022, 2026, Oracle and/or its affiliates.

Element	Description
Document Type	Select the type of document from which to extract information. Key value extraction can identify values for predefined keys in a document. • **Invoice**: Identifies values for predefined keys in an invoice. For example, if an invoice includes a vendor name, total, and invoice ID, Oracle Cloud Infrastructure Document Understanding can identify these values and return them as a key-value pair. • **Receipt**: Identifies values for predefined keys in a receipt. For example, if a receipt includes a merchant name, merchant address, or merchant phone number, Oracle Cloud Infrastructure Document Understanding can identify these values and return them as a key-value pair. • **Driver license**: Identifies values for predefined keys in a US or UK driver's documentation. For example, if a driver ID includes an issue date, region, and expiry date, Oracle Cloud Infrastructure Document Understanding can identify these values and return them as a key-value pair. • **Passport**: Identifies values for predefined keys in an MRZ-supported passport. For example, if a passport includes nationality and date of issue, Oracle Cloud Infrastructure Document Understanding can identify these values and return them as a key-value pair. • **Healthcare Insurance ID**: Identifies values for predefined keys in a healthcare insurance ID card and returns these values as a key-value pair. Your selection is passed into the mapper for you to map.
Input source	Select the source location from which to get the input document. • **Inline**: Accepts the document through base64 format in the request payload of the mapper. • **Object storage**: Reads the document from the object storage bucket. – **Input storage bucket**: Select the storage bucket from which to read the file. – **Namespace**: Select the namespace for the object storage bucket. After completing configuration in the wizard, you must specify the file name to read from the object storage bucket in the mapper.

3. If you selected the **Table extraction**, **Text extraction**, or **Document classification** action:
   a. Select the following information, then click **Continue**.

Copyright © 2022, 2026, Oracle and/or its affiliates.

Element	Description
Compartment Name	Select the Oracle Cloud Infrastructure compartment in which Oracle Integration is installed.
Input source	Select the source location from which to get the input document. • **Inline**: Accepts the document through base64 format in the request payload of the mapper. • **Object storage**: Reads the document from the object storage bucket and returns the response based on the selected document type. – **Input storage bucket**: Select the storage bucket from which to read the file. – **Namespace**: Select the namespace for the object storage bucket. After completing configuration in the wizard, you must specify the file name to read from the object storage bucket in the mapper.

4. On the Summary page, click **Finish**.

Custom models

You can build a custom model to detect and extract your own defined fields in documents. For example, you can create a custom key-value model for a job application form that includes name, job ID, and email address fields. Before you can use a custom model in an integration, you must create your own model. See Custom Models.

Follow these steps to configure a custom model for use.

1. From the **Action** list, note that **Analyze document** is the only option available and cannot be deselected, then click **Continue**.
2. Select the following information, then click **Continue**.

Element	Description
Compartment	Select the Oracle Cloud Infrastructure compartment in which Oracle Integration is installed.
Model	Select the model that you created in Oracle Cloud Infrastructure.

Copyright © 2022, 2026, Oracle and/or its affiliates.

Element	Description
Input source	Select the source location from which to get the input document. • **Inline**: Accepts the document through base64 format in the request payload of the mapper. • **Object storage**: Reads the document from the object storage bucket and returns the response based on the selected document type. – **Input storage bucket**: Select the storage bucket from which to read the file. – **Namespace**: Select the namespace for the object storage bucket. After completing configuration in the wizard, you must specify the file name to read from the object storage bucket in the mapper.

3. On the Summary page, click **Finish**.

Processor Job

Follow these steps to configure a processor job action for use.

1. Select an action to perform, then click **Continue**.

Action	Description	See...
Create a processor job	Creates a processor job. This action reads a file from object storage and provides a response with a processor job ID.	Step 2 to configure this selection in the wizard.
Get a processor job status	Gets the status of the processor job. This action takes the processor job ID as a parameter from the **Create a processor job** action and returns the status of the processor job.	Step 3. The Summary page is displayed and no further wizard configuration is required.
Cancel a processor job	Cancels the processor job submitted through a **Create a processor job** action.	Step 3. The Summary page is displayed and no further wizard configuration is required.

2. Select the following information, then click **Continue**.

Element	Description
Compartment Name	Select the Oracle Cloud Infrastructure compartment in which Oracle Integration is installed.
Output storage bucket	Specify the output storage location for the file.
Namespace	Select the namespace for the object storage bucket name.

3. On the Summary page, click **Finish**.
Upon completing configuration in the wizard, you must perform specific tasks in the mapper based on the action and file source you configured. The following sections provide examples of some of the above selections.

Copyright © 2022, 2026, Oracle and/or its affiliates.

- Configure the Mapper to Read the Inline Document for the Analyze Document Action
- Configure the Mapper to Read the Document from a Storage Bucket for the Analyze Document Action
- Configure the Mapper to Read the Inline Document for the Analyze Document (with confidence score) Action
- Configure the Mapper to Read the Inline Document for the Document Classification Action
- Configure the Mapper to Read the Document from a Storage Bucket for the Document Classification Action
- Configure the Mapper to Read the Inline Document for the Table Extraction Action
- Configure the Mapper to Read the Inline Document for the Text Extraction Action
- Configure the Mapper for the Create a Processor Job Action
- Configure the Mapper for the Get a Processor Job Status Action
- Configure the Mapper for the Cancel a Processor Job Action
- Configure the Mapper to Read the Inline Document for the Custom Model Action

Configure the Mapper to Read the Inline Document for the Analyze Document Action

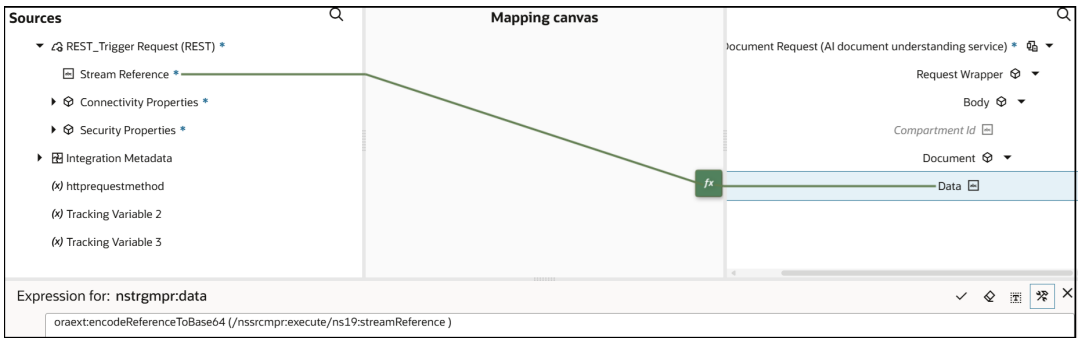
You send the data in base-64 format for any document type that you selected in the wizard (for example, invoice, receipt, passport, driver's license, or healthcare insurance ID).

1. Expand the target **Document** element.
2. Right-click **Data**, and select **Create target node**.
3. Click **Functions** .
4. Click **Design View** in the Expression Builder.
5. In the **Functions** section, expand **Advanced**, and drag **encodeReferenceToBase64** into the Expression Builder. This step is required for all document types (invoice, receipt, drivers license, and passport).

   ```
   oraext:encodeReferenceToBase64 ( )
   ```

6. Drag **Stream Reference** from the **Sources** section into the Expression Builder.

   ```
   oraext:encodeReferenceToBase64 (/nssrcmpr:execute/ns19:streamReference )
   ```



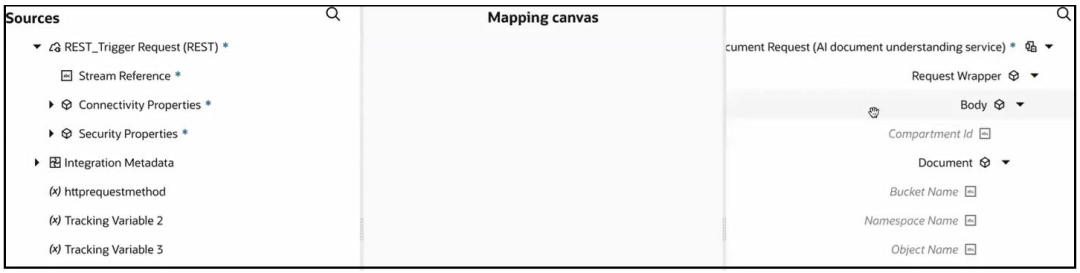
Copyright © 2022, 2026, Oracle and/or its affiliates.

7. Exit the mapper.
8. Add a log action and select fields to log to the activity stream.
   The document understanding action is now configured.

Configure the Mapper to Read the Document from a Storage Bucket for the Analyze Document Action

You must select the file name to read from the object storage bucket in the mapper (for example, invoice, receipt, passport, driver's license, or healthcare insurance ID).

1. Drag the file name from the **Sources** section to the **Object Name** target element.
   If you also want to override other values you defined in the wizard, you can specify values for the **Component id**, **Bucket Name**, and **Namespace Name** target elements.



2. When complete, exit the mapper.
   The document understanding action is now configured.

Configure the Mapper to Read the Inline Document for the Analyze Document (with confidence score) Action

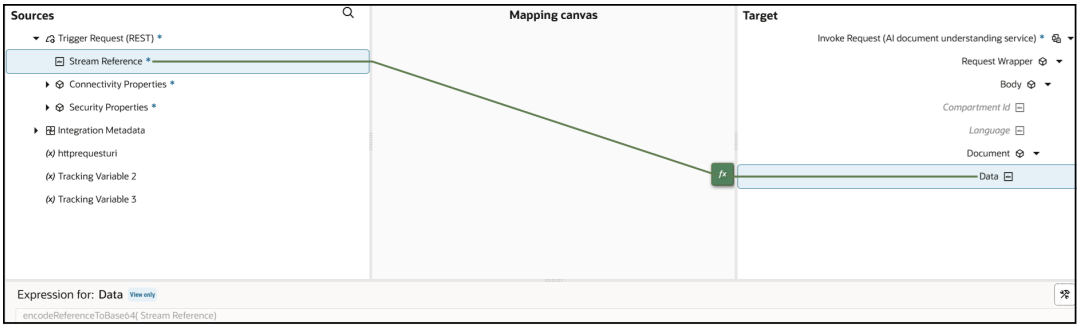
You send the data in base-64 format in the request mapper for any document type that you selected in the wizard (for example, invoice, receipt, passport, driver's license, or healthcare insurance ID).

1. Expand the target **Document** element.
2. Right-click **Data**, and select **Create target node**.
3. Click **Functions** .
4. Click **Design View** in the Expression Builder.
5. In the **Functions** section, expand **Advanced**, and drag **encodeReferenceToBase64** into the Expression Builder. This step is required for all document types (invoice, receipt, drivers license, passport, and healthcare ID).

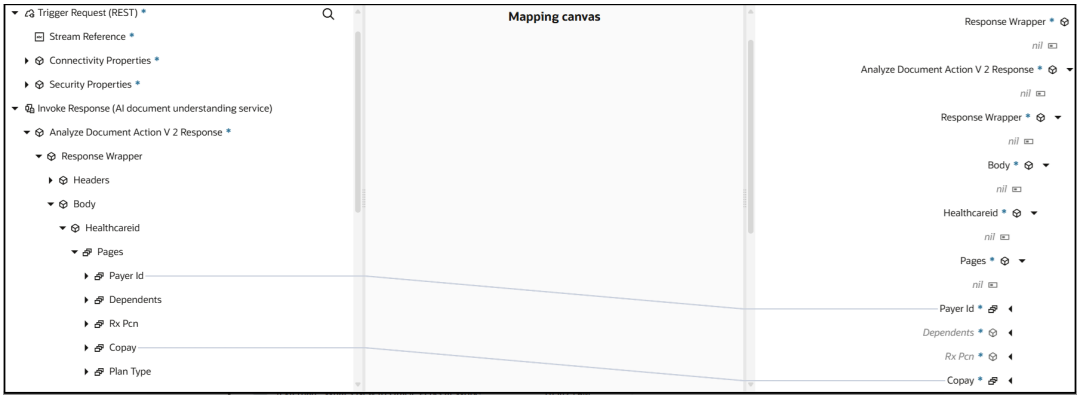
   ```
   oraext:encodeReferenceToBase64 ( )
   ```

6. Drag **Stream Reference** from the **Sources** section into the Expression Builder.

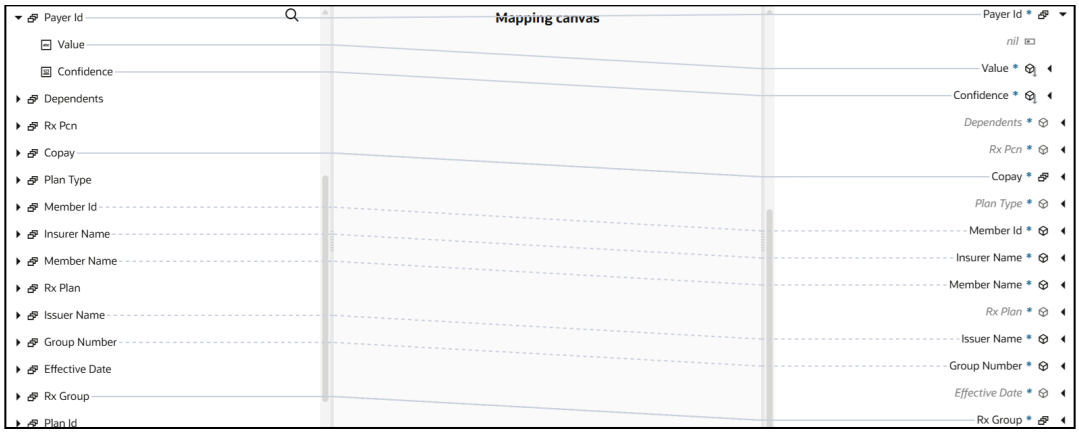
Copyright © 2022, 2026, Oracle and/or its affiliates.



7. In the response mapper, expand the source and target elements. For this example, the payload is the contents of a healthcare ID.



8. Map the source and target elements. Note the **confidence** elements.



The document understanding action is now configured.

Configure the Mapper to Read the Inline Document for the Document Classification Action

You send the data in base-64 format for any document type that you selected in the wizard for the **Analyze document** action.

1. Expand the target **Document** element.
2. Right-click **Data**, and select **Create target node**.

Copyright © 2022, 2026, Oracle and/or its affiliates.

3. Click **Functions** .

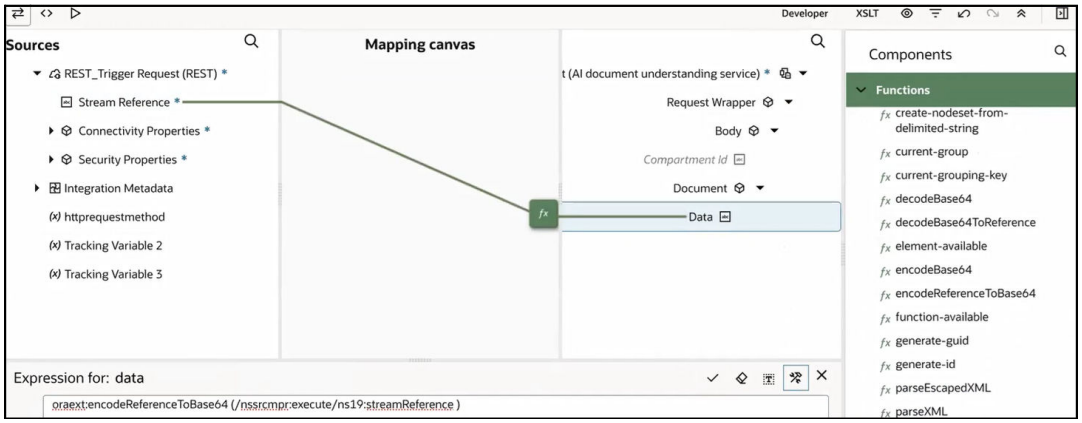
4. Click **Design View** in the Expression Builder.

5. In the **Functions** section, expand **Advanced**, and drag **encodeReferenceToBase64** into the Expression Builder.

   ```
   oraext:encodeReferenceToBase64 ( )
   ```

6. Drag **Stream Reference** from the **Sources** section into the Expression Builder.

   ```
   oraext:encodeReferenceToBase64 (/nssrcmpr:execute/nssrcmpr:attachments/
   ns19:attachment/ns19:attachmentReference )
   ```



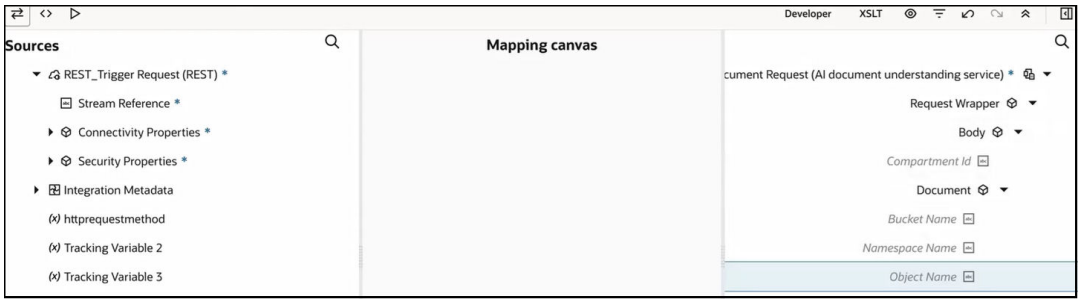
7. Exit the mapper.

8. Add a log action and select fields to log to the activity stream.
   The document understanding action is now configured.

Configure the Mapper to Read the Document from a Storage Bucket for the Document Classification Action

You must select the file name to read from the object storage bucket in the mapper.

1. Drag the file name from the **Sources** section to the **Target Object Name** element.
   If you also want to override other values you defined in the wizard, you can specify values for the **Component id**, **Bucket Name**, and **Namespace Name** target elements.



Copyright © 2022, 2026, Oracle and/or its affiliates.

2. Exit the mapper.
3. Add a log action and select fields to log to the activity stream.
   The document understanding action is now configured.

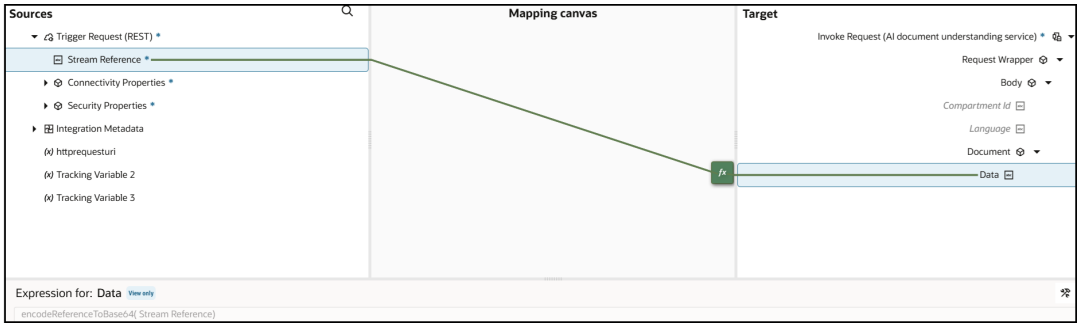
Configure the Mapper to Read the Inline Document for the Table Extraction Action

You send the data in base-64 format for any document type that you selected in the wizard for the **Table Extraction** action.

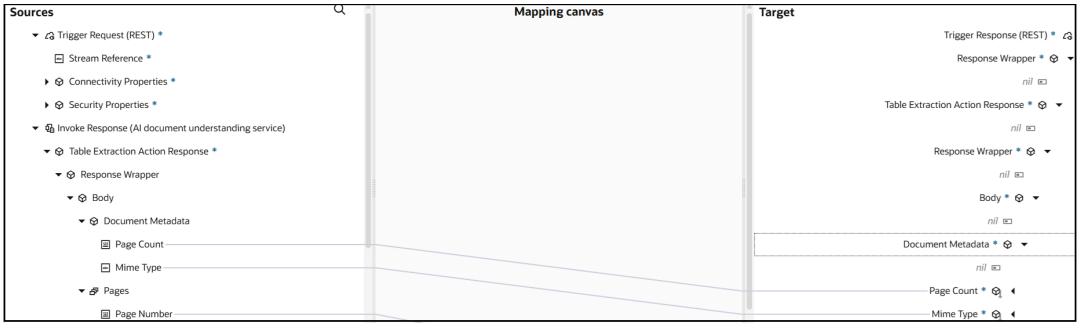
1. Expand the target **Document** element.
2. Right-click **Data**, and select **Create target node**.
3. Click **Functions**  .
4. Click **Design View**  in the Expression Builder.
5. In the **Functions** section, expand **Advanced**, and drag **encodeReferenceToBase64** into the Expression Builder.

   ```
   oraext:encodeReferenceToBase64 ( )
   ```

6. Drag **Stream Reference** from the **Sources** section into the Expression Builder.

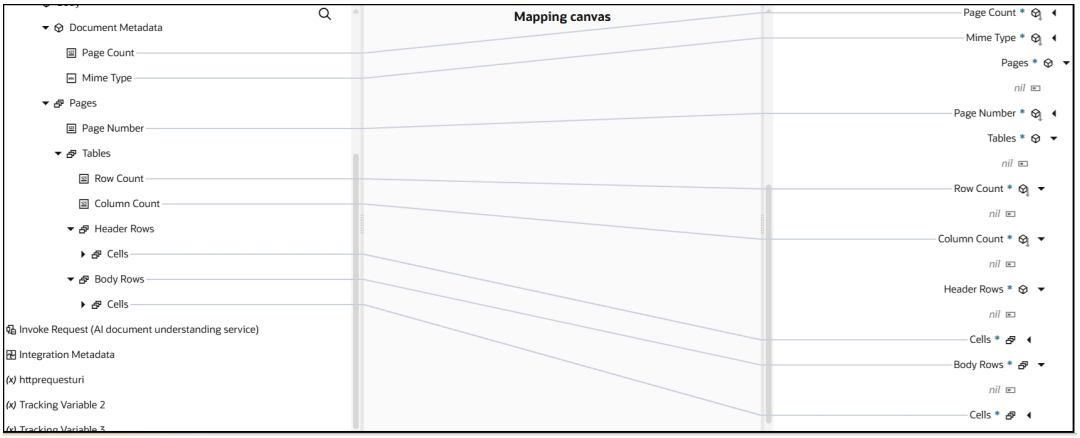


7. In the response mapper, expand the source and target elements.



8. Map the source and target elements. For this example, the payload is in tabular format.

Copyright © 2022, 2026, Oracle and/or its affiliates.



Configure the Mapper to Read the Inline Document for the Text Extraction Action

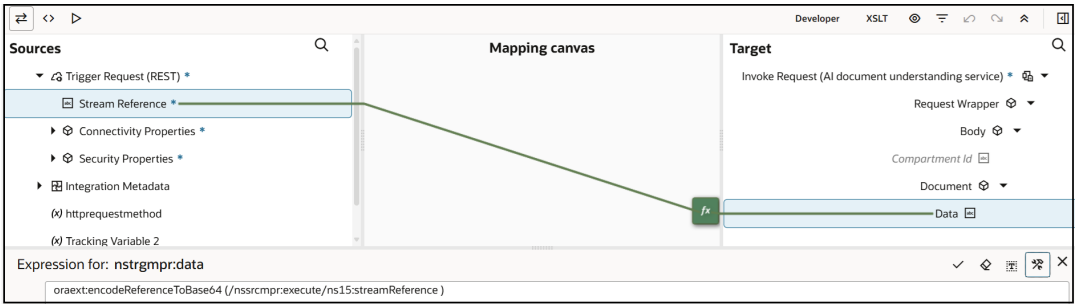
You send the data in base-64 format for any document type that you selected in the wizard for the **Text Extraction** action.

1. Expand the target **Document** element.
2. Right-click **Data**, and select **Create target node**.
3. Click **Functions** .
4. Click **Design View** in the Expression Builder.
5. In the **Functions** section, expand **Advanced**, and drag **encodeReferenceToBase64** into the Expression Builder.

   ```
   oraext:encodeReferenceToBase64 ( )
   ```

6. Drag **Stream Reference** from the **Sources** section into the Expression Builder.

   ```
   oraext:encodeReferenceToBase64 (/nssrcmpr:execute/ns19:streamReference )
   ```



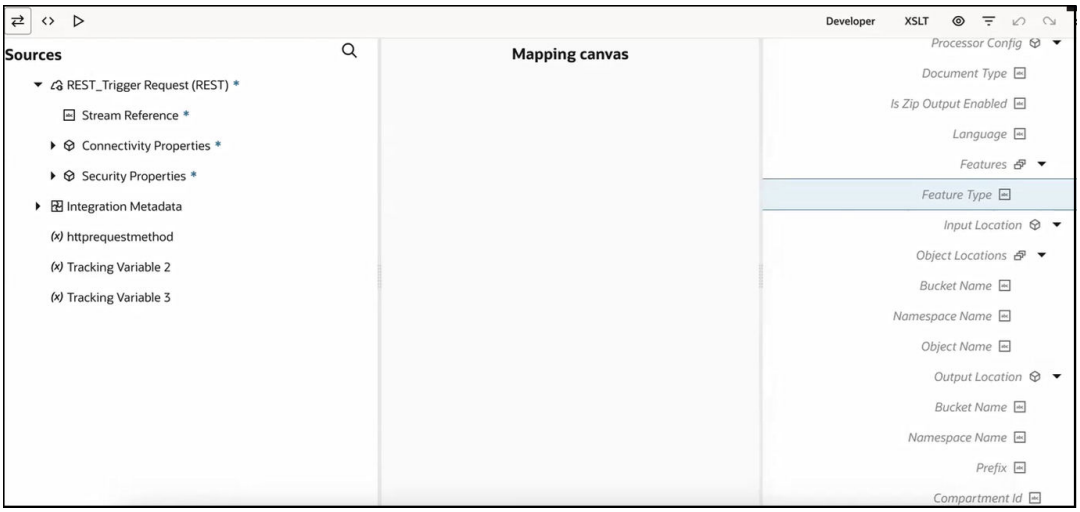
7. Add a log action and select fields to log to the activity stream.
   The document understanding action is now configured.

Copyright © 2022, 2026, Oracle and/or its affiliates.

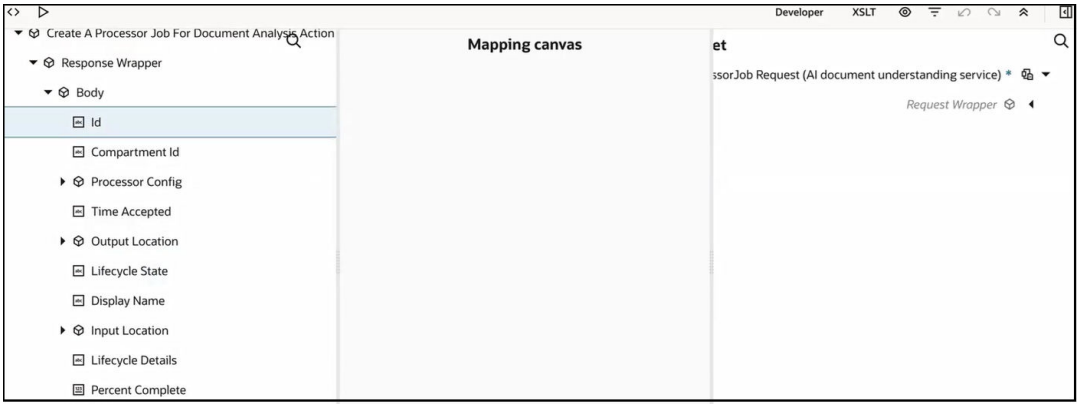
Configure the Mapper for the Create a Processor Job Action

This action returns a processor job ID. The input from the storage bucket is sent through the request mapper. The output is sent to the object storage bucket.

1. Expand the **Target Request Wrapper** to specify the target values. Most values are optional such as **Display Name**, **Processor Config**, and **Document Type** (for example, an invoice). However, the following elements are mandatory:
   a. **Feature Type**: Specify the document analysis type (for example, **TEXT_EXTRACTION** or others). For a complete list of available types, see DocumentFeature Reference.
   b. **Bucket Name** (under **Input Location**): Specify the input object storage bucket from which to get the file.
   c. **Namespace** (under **Input Location**): Specify the namespace for the input object storage bucket.
   d. **Prefix** (under **Output Location**): Specify the prefix. The folder in object storage is created with the prefix that you specify. If you also want to override values you defined in the wizard, you can specify values for the **Bucket Name** and **Namespace Name** target elements under **Output Location**.



The response mapper returns a job ID (**id** element) and a job status (**Percent Complete** element).



Copyright © 2022, 2026, Oracle and/or its affiliates.

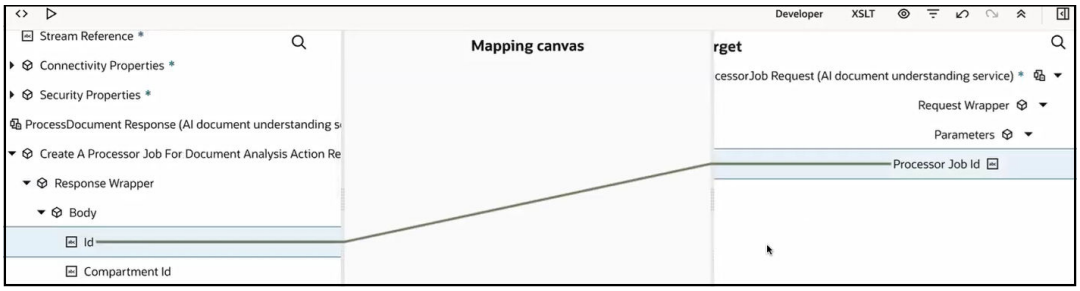
2. Exit the response mapper.
   The document understanding action is now configured.

Configure the Mapper for the Get a Processor Job Status Action

This action takes the job ID returned in the response mapper of the **Create a processor job** action. This action uses that value to get the status of the processor job. You configure this use case as follows:

- Configure an initial OCI Document Understanding action with the **Create a processor job** action. This action returns the job ID.
- Configure a second OCI Document Understanding action with the **Get a processor job status** action. This action gets the status of the processor job.

1. Map the **Sources Create A Processor Job id** element to the **Target Processor Job id** element.



2. Exit the mapper.
3. Add a log action and select fields to log to the activity stream.
   The document understanding action is now configured.

Configure the Mapper for the Cancel a Processor Job Action

This action takes the job ID returned in the response mapper of the Create a Processor Job action. This action uses that value to cancel the processor job. You configure this use case as follows:

- Configure an initial OCI Document Understanding action with the **Create a processor job** action. This action returns the job ID.
- Configure a second OCI Document Understanding action with the **Get a processor job status** action. This action gets the status of the processor job.
- Configure a third OCI Document Understanding action with the **Cancel a processor job** action. This action cancels the processor job. The job status must be in progress.

1. Map the **Sources Create A Processor Job id** element to the **Target Processor Job id** element.

Configure the Mapper to Read the Inline Document for the Custom Model Action

You send the data in base-64 format for any custom model that you selected in the wizard for the **Custom model** action.

1. Expand the target **Document** element.
2. Right-click **Data**, and select **Create target node**.

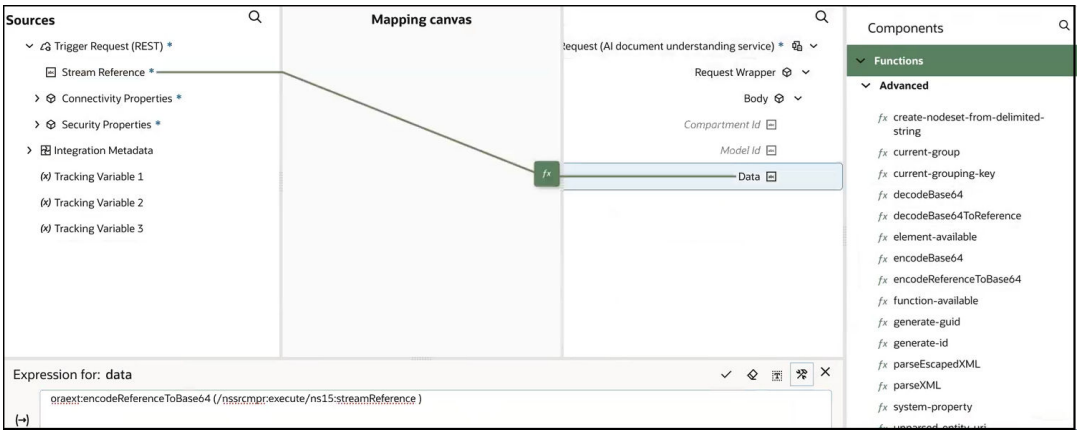
Copyright © 2022, 2026, Oracle and/or its affiliates.

3. Click **Functions** .
4. Click **Design View** in the Expression Builder.
5. In the **Functions** section, expand **Advanced**, and drag **encodeReferenceToBase64** into the Expression Builder.

   ```
   oraext:encodeReferenceToBase64 ( )
   ```

6. Drag **Stream Reference** from the **Sources** section into the Expression Builder.

   ```
   oraext:encodeReferenceToBase64 (/nssrcmpr:execute/ns15:streamReference )
   ```



Several design-time to runtime use cases using the document understanding action are provided. See Extract Content from an Invoice PDF Document with a Document Understanding Action and Extract Information from a Custom Document with a Document Understanding Action.

Watch a video to learn more:

Video

Analyze and Extract Information from Images with a Vision Action

You can use AI to perform deep-learning–based image analysis with a vision action. Oracle Cloud Infrastructure Vision enables you to automatically extract textual or visual information from images.

- Capabilities
- Prerequisites
- Invoke Oracle Cloud Infrastructure Vision from an Integration

Capabilities

OCI Vision is an AI service that enables you to detect objects and text in images, classify images using labels, and detect faces in images. You can use OCI Vision to detect visual anomalies (for example, anomalies in manufacturing), identify objects in images to automate counting of common items (such as packages, products, shipments, vehicles, and so on),

Copyright © 2022, 2026, Oracle and/or its affiliates.

identify items that need attention (such as overgrowth of vegetation near an electric power line), classify items (for example, digital media assets) to organize them, identify textual/visual data in images to power analytic applications, and much more. Oracle Integration supports using OCI Vision in an integration with the vision action.

See AI Vision.

Prerequisites

See Prerequisites for information on the prerequisites you must satisfy in the Oracle Cloud Console.

Invoke Oracle Cloud Infrastructure Vision from an Integration

1. Add a **Vision** action to an integration in either of the following ways:
   - On the side of the canvas, click **Actions** and drag the **OCI Vision** action to the appropriate location.
   - Click + at the location where you want to add the OCI Vision action, then select **OCI Vision**.
2. Enter a name and optional description, then click **Continue**.
3. Select the following information, then click **Continue**.

Element	Description
Categories	Select one of the following categories to use: • **Pre-trained models**: Pre-trained models let you perform image analysis without the need to create your own model. Pre-trained models enable you to use AI with no data science experience. You can provide an image to the OCI Vision action service and get information about the objects, text, and faces in the image. You can also classify the objects in an image using labels. • **Image Jobs**: Image jobs let you perform analysis on a set of images by creating batch jobs. It also enables you to get information about a batch image job and cancel a batch image job. • **Custom models**: Custom models let you perform custom object detection. For example, you can build and use a model that detects the location of objects (such as a car, a tree, a dog, and so on) in images. The output of this example custom model can include a confidence score, and the coordinates of the bounding box for each object found. You can create a data set, train a model based on the data set, and use that custom model for your scenario. For information about building and training custom models for object detection, see Building a Custom Model and Train the Custom Model.

Copyright © 2022, 2026, Oracle and/or its affiliates.

Element	Description
Action	If you selected the **Pre-trained models** or **Custom models** category, select the **Analyze Image** action to perform image analysis. If you selected the **Image Jobs** category, select one of the following jobs: • **CancelImageJob**: Cancels the image job using the image job ID you specify. • **CreateImageJob**: Creates an image job. This action reads the images from object storage and returns the results of the analysis along with the image job ID. • **GetImageJob**: Gets the status of the image job using the image job ID you specify. This action returns the status of the image job.

4. You can perform the following image analysis tasks using the OCI Vision action:

   - **Image Classification**: Classifies objects in an image using labels. For example, if an image is a surveillance image of a street that has a vehicle, an electric power line, and a tree, OCI Vision can identify these objects, classify them using labels, and return this information. This might be helpful to identify items that need attention such as overgrowth of vegetation near an electric power line.
   - **Object Detection**: Identifies objects in an image. For example, if an image includes vehicles such as a car, a bus, or a truck, OCI Vision can identify these objects and return this information.
   - **Text Detection**: Identifies words and lines of text in an image. For example, if an image is a visa application form and it contains text such as name, city, citizenship, country and other text content, OCI Vision can recognize the text and return this information.
   - **Face Detection**: Identifies faces and facial features of people in an image. For example, if an image is a photograph of a group of people, OCI Vision can identify the faces and facial features of the people in the image and return this information.

   To select an image analysis task, perform the following.

   a. If you selected the **Pre-trained models** category in Step 3, then select the following information, and click **Continue**.

Element	Description
Compartment	Select the Oracle Cloud Infrastructure compartment in which your Oracle Integration is installed.
Task	Select the task to perform: • **Image Classification** • **Object Detection** • **Text Detection** • **Face Detection**
Input source	Select an input source where the image to analyze will be available: • **Inline**: If you select this option, you need to provide the image in base-64 format in the request payload of the mapper. • **Object storage**: If you select this option, you need to make the image available in an object storage bucket. You can select the object storage bucket that has the image in the **Input storage bucket** field.

Copyright © 2022, 2026, Oracle and/or its affiliates.

Element	Description
Input storage bucket	This field is available only when you select **Object storage** as the input source. Select the input storage bucket that contains the image to analyze from the list of buckets.
Namespace	This field is available only when you select **Object storage** as the input source. Select a namespace. To understand namespaces, see Understanding Object Storage Namespaces.
Maximum no. of results to be returned	Select the maximum number of results to be returned after the image has been analyzed using the selected task.

b. If you selected the **Image Jobs** category in Step 3, then select the following information, and click **Continue**.

Element	Description
Compartment	Select the Oracle Cloud Infrastructure compartment in which your Oracle Integration is installed.
Task	Select the task to perform: • **Image Classification** • **Object Detection** • **Text Detection** • **Face Detection**
Output storage bucket	Select the output storage bucket where you want to store the results of the image analysis task.
Namespace	Select a namespace. To understand namespaces, see Understanding Object Storage Namespaces.
Maximum no. of results to be returned	Select the maximum number of results to be returned after the images have been analyzed using the selected task.

5. If you selected the **Custom models** category in Step 3, then select the following information, and click **Continue**.

Element	Description
Compartment	Select the Oracle Cloud Infrastructure compartment in which your Oracle Integration is installed.
Task	Select the task to perform: • **Object Detection**
Model	Select a custom model ID to use for object detection from the list of models available in the selected compartment.
Input source	Select an input source where the image to analyze will be available: • **Inline**: If you select this option, you need to provide the image in base-64 format in the request payload of the mapper. • **Object storage**: If you select this option, you need to make the image available in an object storage bucket. You can select the object storage bucket that has the image in the **Input storage bucket** field.

Copyright © 2022, 2026, Oracle and/or its affiliates.

Element	Description
Input storage bucket	This field is available only when you select **Object storage** as the input source. Select the input storage bucket that contains the image to analyze from the list of buckets.
Namespace	This field is available only when you select **Object storage** as the input source. Select a namespace. To understand namespaces, see Understanding Object Storage Namespaces.
Maximum no. of results to be returned	Select the maximum number of results to be returned after the image has been analyzed using the selected task.

6. On the Summary page, click **Finish**.

Upon completing the configuration in the wizard, you must perform specific tasks in the mapper based on the category and input source you configured.

- Configure the Mapper to Read the Inline Image for the Pre-trained Models and Custom Models Categories
- Configure the Mapper to Read the Image from a Storage Bucket for the Pre-trained Models and Custom Models Categories
- Configure the Mapper for the CancelImageJob Action
- Configure the Mapper for the CreateImageJob Action
- Configure the Mapper for the GetImageJob Action

Configure the Mapper to Read the Inline Image for the Pre-trained Models and Custom Models Categories

You must send the data in base-64 format. Open the mapper and define the mappings between the source and target elements as needed.

1. Expand the target **Image** element.
2. Right-click **Data**, and select **Create target node**.
3. Click **Functions**  .
4. Click **Design View**  in the Expression Builder.
5. In the **Functions** section, expand **Advanced**, and drag **encodeReferenceToBase64** into the Expression Builder for the **Data** target element.

   ```
   oraext:encodeReferenceToBase64 (   )
   ```

6. Drag **Stream Reference** from the **Sources** section into the Expression Builder.

   ```
   oraext:encodeReferenceToBase64 (/nssrcmpr:execute/ns23:streamReference)
   ```

> **Note**
>
> You can optionally specify **Compartment Id** in the mapper to override the value you selected initially for **Compartment**.

Copyright © 2022, 2026, Oracle and/or its affiliates.

7. Exit the mapper.
   The vision action is now configured.

Configure the Mapper to Read the Image from a Storage Bucket for the Pre-trained Models and Custom Models Categories

You must select the object name (of the image) to read from the object storage bucket in the mapper. Open the mapper and define the mappings between the source and target elements as needed.

1. Drag the object name (file name of the image) from the **Sources** section to the **Object Name** element in the **Target** section.

> **Note**
>
> - Object name is the unique identifier you defined when uploading the image (object) to the object storage bucket. For more information about object names, see Object Names.
> - You can optionally specify **Compartment Id**, **Namespace Name**, and **Bucket Name** in the mapper to override the value you selected initially for these elements in Step 4.

2. Exit the mapper.
   The vision action is now configured.

Configure the Mapper for the CancelImageJob Action

This action cancels the image job using the image job ID you specify in the request payload. You can get the image job ID from the response of the **CreateImageJob** action. You can only cancel an image job that is in progress.

1. Map the job ID from the **Sources** section to the target **Image Job Id** in the **Target** section.
2. Exit the mapper.
   The vision action is now configured.

Configure the Mapper for the CreateImageJob Action

This action creates an image job to read a set of images from the object storage bucket and perform the analysis tasks specified. The input from the storage bucket is sent through the request mapper. The output is sent to the object storage bucket. The output returned also includes the image job ID. Open the mapper and define the mappings between the source and target elements as needed.

1. Map the appropriate source elements to the following target elements:
   - **Features**
   - **Feature Type**
   - **Source Type** under **Input Location**
   - **Object Locations** under **Input Location**
   - **Bucket Name** under **Object Locations**
   - **Namespace Name** under **Object Locations**
   - **Object Name** under **Object Locations**
   - **Prefix** under **Output Location**

Copyright © 2022, 2026, Oracle and/or its affiliates.

- **Display Name** under **Output Location**
- **Is Zip Output Enabled** under **Output Location**

The values such as **Display Name** and **Is Zip Output Enabled** are optional. For information about the above parameters, see ImageJob Reference.

> **Note**
>
> - Feature type is the task you selected in Step 4b.
> - Prefix is used in the file name of the output files.

2. Exit the mapper.
   The vision action is now configured.

Configure the Mapper for the GetImageJob Action

This action returns the status of the image job using the image job ID you specify in the request payload. You can get the image job ID from the response of the **CreateImageJob** action.

1. Map the job ID from the **Sources** section to the target **Image Job Id** in the **Target** section.
2. Exit the mapper.
   The vision action is now configured.

Generate, Embed, or Rerank Text with a Generative AI Action

You can use AI to generate, embed, or rerank text with a generative AI action. Oracle Cloud Infrastructure Generative AI provides answers or information to the questions you ask by generating text. It also enables you to create text embeddings and rerank text.

- Capabilities
- Prerequisites
- Invoke Oracle Cloud Infrastructure Generative AI from an Integration

Capabilities

OCI Generative AI is a fully managed AI service that offers customizable large language models (LLMs) for various use cases such as chat, generating text embeddings, and reranking text. Oracle Integration supports using OCI Generative AI in an integration with the generative AI action.

OCI Generative AI enables you to perform the following actions:

- **Chat**: You can prompt the OCI Generative AI chat model to generate text. You can ask questions and get text responses. For this action, OCI Generative AI returns a text response to your question or request. You can also include an image as part of your chat, and it will be used to provide the response. The OCI Generative AI's chat model enables you to brainstorm ideas, get answers to questions, write descriptions or content, explain concepts, perform image analysis, and much more.
- **EmbedText**: You can create embeddings for the text you provide using OCI Generative AI. The text can be a phrase, a sentence, or one or more paragraphs. An embedding is a numeric representation of a text. Embeddings are useful to find similarity between text, for semantic searches when the search is based on the meaning of the text rather than

Copyright © 2022, 2026, Oracle and/or its affiliates.

keywords, text classification, and so on. For this action, OCI Generative AI returns the embedded result.

- **Rerank**: You can rerank text responses based on input documents and a prompt. Rerank assigns an index and a relevance score to each document, indicating which document is most related to the prompt.

See Generative AI.

Prerequisites

See Prerequisites for information on the prerequisites you must satisfy in the Oracle Cloud Console.

Invoke Oracle Cloud Infrastructure Generative AI from an Integration

1. Add a **Generative AI** action to an integration in either of the following ways:
   - On the side of the canvas, click **Actions** and drag the **OCI Generative AI** action to the appropriate location.
   - Click + at the location where you want to add the OCI Generative AI action, then select **OCI Generative AI**.
2. Enter a name and optional description, then click **Continue**.
3. Select the following information, then click **Continue**.

Element	Description
Select Categories	Select one of the following categories to use: • **On demand**: This option enables you to use pre-trained foundational models (on-demand models) to perform actions using the OCI Generative AI without the need to create your own model. Pre-trained models enable you to use AI with no data science experience. • **Dedicated**: This option allows you to use compute resources for fine-tuning custom models or for hosting endpoints. The AI clusters are dedicated to your models and are not shared with other customers. You can provision your own endpoint if you use this option.
Action	If you selected the **On demand** category, select one of the following actions: • **Chat** • **EmbedText** If you selected the **Dedicated** category, select one of the following actions: • **Chat** • **Rerank** • **EmbedText**

4. Select the following information, then click **Continue**.

Element	Description
Region	Select the region you want to use. **Note**: Ensure you select the region in which OCI is provisioned for you.

Copyright © 2022, 2026, Oracle and/or its affiliates.

Element	Description
Compartment	Select the Oracle Cloud Infrastructure compartment in which your Oracle Integration is installed.
Model	This field is available only when you select the **On demand** category and the **Chat** action in Step 3. Select a model for your generative AI action. • If you select **GENERIC (llama)**, all the Meta Llama models are listed for selection in the **Model ID** field. • If you select **COHERE**, all the Cohere models are listed for selection in the **Model ID** field.
Model ID	This field is available only when you select the **On demand** category and the **Chat** or **EmbedText** action in Step 3. Select a model ID to use.
Endpoint	This field is available only when you select the **Rerank** action in Step 3. Select an endpoint to use.
Endpoint ID	This field is available only when you select the **Dedicated** category and the **Chat** or **EmbedText** action in Step 3. Select an endpoint ID to use.

5. On the Summary page, click **Finish**.
6. Open the mapper and define the mappings between the source and target elements as needed.
   For **Chat** action, perform the following steps.
   a. Expand the topmost request node in the **Target** section.
   b. Within the topmost request node in the **Target** section, expand **Request Wrapper**, expand **Chat Request**, expand **Messages**, and then expand **Content**.
   c. Within **Content**, right-click **Text**, and select **Create target node**.
   d. Click **Design View**  in the Expression Builder.
   e. You can prompt the OCI Generative AI chat model to generate text by specifying your request for the **Text** target element in the Expression Builder. In the Expression Builder, enter the question you want to ask or specify what information you need (for example, `'Tell me about ELE'`).
   f. You can also include an image as part of your chat. You need to send it in base-64 format. You need to specify the image URL in the target **Url** element (within **Image URL**).

Copyright © 2022, 2026, Oracle and/or its affiliates.

> **Note**
>
> In the mapper, you can optionally override the following values you initially selected when you performed Step 4:
>
> - **Compartment Id**: Use this to specify a compartment Id to override the default compartment that was selected for the model to be deployed.
> - **Api Format** (model): Use this to override the default API format used for model serving.
> - **Model Id**: Use this to override the initially selected model identifier.
>
> You can also optionally configure other fields as needed in the mapper based on the model you selected, including setting **servingType**, **Citation Quality**, **Preamble Override**, tuning parameters such as **Top p**/**Temperature**, and so on. See CohereChatRequest Reference.

For **EmbedText** action, perform the following steps.

a. Expand the topmost request node in the **Target** section.

b. Within the topmost request node in the **Target** section, expand **Request Wrapper**, and then expand **Body**.

c. Within **Body**, right-click **Inputs**, and select **Create target node**.

d. Click **Design View**  in the Expression Builder.

e. For the **Inputs** target element, in the Expression Builder, specify the text for which OCI Generative AI needs to generate an embedding.

f. Right-click **Input Type**, and select **Create target node**.

g. Optionally, you can specify the input type using the **Input Type** target element.
You can also optionally configure other fields as needed in the mapper, such as **Compartment Id**, **Model Id**, **Is Echo**, and so on.

For **Rerank** action, perform the following steps.

a. Expand the topmost request node in the **Target** section.

b. Within the topmost request node in the **Target** section, expand **Request Wrapper**, and then expand **Body**.

c. Within **Body**, right-click each of the **Documents** element, and select **Create target node**.

d. Click **Design View**  in the Expression Builder.

e. For each of the **Documents** target element, in the Expression Builder, specify the documents that you want to rerank. This is a list of document strings to rerank based on the query you specify.

f. Right-click **Input**, and select **Create target node**.

g. For the **Input** target element, in the Expression Builder, specify the input text (input query for search in the documents).
You can also optionally configure other fields as needed in the mapper, such as **Compartment Id**, **Endpoint Id**, **Is Echo**, and so on.

7. Exit the mapper.
The generative AI action is now configured.

Copyright © 2022, 2026, Oracle and/or its affiliates.

Here's what happens when you activate and run the integration based on the action you selected:

- **Chat**: Provides a text response to your question.
- **EmbedText**: Provides an embedding for the text you specified.
- **Rerank**: Provides an index and a relevance score to each document based on the input query you specified.

Perform Text Analysis and Translation with a Language Action

You can use AI to perform in-depth text analysis and machine translation with a language action. Oracle Cloud Infrastructure Language enables you to process unstructured text for sentiment analysis, entity recognition, classification, translation, and more.

- Capabilities
- Prerequisites
- Invoke Oracle Cloud Infrastructure Language from an Integration

Capabilities

OCI Language is a cloud-based AI service that enables you to build intelligent applications by using REST APIs and SDKs to process unstructured text for language detection, text classification, recognition of named entities, key phrase extraction, sentiment analysis, text translation, and detection of personal identifiable information. OCI Language can identify more than 100 languages in text. It also automatically recognizes at least 18 entity types, including the names of organizations and products. It makes text analysis easier for large volumes of text data. Oracle Integration supports using OCI Language in an integration with the language action.

See AI Language.

Prerequisites

See Prerequisites for information on the prerequisites you must satisfy in the Oracle Cloud Console.

Invoke Oracle Cloud Infrastructure Language from an Integration

1. Add a **Language** action to an integration in either of the following ways:
   - On the side of the canvas, click **Actions** and drag the **OCI Language** action to the appropriate location.
   - Click + at the location where you want to add the OCI Language action, then select **OCI Language**.
2. Enter a name and optional description. Select the following information, then click **Continue**.

Copyright © 2022, 2026, Oracle and/or its affiliates.

Element	Description
Select categories	Select one of the following categories: • **Actions** • **Jobs** The **Actions** category lets you perform analysis and translation on unstructured text. You will need to provide the text to process in the mapper using the **Text** target element. The **Jobs** category lets you perform analysis on unstructured text data at scale by enabling you to create asynchronous language jobs. It also enables you to get information about a language job or list language jobs. You will be able to process large volumes of textual information. You need to provide the documents to be processed in object storage. It also enables you to update, delete, or cancel a language job.

Copyright © 2022, 2026, Oracle and/or its affiliates.

Element	Description
Action	If you selected the **Actions** category, select one of the following actions to perform. You can enter the action name to filter the list. • **Language detection**: Detects languages based on the text provided, and includes a confidence score. OCI Language detects the language and returns the detected language along with a related confidence score (between 0 and 1). You can also specify a batch of records. • **Named entity recognition**: Identifies common entities, people, places, locations, email, and so on. OCI Language extracts the entities in the text records. It returns the type/subtype and confidence score (between 0 and 1) for each entity. • **Key phrase extraction**: Extracts an important set of phrases from a block of text. OCI Language extracts the key phrases from the text. For each key phrase, it returns a score (between 0 and 1) that highlights the importance of the key phrase in the context of the text. • **Sentiment analysis**: Identifies the tone of the text and classifies the expressions in the text into positive, negative, neutral, or mixed polarity. OCI Language supports both aspect-based and sentence-based sentiment analysis. For example, opinions, appraisals, emotions, or attitudes toward a topic, person, or entity. After the analysis, it returns a confidence score for each of the classes (positive, negative, neutral, or mixed). • **Personal identifiable information (PII)/ Private Health Information (PHI)**: Identifies, classifies, and de-identifies private information in unstructured text. It can also process healthcare records and detect private health information such as healthcare plan IDs, medical records numbers, and so on. OCI Language supports masking of PII. You can configure masking for entities such as bank account number, telephone number, age, and so on. The entities that you can configure are available in the mapper under the **MASKING** element. You can also specify masking options to specify the masking character, whether PII should be unmasked from the start/end, and the number of characters to leave unmasked. OCI Language helps identify and classify personal identifiable information such as name, age, address, email, telephone number, and so on. It returns the information that was identified and classified. • **Text classification**: Identifies the document category and subcategory that the text belongs to. OCI Language analyzes the text and automatically classifies it into a set of predetermined categories and sub-categories. It returns this information for each record that was classified. • **Text translation**: Translates text into the language of your choice. OCI Language translates the text you provide from the source language to the specified language. It returns the translated text. • **Health entity**: Detects and extracts healthcare entities from healthcare records such as electronic healthcare records (EHR), progress notes, and clinical trial documents. See Healthcare NLP Models and Analyzing Healthcare Data Using Healthcare NLP Models.

Copyright © 2022, 2026, Oracle and/or its affiliates.

Element	Description
	– You need to create a project in the Oracle Cloud Console to host your model. – You need to create a model. – To access the model, you need to create an endpoint for the model in the Oracle Cloud Console. If you selected the **Jobs** category, select one of the following jobs to perform. • **Create job**: Creates a new asynchronous language job. It reads the documents from object storage and returns the results of the analysis along with the job ID. • **Get job**: Gets the status of the language job using the job ID you specify. This action returns the status of the language job. • **List jobs**: Returns a list of language jobs in the specified compartment. • **Update job**: Updates the specified language job with details specified in the request payload. • **Delete job**: Deletes the specified language job. • **Cancel job**: Cancels the specified language job (including its tasks).

3. If you selected the **Actions** category and an action, select the following information:

Element	Description
Compartment	Select the Oracle Cloud Infrastructure compartment in which your Oracle Integration is installed. This field is not applicable if you selected the **Health entity** action.
endpointid	This field is available only if you selected the **Health entity** action. Specify an endpoint ID to use for inferencing. You need to specify the ID (OCID) of the model endpoint you created in the Oracle Cloud Console in Step 2.
Level	This field is available only if you selected the **Sentiment analysis** action. Select the level of sentiment analysis you want to perform from the **Available Options** list and move it to the **Selected Options** list. • **Sentence**: Enables you to perform sentence-level sentiment analysis with confidence scores for each sentence in the text. • **Aspect**: Enables you to extract the individual aspects in the input document and classify each of the aspects into one of the polarity classes (positive, negative, mixed, or neutral). With the predicted sentiment for each aspect, it also provides a confidence score for each of the classes, and their corresponding offsets in the input. Confidence scores closer to 1 indicate higher confidence in the label's classification and lower scores indicate lower confidence. The range of the confidence score for each class is between 0 to 1, and the cumulative scores of all the four classes sum to 1. This option is available in the **Selected Options** list by default. You can select one of the options or both the options, as needed. You can also specify the `"SENTENCE"` or `"ASPECT"` option using the **Basis** target element in the mapper.

Copyright © 2022, 2026, Oracle and/or its affiliates.

Element	Description
Entities to be detected	This field is available only if you selected the **Personal identifiable information (PII)/ Private Health Information (PHI)** action. Select an entity to detect from the list of entities (such as **Person name**, **Address**, **Age**, and so on) available in the drop-down list. If you want all the entities to be detected, you can select **All** from the drop-down list.
Masking configuration (optional)	This field is available only if you selected the **Personal identifiable information (PII)/ Private Health Information (PHI)** action. For PII masking, you can optionally select masking modes such as the following: • **MASK**: Masks (hides) PII with masking characters like '***', 'X', or by showing a portion of the information (for example: last four characters). – **Masking character (Optional)**: Enter a masking character to use. – **Number of characters to be left unmasked (Optional)**: Enter the number of characters to be left unmasked if you want to show only a portion of the information. – **Unmask options**: Select **From the start** or **From the end**. • **REPLACE**: If you select this option, then, in the **Replace with** field, specify the characters that will replaces PII. • **REMOVE**: Removes the PII entities from the output text. You can also specify masking modes and masking options in the request payload. The options specified in the request payload override the options specified in the wizard.

4. If you selected the **Jobs** category and the **Create job** action, select the following information.

Element	Description
Compartment	Select the Oracle Cloud Infrastructure compartment in which your Oracle Integration is installed.
Feature type	Select one of the following feature types: • **Pretrained language detection** • **Pretrained sentiment analysis** • **Pretrained keyphrase extraction** • **Pretrained language pii entities** • **Pretrained language translation** • **Pretrained named entity recognition** • **Pretrained text classification** • **Text classification** • **Named entity recognition** Pretrained models are ready-to-use AI models that perform the selected task (feature type) on the text you provide in the request payload.

Copyright © 2022, 2026, Oracle and/or its affiliates.

Element	Description
Configuration	This field is available only if you selected the **Pretrained sentiment analysis** feature type. Select sentiment analysis options from the **Available Options** list and move them to the **Selected Options** list. • **Sentence**: Enables you to perform sentence-level sentiment analysis with confidence scores for each sentence in the text. • **Aspect**: Enables you to extract the individual aspects in the input document and classify each of the aspects into one of the polarity classes (positive, negative, mixed, or neutral). With the predicted sentiment for each aspect, it also provides a confidence score for each of the classes, and their corresponding offsets in the input. Confidence scores closer to 1 indicate higher confidence in the label's classification and lower scores indicate lower confidence. The range of the confidence score for each class is between 0 to 1, and the cumulative scores of all the four classes sum to 1. This option is available in the **Selected Options** list by default. You can also specify the `"SENTENCE"` or `"ASPECT"` option using the **Basis** target element in the mapper.
Entities to be detected	This field is available only if you selected the **Pretrained language pii entities** feature type. Select **All** from the drop-down list.
Masking configuration (optional)	This field is available only if you selected the **Pretrained language pii entities** feature type. For PII masking, you can optionally select masking modes such as the following: • **MASK**: Masks (hides) PII with masking characters like '***', 'X', or by showing a portion of the information (for example: last four characters). – **Masking character (Optional)**: Enter a masking character to use. – **Number of characters to be left unmasked (Optional)**: Enter the number of characters to be left unmasked if you want to show only a portion of the information. – **Unmask options**: Select **From the start** or **From the end**. • **REPLACE**: Replaces PII with a designated sequence of characters. • **REMOVE**: Removes the PII entities from the output text. You can also specify masking modes and masking options in the request payload. The options specified in the request payload override the options specified in the wizard.
Source language	This field is available only for the **Pretrained language translation** feature type. Select the source language of the text to be translated from the drop-down list.
Target language	This field is available only for the **Pretrained language translation** feature type. Select one or more target languages from the **Available Options** list and move them to the **Selected Options** list to translate the text from the source language to the selected target language(s).

Copyright © 2022, 2026, Oracle and/or its affiliates.

Element	Description
Document Type	Select one of the following document types: • **TXT**: Use this option to provide text files as input files. • **CSV**: Use this option to provide comma-separated value (CSV) files as input files. This field is not available for the **Pretrained language translation** feature type.
Input bucket	Select the input storage bucket that contains the document(s) to analyze. **Note**: • For **TXT** document type, all the text files in the bucket are selected for processing. • For **CSV** document type, you need to specify the particular CSV file to use in the request payload along with other parameters (such as row/column details). See DocumentsConfiguration Reference. • For the **Pretrained language translation** feature type, you can specify a particular text or CSV file to use in the request payload.
Input prefix (Optional)	This field is available only if you select the **TXT** document type. Optionally, you can specify an input prefix (which is similar to a folder name). This field is available for the **Pretrained language translation** feature type too.
Output bucket	Select the output storage bucket to store the results.
Output prefix (Optional)	Optionally, you can specify an output prefix (which is similar to a folder name).

If you selected the **Jobs** category and the **List jobs** action, select the following information.

Element	Description
Compartment	Select the Oracle Cloud Infrastructure compartment in which your Oracle Integration is installed.

Click **Continue**.

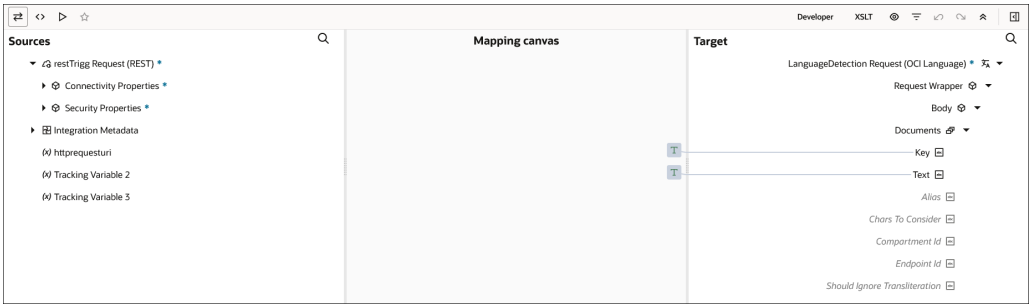
5. On the Summary page, click **Finish**.
6. Open the mapper and complete the configuration by mapping appropriate source elements to the target elements.
   a. If you selected the **Actions** category in Step 2:
   For **Language detection**, **Key phrase extraction**, **Sentiment analysis**, **Text classification**, **Text translation**, and **Named entity recognition** actions, complete the following steps.
      i. Expand the topmost node in the **Target** section.
      ii. Within that node, expand **Request Wrapper**, expand **Body**, and then expand **Documents**.
      iii. Right-click **Key**, and select **Create target node**.
      iv. Click **Design View**  in the Expression Builder.
      v. In the Expression Builder, specify a value for **Key**.

Copyright © 2022, 2026, Oracle and/or its affiliates.

> **Note**
>
> A key is an identifier that is used to distinguish the documents within a request. It should be unique within that request.

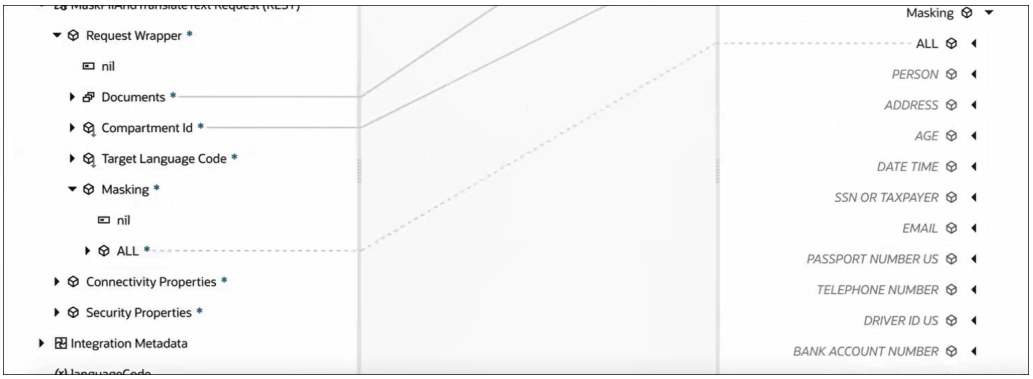
vi. In the **Target** section, within **Documents**, right-click **Text**, and select **Create target node**.

vii. In the Expression Builder, specify the text on which you want to perform the action you selected in Step 2.



If you selected the **Personal identifiable information (PII)/ Private Health Information (PHI)** action in Step 2, then map the appropriate source element to the **Documents** target element. You also need to map the appropriate source element to the **Profile** target element. Using the Expression Builder, specify values for attributes of the **Profile** element. See Profile Reference. For specific masking entities (such as BANK ACCOUNT NUMBER, AGE, EMAIL, and so on), complete the mappings for target elements such as **Mode**, **Is Unmasked From End**, **Leave Characters Unmasked**, **Masking Character**, and **Replace With** as needed. Or, to configure all the entities, map the appropriate source element to the **All** target element within **Masking** in the **Target** section, and complete one of the following mappings as needed:

- Mapping for **Mode**, **Is Unmasked From End**, and **Leave Characters Unmasked** target elements.
- Mapping for **Mode** and **Replace With** target elements.
- Mapping for **Mode** target element.



If you selected the **Health entity** action in Step 2, then map the appropriate source elements to the **Documents** and **Endpoint Id** target elements. Optionally, you can

Copyright © 2022, 2026, Oracle and/or its affiliates.

also perform mappings for the other target elements such as **Is Detect Assertions** and **Is Detect Relationships**. See BatchDetectHealthEntityDetails Reference.

> **Note**
>
> You can optionally specify **Compartment Id** in the mapper to override the value you selected initially for **Compartment** in Step 4.

b. If you selected the **Jobs** category, then complete the following steps.
If you selected the **Create job** action, then map the appropriate source elements to the **Compartment Id**, **Input Location**, **Model Metadata Details**, and **Output Location** target elements. See CreateJobDetails Reference.

If you selected the **List jobs** action, then map the appropriate source element to the **Compartment Id** target element.

If you selected the **Get job**, **Delete job**, **Cancel job**, or **Update job** action, then map the appropriate source element to the **Job Id** target element.

> **Note**
>
> - If you selected the **Get job**, **Delete job**, or **Cancel job** action, then you need to specify the job ID (on which to perform the action) as a path parameter.
> - If you selected the **Jobs** category and the **Update job** action, then you need to specify the job ID as a path parameter, and specify the updates in the request payload.

See Job Reference, CreateJob, UpdateJob, GetJob, ListJobs, DeleteJob, and CancelJob.

7. Exit the mapper.
The language action is now configured.

Example

1. Send the following text, masking configuration, and profile domain value in the request payload.
Text: User passport is A123456789, email example@example.com, age is 25 years, IFSC_CODE is I12345.

Request payload:

```
{
"compartmentId":
"ocid1.compartment.oc1..aaaaakxpkqgmdmspdcf2smlvkph7memlu3gwe5c7aj7pqozyndl
wy5a",
"documents": [
{
"key": "doc1",
"text": "User passport is A123456789, email example@example.com, age is 25
years, IFSC_CODE is I12345.",
"languageCode": "en"
}
],
```


Copyright © 2022, 2026, Oracle and/or its affiliates.

```
"masking": {
"PASSPORT_NUMBER_US": { "mode": "MASK", "maskingCharacter": "*",
"leaveCharactersUnmasked": 3, "isUnmaskedFromEnd": true },
"EMAIL": { "mode": "REMOVE" },
"AGE": { "mode": "REPLACE", "replaceWith": "30 years" }
},
"profile": {
   "domain": "ALL"
}
}
```

2. Select the **Personal identifiable information (PII)/ Private Health Information (PHI)** action.
3. Complete the mapper configuration as needed.
   The language action returns the text with passport number partially masked (showing only the last 3 characters), email removed, and age replaced with the text specified.

Ask Questions and Receive Answers Based on Current Enterprise Data with a Generative AI Agents RAG Action

Oracle Cloud Infrastructure Generative AI Agents RAG combines the power of large language models (LLMs) and retrieval-augmented generation (RAG) with enterprise data stores, enabling you to query diverse enterprise knowledge bases and receive context-specific answers based on the current data.

- Capabilities
- Prerequisites
- Invoke Oracle Cloud Infrastructure Generative AI Agents RAG from an Integration

Capabilities

OCI Generative AI Agents RAG is an AI service that provides answers to your questions or requests based on enterprise data. OCI Generative AI Agents RAG provides a way to optimize the responses from an LLM without modifying the underlying model itself; the responses are based on the latest data and can be specific to a particular organization or industry. This enables the generative AI system to provide more contextually appropriate answers to your queries as well as base those answers on the current data. It lets you query diverse enterprise knowledge bases and receive up-to-date information. Oracle Integration supports using OCI Generative AI Agents RAG in an integration with the generative AI agents RAG action.

Prerequisites

See Prerequisites for information on the prerequisites you must satisfy in the Oracle Cloud Console.

Invoke Oracle Cloud Infrastructure Generative AI Agents RAG from an Integration

1. Add a **Generative AI Agents RAG** action to an integration in either of the following ways:
   - On the side of the canvas, click **Actions** and drag the **OCI Generative AI Agents RAG** action to the appropriate location.
   - Click + at the location where you want to add the OCI Generative AI Agents RAG action, then select **OCI Generative AI Agents RAG**.

Copyright © 2022, 2026, Oracle and/or its affiliates.

2. Enter a name and optional description.
3. Click **Continue**.
4. Select the following information, then click **Continue**.

Element	Description
Region	Select the region you want to use. **Note**: Ensure you select the region in which OCI is provisioned for you.
Compartment	Select the Oracle Cloud Infrastructure compartment in which your Oracle Integration is installed. **Note**: Ensure you select a compartment where the AI agent is available.
Endpoints	This drop-down list shows the agent endpoints that are available in the selected compartment. Agent endpoints are endpoints that contain the knowledge bases that you want to query and receive responses. Select an agent endpoint to use.

> **Note**
>
> For every new chat that you want to perform, the OCI Generative AI Agents RAG action creates a session, uses the knowledge base that's available in the agent endpoint you specify, and then enables chat.

5. On the Summary page, click **Finish**.
6. Open the mapper and complete the configuration.
   a. Map the source **User Prompt** element to the target **User Message** element.
   You need to specify your question/request in the request payload body.
   b. Map the source **Session Id** element (of a previous response) to the target **Session Id** element if you want to continue the chat in the same session (session of a previous response).

   > **Note**
   >
   > If you want to continue your chat using the same session of a previous response, then you can specify the Session Id of that previous response in the request payload and perform the above mapping in the mapper. To get the Session Id of a previous response, check the information returned for the previous response as it will contain the Session Id. If you don't specify a Session Id in the request payload and don't perform the above mapping, then the OCI Generative AI Agents RAG action creates a new Session Id.

   c. Set the target **Should Stream** element to `false` in the expression builder.
7. Exit the mapper.
   The generative AI agents RAG action is now configured.

   When you activate and run the integration, the AI agent provides a response to your request/question based on the knowledge base you had specified.

Copyright © 2022, 2026, Oracle and/or its affiliates.

See Approve Invoices Using OCI AI Services for a recipe use case that uses the Generative AI Agents RAG action.

Transcribe Speech to Text and Synthesize Text to Speech with a Speech Action

Oracle Cloud Infrastructure Speech transcribes natural language speech to text and synthesizes text to speech. You can get accurate, text-normalized, and time-stamped transcriptions, and synthesized voice.

- Capabilities
- Prerequisites
- Invoke Oracle Cloud Infrastructure Speech from an Integration

Capabilities

OCI Speech is an AI service that can transcribe speech to text and synthesize text to speech.

OCI Speech harnesses the power of spoken language by enabling you to easily convert audio files containing human speech into highly exact text transcriptions. It uses automatic speech recognition (ASR) technology to provide a grammatically correct transcription. It can handle low-fidelity media recordings and transcribe challenging recordings such as meetings or call center calls.

OCI Speech lets you synthesize text to human-like speech across applications. This feature enables customer conversations, multi-language voice translations, and improved accessibility. Based on neural networks deep learning technology, this feature learns from a vast data set of human speech, capturing subtleties such as intonation, emotion, and rhythm to generate speech that closely mimics natural human expression. For example, a pause is added to the generated voice after a complete sentence followed by a period. You can use this feature in various ways, including improving accessibility for users with visual impairments, enhancing the experience in gaming, and accelerating the creation of educational content.

Oracle Integration supports using OCI Speech in an integration with the speech action.

Prerequisites

See Prerequisites for information on the prerequisites you must satisfy in the Oracle Cloud Console.

Invoke Oracle Cloud Infrastructure Speech from an Integration

1. Add a **Speech** action to an integration in either of the following ways:
   - On the side of the canvas, click **Actions** and drag the **OCI Speech** action to the appropriate location.
   - Click + at the location where you want to add the OCI Speech action, then select **OCI Speech**.
2. Enter a name and optional description.
3. Select the following information.

Copyright © 2022, 2026, Oracle and/or its affiliates.

Element	Description
Select categories	Select one of the following categories: • **Speech to text** • **Text to speech**
Action	If you selected the **Speech to text** category, select a transcription job to perform. You can create, update, or delete a transcription job. You can also get information about a transcription job or list the transcription jobs that are available in a compartment. • **Create transcription job**: If you select this action, Oracle Integration will accept a request payload containing details such as compartment ID, model, input location (of the speech to be transcribed to text), and output location to create a transcription job. Compartment ID can also be specified in the **Compartment** field when performing Step 5. • **Get transcription job**: If you select this action, Oracle Integration will accept the transcription job ID (of the transcription job) as a path parameter to retrieve the transcription job. • **List transcription jobs**: If you select this action, Oracle Integration will accept the compartment ID (that contains the transcription jobs) as a query parameter to return a list of transcription jobs available in the compartment. It can also be specified in the **Compartment** field when performing Step 5. • **Update transcription job**: If you select this action, Oracle Integration will accept the transcription job ID (of the transcription job you want to update) as a path parameter and a request payload with details such as display name, description, and so on that need to be modified. This action updates the specified transcription job with the new details you provide. • **Delete transcription job**: If you select this action, Oracle Integration will accept the transcription job ID (of the transcription job you want to delete) as a path parameter. This action deletes the specified transcription job (but it will not delete the output transcription file that is stored in the output location bucket in the object store). If you selected the **Text to speech** category, select the **Synthesize Speech** action. If you select this action, Oracle Integration will accept a request payload with details such as compartment ID, model name, text (the text to be synthesized to speech), output format, and so on. It returns a stream reference that you can download. See Using Text to Speech.

4. Click **Continue**.
5. Select the following information, then click **Continue**.

Element	Description
Compartment	This field is available only when you select **Create transcription job**, **List transcription jobs**, or **Synthesize Speech** action in Step 3. Select the Oracle Cloud Infrastructure compartment in which your Oracle Integration is installed.
Output bucket	This field is available only when you select **Create transcription job** action in Step 3. Select a bucket to store the text output generated by the OCI Speech action.

Copyright © 2022, 2026, Oracle and/or its affiliates.

Element	Description
Speaker	This field is available only when you select **Synthesize Speech** action in Step 3. Select a speaker (predefined voice) from the drop-down list.

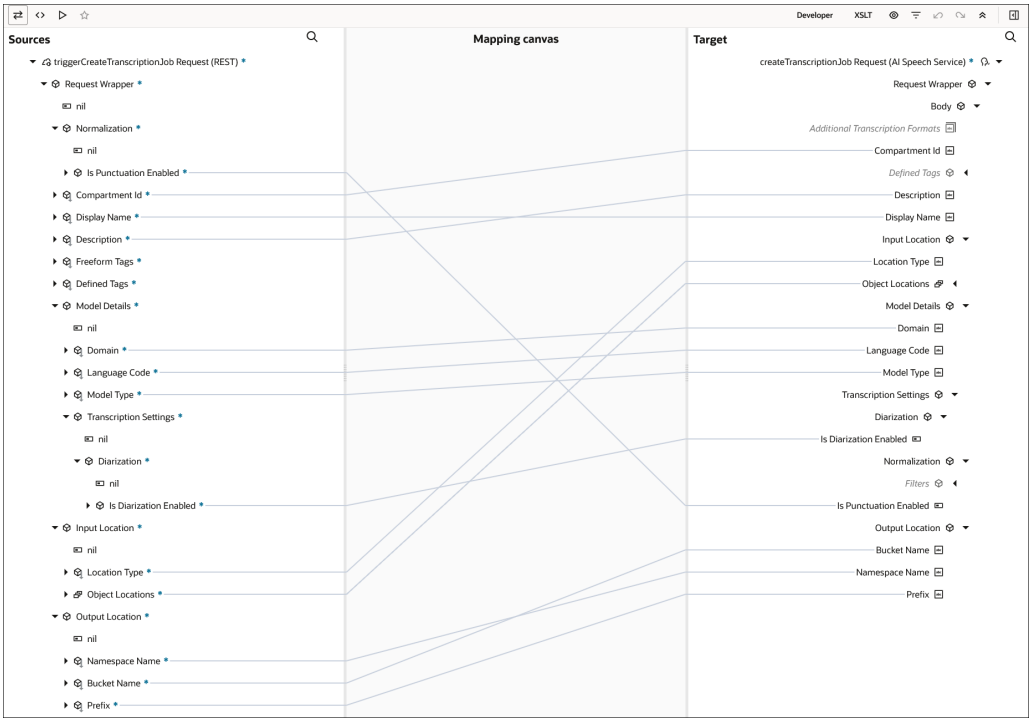
6. On the Summary page, click **Finish**.
7. Open the mapper and define the mappings between the source and target elements as needed for the action you selected in Step 3.

> **Note**
>
> You can optionally specify **Compartment Id** and **Bucket Name** in the mapper to override the value you selected initially for **Compartment** and **Output Bucket** respectively (in Step 5).

a. Perform the following source-to-target mappings for the **Create transcription job** action:

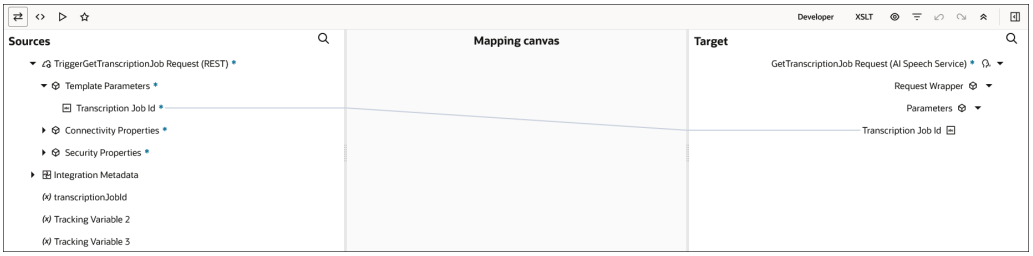
- Map the source **Is Punctuation Enabled** to the target **Is Punctuation Enabled**.
- Map the source **Compartment Id** to the target **Compartment Id**.
- Map the source **Display Name** to the target **Display Name**.
- Map the source **Description** to the target **Description**.
- Map the source **Domain** to the target **Domain**.
- Map the source **Language Code** to the target **Language Code**.
- Map the source **Model Type** to the target **Model Type**.
- Map the source **Is Diarization Enabled** to the target **Is Diarization Enabled**.
- Map the source **Location Type** to the target **Location Type**.
- Map the source **Object Locations** to the target **Object Locations**.
- Map the source **Namespace Name** to the target **Namespace Name**.
- Map the source **Bucket Name** to the target **Bucket Name**.
- Map the source **Prefix** to the target **Prefix**.

Copyright © 2022, 2026, Oracle and/or its affiliates.



b. Perform the following source-to-target mapping for the **Get transcription job** action:

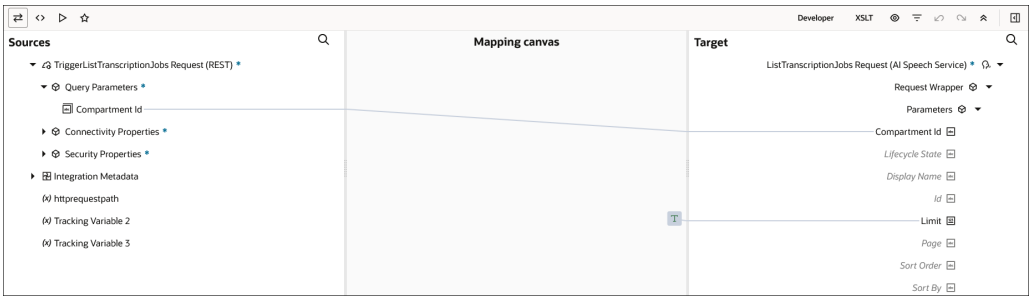
- Map the source **Transcription Job Id** to the target **Transcription Job Id**.



c. Perform the following source-to-target mapping for the **List transcription jobs** action:

- Map the source **Compartment Id** to the target **Compartment Id**.

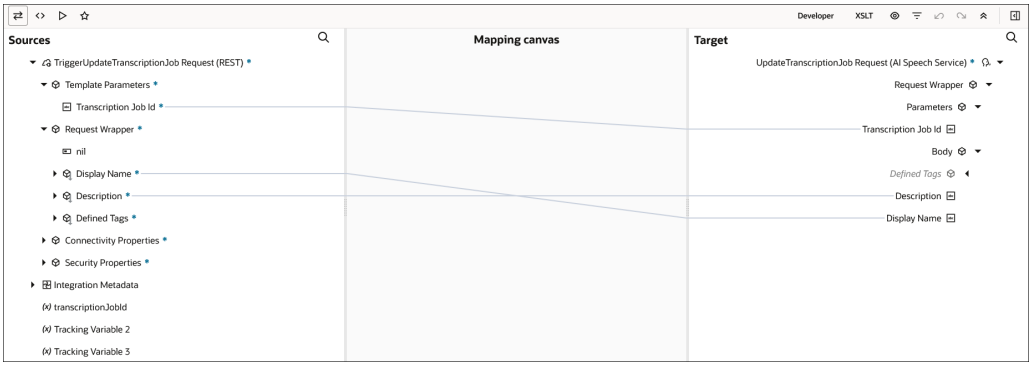
You can optionally configure target elements such as **Lifecycle State**, **Display Name**, **Id**, **Limit**, **Page**, **Sort Order**, and **Sort By**.



Copyright © 2022, 2026, Oracle and/or its affiliates.

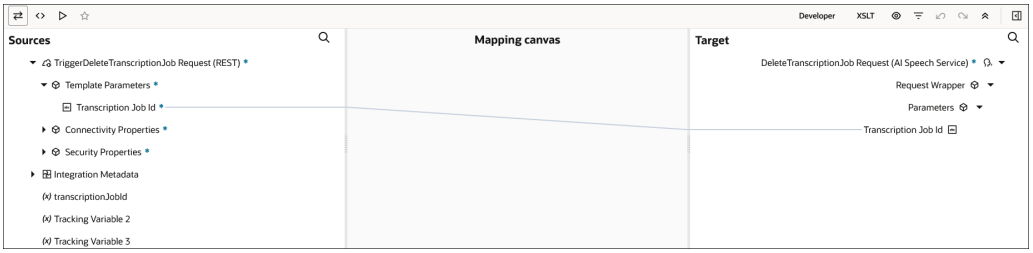
d. Perform the following source-to-target mappings for the **Update transcription job** action:

- Map the source **Transcription Job Id** to the target **Transcription Job Id**.
- Map the source **Display Name** to the target **Display Name**.
- Map the source **Description** to the target **Description**.



e. Perform the following source-to-target mapping for the **Delete transcription job** action:

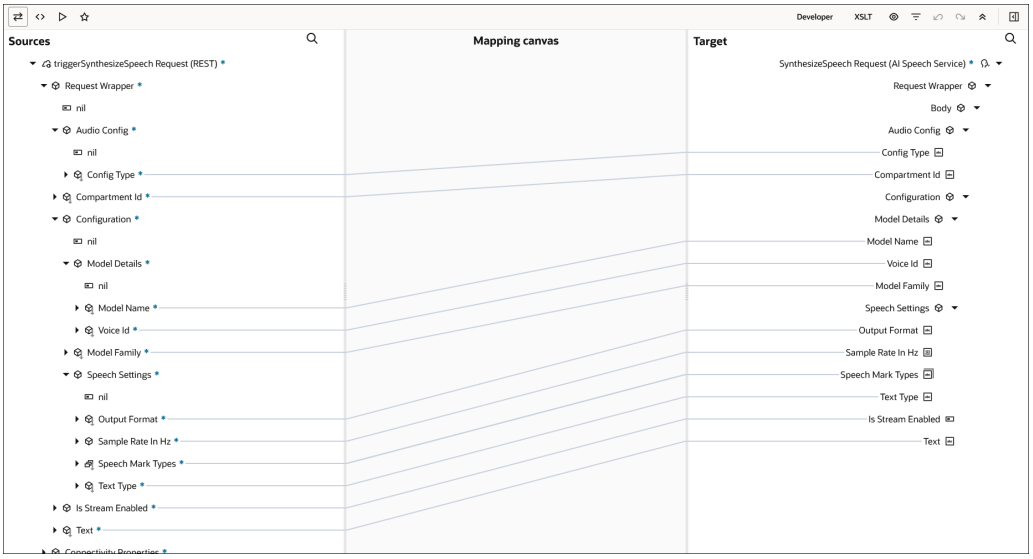
- Map the source **Transcription Job Id** to the target **Transcription Job Id**.



f. Perform the following source-to-target mappings for the **Synthesize Speech** action:

- Map the source **Is Punctuation Enabled** to the target **Is Punctuation Enabled**.
- Map the source **Config Type** to the target **Config Type**.
- Map the source **Compartment Id** to the target **Compartment Id**.
- Map the source **Model Name** to the target **Model Name**.
- Map the source **Voice Id** to the target **Voice Id**.
- Map the source **Model Family** to the target **Model Family**.
- Map the source **Output Format** to the target **Output Format**.
- Map the source **Sample Rate In Hz** to the target **Sample Rate In Hz**.
- Map the source **Speech Mark Types** to the target **Speech Mark Types**.
- Map the source **Text Type** to the target **Text Type**.
- Map the source **Is Stream Enabled** to the target **Is Stream Enabled**.
- Map the source **Text** to the target **Text**.

Copyright © 2022, 2026, Oracle and/or its affiliates.



8. Exit the mapper.
The speech action is now configured.

   Here's what happens when you activate and run the integration based on the action you selected in Step 3:

   - **Create transcription job**: Converts the speech you provided to text and the text output is stored in the output bucket you selected.
   - **Get transcription job**: Retrieves the transcription job with the specified ID.
   - **List transcription jobs**: Returns a list of transcription jobs available in the specified compartment.
   - **Update transcription job**: Updates the specified transcription job with the new details you provided.
   - **Delete transcription job**: Deletes the specified transcription job (but it will not delete the output transcription file that is stored in the output location bucket in the object store).
   - **Synthesize Speech**: Converts the text you provided to speech. You can download the output file from the response.

Logic Category

The Logic category enables you to add for-each, parallel, scope, switch, and while actions to an integration.

Topics:

- Loop over Repeating Elements with a For-Each Action
- Process Tasks in Parallel with a Parallel Action
- Manage a Group of Actions and Fault Handlers with a Scope Action
- Route Expressions with Switch Action Branches
- Loop Over Actions or Invoke Connections While a Condition is Satisfied with a While Action

Copyright © 2022, 2026, Oracle and/or its affiliates.

Loop over Repeating Elements with a For-Each Action

The for-each action enables you to loop over a repeating element and process 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 download 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**
>
> - You *cannot* use an unqualified repeating element in a for-each action.
> - When you configure a stop action inside a for-each action, the entire integration is terminated when the for-each action is processed for the first time. The for-each action is not allowed to process more than once. The stop action does not display any message describing this behavior during design time.

Creating a For-Each Action

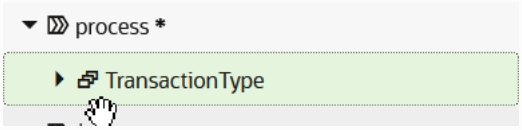
1. Add a for-each action to an integration in either of the following ways:

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

   The Configure For Each panel opens. This includes a **Sources** tree of elements.

2. Click **Edit** under **Configure For Each** to enter a name and optional description for the for-each action.
3. Expand the **Sources** tree to select an element.
4. Drag a repeatable element to the **Repeating Element** field. This is the element over which to loop.

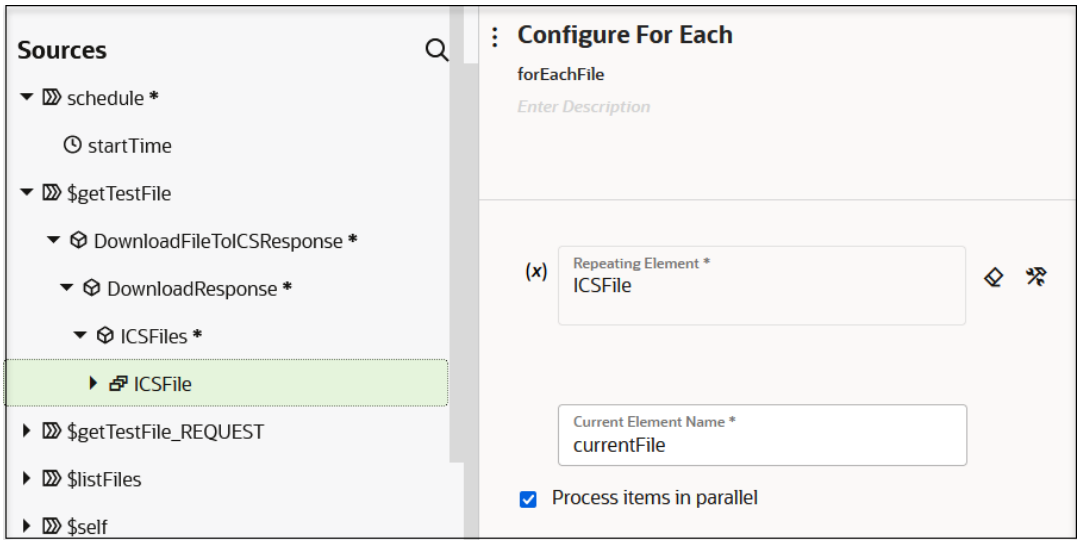
Copyright © 2022, 2026, Oracle and/or its affiliates.

> **Note**
>
> Note the following restrictions:
>
> - The selected element must be repetitive. You can identify repetitive elements by the three-box 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.

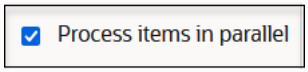
For this example, the element over which to loop is **ICSFile**.

For every iteration of the loop, there is a single reference to the repeating element (**ICSFile**). A current element name file is required for this action to occur.

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



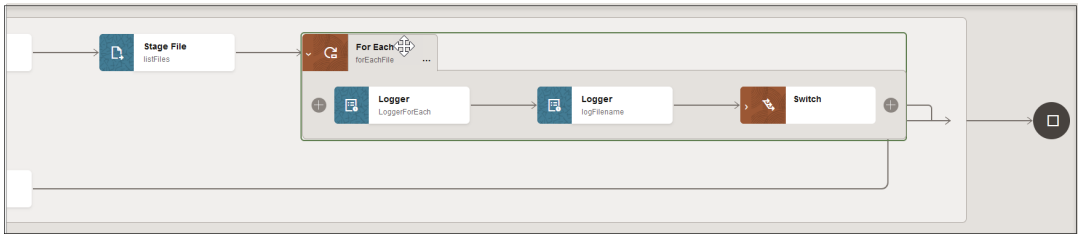
Copyright © 2022, 2026, Oracle and/or its affiliates.

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

6. Click **Save**.

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

7. Click **+** inside the for-each action to add additional actions. These actions define what happens during each iteration of the loop. For this example, scope and switch actions are defined within the for-each action.



Using Global Variables in For-Each Loops

When global variables are used in for-each loops (for example, for logging purposes), they can lead to the following runtime exception when bad or invalid data is associated with the variable:

```
FORG0001: invalid value for cast/constructor, Cikey=13695578,
FlowId=10025204, Current Activity Key=13695578-BpAss43-BpSeq8.240-4, Current
Activity Label=assignment_ics_api_internal_variable23,
```

As a recommendation:

- Limit the use of global variables in for-each loops.
- Global variable handling requires database access, which results in an added performance overhead.
- Local variables can be used for logic specific to a local scope.

Tracking the Status of a For-Each Action During Runtime

You can track the status of the for-each action on the Instances page through the tracking diagram and activity stream for an activated integration.

Copyright © 2022, 2026, Oracle and/or its affiliates.

1. In the navigation pane, click **Observability**, then **Instances**.
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. The activity stream is displayed on the right side.

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

- Use Bulk Response Operations in an Integration in *Using the Salesforce Adapter with Oracle Integration 3*
- Process Large Data Sets Asynchronously with Different Bulk Import Operations in *Using the Marketo Adapter with Oracle Integration 3*
- Subscribe to Atom Feeds in a Schedule Integration in *Using the Oracle HCM Cloud Adapter with Oracle Integration 3*
- Invoke an Endpoint Dynamically in *Using the Oracle ERP Cloud Adapter with Oracle Integration 3*
- Create an Integration to Import and Process Bulk Files in *Using the FTP Adapter with Oracle Integration 3*

Route Expressions with Switch Action Branches

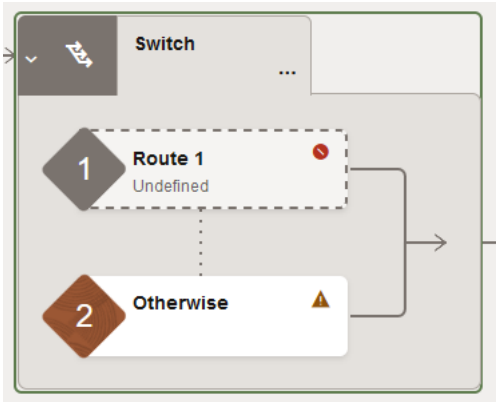
You can define switch action branches to add routing expressions in your integration. The switch action takes the first branch that evaluates to true. The other branches are ignored. A switch is essentially single-threaded.

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

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

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

Two branches are automatically created:



Copyright © 2022, 2026, Oracle and/or its affiliates.

- **Route 1** (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.

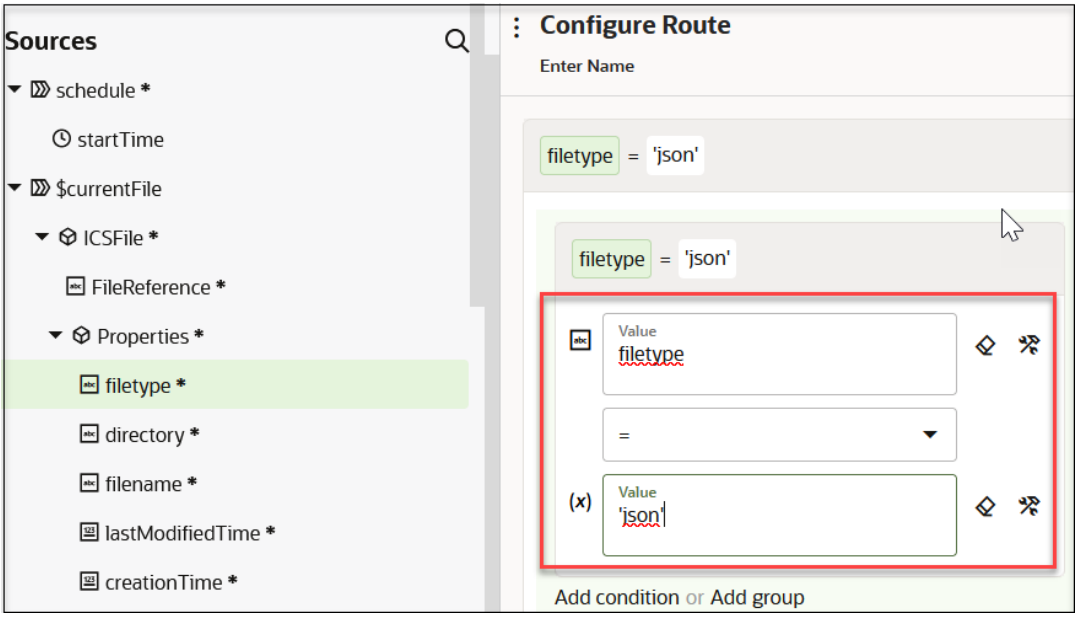
> **Note**
>
> To add more branches, click **Actions** ⋯ in the switch action, then **Add**.

2. From the **Route 1** branch icon, double-click the branch or click **Actions** ⋯, then **Edit**.

   The Configure Route panel opens. This includes a **Sources** tree of elements.

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

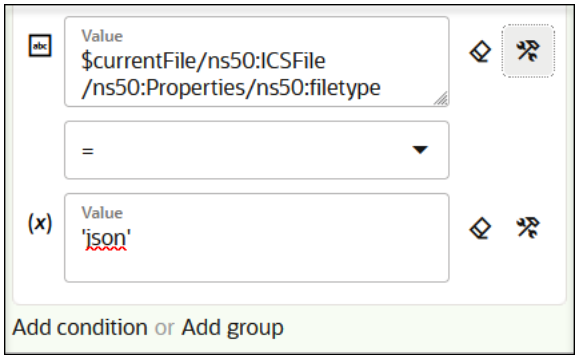
3. Click **Edit**  under **Configure Route** to enter a name for the Route 1 branch.
4. Define a routing expression. For this example:
   - **filetype** is dragged to the upper **Value** field
   - **=** is selected as the operator
   - `'json'` is manually entered in the lower **Value** field. You can also drag a value from the **Sources** tree or select a value from the drop-down list.



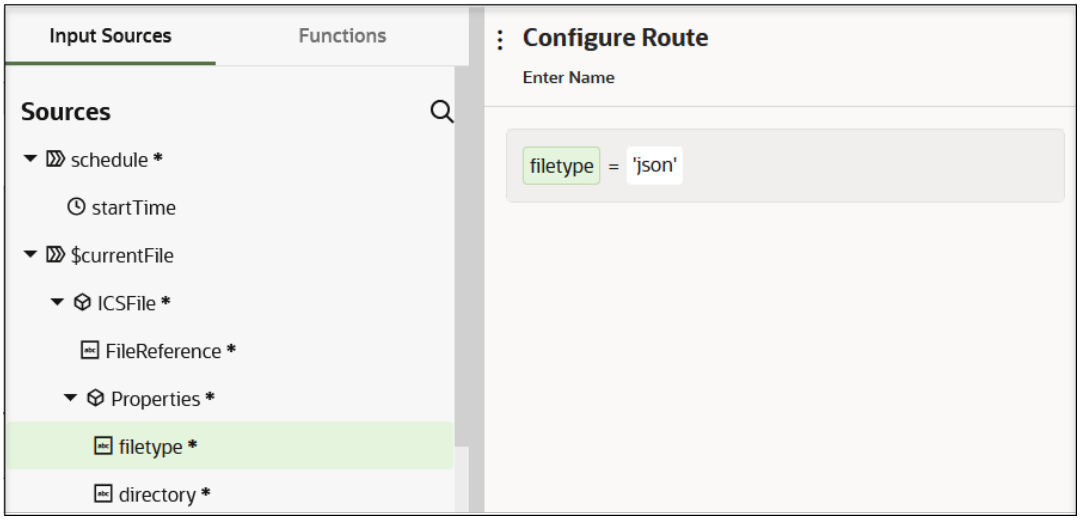
You can add functions to your expression under the **Functions** tab. XPath version 2.0 functions are supported. Base 64 encode and decode functions that process data confidentially across layers and functions that return boolean results are also supported. Functions that return nonboolean values are not supported.

5. Click **Switch to Developer Mode**  to view the complete path name.

Copyright © 2022, 2026, Oracle and/or its affiliates.



6. Click **Add condition** to add more conditions or **Add group** to add group conditions, when needed.
7. Click **Save**.
   For this example, the following expression is defined:



8. Drag other actions and invokes into the branches of the switch action, when required by your integration. For this example, two switch action branches are defined.



Related Documentation

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

- Invoke an Endpoint Dynamically in *Using the Oracle ERP Cloud Adapter with Oracle Integration 3*
- Design an Asynchronous Search Operation with Pagination in *Using the Oracle NetSuite Adapter with Oracle Integration 3*

Copyright © 2022, 2026, Oracle and/or its affiliates.

- Use the Extract API in a Schedule Integration in *Using the SAP Concur Adapter with Oracle Integration 3*
- Use the Pagination Concept in an Integration in *Using the Jira Adapter with Oracle Integration 3*

Process Tasks in Parallel with a Parallel Action

You can use a parallel action to process tasks in parallel to improve integration performance and response times. A parallel action allows the path of an integration to be split into multiple branches. Each branch is processed in parallel due to their independence from each other. Messages are sent to each service endpoint in parallel.

When all tasks are completed, all branches are automatically synchronized and merged at their termination points in the parallel action and the main path of the integration is resumed. Any variables created inside the parallel branches (for example, invoke responses) are available for use in the main path.

- Use Cases
- Guidelines and Restrictions
- Design Time Behavior
- Runtime Behavior

Want to see a video demo of the parallel action?

Video

Use Cases

A parallel action is useful for use cases such as the following:

- An order processing integration contains individual tasks for checking inventory, customer credit, and legal constraints. These tasks are processed in parallel to ensure a better response time.
- An integration obtains the lowest price for an item from suppliers. The integration checks three suppliers for prices and takes the best offer. The three price checks are done in parallel because they are independent of each other. Once the three suppliers have responded, the branches are terminated. The three prices are compared to identify the lowest. The integration then continues its main path by placing the order with that supplier.
- An integration processes an incoming order from both an inventory and legal perspective (for example, can the product be shipped to the customer's country). Both checks are done in parallel.

Guidelines and Restrictions

- You can create a maximum of five branches to a parallel action. If you attempt to create a sixth branch by selecting **Actions** ⋮, and then **Add** on the parallel action, the selection is disabled.
- There is no limit to the number of parallel actions you can include in an integration.
- Synchronous and asynchronous integrations are supported.
- You cannot add a parallel action inside another parallel action.
- Parallel actions can be used in application integrations and schedule integrations.
- Only map, scope, switch, logger, JavaScript, B2B, and notification actions and invoke connections are supported in a parallel action branch.

Copyright © 2022, 2026, Oracle and/or its affiliates.

- You can add parallel actions to the branches of a pick action.
- You can add parallel actions inside scope and switch actions.
- Nesting of parallel actions in for-each actions, while actions, or stage file actions with chunking is not supported.
- No independent branch-specific fault handling is supported. You can catch faults at the integration level with a global fault handler.
- Synchronization of global variables is not supported.
- Branch-specific timeouts are not supported.

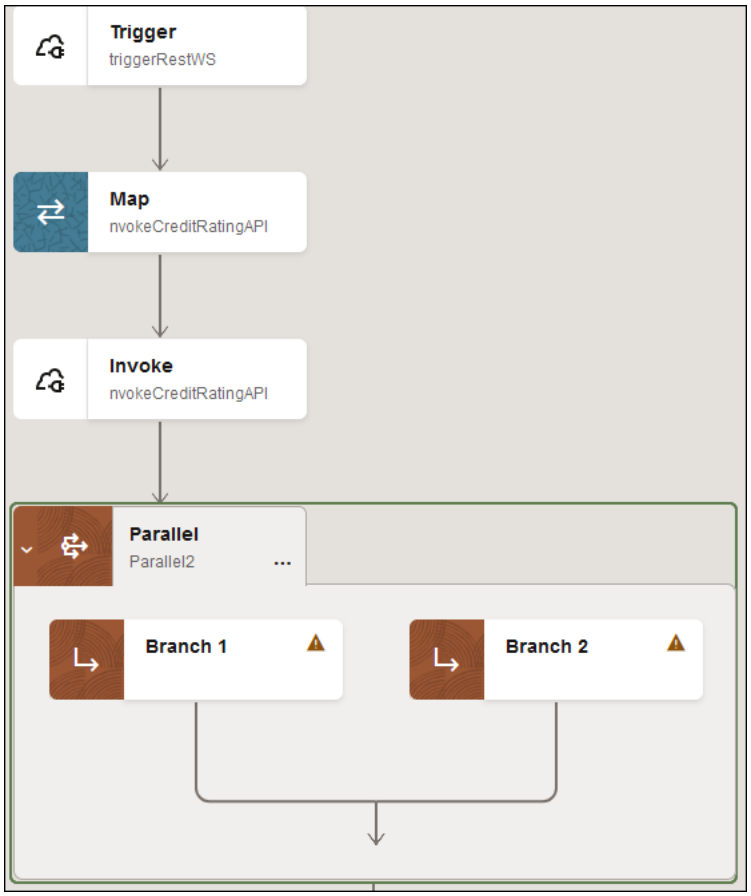
Design Time Behavior

The following example provides a high-level overview of how to design a specific parallel action. This is just one example of parallel action use.

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

   - On the side of the canvas, click **Actions** and drag the **Parallel** action to the appropriate location.
   - Click + at the location where you want to add the parallel action, then select **Parallel**.

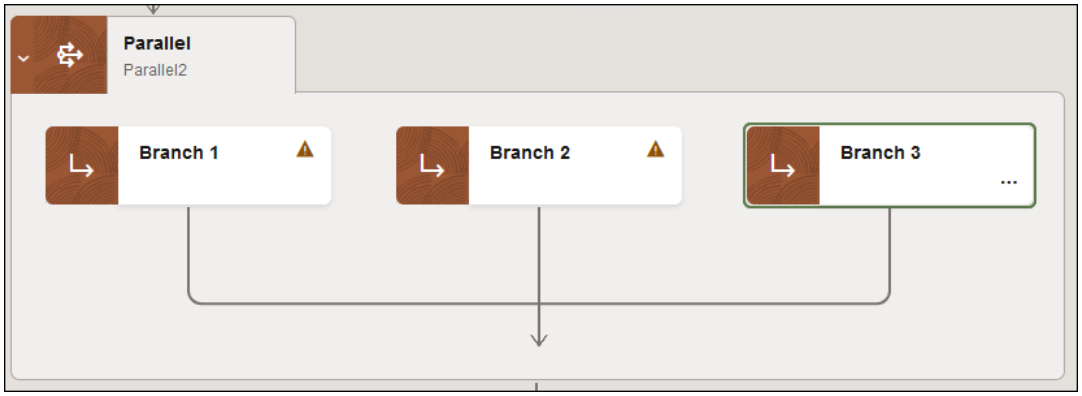
   A parallel action with two branches is created.



Copyright © 2022, 2026, Oracle and/or its affiliates.

2. Select **Actions** ⋮ to perform the following tasks.
   - **Edit**: Change the name of the parallel action.
   - **Add**: Add more branches. You can add a maximum of five branches.
   - **Delete**: Delete the parallel action.
   - **Collapse/Expand**: Collapse or expand the parallel action.
   - **Cut**: Cut the parallel action and paste it elsewhere in the integration. You can also cut and paste existing maps/invokes into the new parallel branches to make them process in parallel. You *cannot* cut an individual branch of a parallel action and paste it into another parallel action.
     See Cut and Paste Elements in the Integration Canvas.

   For this example, a third branch is added.

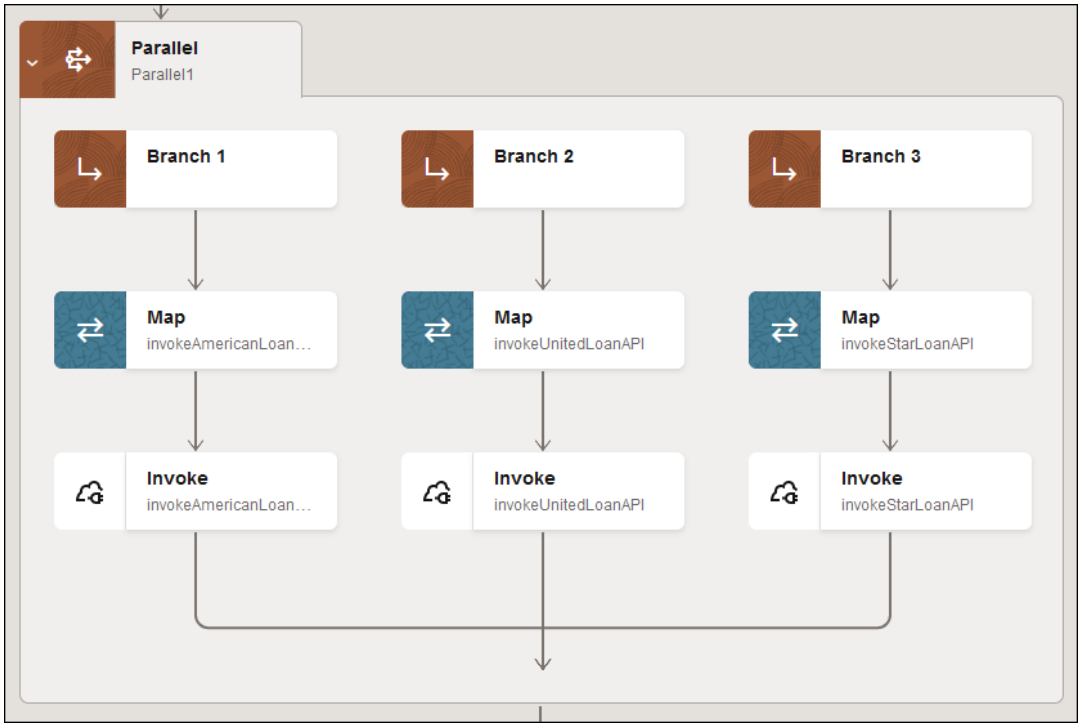


   For this example, the three branches are designed to obtain loan offers from three service endpoints and then select the best one.

3. Add and configure three REST Adapter connections to invoke the following service endpoints. At runtime, all three loan service endpoints are invoked in parallel to identify their specific loan offers.
   - American Loan
   - United Loan
   - Star Loan

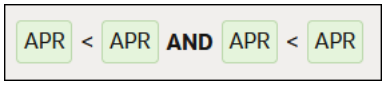
4. In the mapper for each invoke action, map your SSN to the loan application for that service endpoint.

Copyright © 2022, 2026, Oracle and/or its affiliates.

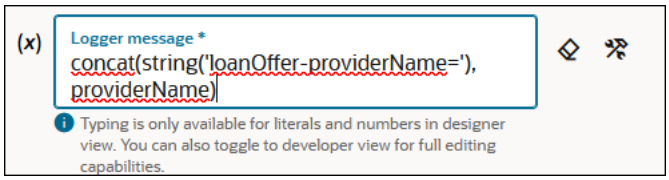


At the bottom, the three branches are synchronized at their termination points and the parallel action rejoins the main branch of the integration. The main integration does not continue until all branches are complete

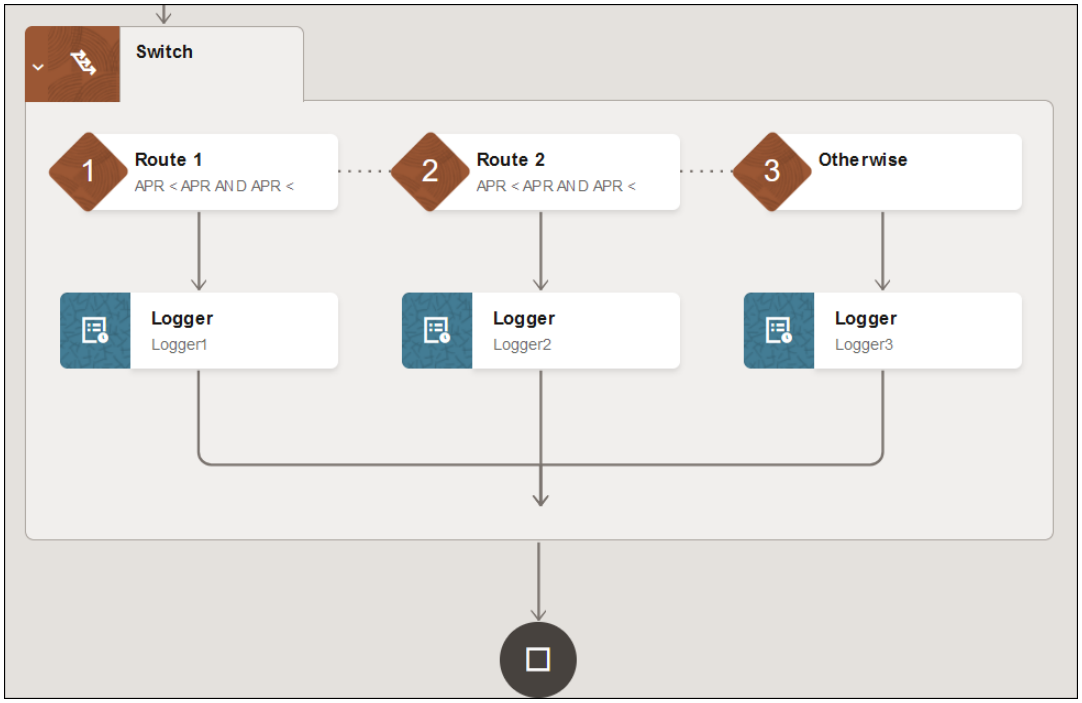
5. Add a switch action to define the routing rules to check for the best offer returned from the loan service endpoints.



6. Define the message in a log action for each switch branch.

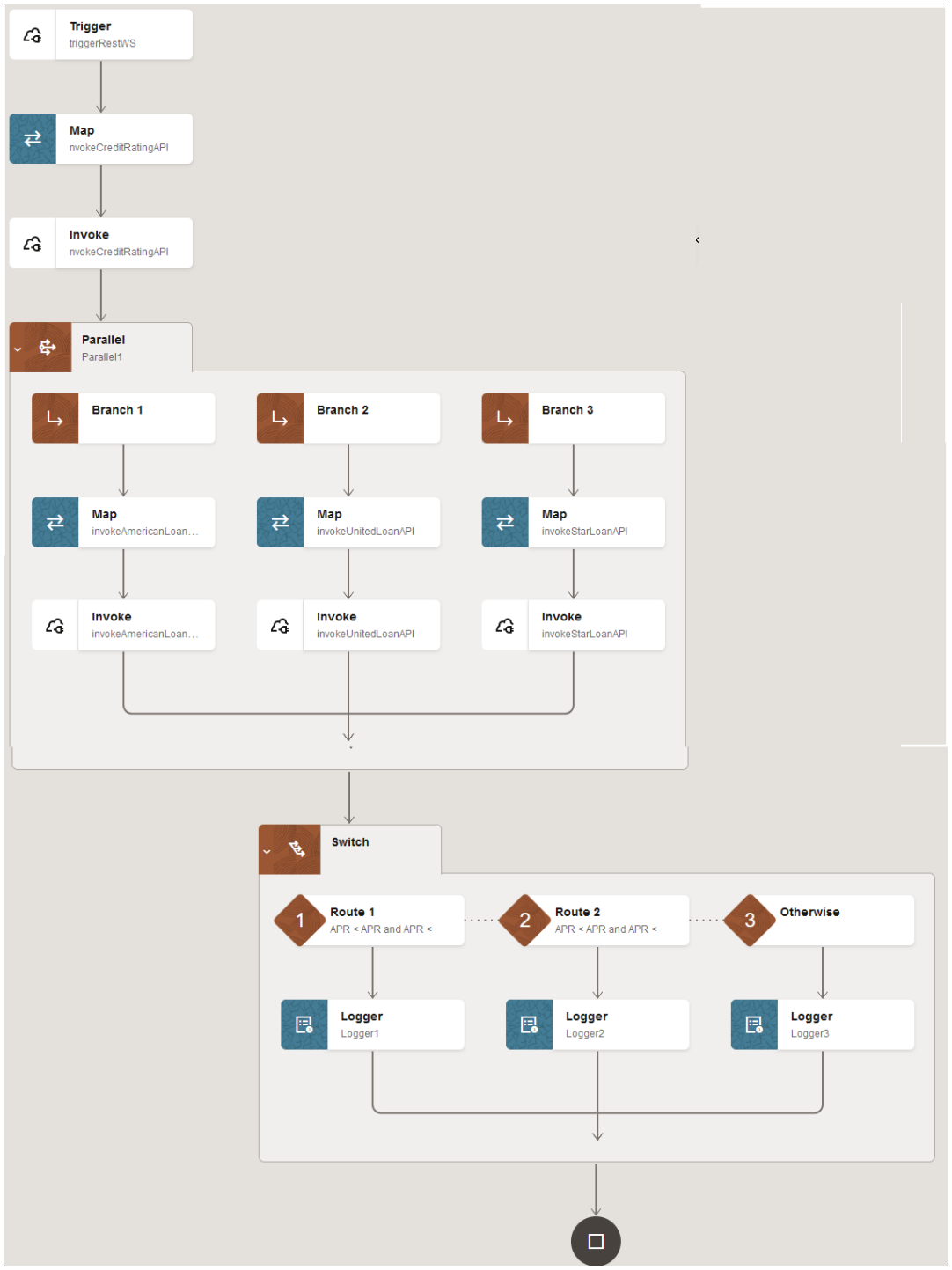


Copyright © 2022, 2026, Oracle and/or its affiliates.



When complete, the integration looks as follows.

Copyright © 2022, 2026, Oracle and/or its affiliates.



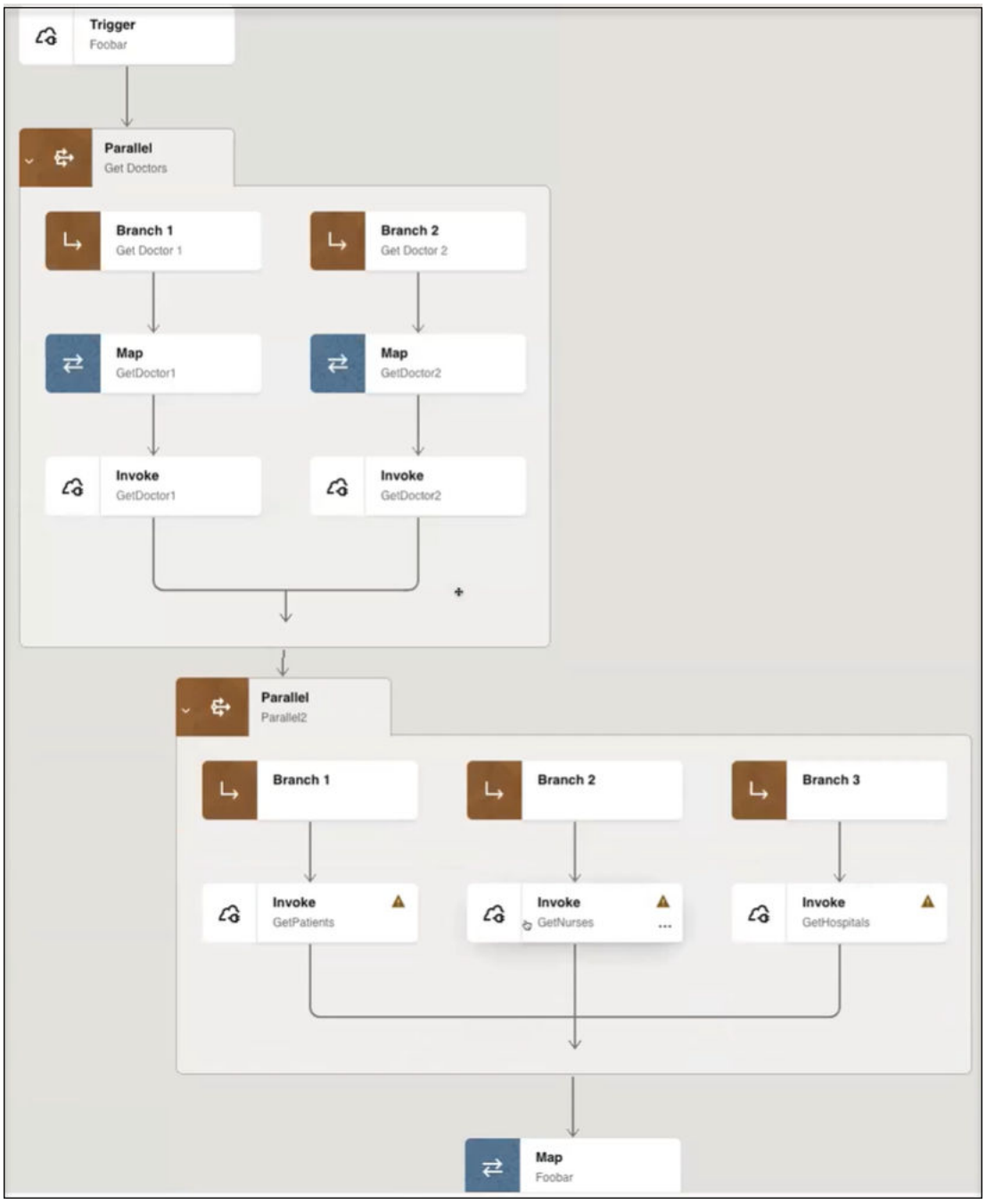
The integration is now ready to be activated and invoked.

Runtime Behavior

You can view the status of parallel branches in the activity stream during runtime. For example, assume your integration is designed with the following:

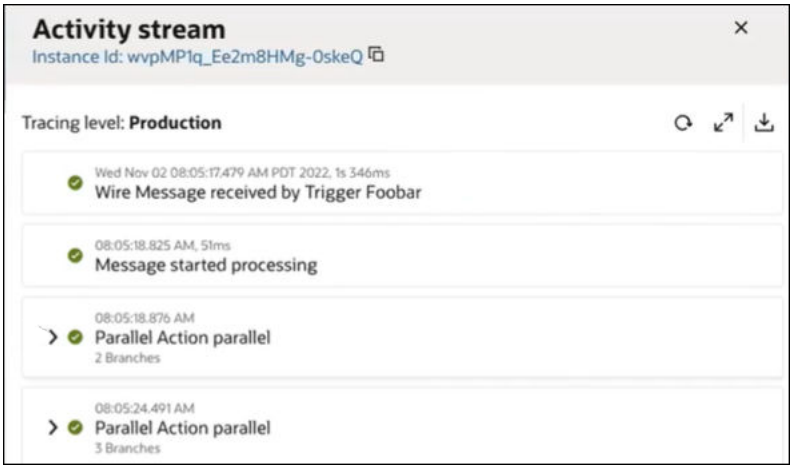
- One parallel action with two branches
- A second parallel action with three branches

Copyright © 2022, 2026, Oracle and/or its affiliates.



1. Invoke an instance of your integration and go to the Instances page.
   During runtime, the activity stream shows both parallel actions and their number of branches. For this example, both parallel actions processed successfully.

Copyright © 2022, 2026, Oracle and/or its affiliates.



2. Expand each branch to view more specific details.

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.

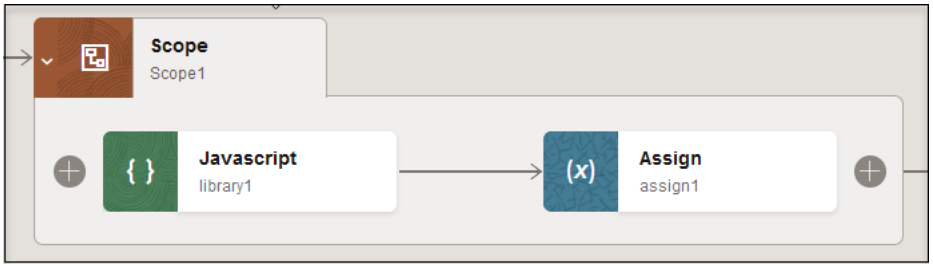
> **Note**
>
> Scopes can have fault handlers in which specific faults can be caught and rethrown. However, in the case of connectivity agent-based invokes, the named fault handlers are not processed. All fault handling must be done in the default fault handler.

- Creating a Scope Action
- Add Nested Scopes to a Scope Action
- Tracking the Status of a Scope Action During Runtime
- Rename a Scope

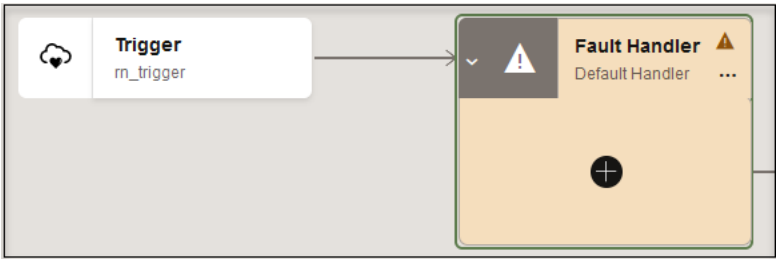
Create a Scope Action

1. Add a scope action to an integration in either of the following ways:
   - On the right side of the canvas, click **Actions** and drag the **Scope** action to the appropriate location.
   - Click + at the location where you want to add the scope action, then select **Scope**.
2. Click **Edit** under **Edit Scope** to enter a name and optional description for the scope action.
3. Drag and design appropriate actions, mappings, and invokes into the scope action.

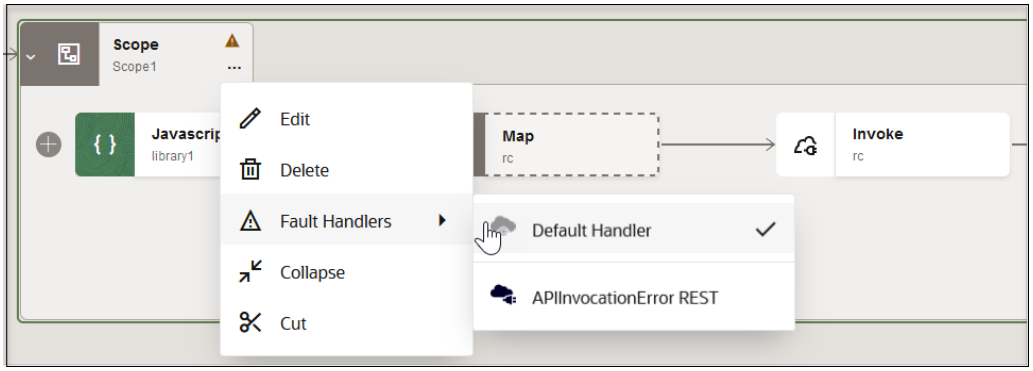
Copyright © 2022, 2026, Oracle and/or its affiliates.



The scope includes a default fault handler that is accessible by clicking the scope action and selecting **Actions** ⋯, then **Fault Handlers**, then **Default Handler**.



4. Click **+** to define actions on the default fault handler.
5. To return to the scope action, click the fault handler, and select **Actions** ⋯, then **Show Scope Block 'scope_name'**.
   When you add invokes to a scope, the named faults associated with the invokes are added to the default fault handler in the scope. For example:
   a. Add and configure an invoke connection inside the scope action (for example, a REST Adapter).
   b. Click the scope and select **Actions** ⋯, then **Fault Handlers**.
      In addition to the **Default Handler**, an invoke fault handler has been added.



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.

6. 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

Copyright © 2022, 2026, Oracle and/or its affiliates.

invoke throws a fault, but there is not a specific catch named for it. See Catch Faults with a Re-throw Fault Action.

7. Collapse the scope action by double-clicking it. You can expand it again by double-clicking it, selecting **Actions**, then **Expand**, or clicking **>**.

> **Note**
>
> For application 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.

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, when necessary.

Track the Status of a Scope Action During Runtime

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

1. In the navigation pane, click **Observability**, then **Instances**.
2. Click the business identifier value of the integration to track.

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

   - Scope processing succeeds and is displayed in green. Because the fault handler is not processed, the **Fault Handler** section of the scope remains hidden.
   - Scope processing fails and is displayed in red, but the fault handler succeeds in handling the fault and is displayed in green. Processing continues after the scope.

Copyright © 2022, 2026, Oracle and/or its affiliates.

Because the fault handler was processed, 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.

See Track the Status of Integration Instances for additional details.

Rename a Scope

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

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 application integrations.

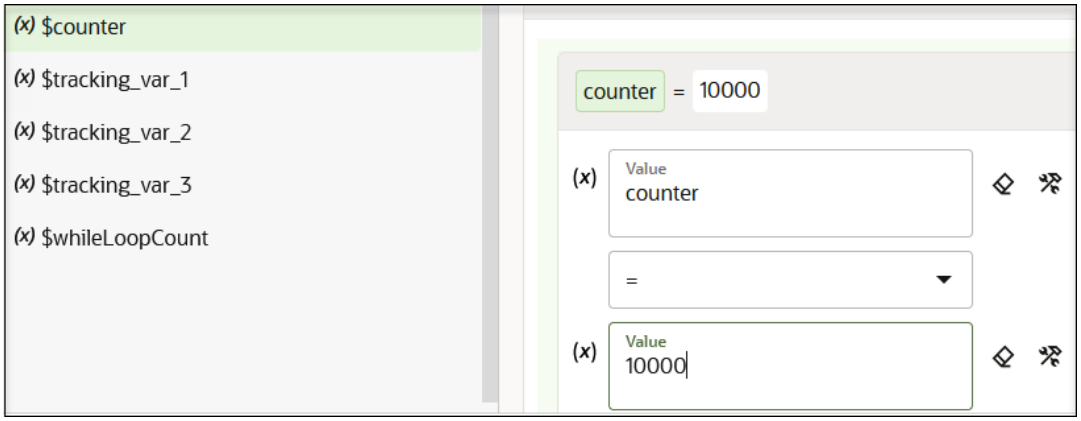
Creating a While Action

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

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

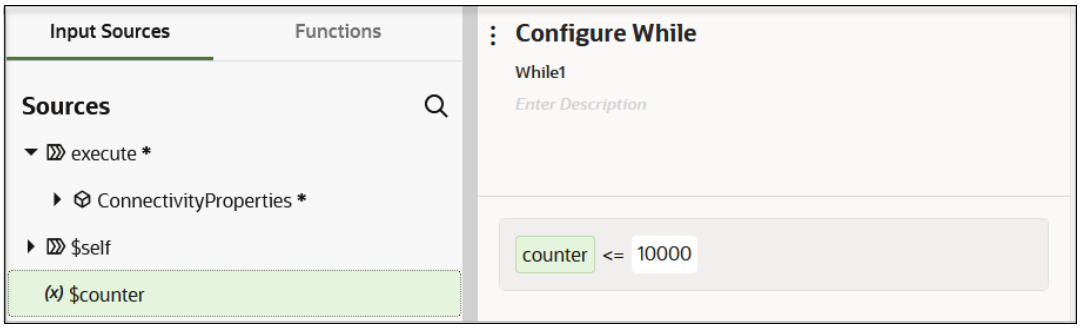
   The Configure While panel opens. This includes a **Sources** tree of elements.
2. Click **Edit**  under **Configure While** to enter a name and optional description for the while action.
3. Manually enter, select from the drop-down list, or drag an element from the **Sources** tree to the **Value** field. You can also add functions.
4. Select an operator.
5. Enter a value in the **Value** field.

Copyright © 2022, 2026, Oracle and/or its affiliates.

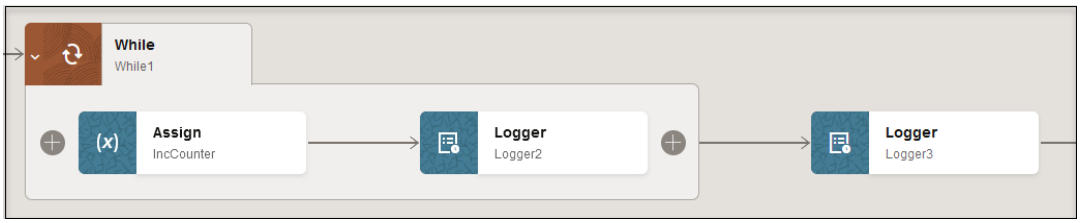


6. Click **Save**.

   For this example, while the counter is less than or equal to 10000, the integration loops over the condition. When the condition is not met, the looping ends.



7. Drag invoke connections or other actions for configuration to the **+** sign that is displayed inside the while action. These invoke connections and actions are processed as long as the condition set in the while action is met.



8. If you want to edit the name or expression, double-click the **While** icon or select **Actions** ⋯, then **Edit**.

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

Copyright © 2022, 2026, Oracle and/or its affiliates.

Track the Status of a While Action During Runtime

During runtime, you can track the status of the while action in the Instances page through the tracking diagram and activity stream for an activated integration.

1. In the navigation pane, click **Observability**, then **Instances**.
2. Click the business identifier value of the integration to track.

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

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

- Retrieve the Name and IDs from Account Records with the queryMore Operation in *Using the Salesforce Adapter with Oracle Integration 3*
- Retrieve Issue Details Using the Jira Query Language (JQL) in *Using the Jira Adapter with Oracle Integration 3*
- Use Pagination in an Integration in *Using the Shopify Adapter with Oracle Integration 3*

Call Category

The Call category enables you to add decision, file server, JavaScript, OCI function action, OCI object storage, process, public event, and robot flow actions to an integration.

Topics:

- Add a Decision Model to an Integration with a Decision Action
- Interact with Files in File Server
- Invoke A Child Integration with an Integration Action
- Add JavaScript Functions to an Integration with a JavaScript Action
- Invoke Oracle Cloud Infrastructure Functions Directly from an Integration with an OCI Function Action
- Invoke Oracle Cloud Infrastructure Object Storage from an Integration with an OCI Object Storage Action
- Invoke a Process from an Integration with a Process Action
- Publish Events in an Integration with a Publish Event Action
- Call a Robot from an Integration

Add a Decision Model to an Integration with a Decision Action

You can select to include an activated decision model in your integration by adding a decision action at the appropriate point.

All information about adding a decision action is covered in a separate guide. See Add a Decision Model to an Integration in *Using Decisions in Oracle Integration 3*.

Copyright © 2022, 2026, Oracle and/or its affiliates.

Interact with Files in File Server

You can obtain a list of files, write or move a file to File Server, get the reference of a file and then stage read or unzip the file, and delete a file on File Server using the File server action.

On This Page

- What You Can Do with the File Server Action
- When Not to Use the File Server Action
- Start Here: Add the File Server Action to an Integration
- Option 1. Obtain a List of Files in File Server
- Option 2. Write a File to File Server
- Option 3. Move a File to File Server
- Option 4. Stage a File and Work with It, or Write a File to Another Location in File Server
- Option 5. Delete a File in File Server
- Next Steps

What You Can Do with the File Server Action

The File server action lets you interact with files in the following ways.

For all of the options, you can hard code values, but most of the time, you'll pass dynamic values into the File server action using the Map action. Complete that task after you finish defining the File server action. See Next Steps.

Goal	Operation to use	Step-by-step instructions
Obtain a list of files in a File Server directory	**List directory**	1. Start Here: Add the File Server Action to an Integration 2. Option 1. Obtain a List of Files in File Server
Create a new file in a File Server directory	**Write File**	1. Start Here: Add the File Server Action to an Integration 2. Option 2. Write a File to File Server
Move a file from one File Server directory to another (similar to a cut and paste operation)	**Move File**	1. Start Here: Add the File Server Action to an Integration 2. Option 3. Move a File to File Server

Copyright © 2022, 2026, Oracle and/or its affiliates.

Goal	Operation to use	Step-by-step instructions
Get a reference to a file in a File Server director, and use the reference to complete either of the following tasks: • Stage the file and then work with the staged file as needed in Oracle Integration • Write the file to another File Server directory	**Get File Reference**	1. Start Here: Add the File Server Action to an Integration 2. Option 4. Stage a File and Work with It, or Write a File to Another Location in File Server
Encrypt or decrypt a file	• For files that must be encrypted and saved to File Server: **Get File Reference**, which gets a reference to a file. After the File server action, add a **stage file action** to encrypt the file, and then another **File server action** using the **Write File** operation. The File server action uploads the file to File Server by passing the encrypted file reference. • For encrypted files in File Server that must be decrypted: **Get File Reference**, which gets a reference to an encrypted file, followed by a **stage file action** to decrypt the file. You can also use the capabilities of the FTP Adapter to encrypt or decrypt a file.	1. Start Here: Add the File Server Action to an Integration 2. Option 4. Stage a File and Work with It, or Write a File to Another Location in File Server 3. Process Files in Schedule Integrations with a Stage File Action 4. Option 2. Write a File to File Server Alternatively, to use the FTP Adapter to encrypt or decrypt a file, see Create a Connection and Add the FTP Adapter Connection to an Integration in *Using the FTP Adapter with Oracle Integration 3*.

Copyright © 2022, 2026, Oracle and/or its affiliates.

Goal	Operation to use	Step-by-step instructions
Sign or verify a file	You can perform signing using the private key to allow the receiver to verify that the file contents were not altered during transit. You can also perform signature verification using the public key on the incoming file to verify that the contents were not altered during transit. • For files that must be signed or verified and then saved to File Server: **Get File Reference**, which gets a reference to a file. After the File server action, add a **stage file action** to perform file signing or verification, and then another **File server action** using the **Write File** operation. The File server action uploads the file to File Server. • Alternatively, you can use the capabilities of the FTP Adapter to perform file signing or verification.	1. Start Here: Add the File Server Action to an Integration 2. Option 4. Stage a File and Work with It, or Write a File to Another Location in File Server 3. Process Files in Schedule Integrations with a Stage File Action 4. Option 2. Write a File to File Server Alternatively, to use the FTP Adapter to perform file signing or verification, see Create a Connection and Add the FTP Adapter Connection to an Integration in *Using the FTP Adapter with Oracle Integration 3*.
Delete a file in a File Server directory	**Delete File**	1. Start Here: Add the File Server Action to an Integration 2. Option 5. Delete a File in File Server

When Not to Use the File Server Action

If you need to write a file to a temporary storage location and then perform work on the file, such as by updating the file, use a connection that is based on the FTP Adapter, or use the stage file action.

Start Here: Add the File Server Action to an Integration

No matter what you need to use the File server action to do, start here by adding the action to your integration.

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

> **Note**
>
> If you don't see the File server action or can't add it to an integration, then an administrator needs to enable File Server for your Oracle Integration instance. See Enable File Server in *Using File Server in Oracle Integration 3*.

Copyright © 2022, 2026, Oracle and/or its affiliates.

2. Fill in the following fields.

Field	Description
What do you want to call your endpoint?	Name of the endpoint, such as **write_file_to_File_Server**.
Select resource	Select one of the following options: • **File:** Interact with a file, such as by writing, moving, downloading, reading, or deleting it. • **Directory:** Obtain a list of files in a specific directory in File Server.

Additional fields, which are related to the operation that you selected, appear.

3. Continue following the steps in the appropriate section:
   - Option 1. Obtain a List of Files in File Server
   - Option 2. Write a File to File Server
   - Option 3. Move a File to File Server
   - Option 4. Stage a File and Work with It, or Write a File to Another Location in File Server
   - Option 5. Delete a File in File Server

Option 1. Obtain a List of Files in File Server

1. Complete the steps in Start Here: Add the File Server Action to an Integration.
2. From the **Select operation** drop-down list, select **List directory**.
3. Fill in the following fields.

Field	Description	Can override later in the mapper?
Input Directory	Name of the directory that contains the files that you need to list. Oracle recommends using an absolute directory. For example, `/HOME/USERS/`.	Yes
File name pattern	Pattern of the files that you want to list, such as ***.csv** for only CSV files. Or, enter ***** to list files of all types.	Yes
Max Files	Maximum number of files to return, up to 1000. **Note:** The File server action returns the files according to the last modified time. If you selected 10 as the maximum number of files and the last modified time of the eleventh file is the same as the tenth file, then the action also returns the eleventh file and any other files with the same last modified time. For example, consider a directory with 15 files, and the Max Files setting is 10. If the tenth, eleventh, twelfth, and thirteenth files have the same time stamp, then the File server action returns thirteen files.	No

Copyright © 2022, 2026, Oracle and/or its affiliates.

Field	Description	Can override later in the mapper?
Minimum Age (seconds)	Minimum age in seconds of the files to return. For example, if a file was last modified at 02:28:45 AM, and the minimum age is 80 seconds, the file isn't returned at 02:29:00 AM and 02:30:00 AM. The file is returned only after 02:30:05.	No
List Files Recursively	Whether to list files within directories recursively beneath the input directory.	No
Sort By, Sort Order	The list of files returned in the response can be sorted using the following options: • **Sort By**: Filename, Last Modified Time, Creation Time, Size • **Sort Order**: ASC, DESC.	No

4. Select **Continue**.
5. Review the summary of the action, and select **Finish**.

The File server action appears on the canvas with a Map action before it.

Option 2. Write a File to File Server

1. Complete the steps in Start Here: Add the File Server Action to an Integration.
2. From the **Select operation** drop-down list, select **Write File**.
3. Fill in the following fields.

Field	Description	Can override later in the mapper?
Specify an Output Directory	Name of the directory to write the files to. Oracle recommends using an absolute directory. For example, `/HOME/USERS/`.	Yes
File Name	Name of the file to create.	Yes

4. Select **Continue**.
5. Review the summary of the action, and select **Finish**.

The File server action appears on the canvas with a Map action before it.

Option 3. Move a File to File Server

1. Complete the steps in Start Here: Add the File Server Action to an Integration.
2. From the **Select operation** drop-down list, select **Move File**.
3. Fill in the following fields.

Field	Description	Can override later in the mapper?
Directory path	Name of the directory that contains the file to be moved. Oracle recommends using an absolute directory. For example, `/HOME/USERS/`.	Yes

Copyright © 2022, 2026, Oracle and/or its affiliates.

Field	Description	Can override later in the mapper?
File name	Name of the file to be moved. The file can be of any size. If the file is missing, an error occurs when the integration attempts to locate it.	Yes
Target directory path	Name of the directory to move the file to. Oracle recommends using an absolute directory. For example, `/HOME/USERS/`. If the directory doesn't exist, File Server creates it.	Yes
Target file name	Name of the file in the target directory.	Yes
Overwrite File	When selected, the target file is overwritten if it already exists. If you deselect this option and the file already exists in the target directory, an error occurs when File Server attempts to move the file to the directory.	No

4. Select **Continue**.
5. Review the summary of the action, and select **Finish**.

The File server action appears on the canvas with a Map action before it.

Option 4. Stage a File and Work with It, or Write a File to Another Location in File Server

This operation gets a reference to a file in File Server. If anyone moves or deletes the file while the integration is running, an error occurs, indicating that the file doesn't exist.

1. Complete the steps in Start Here: Add the File Server Action to an Integration.
2. From the **Select operation** drop-down list, select **Get File Reference**.
3. Fill in the following fields.

Field	Description	Can override later in the mapper?
Input Directory	Name of the directory that contains the files that you need to download. Oracle recommends using an absolute directory. For example, `/HOME/USERS/`.	Yes
File Name	Name of the file in the input directory.	Yes

4. Select **Continue**.
5. Review the summary of the action, and select **Finish**.

The File server action appears on the canvas with a Map action before it.

Option 5. Delete a File in File Server

1. Complete the steps in Start Here: Add the File Server Action to an Integration.
2. From the **Select operation** drop-down list, select **Delete File**.
3. Fill in the following fields:

Copyright © 2022, 2026, Oracle and/or its affiliates.

Field	Description	Can override later in the mapper?
Directory path	Name of the directory to delete. Oracle recommends using an absolute directory. For example, `/HOME/USERS/`.	Yes
File name	Name of the file to delete.	Yes

4. Select **Continue**.
5. Review the summary of the action, and select **Finish**.

The File server action appears on the canvas with a Map action before it.

Note that the activity stream doesn't detect the file size, so the file size always appears as 0 in the activity stream for deleted files.

Next Steps

- Pass dynamic values into the File server action using the Map action, which appears before the File server action on the canvas. See About Mapping Data Between Applications in *Using the Oracle Mapper with Oracle Integration 3*.
- If you need to encrypt a file and then write it to File Server, perform the following steps:
  1. After the File server action, add a stage file action to encrypt the file.

     See Process Files in Schedule Integrations with a Stage File Action.
  2. After the stage file action, add another File server action using the Write File operation. This action uploads the file to File Server by passing the encrypted file reference.

     See Option 2. Write a File to File Server.

  Alternatively, use the FTP Adapter to encrypt the file and write it to File Server. See Create a Connection and Add the FTP Adapter Connection to an Integration in *Using the FTP Adapter with Oracle Integration 3*.
- If you need to decrypt a file that is in File Server, use the stage file action or the FTP Adapter. See the following links:
  - Process Files in Schedule Integrations with a Stage File Action
  - Create a Connection and Add the FTP Adapter Connection to an Integration in *Using the FTP Adapter with Oracle Integration 3*

Invoke A Child Integration with an Integration Action

You can invoke a co-located (child), active integration from a parent integration. Co-located means the integration is running on the same host instance or in the same domain.

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

See Invoke a Child Integration from a Parent Integration.

Copyright © 2022, 2026, Oracle and/or its affiliates.

Add JavaScript Functions to an Integration with a JavaScript Action

You can add JavaScript functions to the integration.

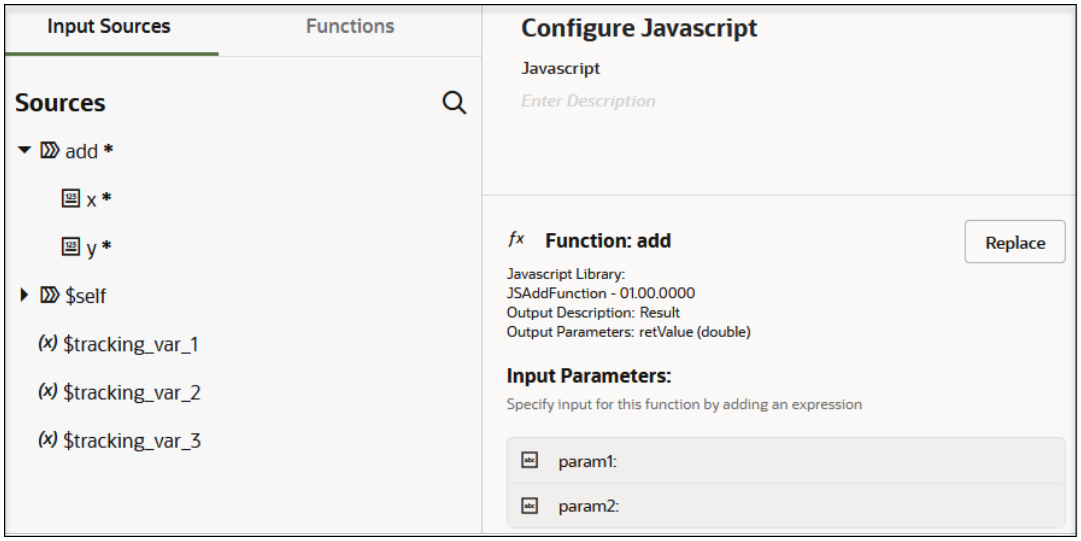
Create a JavaScript Action

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

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

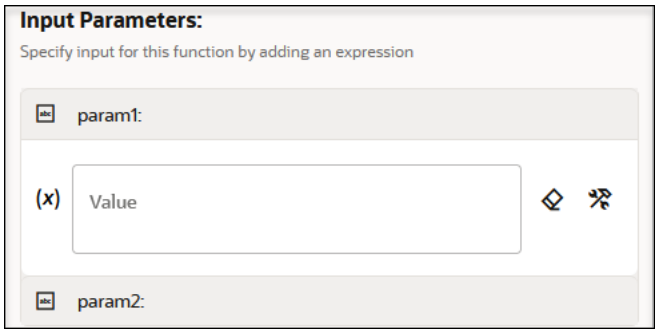
   The **Functions List** appears.
4. Select a function.

   The function and **Sources** tree are displayed. Function details are shown, including the input and output parameters.

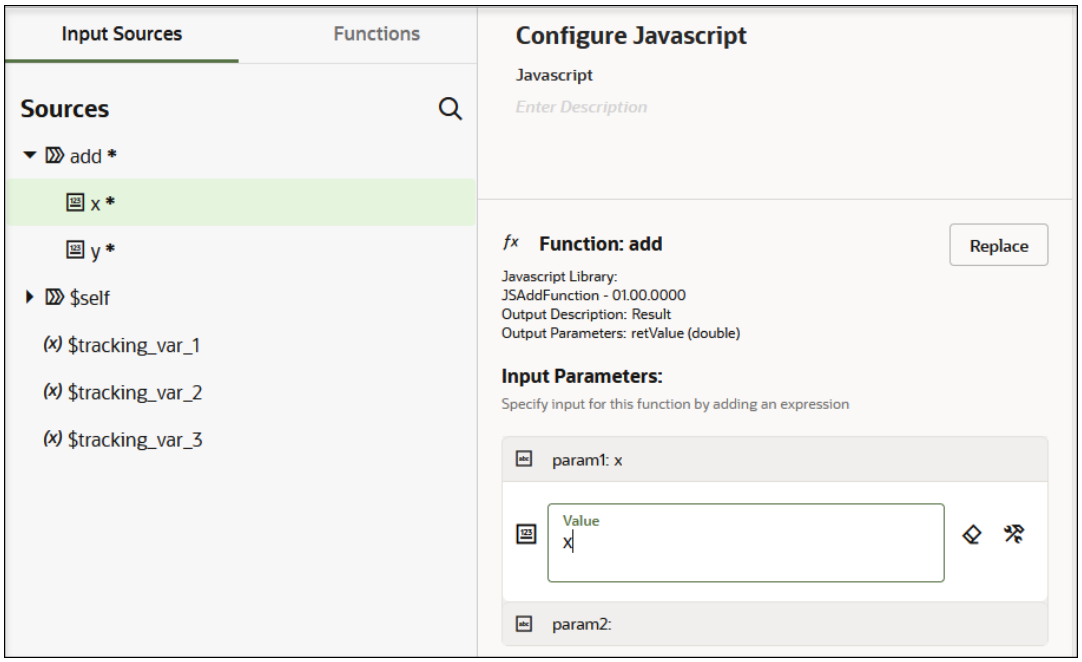


Copyright © 2022, 2026, Oracle and/or its affiliates.

5. Click the first input parameter to open a **Value** field.



6. Drag a value from the **Sources** tree.



7. Perform the same steps to add a value for the second input parameter.
8. Click **Save**.

Track the Status of a Javascript Action During Runtime

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

1. In the navigation pane, click **Observability**, then **Instances**.
2. Click the business identifier value of the integration to track.
3. The integration (including any JavaScript actions) is displayed. Any JavaScript action failures are identified by red.

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

Copyright © 2022, 2026, Oracle and/or its affiliates.

See Use Libraries to Manage Integration Functions.

Invoke Oracle Cloud Infrastructure Functions Directly from an Integration with an OCI Function Action

Oracle Integration provides native support for invoking Oracle Cloud Infrastructure functions. This support enables you to invoke Oracle Cloud Infrastructure functions with an OCI Function action in an integration.

Oracle Cloud Infrastructure functions are part of a fully managed, multi-tenant, highly scalable, on-demand, Functions-as-a-Service platform. Use Oracle Cloud Infrastructure functions to focus on writing code to meet business needs. See Overview of Functions and Functions QuickStart Guides.

> **Note**
>
> Ensure that your cloud tenancy uses identity domains. The OCI Function action in the integration canvas does *not* work in cloud tenancies that are not enabled for identity domains.

- Capabilities
- Prerequisites
- Invoke Oracle Cloud Infrastructure Functions
- Override the Endpoint URL Setting for the OCI Function Action at Runtime

Want to see a video demo of the OCI Function action?

Video

Capabilities

Oracle Cloud Infrastructure functions provide serverless execution of functions within Oracle Integration to process complex logic not easily performed in Oracle Integration, such as the following:

- Creating a digital signature based on a key against a specific payload.
- Supporting a custom ZIP format such as GZIP or TAR.
- Taking and parsing a PDF file for data fields that can then participate in an integration.

Oracle Integration supports whatever functions you create using Oracle Cloud Infrastructure functions. Oracle Cloud Infrastructure also provides a catalog of prebuilt functions. See Creating Functions Using Pre-Built Functions.

Copyright © 2022, 2026, Oracle and/or its affiliates.

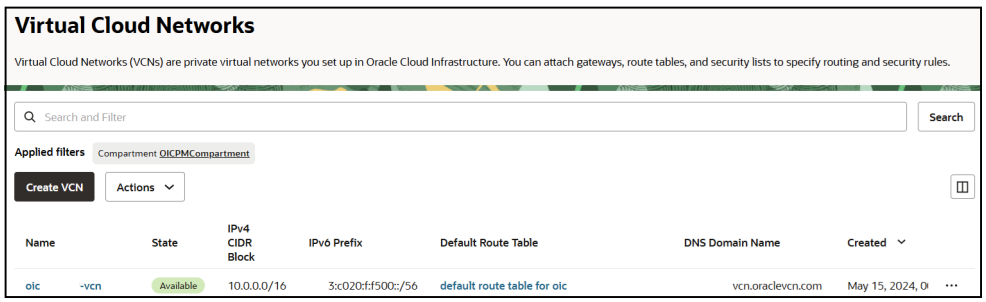
Prerequisites

> **Note**
>
> When making cross-region calls (the function is in a different region than that of the Oracle Integration service instance), ensure you do the following:
>
> - The tenancy is subscribed to both regions. See Subscribing to an Infrastructure Region.
> - The identity domain in which the Oracle Integration service instance resides is replicated to both regions. See Replicating an Identity Domain to Multiple Regions.

You must perform the following prerequisites from the Oracle Cloud Console before you can invoke functions from an integration. Not performing these prerequisites results in the following error:

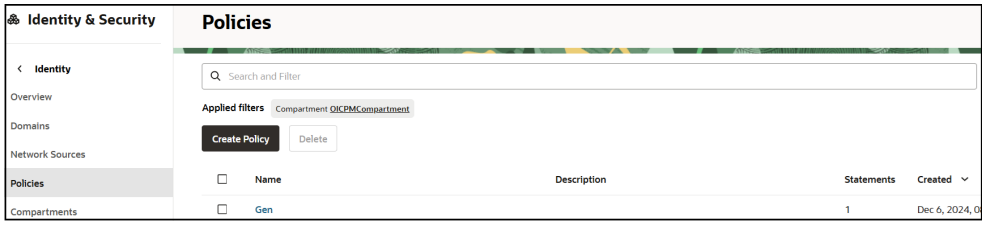
```
An authentication issue was encountered when invoking Function. Please
check if the prerequisites for calling OCI services have been properly setup
for
this service instance. OCI service integration requires dynamic groups and
policies
granting access to be created in OCI Console. Please consult the
documentation for
more information.
```

1. Ensure that your cloud tenancy uses identity domains. Invoking Oracle Cloud Infrastructure functions from an integration does *not* work in cloud tenancies that are not enabled for identity domains. If you are unsure, ask your administrator for details.
2. Set up your environment to invoke Oracle Cloud Infrastructure functions from an integration. See Functions QuickStart on Cloud Shell. This section describes how to set up the following:
   - Set up your tenancy:
     - Create groups and users (if not already done so).
     - Create a compartment in which to create functions. You specify this compartment when configuring the **OCI Function** action in the integration canvas.
     - Create a virtual cloud network (VCN) and subnets. For example:

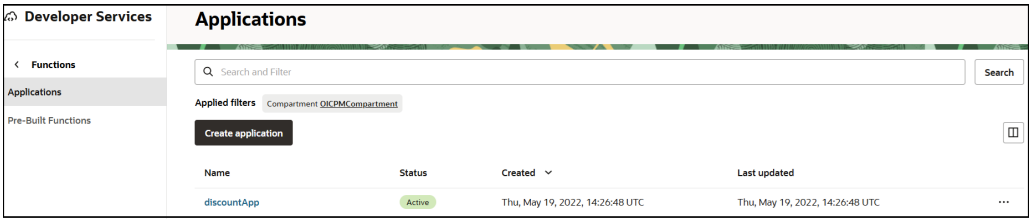


     - Create a policy for groups and services. This policy enables you to create functions. For example:

Copyright © 2022, 2026, Oracle and/or its affiliates.

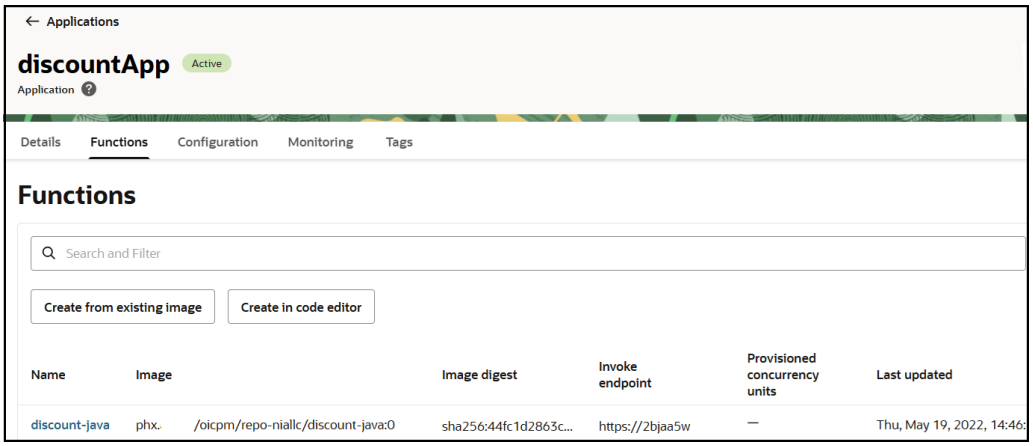


- Create an application in which to create the functions to invoke. You specify this application when configuring the **OCI Function** action in the integration canvas. For example:



- Set up your Cloud Shell development environment.
- Create, deploy, and invoke your function. For example:

> **Note**
>
> You must create your own functions in your tenancy. Oracle Cloud Infrastructure doesn't deploy or create any functions for you by default. Oracle Cloud Infrastructure does provide prebuilt functions that you can deploy to your own instance.

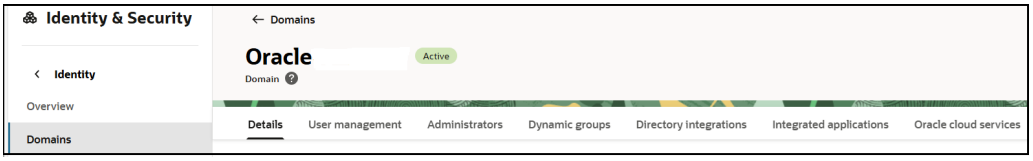


3. Create a dynamic group.
   a. Obtain the client ID of the OAuth application for the Oracle Integration instance.
   b. In the navigation pane, select **Identity & Security**.
   c. Under **Identity**, select **Domains**.

Copyright © 2022, 2026, Oracle and/or its affiliates.

d. In the upper right corner, select **Profile**, then click the identity domain.

e. In the menu bar, click **Oracle cloud services**.



The Oracle Cloud Services page for your domain appears.

f. In the **Name** column, click your service instance.

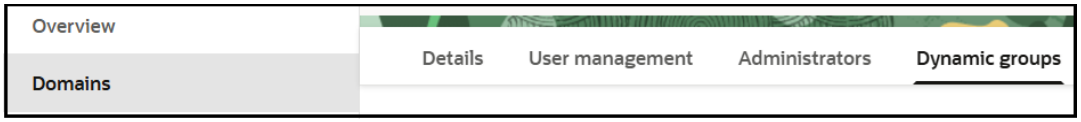
g. Click **OAuth configuration**.

h. Scroll down to the **General Information** section and copy the client ID value to use to create your dynamic group.

4. Scroll to the breadcrumbs at the top and click **Oracle cloud services** to return to the main page for the identity domain.



5. In the menu bar, click **Dynamic groups**.



6. Click **Create dynamic group**.

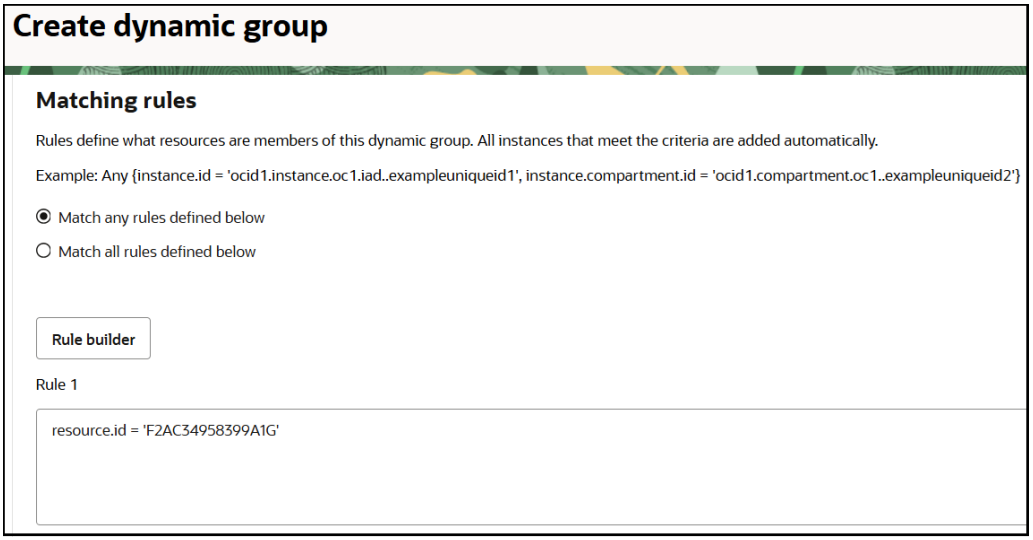
7. Enter the following details:

a. In the **Name** and **Description** fields, enter values. These fields are required.

b. In the **Matching Rules** section, enter the required rule. The resource ID you specify must match the client ID of the OAuth application of your Oracle Integration instance. Ensure that you enclose the value in single quotes. For example:

```
resource.id = 'client_ID'
```


Copyright © 2022, 2026, Oracle and/or its affiliates.



8. In the left navigation pane, click **Policies** to assign a policy to the dynamic group to enable you to invoke Oracle Cloud Infrastructure functions from an integration.
9. Click **Create Policy**.
10. Enter the following details:
    a. In the **Name** and **Description** fields, enter values. These fields are required.
    b. In the **Policy Builder** section, click **Show manual editor** to build the required policy for the dynamic group. The following example allows the dynamic group to use the entire functions family, but you can set more granular policies, as needed. See Controlling Access to Invoke and Manage Functions and Policy Statements to Give OCI Functions Users Access to Function-Related Resources.

       ```
       Allow dynamic-group group_name to use functions-family in compartment
       compartment_name
       ```

       Where:

       - *group_name*: Is the dynamic group name you created.
       - *compartment_name*: Is the compartment where the functions to invoke are hosted.

       > **Note**
       >
       > If you are not creating the policy in the default domain, then specify your domain name in front of the group name. For example:
       >
       > ```
       > Allow dynamic-group identity_domain/group_name to use functions-
       > family in compartment compartment_name
       > ```

       This enables the Oracle Integration instance associated with the dynamic group to invoke Oracle Cloud Infrastructure functions with the **OCI Function** action in an integration.

Copyright © 2022, 2026, Oracle and/or its affiliates.

Invoke Oracle Cloud Infrastructure Functions

This section provides an overview of how to configure an **OCI Function** action in the integration canvas to invoke the Oracle Cloud Infrastructure functions available in your application.

1. Add an **OCI Function** action to an integration in either of the following ways:

   - On the side of the canvas, click **Actions**  and drag the **OCI Function** action to the appropriate location.
   - Click  at the location where you want to add the parallel action, then select **OCI Function**.

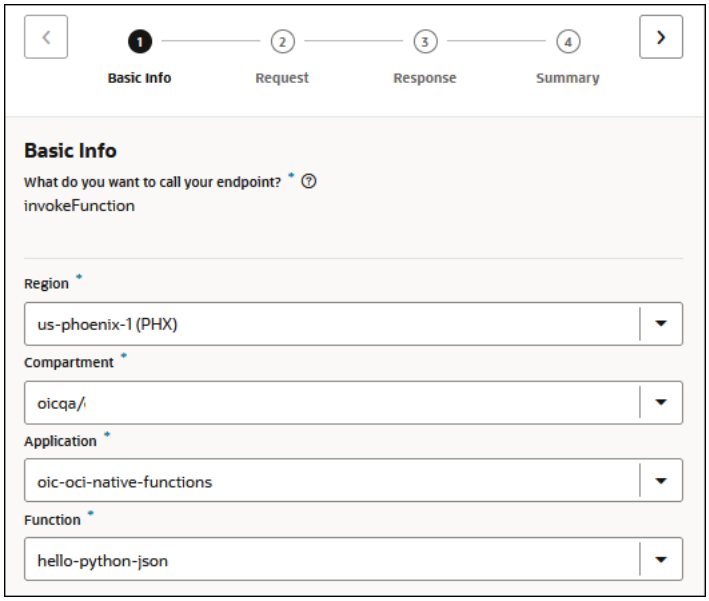
   > **Note**
   >
   > If your cloud tenancy is not using identity domains, an error message appears and the page is blank.

   The OCI Functions wizard appears.

2. On the Basic Info page, enter the following information to identify the function to invoke, then click **>**. When you enter details in a field, the page refreshes to show more fields.

Element	Description
Region	Select the region where the functions are deployed.
Compartment	Select the compartment where the functions to invoke are hosted.
Application	Select the application. Oracle Cloud Infrastructure functions are organized in sets called applications. An application can contain multiple functions.
Function	Select the Oracle Cloud Infrastructure function to invoke. The functions you are entitled to invoke by your application are displayed for selection.

Copyright © 2022, 2026, Oracle and/or its affiliates.



3. On the Request page, select the payload type and specify the content, then click **>**.
   - **JSON Sample**: Drag and drop or select a file, or manually enter the payload.
   - **XML Schema**: Drag and drop or select a file.
   - **XML Sample (Single or No NameSpace)**: Drag and drop or select a file.
   - **Binary**:
     - For files, simply select **Binary** and map the content.
     - For plain/text request/response payloads, select **Binary** and perform the following encoding:
       * To provide text as input, you must first encode the text as `Base64`, then decode that `Base64` to a stream reference.

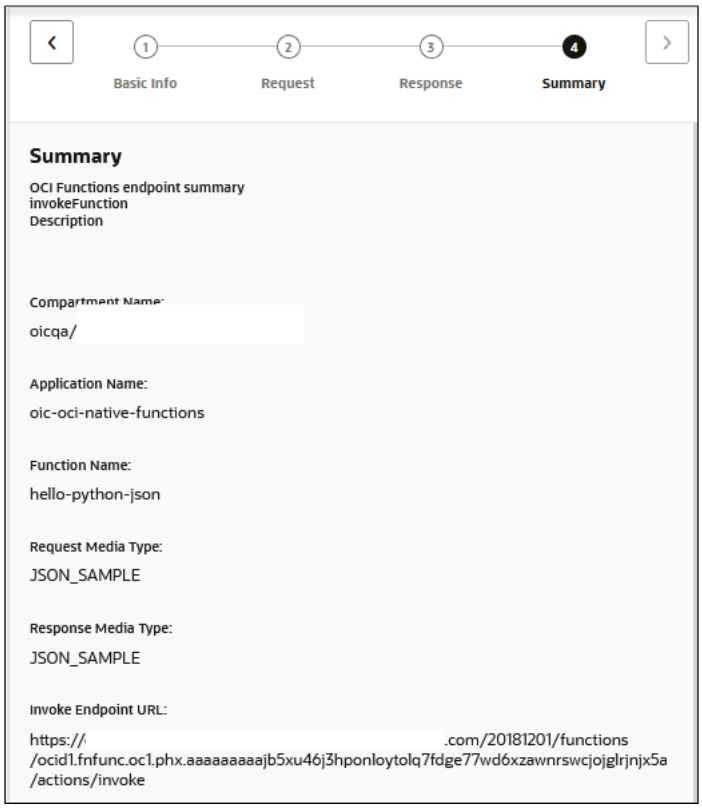
         ```
         decodeBase64ToReference(encodeBase64(Name))
         ```

       * To receive the text output, you need to encode the output stream reference to `Base64`, then decode that `Base64` to text.

         ```
         decodeBase64(encodeReferenceToBase64(Stream Reference))
         ```

4. On the Response page, select the response payload type and specify the content, then click **>**. The available response payload types are the same as the request payload types.
5. Review your selections on the Summary page. For this example, the **hello-python-json** function is invoked. The request and response media is in JSON format.

Copyright © 2022, 2026, Oracle and/or its affiliates.



6. Click **Done**.

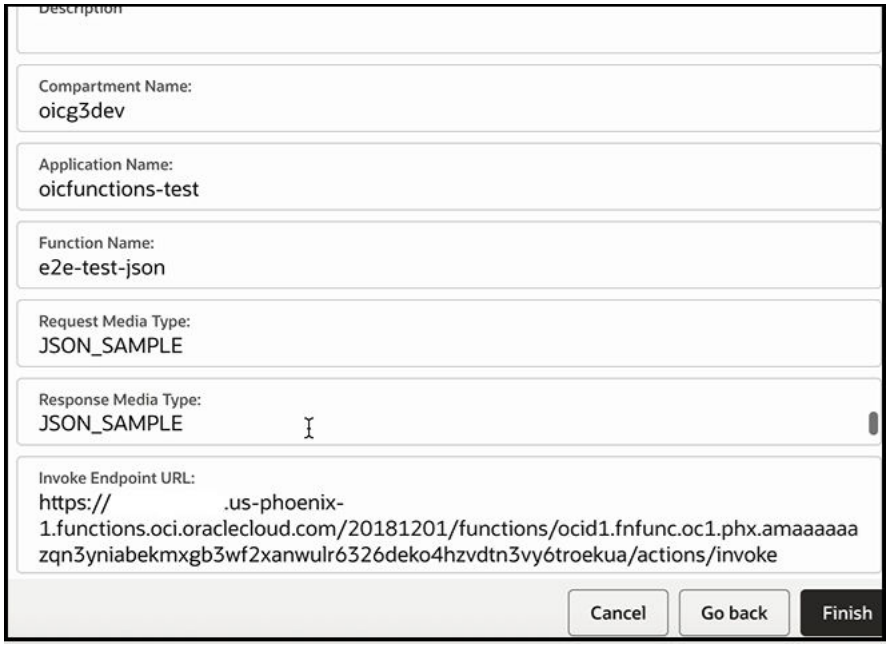
Override the Endpoint URL Setting for the OCI Function Action at Runtime

You can specify the invoke endpoint URL to use at runtime in the mapper. This setting overrides the invoke endpoint URL specified in the OCI Function action at design time.

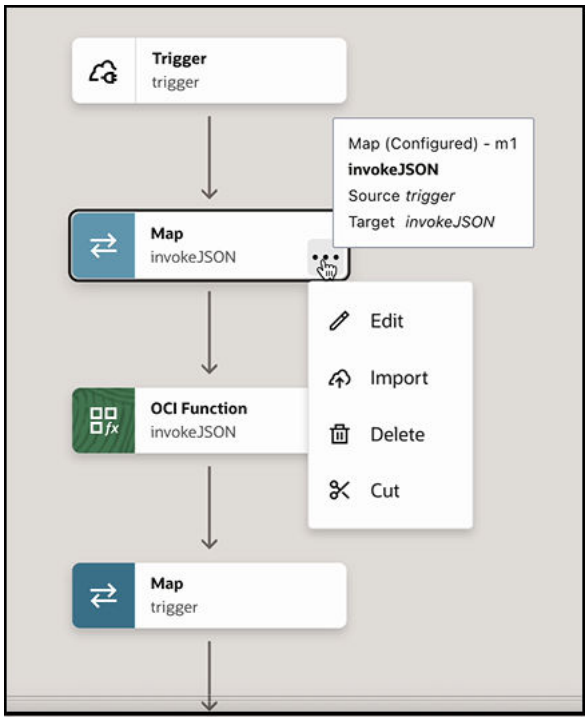
The following steps provide an overview of configuring this feature:

1. Configure the **OCI Function** action. The **Invoke Endpoint URL** field on the Summary page shows the URL specified.

Copyright © 2022, 2026, Oracle and/or its affiliates.

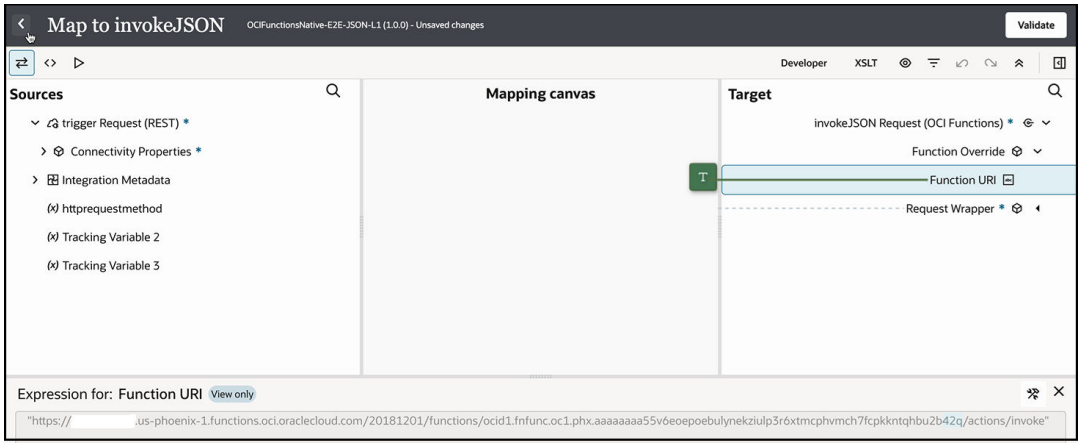


2. Open the mapper for editing.

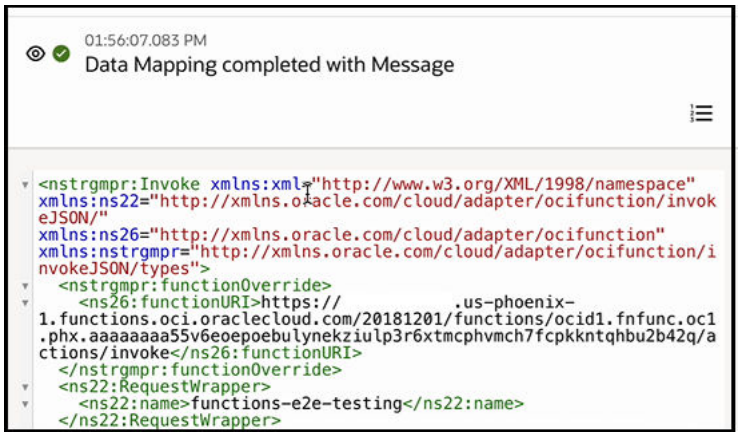


3. Expand the target **Function Override** element.
4. Right-click **Function URI** and select **Create target node**.
5. Enter the endpoint URL in the Expression Builder. This value overrides the value configured in the OCI Function action for the **Invoke Endpoint URL** field.

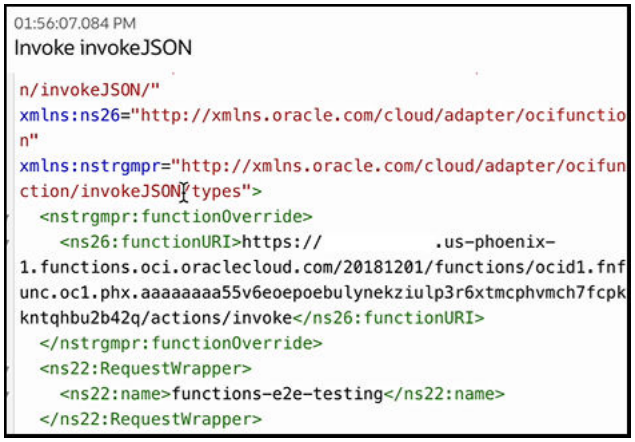
Copyright © 2022, 2026, Oracle and/or its affiliates.



6. Complete integration design, then activate and run the integration.
7. View the mapper in the activity stream and note that the function URL specified in the mapper was used.



8. Expand the invoke connection for the OCI function action in the activity stream and note that the function URL provided in the mapper was also used.



Copyright © 2022, 2026, Oracle and/or its affiliates.

Invoke Oracle Cloud Infrastructure Object Storage from an Integration with an OCI Object Storage Action

You can natively invoke Oracle Cloud Infrastructure Object Storage from an integration without the need to configure an explicit REST Adapter connection.

- Oracle Cloud Infrastructure Object Storage Concepts
- Prerequisites
- Invoke Oracle Cloud Infrastructure Object Storage from an Integration
- Override the Object Storage Bucket Name at Runtime

Oracle Cloud Infrastructure Object Storage Concepts

Oracle Cloud Infrastructure Object Storage is an internet-scale, high-performance storage platform that offers reliable and cost-efficient data durability. Oracle Cloud Infrastructure Object Storage enables you to securely store any type of data in its native format. An unlimited amount of unstructured data of any content type is supported, including analytic data and rich content such as images and videos. See Object Storage and Overview of Object Storage.

Prerequisites

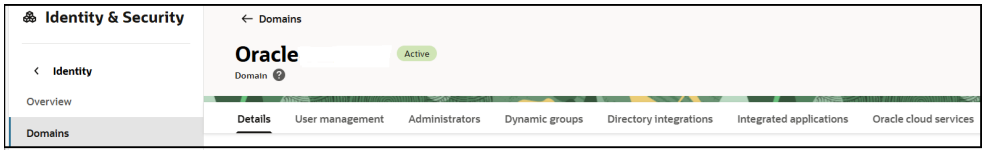
To natively invoke Oracle Cloud Infrastructure Object Storage, you must satisfy the following Resource Principal Session Token (RPST) prerequisites in the Oracle Cloud Console.

> **Note**
>
> The OCI object storage action doesn't support cross tenancy calls. To make cross tenancy calls, configure the REST Adapter with the OCI Signature Version 1 security policy to call the object storage REST APIs.

- Create a dynamic group
- Create a policy to grant access to Oracle Cloud Infrastructure Object Storage.
- Ensure that your cloud tenancy uses identity domains. If you are unsure, ask your administrator for details.
- Create the required dynamic group and assign a policy to that group to allow your Oracle Integration instance to access Oracle Cloud Infrastructure Object Storage. The policy defines the permissions for the dynamic group and determines which operations the dynamic group can perform on Oracle Cloud Infrastructure Object Storage and other services such as Oracle Cloud Infrastructure Functions, Oracle Cloud Infrastructure Vision, and more

  1. Log in to the Oracle Cloud Console.
  2. Obtain the client ID of the OAuth application for the Oracle Integration instance.
     a. In the navigation pane, select **Identity & Security**.
     b. Under **Identity**, select **Domains**.
     c. In the upper right corner, select **Profile**, then click the identity domain.
     d. In the menu bar, click **Oracle cloud services**.

Copyright © 2022, 2026, Oracle and/or its affiliates.



The Oracle Cloud Services page for your domain appears.

e. In the **Name** column, click your service instance.

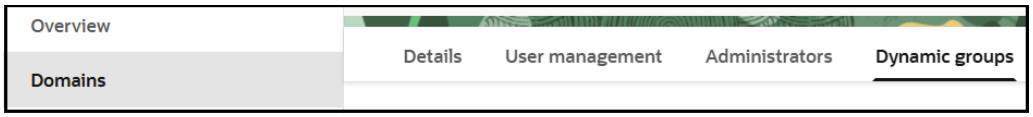
f. Click **OAuth configuration**.

g. Scroll down to the **General Information** section and copy the client ID value to use to create your dynamic group.

3. Scroll to the breadcrumbs at the top and click **Oracle cloud services** to return to the main page for the identity domain.



4. In the menu bar, click **Dynamic groups**.



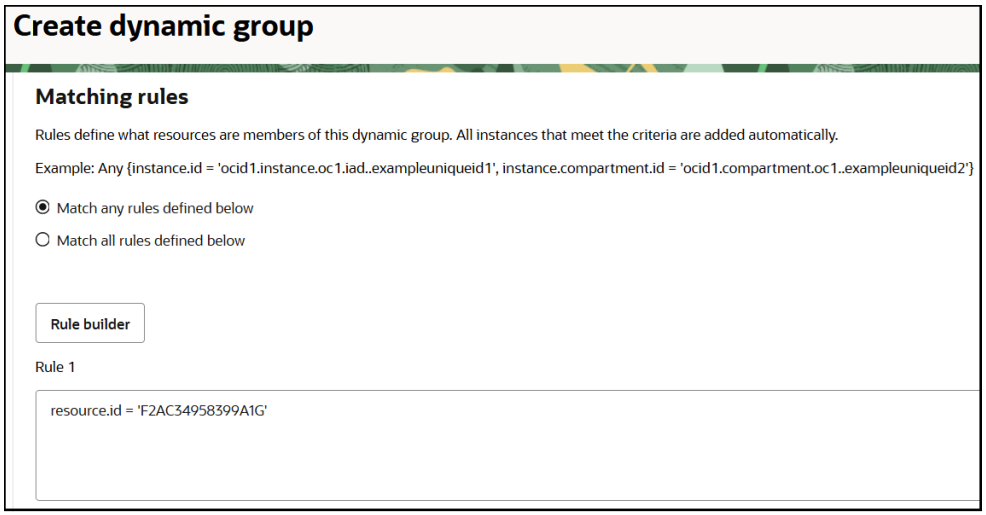
5. Click **Create dynamic group**

6. Enter the following details:

a. In the **Name** and **Description** fields, enter values. These fields are required.

b. In the **Matching Rules** section, enter the required rule. The resource ID you specify must match the client ID of the OAuth application of your Oracle Integration instance. Ensure that you enclose the value in single quotes. For example:

```
resource.id = 'client_ID'
```



Copyright © 2022, 2026, Oracle and/or its affiliates.

7. In the left navigation pane, click **Policies**.
8. Click **Create Policy**.
9. Select the compartment in which to create the policy.
10. Enter the following details:
    a. In the **Name** and **Description** fields, enter values. These fields are required.
    b. In the **Policy Builder** section, build the required policy for the dynamic group. Examples of the minimum policy required are as follows:

```
allow dynamic-group dynamic_group to manage object-family in
compartment compartment_name
```

```
allow dynamic-group dynamic_group to inspect compartments in
compartment compartment_name
```

    Where:

    – *dynamic_group*: Is the dynamic group name you specified in Step 5.
    – *compartment_name*: Is the compartment in which your Oracle Integration instance is located.

    This enables the Oracle Integration instance associated with the dynamic group to call Oracle Cloud Infrastructure Object Storage in this particular compartment. The RPST token is only valid for resources to which the dynamic group has been granted access using this policy.

    Details about more granular permissions for object storage are provided. See Securing Object Storage.

Invoke Oracle Cloud Infrastructure Object Storage from an Integration

1. Add an **OCI Object storage** action to an integration in either of the following ways:
    - On the side of the canvas, click **Actions** and drag the **OCI Object storage** action to the appropriate location.
    - Click + at the location where you want to add the action, then select **OCI Object storage**.

    The Configure object storage panel opens.
2. Enter a name.
3. Select the resource to manage.
    - **Manage buckets**: Buckets are logical containers for storing objects. Buckets are region- and compartment-specific. You can define policies at a bucket level to control access.
    - **Manage objects**: Objects are stored within buckets. Objects can be any data type and consist of the object itself and any metadata.
4. If you select **Manage buckets**:
    a. From the **Operations** list, select the operation to perform.

Copyright © 2022, 2026, Oracle and/or its affiliates.

- **Create bucket**: Create a bucket in Oracle Cloud Infrastructure Object Storage. In the mapper, you map the bucket name to create.
  i. Select the compartment in which to create the bucket. This selection during design-time is overridden if you map anything to the **compartment Id** field in the request mapper for this action.
- **List buckets**: Retrieve a list of objects in a bucket in Oracle Cloud Infrastructure Object Storage. In the mapper, you map the bucket name to retrieve the objects list.
  i. Select the compartment in which to list the buckets. This selection during design-time is override if you map anything to the **compartment Id** field in the request mapper for this action.
- **Delete bucket**: Delete a bucket in Oracle Cloud Infrastructure Object Storage. In the mapper, you map the bucket name to delete.

b. Click **Continue**.

5. If you select **Manage objects**:
   a. From the **Operations** list, select the operation to perform:
      - **Upload object**: Upload an object into Oracle Cloud Infrastructure Object Storage. In the mapper, you map the object name to upload. This is how you update an object (it overrides an existing object with the same name). The file size limit is 1 GB. However, the API can handle objects up to 50 GB in size. See PutObject.
      - **Download object**: Retrieve an object from Oracle Cloud Infrastructure Object Storage. In the mapper, you map the object name to retrieve. The size limit is 1 GB. Otherwise, the download fails. Object storage returns a stream reference. You need to use the encodeReferenceToBase64 function and then the decodeBase64 function to get the actual content.
      - **Delete object**: Delete an object in Oracle Cloud Infrastructure Object Storage. In the mapper, you map the object name to delete.
      - **List objects**: List all objects in Oracle Cloud Infrastructure Object Storage. In the mapper, you map the bucket name to retrieve the objects list.
   b. Select the compartment in which to perform the operation.
   c. Select the bucket in which to perform the operation. This selection during design-time is overridden if you map anything to the **Bucket Name** field in the request mapper for this action.
   d. Click **Continue**.
6. Review your selections on the Summary page, then click **Finish**.
   This creates a map action in front of the **OCI Object storage** action.

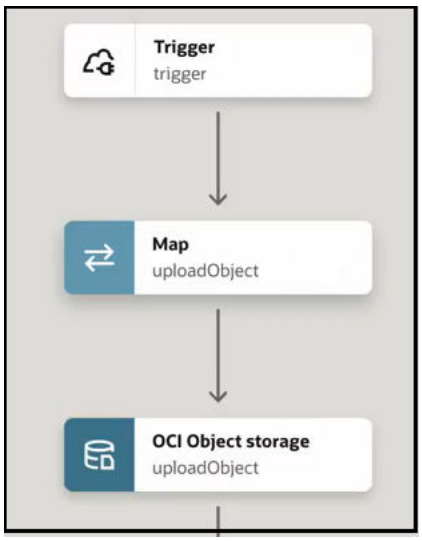
Several use cases that natively invoke Oracle Cloud Infrastructure Object Storage are provided. See Manage Oracle Cloud Infrastructure Buckets and Objects from an Integration.

Override the Object Storage Bucket Name at Runtime

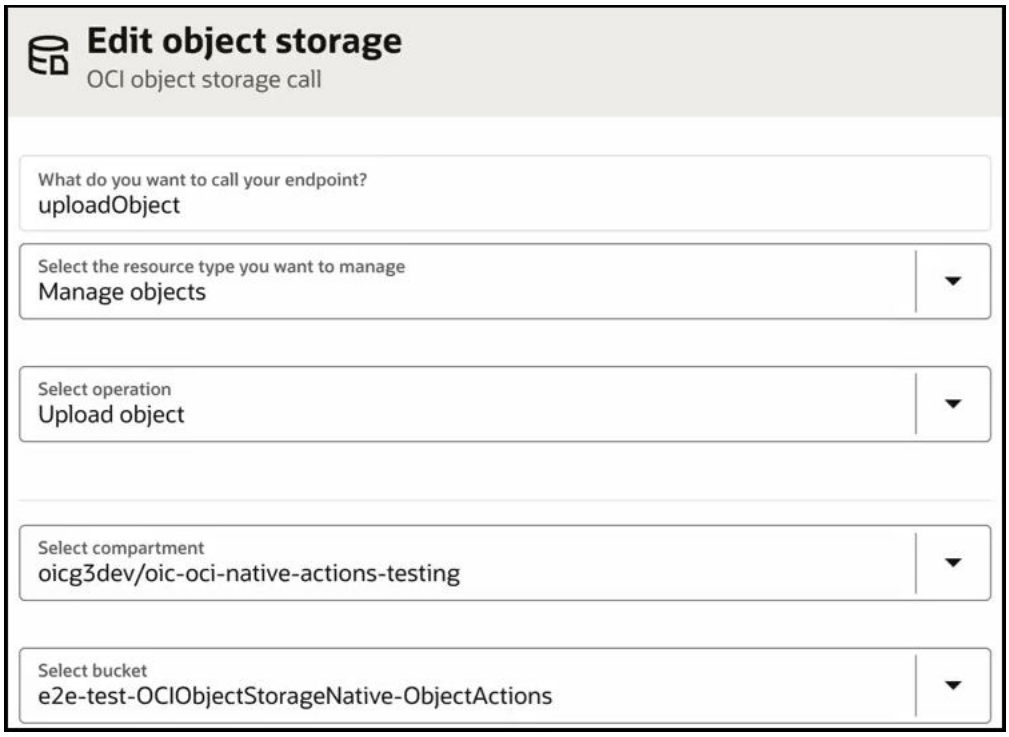
You can override the object storage bucket selected when configuring the OCI Object storage action by specifying a different object storage bucket to use at runtime in the mapper. This capability also implicitly overrides the compartment ID value for object storage bucket operations.

The following example shows part of an integration that includes a map action and an OCI Object storage action.

Copyright © 2022, 2026, Oracle and/or its affiliates.

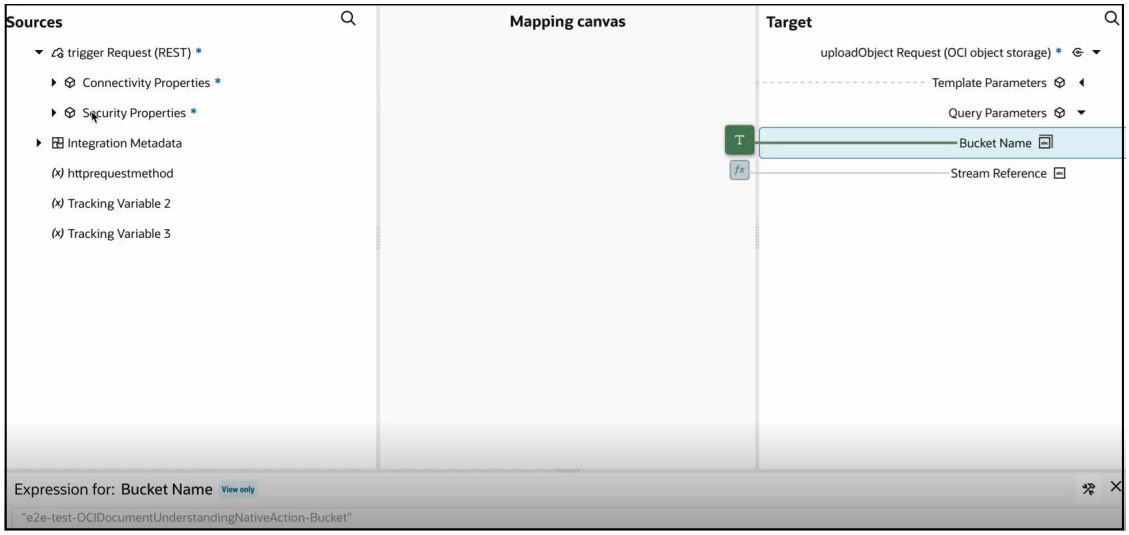


When configuring the OCI Object storage action (**uploadObject**), the **e2e-test-OCIObjectStorageNative-ObjectActions** bucket is selected.

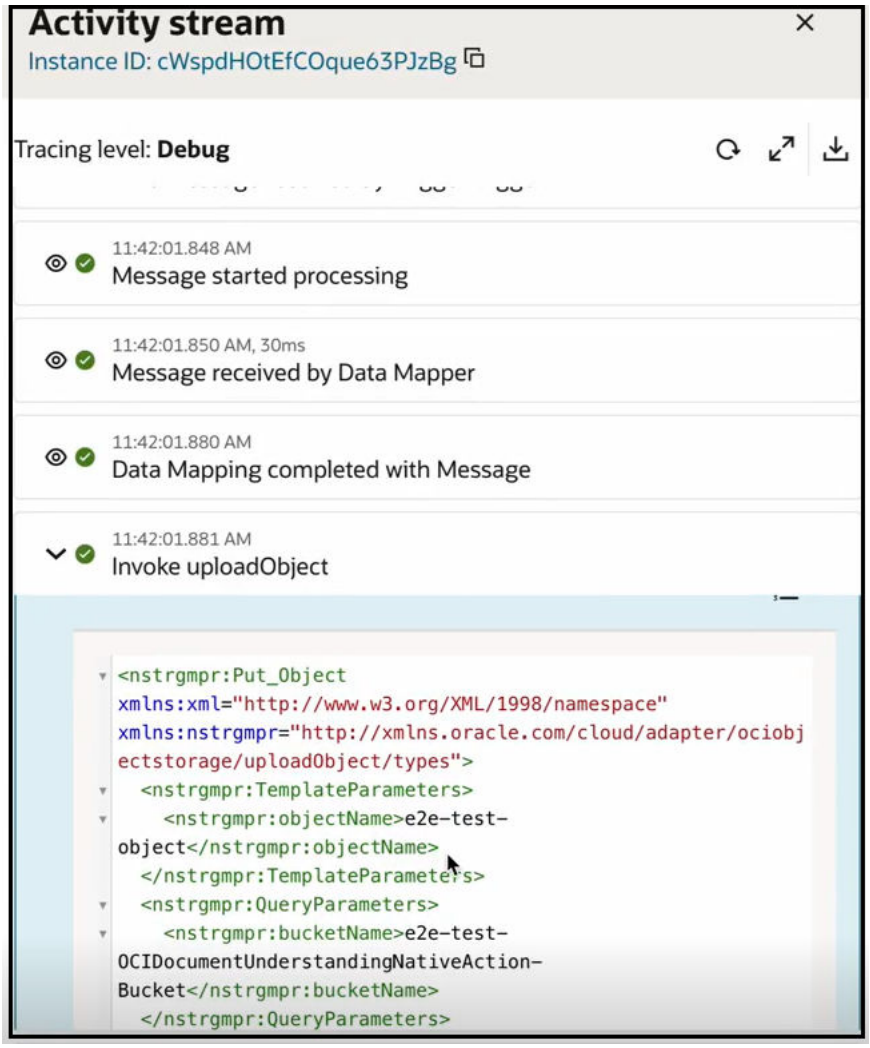


You can override this selection in the mapper. For this example, the target **Bucket Name** element is set to a different object storage bucket in the Expression Builder to use at runtime (**"e2e-test-OCIDocumentUnderstandingNativeAction-Bucket"**).

Copyright © 2022, 2026, Oracle and/or its affiliates.



After activating and invoking the integration, the bucket name set in the mapper (**e2e-test-OCIDocumentUnderstandingNativeAction-Bucket**) is used and is visible in the activity stream.



Copyright © 2022, 2026, Oracle and/or its affiliates.

Invoke a Process from an Integration with a Process Action

You can invoke process automation from an integration. When you drag the process action into an integration, the Process Automation call wizard is invoked and prompts you to select a process application and a process to invoke.

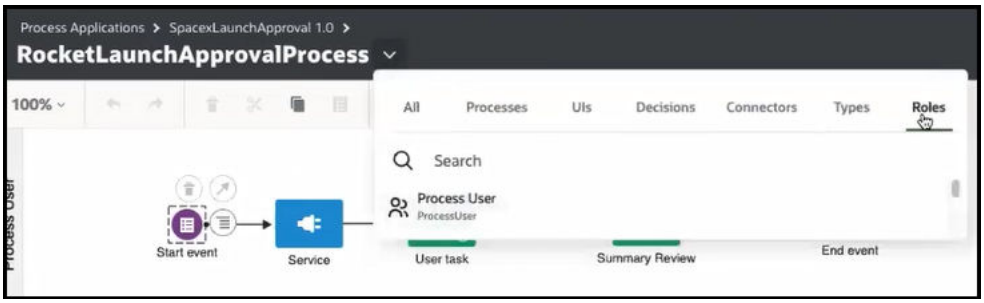
> **Note**
>
> If your Oracle Integration Generation 2 instance invoked processes from integrations and was upgraded to Oracle Integration 3, the following functionality is available:
>
> - You can invoke both Oracle Integration 3 processes and existing Generation 2 processes from integrations in Oracle Integration 3.
> - You can continue to create processes in Oracle Integration Generation 2, and then invoke them from integrations in Oracle Integration 3.

- Prerequisites
- Add a Process Action to the Integration

Prerequisites

- You must enable an Oracle Integration - Process Automation process in the Oracle Cloud Console for your service instance. Otherwise, you cannot drag the process action into the integration canvas. See Enable Process Automation with Oracle Integration 3 in *Administering Oracle Cloud Infrastructure Process Automation*.
- Ensure that you perform one of the following tasks to enable the integration to invoke the process that you select. Otherwise, you receive a `403 NOT AUTHORIZED` error at runtime.
  To provide access to all authenticated users:
  1. Click **Start event** in your process.
  2. From the **Who can run this activity** list, select **All users and external applications**.

  To provide access to Oracle Integration to invoke the process:
  1. From the drop-down list for your process, select **Roles**.



  2. From the **Category** list, select **External applications**.
  3. Copy the name of your service instance. The name is visible along with the instance shape at the top of the Oracle Integration page. See Get Familiar with the Home Page in *Getting Started with Oracle Integration 3*.

Copyright © 2022, 2026, Oracle and/or its affiliates.

4. Search for and select the instance name to add.



Add a Process Action to the Integration

1. Decide the type of process to invoke:

If You Want to Invoke ...	Then Drag this Process Icon into the Canvas...
An Oracle Integration 3 process	
An Oracle Integration Generation 2 process **Note**: This additional process icon is only available if your Oracle Integration Generation 2 instance invoked processes from integrations and was upgraded to Oracle Integration 3.	

2. Add a process action to an integration in either of the following ways:

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

   The Process Automation call wizard is displayed.

3. On the Basic Info page, enter a name and optional description, then click **Continue**.
4. On the Configuration page, enter the following details, then click **Continue**.

Copyright © 2022, 2026, Oracle and/or its affiliates.

Element	Description
Process Application	Select the process application. Process instances enabled as part of Oracle Integration 3 are available for selection. Process instances that are part of standalone Oracle Cloud Infrastructure Process Automation are not available for selection.
Version	**Default** is selected automatically and cannot be changed.
Process	Select the process name. To be visible for selection, the process must already be activated and have been started by a form or a message. Message- and form-started processes are retrieved by the process action for display.
Operation	**Create Instance** is selected automatically and cannot be changed.

5. On the Summary page, confirm your selections and click **Finish**. 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.
6. Complete the design and save the integration.
7. Activate and invoke the integration.
   Every time a process is invoked, a new instance of the process is created.

Publish Events in an Integration with a Publish Event Action

You can publish an event in an integration. Subscription integrations can then subscribe to this event.

1. Add a **Publish Event** action to an integration in either of the following ways:
   - On the side of the canvas, click **Actions** and drag the **Publish Event** action to the appropriate location.
   - Click + at the location where you want to add the publish event action, then select **Publish Event**.

   The Choose event panel opens.
2. Specify an event in either of two ways:
   a. Select the event, then click **Choose**. If you are unsure of the event contents, click **View details**.
   b. Click **Define new event** to create a new event. See Create Events to Publish and Subscribe to in Integrations.
3. Review your selections, then click **Finish**.

See Publish Events in an Integration.

Copyright © 2022, 2026, Oracle and/or its affiliates.

Call a Robot from an Integration

A robot runs only when an integration calls it. Call a robot from an integration using the robot process automation action.

All information about robots is in a separate guide. For guidance about using the robot process automation action, see Design an Integration That Calls a Robot in *Using Robots in Oracle Integration 3*.

Receive Requests for Multiple Resources in a Single REST Adapter Trigger with a Pick Action

You can expose multiple entry points to a single integration that uses the REST Adapter as the trigger connection. Each entry point can be configured with a different resource URI and HTTP action/verb, when necessary. This feature eliminates the need to create multiple integrations (each with a separate resource URI and verb) to perform different operations.

This feature is useful in the following scenarios:

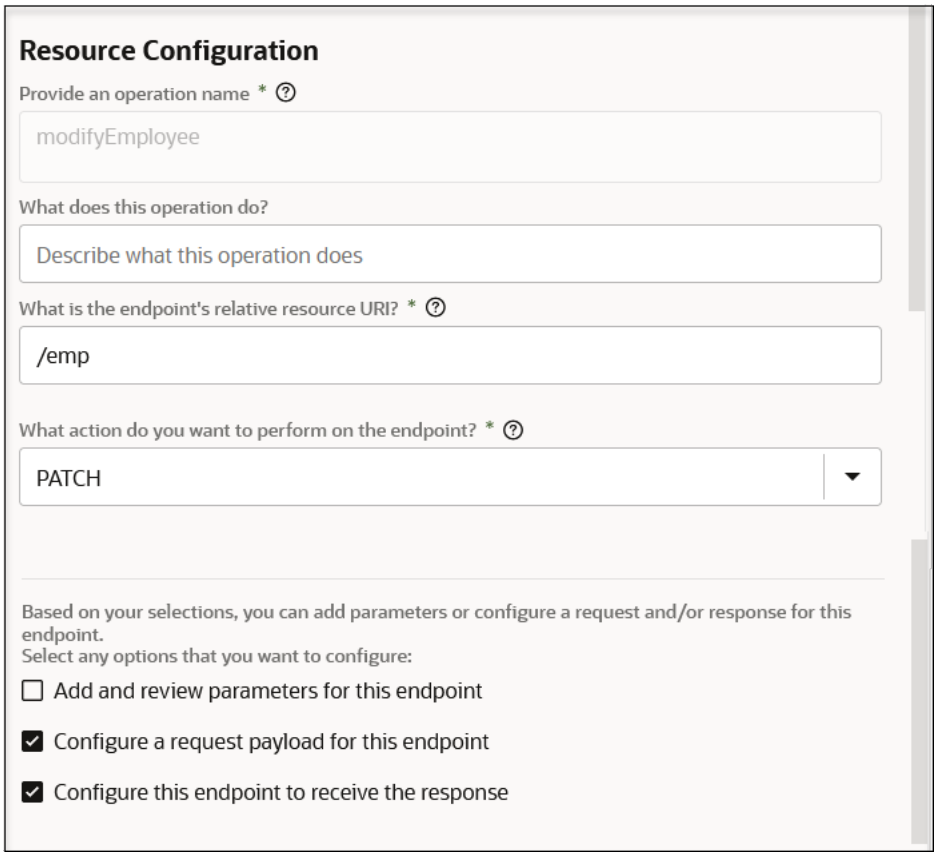
- Supports multiple resource URIs and actions/verbs in a specific integration through the REST Adapter.
- Supports multiple integration patterns (for example, synchronous and asynchronous). The combination of a resource and a verb is called an operation. Each operation can be configured as a synchronous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

> **Note**
>
> This feature is only available when creating an application integration.

1. Drag a REST Adapter into the canvas as a trigger connection of an application integration.

   This invokes the Adapter Endpoint Configuration Wizard.
2. On the Basic Info page, specify the following:
   a. Enter a name and optional description.
   b. Select the **Select to configure multiple resources or verbs (maximum 11)** check box.
3. On the Resource Configuration page, specify the operation name, the endpoint relative resource URI, and the action/verb to perform, then click **Next**.

Copyright © 2022, 2026, Oracle and/or its affiliates.

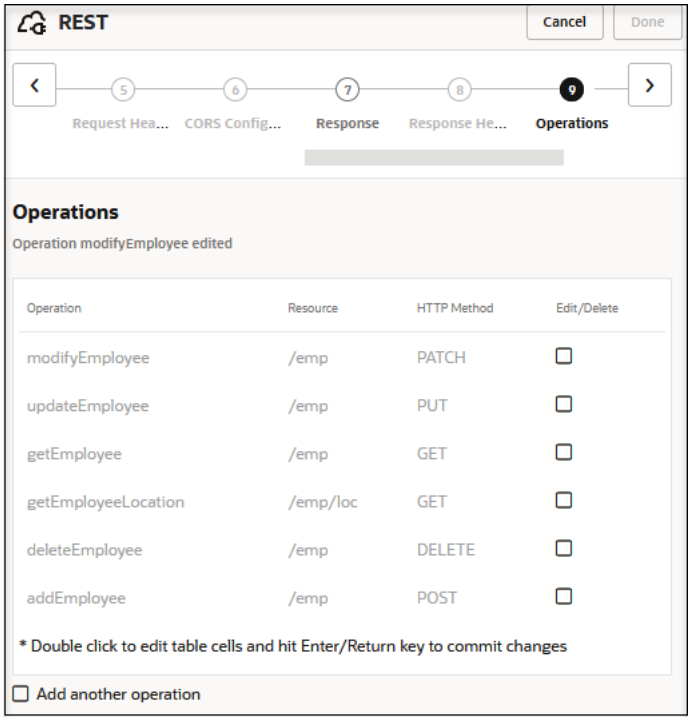


4. On the Operations page, click **Add another operation** to add another resource URI and action/verb, then click **Next**.

   You are returned to the Resource Configuration page.

5. On the Resource Configuration page, specify a different operation name, endpoint relative resource URI, and action/verb to perform, then click **Next**.
6. Continue this process to add as many operations as necessary.
7. On the Operations page, note that all operations are displayed. Each operation represents a different branch in a single integration.

Copyright © 2022, 2026, Oracle and/or its affiliates.

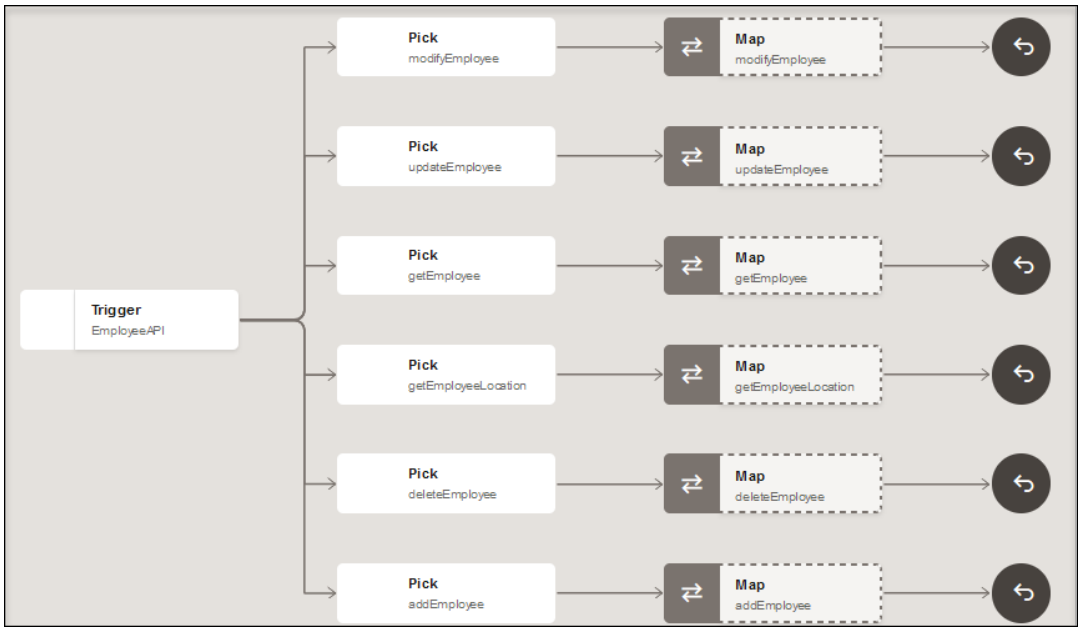


8. On the Summary page, review your selections, and click **Done**.

   For this example, there are six branches (**modifyEmployee**, **updateEmployee**, **getEmployee**, **getEmployeeLocation**, **deleteEmployee**, and **addEmployee**).

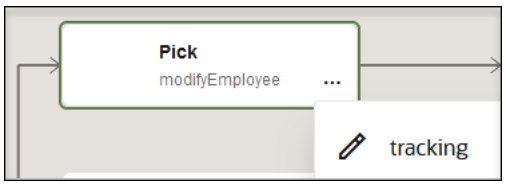
9. Place your cursor over each branch to show details about the operation/resource name, action/verb, and resource URI.

10. Further configure your integration by dragging appropriate actions *after* each operation name. You cannot add actions between the REST Adapter icon and the operation name.



Copyright © 2022, 2026, Oracle and/or its affiliates.

11. Select **Actions** ⋮, then **tracking** for the pick action operation name for which to define business identifier tracking variables.



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

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

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

Add Global Fault Handling to Integrations

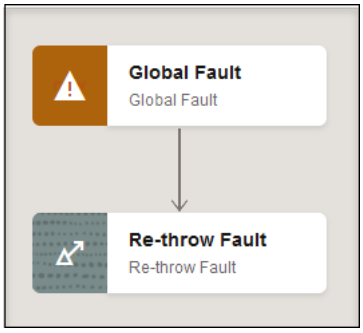
You can add global fault handling to integrations. This functionality enables you to direct business faults back to the caller or apply business logic before sending faults to the error handling framework. You can add fault handling to any integration type (for example, asynchronous, synchronous, or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

Add a Global Fault

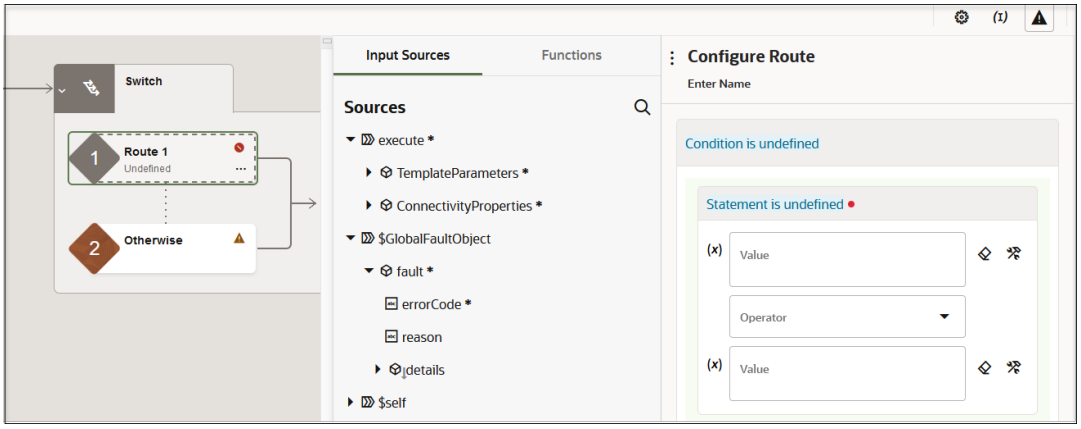
1. Design an integration. The integration does not need to be complete. You can add fault handling at any time. However, the integration must include an invoke connection.
2. In the integration canvas, click **View Global Fault Handler**  .

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

Copyright © 2022, 2026, Oracle and/or its affiliates.



3. Click **Actions** to add actions to design specific fault handling logic in the integration. For example, you can add **Switch**, **Stop**, or additional actions, when needed.
4. To build an expression to capture fault handling information, select the action, then click ..., then **Edit** .



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

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

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

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

Copyright © 2022, 2026, Oracle and/or its affiliates.

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

7. Click **Save**.
8. Click **View Main Integration**  to return to edit mode in the integration canvas.
9. Complete the design of the integration.
10. Activate the integration.

Track the Status of a Global Fault During Runtime

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

1. In the navigation pane, click **Observability**, then **Instances**.
2. On the Instances page, 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.

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

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. 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:

     ```
     Answer
     ```

     .

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

     Or, to select all elements:

     ```
     /*
     ```

   - Axis expression:

Copyright © 2022, 2026, Oracle and/or its affiliates.

For this example,

```
descendant
```

is entered to select all descendants (child, grandchildren, and so on) of the current node. Any descendant with this ID and namespace:

```
mb_v1_3:ID
```

is retrieved by the expression.

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

Axis expressions adhere to the following syntax:

axisname`::`*nodetest*`[`*predicate*`]`

The following table provides examples of axis expressions:

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

Use Lookups in Variable Assignments

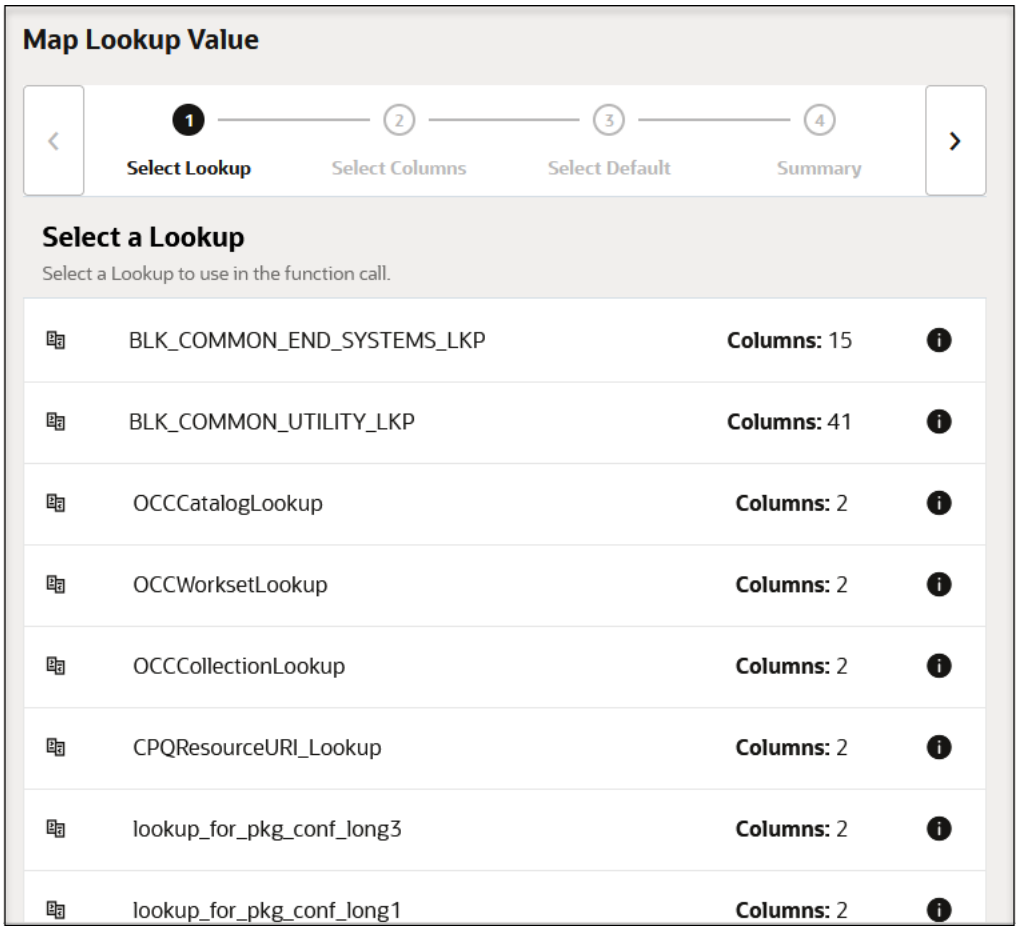
You can create variable assignments that use lookups in the Expression Builder. You must have already created the lookup.

1. Create a lookup in Oracle Integration. See Manage Lookups.
2. Drag an **Assign** action to the **+** sign. You can also create variable assignments that use lookups in other actions that use functions (for example, a data stitch action).
3. Enter a name and optional description.
4. Select **String** from the **Datatypes** drop-down list.

Copyright © 2022, 2026, Oracle and/or its affiliates.

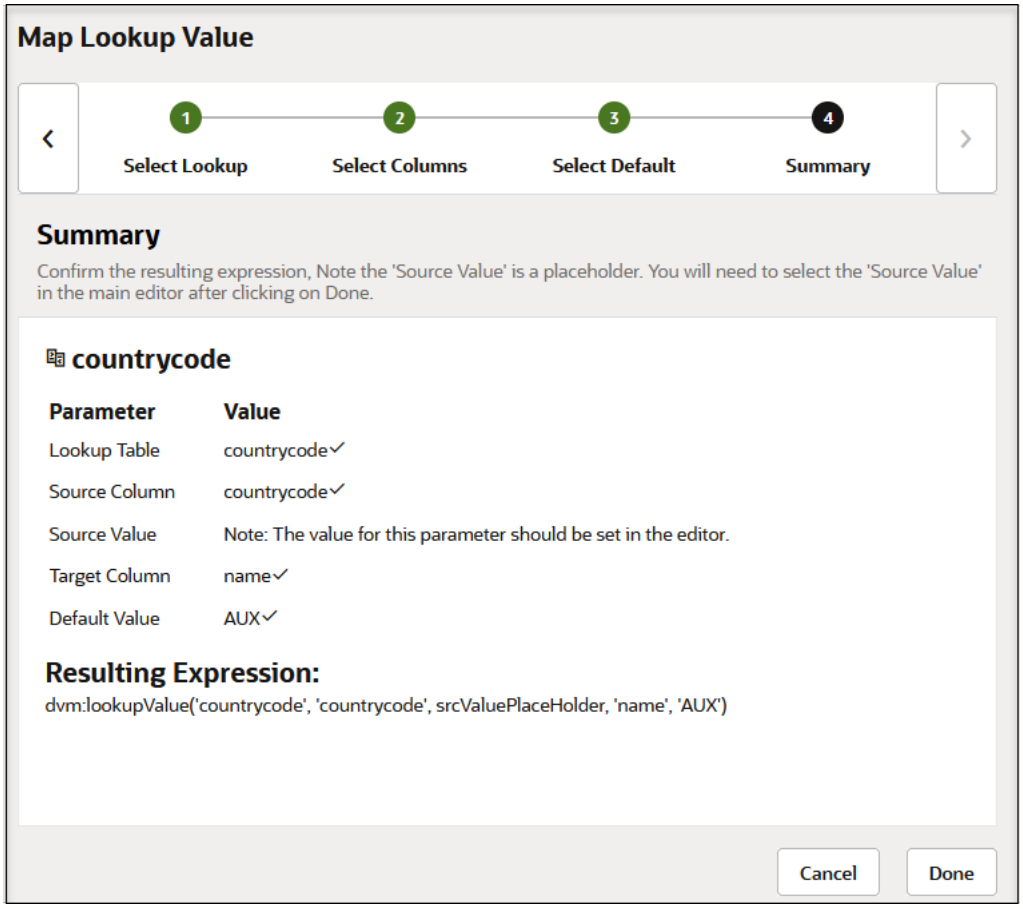
5. Click **Switch to Developer Mode** 
6. Click the **Functions** tab in the **Sources** tree.
7. Drag the **lookupValue** function to the **Value** field. You can use search to access this function.

   This starts the Map Lookup Value Wizard.



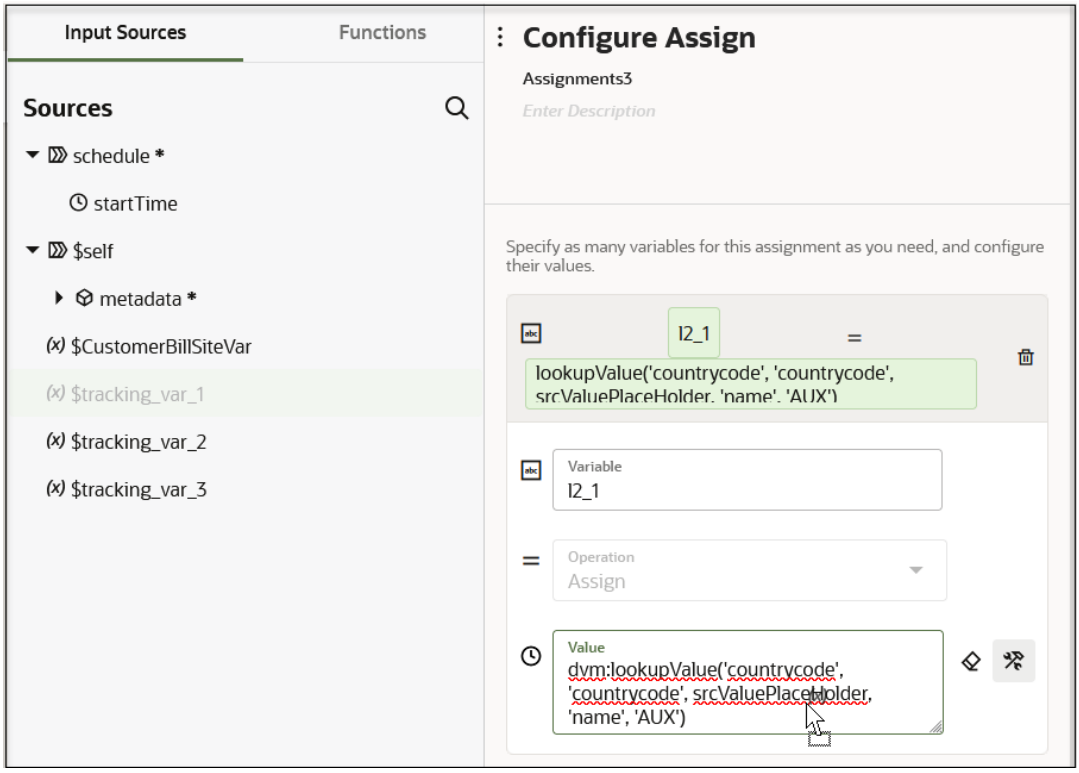
8. Select the lookup to use, and click **Next >**.
9. Select the source and target columns, and click **Next >**.
10. Enter a default value, and click **Next >**.
11. Review your selections on the Summary page and note the **srcValuePlaceHolder** value to replace in the expression, then click **>**.

Copyright © 2022, 2026, Oracle and/or its affiliates.



12. Click the **Input Sources** tab.
13. In the **Sources** tree, drag a value (for this example, **$tracking_var_1**) on top of the **srcValuePlaceHolder** value.

Copyright © 2022, 2026, Oracle and/or its affiliates.



The expression appears as follows:



14. Click **Save**.
Lookup functions are available for use in the target tree of the mapper.

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 schedule 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 schedule integrations.

Two fault aggregation operations are available for selection in schedule integrations:

- **append**: Adds a new fault to the end of a fault list.

Copyright © 2022, 2026, Oracle and/or its affiliates.

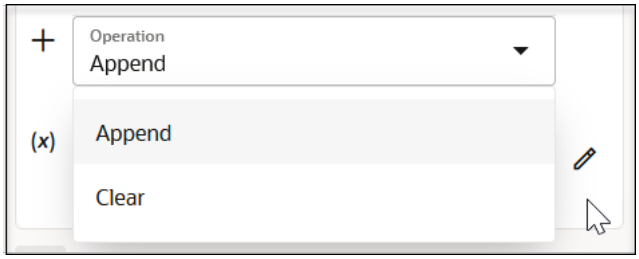
- **clear**: Removes all entries in a fault list.

These options are available for selection under the following scenarios:

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

1. Create a schedule 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 **Datatypes** list.

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



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

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

Copyright © 2022, 2026, Oracle and/or its affiliates.

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
- Maintain Business Identifier Definitions During Trigger Connection/Schedule Integration Parameter Changes

Assign Business Identifiers

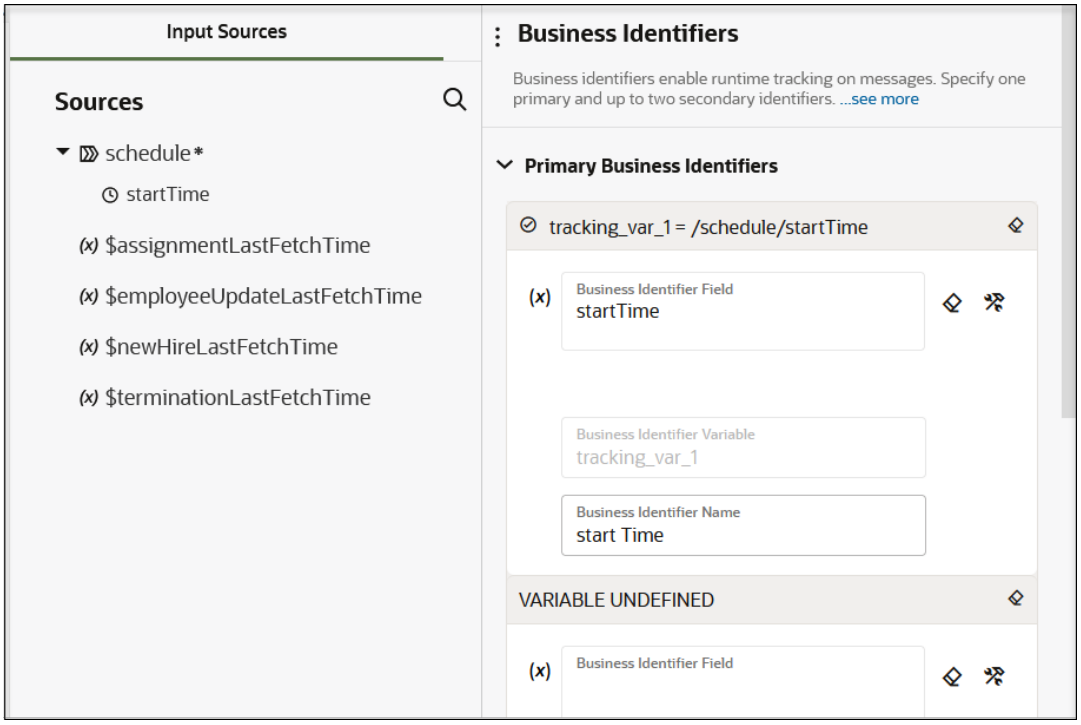
Business identifiers enable you to track payload fields in messages during runtime. You can specify up to three business identifier fields for tracking during design time. One of these fields must be selected as the primary business identifier field. The primary business identifier enables you to track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

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

1. In the navigation pane, click **Design**, then **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. Click **Business Identifiers** (I).

   The Business Identifiers panel 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 **Sources** tree, drag the payload field that you want to track to **Business Identifier Field**. At least one identifier is required. If you only add one, it is automatically selected as the primary key. A check mark to the left of the name indicates that this is the primary business identifier. If you created scheduled parameters, they are available for assignment as a primary or secondary business identifier.

Copyright © 2022, 2026, Oracle and/or its affiliates.



5. Drag additional business identifiers to the fields, when needed.

Delete Business Identifiers

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

1. In the navigation pane, click **Design**, then **Integrations**.
2. Click the specific integration in which to delete a business identifier. You can only delete business identifiers in integrations that are *not* active. If an integration is active, you can view, but not delete, the contents of the Business Identifiers panel.
3. Click **Business Identifiers** (I).

   The Business Identifiers panel is displayed.
4. For the business identifier to delete, click **Clear** . 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 Instances page.

Maintain Business Identifier Definitions During Trigger Connection/Schedule Integration Parameter Changes

Beginning with new integrations created in Release 23.08, when a trigger connection changes or a schedule integration parameter is deleted, business identifier definitions are not removed. Instead, the definitions are retained and validated based on the changes.

The following table describes business identifier definition status based on the release in which the integration was created.

Copyright © 2022, 2026, Oracle and/or its affiliates.

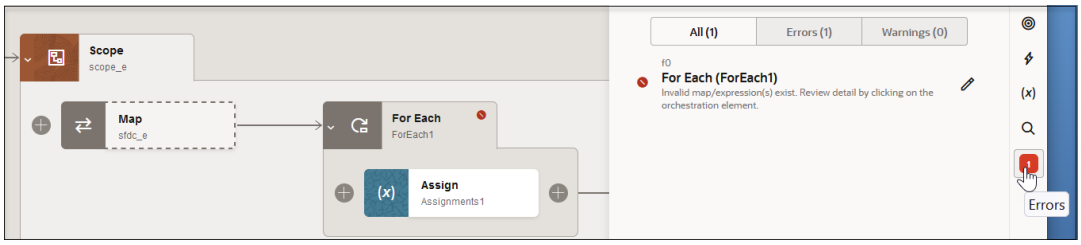
If You ...	Existing Integrations Created Prior to Release 23.08	New Integrations Created in Release 23.08 and Later
Edit a trigger connection in the Adapter Endpoint Configuration Wizard and either make changes or simply click through the wizard pages without making changes.	The business identifier is removed. You must redefine the business identifier. After that, the business identifier behavior described for new integrations created in Release 23.08 is followed.	The business identifier is retained and validated. If you remove a parameter used by the business identifier (for example, a query parameter in the wizard that is used as the business identifier definition), a validation error is displayed in the Errors panel: **`Primary business identifier`** `Invalid tracking variables definition` Re-define the business identifier to resolve the error.
Delete a scheduling parameter used by the business identifier.	Validation is performed and an error is displayed in the Errors panel.	Validation is performed and an error is displayed in the Errors panel.

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 side. These errors and warnings prevent you from activating an integration. You must first resolve these issues to activate an integration.

1. Design an integration.

   If there are errors or warnings, an **Errors** section is displayed on the side.



2. Click the **Errors** section to display error and warning details. For this example, a for-each action has an invalid mapping/expression.
3. Click **Edit** to access the page to edit.
4. Resolve the issue. Once successfully resolved, the **ERRORS** section disappears.

   You can now activate the integration.

Copyright © 2022, 2026, Oracle and/or its affiliates.

8

Create Schedule Integrations

You can schedule the running of integrations. For example, you can create an integration that is triggered by a schedule in which an FTP Adapter reads a file and another 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 and run ad-hoc, schedule integrations. When you create a schedule integration, a schedule icon is displayed with the integration entry on the Integrations page.

Topics:

- Create a Schedule Integration
- Define the Integration Schedule
- Create and Update Parameters in Schedule Integrations
- Run a Schedule Integration Immediately
- Start and Pause an Integration Schedule
- Change the User Submitting an Integration Schedule
- Globally Change the Submitter of Integration Schedules
- View Past and Present Schedule Integration Runs
- View Future Schedule Integration Runs
- Edit an Integration Schedule
- Delete an Integration Schedule
- Unlock an Integration Schedule
- Convert a Schedule Integration to a REST Adapter-Triggered, Application Integration
- Delete Scheduled or Ad-Hoc Runs
- Monitor Schedule Integration Runs
- Understand Schedule Integration Behavior When Using the Local Time Zone

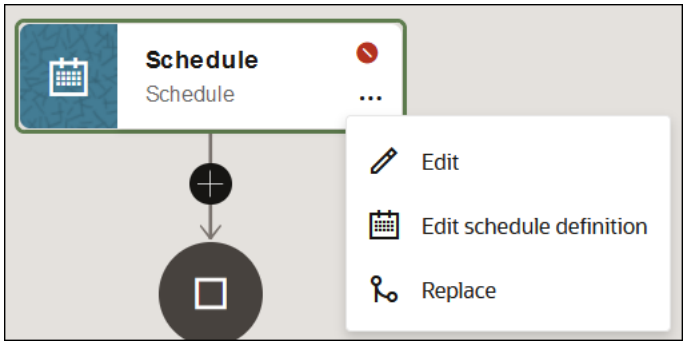
See *Understand the FTP Adapter* in *Using the FTP Adapter with Oracle Integration 3*.

Create a Schedule Integration

This section describes how to create a schedule integration.

1. Follow the steps in Create an Integration to create a schedule integration.
   An empty integration canvas is displayed with a schedule icon at the top.
2. To define the schedule for running the integration, click **Actions** ⋮, then select **Edit schedule definition**.

Copyright © 2022, 2026, Oracle and/or its affiliates.



See Define the Integration Schedule. You can also define a schedule on an integration at a later time from the Integrations page by clicking **Actions** ⋮, then selecting **Add schedule**.

3. See Design an Application Integration to design the entire flow of your schedule integration and Assign Business Identifiers for Tracking Fields in Messages to assign business identifiers. Schedule integration design differs from application integration design only in that you define an adapter trigger connection in an application integration. In schedule integrations, the schedule is the trigger connection in the integration.

Define the Integration Schedule

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

- Decide When to Create the Integration Schedule
- Define the Type of Integration Schedule to Run
- Select When to Run the Integration Schedule

Decide When to Create the Integration Schedule

You can decide to first design and activate an integration and then create the schedule later or first create a schedule and then design and activate the integration later.

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

1. In the navigation pane, click **Design**, then **Integrations**.
2. Find the schedule integration in the Integrations list.

   These integrations are identified by **Schedule integration**  . There are several ways in which to create a schedule on an integration.
3. If you want to first design and activate the integration and then create the schedule:

Copyright © 2022, 2026, Oracle and/or its affiliates.

a. Hover over the integration.

b. Click **Activate** .

The Activate Integration dialog is displayed.

c. Click **Activate And Schedule**.

4. If you want to first create a schedule for an integration that you design and activate later:

a. Hover over the integration on which to create a schedule.

b. Click **Actions** , then select **Add Schedule**.

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

The page for defining the schedule processing details is displayed.

> **Note**
>
> - Assume you create a schedule and generate several integration runs, then deactivate and delete the integration. If you then create and activate a new integration with the same name as the deleted integration, and go to the Instances page, the past runs of the deleted integration are displayed. This is by design.
> - Deactivation of an active, schedule integration is not allowed. You must stop the schedule, then retry.

Define the Type of Integration Schedule to Run

You define a schedule for running integrations, such as a basic schedule, a reoccurring AI-generated iCal schedule created by entering natural language text, or a reoccurring schedule created by directly entering iCal parameters.

- Understand How to Use iCal Expressions
- Create a Basic Integration Schedule
- Create an AI-Generated iCal Schedule by Entering Natural Language Text
- Create a Schedule by Directly Entering iCal Parameters

Understand How to Use iCal Expressions

- Oracle Integration allows up to five recurrences in an iCal expression when defining the schedule.
- There is a one-minute limitation on how frequently you can run schedule integrations with an iCal expression. Anything under this limit is not supported.
- You cannot use lookups in iCal expressions.
- To ensure that a schedule starts at the exact time specified (for example, `14:00` and `15:00` hours), specify `BYMINUTE=0` in the iCal expression (for example, `…;BYHOUR=14,15;BYMINUTE=0;`). If `BYMINUTE` is not specified, other factors determine the

Copyright © 2022, 2026, Oracle and/or its affiliates.

scheduled minute (either the start date specified at schedule creation or the actual time the schedule starts).

Create a Basic Integration Schedule

1. Click **Simple**. For this type, there is a minimum frequency limit of ten minutes. If you try to define a schedule frequency of under ten minutes, a validation error occurs.
2. In the **Frequency** section for a **Simple** schedule, click the list to select the frequency with which to run the integration. When you define one frequency, you can specify additional values by clicking the icon to the right of the **Frequency** section.
3. Select a frequency.
   - **Only Once**: This is the default selection.

     If you select this option, you cannot select **When schedule starts** as the start date. This option is disabled.
   - **Hours and Minutes**: Specify the hour and minute 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.

Create an AI-Generated iCal Schedule by Entering Natural Language Text

You can define an AI-generated, reoccurring schedule by entering natural language. This eliminates the need to manually enter an iCal expression completely from scratch. See AI Innovation and Oracle Integration.

1. Click **iCal**.
2. Enter a reoccurring schedule clearly and explicitly in English, then click **Validate** to generate the equivalent iCal expression. For example:

Entering the Following Natural Language Text ...	Generates This iCal Expression ...
Run the schedule every two weeks on Friday at 11 pm	`FREQ=WEEKLY;INTERVAL=2;BYDAY=FR;BYHOUR=23;`
Run 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;BYMONTHDAY=1,10,15;BYHOUR=5,10,15,20;BYMINUTE=15;`

Run daily at 530 PM every ten minutes. Run daily at 600 PM every ten minutes. Run daily at 7 pm every ten minutes until 730 PM.	`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;`

3. If validation is successful, view the following message that appears at the top: `iCal expression is valid.`

Copyright © 2022, 2026, Oracle and/or its affiliates.

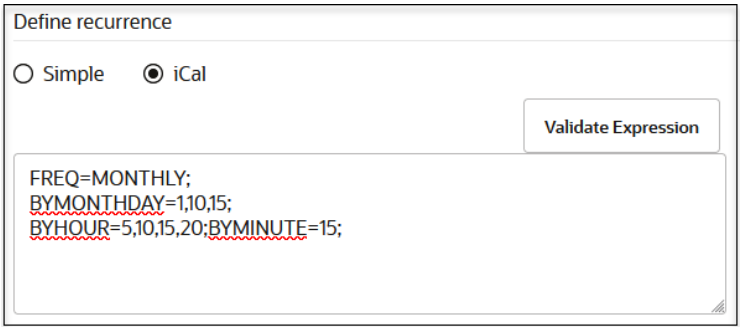
4. Review the generated iCal expression carefully. The expression may be close to what you want, but may still require updating. Copy the generated iCal expression to the text area to update further, as needed. When complete, click **Validate** again.

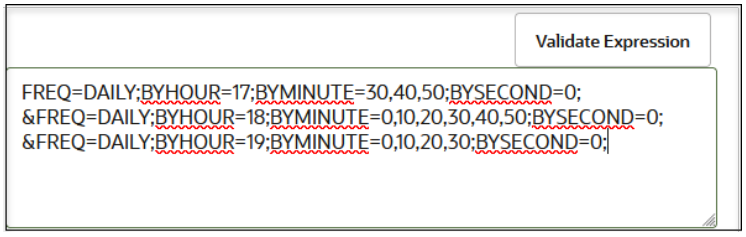
Create a Schedule by Directly Entering iCal Parameters

If you are comfortable with creating iCal expressions completely from scratch, you can use this approach.

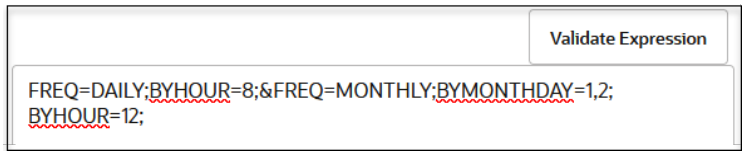
1. Click **iCal**.
2. Enter an iCal expression, and click **Validate**. For example:

Expression	Syntax

This 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;`
	`BYMONTHDAY=1,10,15;`
	`BYHOUR=5,10,15,20;BYMINUTE=15;`
	 

This expression defines multiple schedule frequencies. The schedule runs every day between the hours of 5:30 PM – 7:30 PM, and during these hours it processes 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;BYS`
	`ECOND=0;`
	`&FREQ=DAILY;BYHOUR=19;BYMINUTE=0,10,20,30;BYSECOND=`
	`0;`
	 

Copyright © 2022, 2026, Oracle and/or its affiliates.

Expression	Syntax
This expression runs daily at 8 AM and also monthly at 12 PM on day 1 and day 2.	`FREQ=DAILY;BYHOUR=8;&FREQ=MONTHLY;BYMONTHDAY=1,2;BYHOUR=12;` 

3. If validation is successful, view the following message that is displayed at the top: `iCal expression is valid.`
4. If an AI engine is configured, more human-readable text is shown for the given iCal expression. This is meant to enhance understanding of the iCal expression you entered.

Select When to Run the Integration Schedule

You can define whether the schedule run never expires, whether the schedule has a fixed expiration date, and so on.

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

   A menu is displayed for defining the start date of the schedule.
2. If you want to start the integration run when the schedule is activated, click **When schedule starts**.
3. 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 **Done**.
4. In the **Until** section, click the **Never (repeat indefinitely)** link to the right.

   A menu is displayed for defining the expiration date.
5. If you want the schedule run to never expire, select **Never (repeat indefinitely)**.
6. 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 **Done**.
7. From the **Time zone** list, select a time zone in which to run the schedule. By default, this field shows the value you selected on the Preferences page.
8. Click **Save**.

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

   `Schedule Run` *`name`* `saved successfully`

Copyright © 2022, 2026, Oracle and/or its affiliates.

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

9. Return to the Schedule and future runs page.

   Details about the integration run schedule are displayed. There are two sections:

   - Schedule section: This section is collapsed by default. Clicking the top bar (showing schedule name and buttons) expands the section and shows the schedule.



   - Future Instances table: This table remains empty until the schedule is started. Once it starts, you see data.



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

    a. Click **Actions** ☰, then select **Run**.

       The Configure and run page is displayed.
       Clicking **Run** triggers the integration immediately.

    b. If you are a user with the *Service*Administrator role, you can change the user with which to run the schedule in the **Run as** field.

    c. Select how to initiate an instance of the integration when prompted.

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

    d. Click **Run**.

    or

    a. Select **Start Schedule** ▶ on the Schedule and future runs page to activate the integration schedule.

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

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

Copyright © 2022, 2026, Oracle and/or its affiliates.

> **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 schedule 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.

11. If you have not yet activated the integration, return to the Integrations page and click **Activate**.
12. Run the schedule integration as described in the previous steps.

Create and Update Parameters in Schedule Integrations

You can create and update scalar-type scheduled parameters in schedule 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 schedule integrations. You can also assign values to these parameters in assign actions. Up to five parameters are supported in the Configure schedule parameters panel.

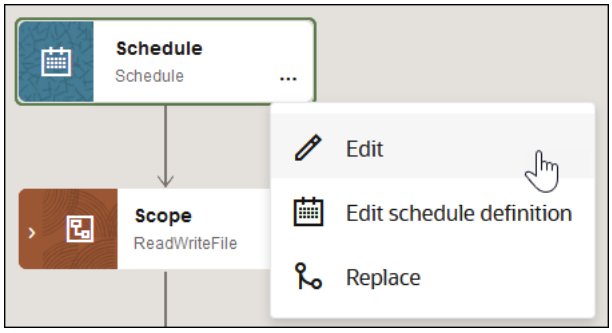
Create Parameters

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

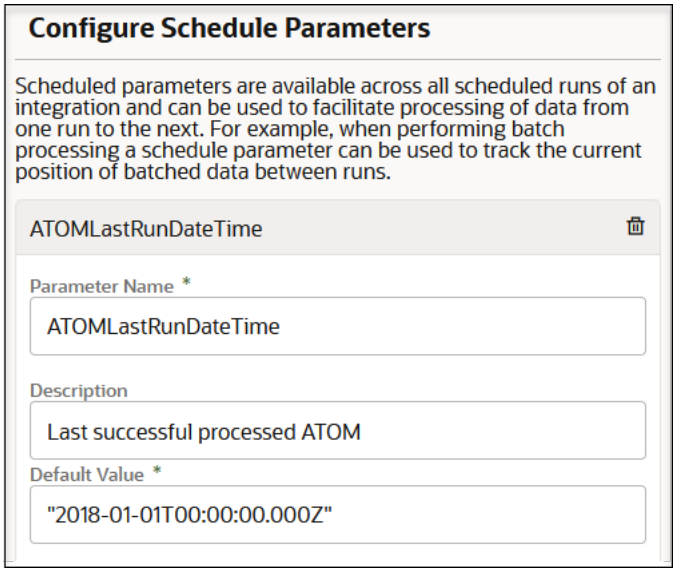
Scheduled parameters are available across all scheduled runs of an integration and can facilitate processing of data from one run to the next. For example, when performing batch processing, a scheduled parameter can track the current position of batch data between runs. This value is available across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

1. In the navigation pane, click **Design**, then **Integrations**.
2. Click **Create**.
3. Select the **Schedule** pattern.
4. Complete the fields of the dialog, and click **Create**.
   The integration canvas is displayed.
5. For **Schedule**, click ⋯, then select **Edit**.

Copyright © 2022, 2026, Oracle and/or its affiliates.

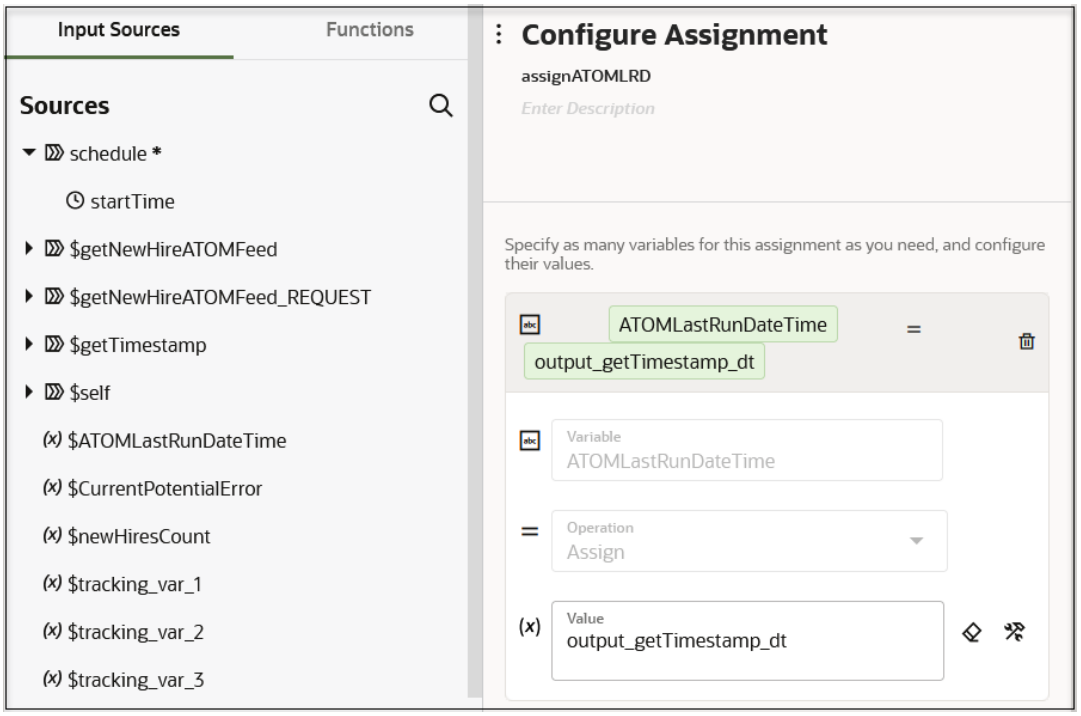


6. Click the field to open fields for entering a parameter name, description, and default value.
7. Enter values. For this example, a parameter of `ATOMLastRunDateTime` and a value of `"2018-01-01T00:00:00.000Z"` are entered.



8. Click **Save**.
   If you enter a parameter without a value, a red warning icon is displayed after the **Default Value** field.
9. Drag an **Assign** action into the integration.
10. Drag the parameter you created in the Configure schedule parameters panel into the **Variable** field and specify a value.

Copyright © 2022, 2026, Oracle and/or its affiliates.

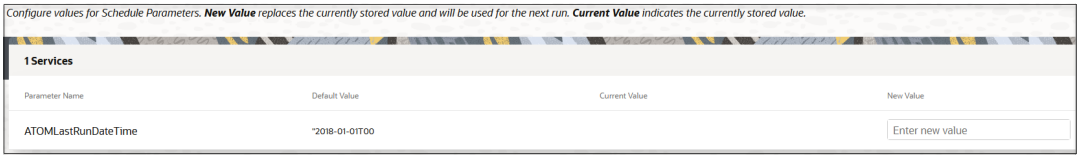


11. Click **Save**.

> **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 schedule 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 cut and paste options to move the assign action outside of the for-each loop.

Update Parameters

1. In the navigation pane, click **Design**, then **Integrations**.
2. Hover over the integration on which the scheduled run is defined.
3. Click **Actions** ⋯, then select **Schedule**.
   The Schedule and future runs page is displayed.
4. Click **Actions** ☰ and select **Update schedule parameters**.
5. Edit the parameter value in the **New value** column, and click **Update**. The value in the **New value** field replaces the currently stored value and is used for the next run. The value in the **Current value** field indicates the currently stored value.



Copyright © 2022, 2026, Oracle and/or its affiliates.

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.

1. Access the dialog to override parameter values in either of two ways. If the schedule integration has defined parameters, the Configure schedule parameters panel is displayed. Otherwise, the open is not displayed and the schedule integration starts.

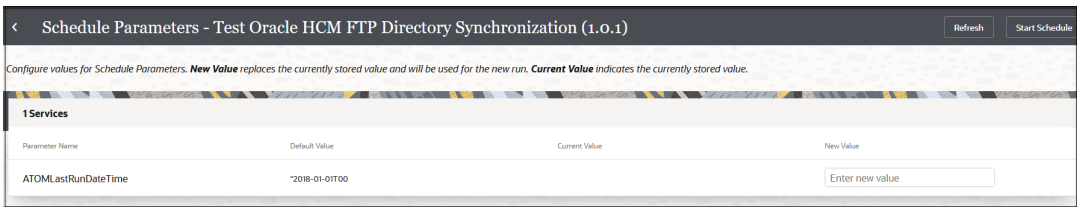
   a. If submitting an ad-hoc schedule integration, select **Run** from one of the following locations:

      - Click **Actions** ☰, then **Run** on the Schedule and future runs page.
      - Click **Actions** ⋮, then **Run** on the Monitor integrations page.

      Both options invoke the Configure and run page, from which you can select **Ad hoc request** and click **Run**.

   b. If scheduling an integration, select **Start schedule** ▶ on the Schedule and future runs page.

   The Schedule parameters page is displayed.



2. Make updates to any necessary parameters.

   The following fields are displayed:

   - **Parameter name**: Displays the parameter name you provided during design-time.
   - **Default value**: Displays the parameter value you assigned during design-time.
   - **Current value**: Displays the value most recently used at runtime.
   - **New value**: Optionally enter new values for the schedule parameters. These values override the current and default values. If these fields remain empty, the current values are used. If there are no new or current values, the default values defined during design-time are 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 values are saved and become the current values for the next schedule run.

4. Save your updates.

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

   b. If submitting an ad-hoc schedule integration, click **Run**.

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

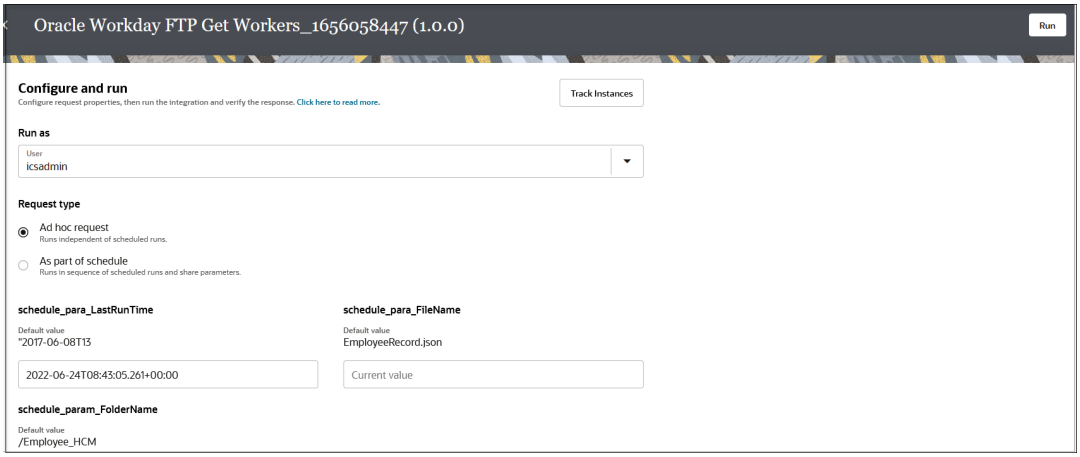
Copyright © 2022, 2026, Oracle and/or its affiliates.

Run a Schedule Integration Immediately

You can run a schedule integration immediately, known as an ad-hoc request.

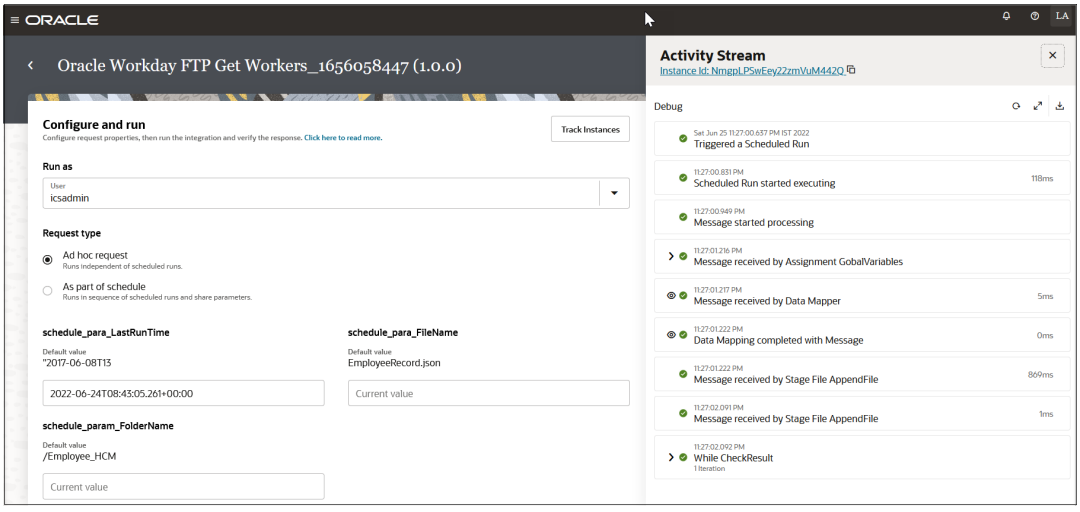
1. In the navigation pane, click **Design**, then **Integrations**.
2. Hover over the row of the schedule integration to run.
3. Click **Actions** ⋮, then select **Run**.

   The Configure and run page is displayed.



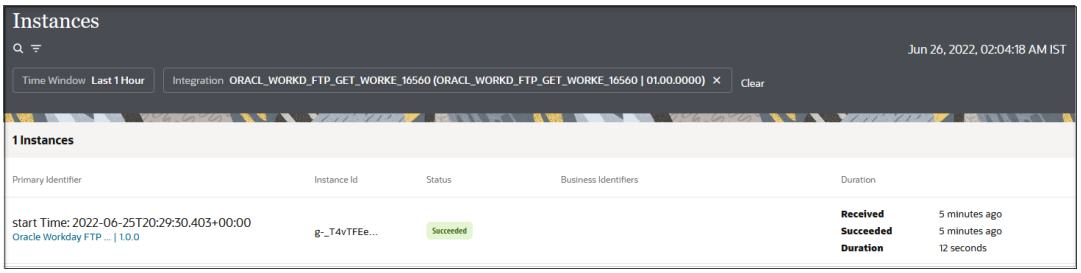
4. If you are a user with the *Service*Administrator role, you can change the user with which to submit the schedule.
5. Ensure **Ad hoc request** is selected. This is the default selection.
6. If any schedule parameters are defined for this integration, they are displayed after. Accept the default value or enter a different value to use for this run in the **Current value** field. Otherwise, this section is empty.
7. Click **Run**.

   The Activity Stream panel opens on the right to show the progress of the run.



Copyright © 2022, 2026, Oracle and/or its affiliates.

8. Click **View details** to view specific details about the movement of the message payload in the activity stream. See Track the Status of Integration Instances.
9. Click **Track instances** to access the Instances page filtered to show the completed instance.



Based on the **Status** column value (for example, **Queued**, **Succeeded**, and others), the **Duration** column provides additional status details about the schedule instance. See View the Status of Future Schedule Integrations.

Start and Pause an Integration Schedule

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

What Happens When I Pause or Stop Scheduled Integration Runs

When you pause a schedule, the start of the execution of the next scheduled run is paused. For example, assume you have the following scheduled integration runs:

- 2 PM
- 3 PM
- 4 PM
- 5 PM

If you pause the schedule after the 3 PM run completes, the execution of the 4 PM run is paused. If you then unpause the scheduled run at 4:30 PM, the 4 PM run is executed.

When you stop a schedule, all runs are cancelled.

Start and Pause an Integration Schedule

1. In the navigation pane, click **Design**, then **Integrations**.
2. Hover over the row of the integration on which the scheduled run is defined.
3. Click **Actions** ..., then select **Schedule**.
   The Schedule and future runs page is displayed.
4. Click **Start** .
5. If you are a user with the *Service*Administrator role, you can change the user with which to submit the schedule.
6. Click **Confirm** to start the schedule.

Copyright © 2022, 2026, Oracle and/or its affiliates.

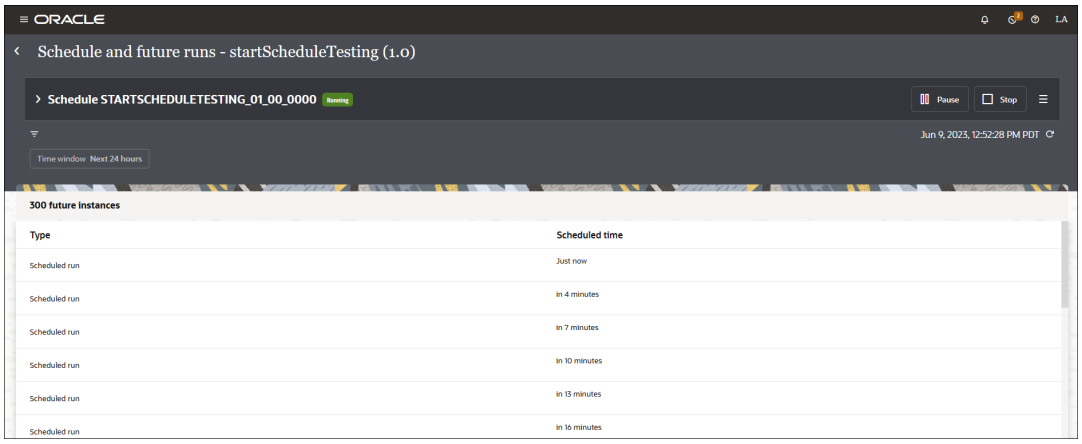
If you have defined scheduled parameters on this integration, the Schedule parameters page is displayed.

7. Make any necessary parameter value updates in the **New value** field, and click **Start Schedule**.

   The **Start** button is changed to **Stop**.

> **Note**
>
> If you want to pause the schedule run, click **Pause** in the upper right corner, then click **Confirm** when prompted.

8. Click **Refresh** to refresh the status of the schedule integration.



The **Pause** and **Stop** buttons are replaced with **Retry** and **Refresh** buttons if the schedule operation doesn't complete immediately after submission. In these cases, the status is displayed as **Stopping**, **Starting**, **Pausing**, or **Resuming**. The **Retry** option can be used to recover from these states. There must be a gap of 10 minutes between two retry attempts.

For example:

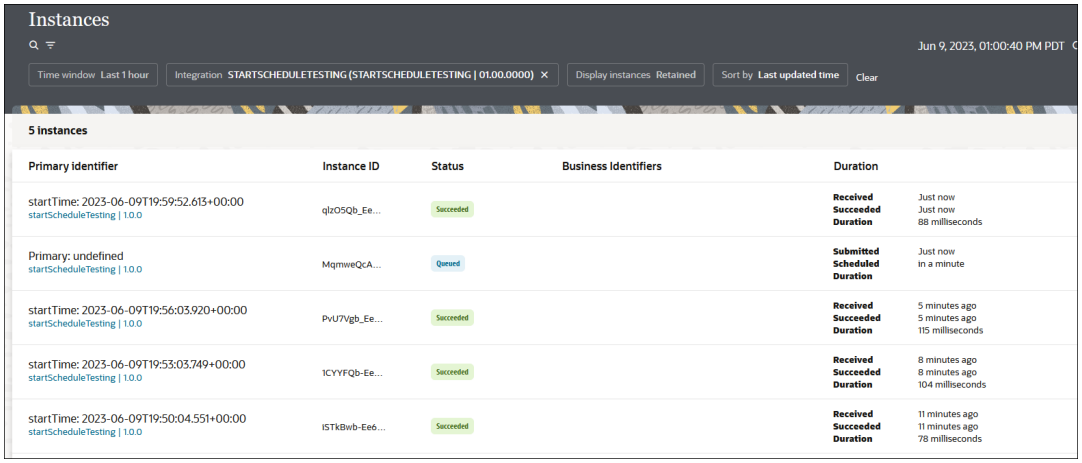


or



9. Click **Schedule menu**, then select **Track instances**.
   The Instances page shows details about the run of the scheduled instance.

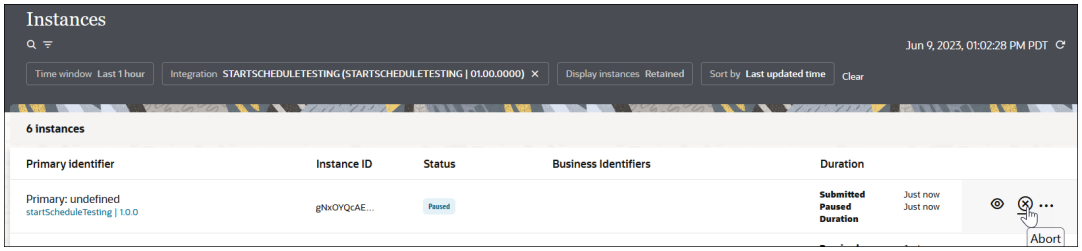
Copyright © 2022, 2026, Oracle and/or its affiliates.



The following status values are displayed during the course of the schedule integration life cycle:

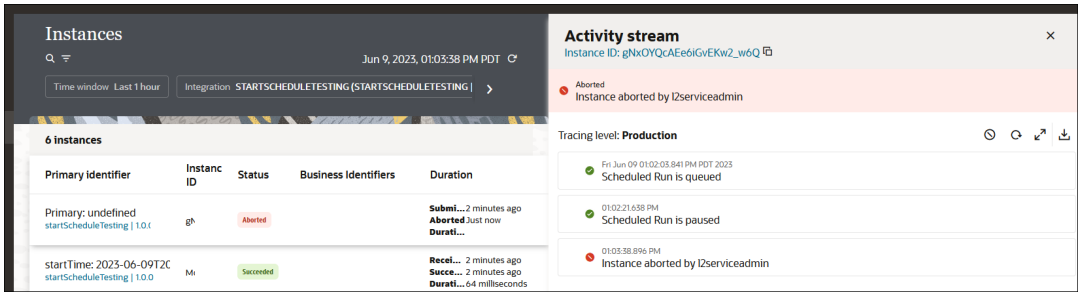
- Queued (Typically the state of an instance when it is scheduled to process sometime in the future.)
- Succeeded
- Errored
- Aborted
- In progress
- Queued
- Paused
  The schedule has been paused, meaning the instance is also paused. Unpausing the schedule moves the instance to different states:
  - If the next scheduled run is in the future, the instance moves to the waiting state.
  - If the scheduled start time has passed, the instance runs immediately.
- Blocked
- Resubmitted

10. If necessary, schedule integration instances in the queued or paused state can be canceled by selecting **Abort**.



The activity stream provides details about the cancellation.

Copyright © 2022, 2026, Oracle and/or its affiliates.



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

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

11. Click **View** to view schedule milestones such as the following in the activity stream.
    - Scheduled run is triggered: Indicates the time at which the request to trigger the schedule integration was submitted.
    - Scheduled run started processing: Indicates the time at which the submitted request started processing.
    - Schedule paused: Indicates the time at which a schedule was paused.
    - Schedule resumed: Indicates the time at which the schedule was resumed (from a paused state).
    - Processing completed successfully: Indicates the time at which schedule processing completed successfully.

Change the User Submitting an Integration Schedule

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

1. Go to the Schedule and future runs page. See Define the Integration Schedule.
2. Click **Actions**, then **Update submitter** to update the user that can submit the schedule.

   The Update submitter dialog is displayed.
3. Change the submitter of the schedule in the drop-down list.

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

   Details about the user who submitted the schedule for the integration are visible in the design-time audit logs on the Dashboard page. For example:

   ```
   User icsadmin triggered integration integration_name as user icsdeveloper
   ```

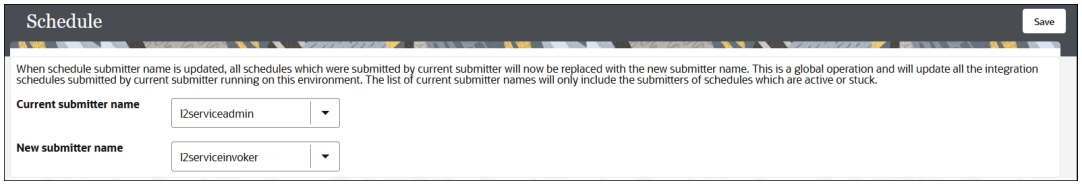

Copyright © 2022, 2026, Oracle and/or its affiliates.

Child (co-located) integrations invoked from a parent schedule integration inherit the same user that submitted the schedule. See Invoke a Child Integration from a Parent Integration.

Globally Change the Submitter of Integration Schedules

You can globally change all schedules submitted by the current user to another user. The list of current submitter names only include submitters of those schedules that are currently active or stuck. Only a user with the *Service*Administrator role can perform this task.

1. In the navigation pane, click **Settings**, then **Schedule**.
2. From the **Current submitter name** list, select the submitter to replace.
3. From the **New submitter name** list, select the new submitter.



For this example, any schedules that the old **l2gen3serviceadmin** user submitted can now be submitted by the new **l2gen3serviceinvoker** user.

4. Click **Save**.

View Past and Present Schedule Integration Runs

You can view the status of past and present schedule integration runs.

1. In the navigation pane, click **Design**, then **Integrations**.
2. Go to the row of the integration on which the scheduled run is defined.
3. Click **Actions**, then select **Schedule**.

   The Schedule and future runs page is displayed.
4. Click **Actions**, then select **Track instances**.

   The Instances page is displayed. By default, all completed integration runs are displayed.
5. Click **Filter** to filter the display of integration runs.
6. Click the file name of the instance.

   A graphical view of the integration flow is displayed.

   For example, if the resubmission resulted in a failure, details are displayed in the activity stream.

View Future Schedule Integration Runs

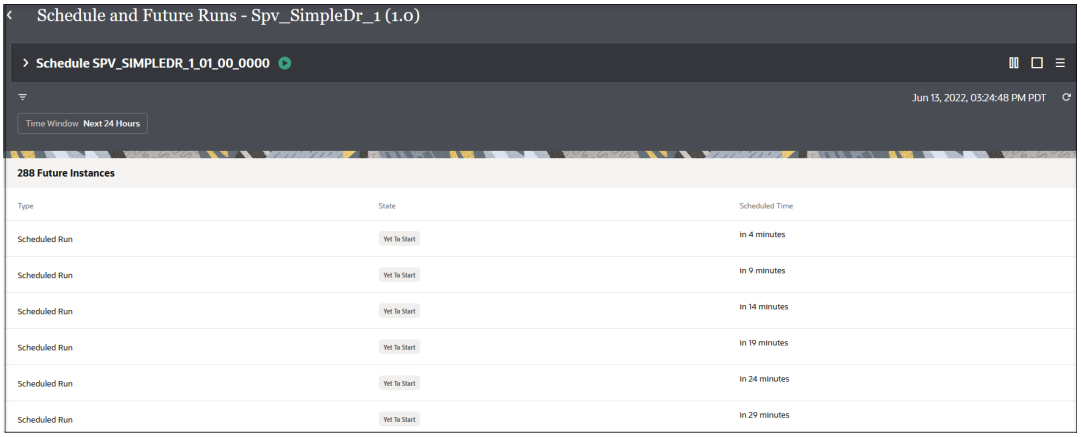
You can view the status of future schedule integration runs.

1. In the navigation pane, click **Design**, then **Integrations**.
2. Hover over the integration on which the scheduled run is defined.

Copyright © 2022, 2026, Oracle and/or its affiliates.

3. Click **Actions** ⋮, then select **Schedule**.

   The Schedule and future runs page is displayed.
   Details about scheduled future runs are displayed. You can filter the display of integration runs. The page only displays manually submitted runs with a state of blocked, queued, ready, paused, or canceling. The time zone in which the schedule was created is also displayed.



> **Note**
>
> If you select a different time zone in another browser session, the **Scheduled time** field continues to show the time zone in which the schedule was created. All future runs process per this time zone, and not the time zone you specified in another browser session.

Edit an Integration Schedule

You can edit a schedule for an integration run.

1. In the navigation pane, click **Design**, then **Integrations**.
2. Hover over the row of the integration on which the scheduled run is defined.
3. Click **Actions** ⋮, then select **Schedule**.

   The Schedule and future runs page is displayed.

4. Click **Edit** ✎.
5. Edit the schedule. See Define the Integration Schedule.
6. Click **Save**.

Delete an Integration Schedule

You can delete a schedule for an integration. This action deletes all future runs of the schedule and stops any currently running schedule integrations.

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

Copyright © 2022, 2026, Oracle and/or its affiliates.

2. Hover over the row of the integration on which the scheduler run is defined.
3. Click **Actions** ⋮, then select **Schedule**.

   The Schedule and future runs page is displayed.
4. Click **Schedule menu** ☰, then select **Delete schedule**.
5. Click **Confirm** when prompted. This deletes the schedule. Any scheduled jobs currently running are stopped.

Unlock an Integration Schedule

When a schedule is in edit mode and the browser fails, the schedule becomes locked, which prevents it from being edited. You can unlock an integration schedule to resume editing.

1. Log in as a user with the Administrator role.
2. In the navigation pane, click **Design**, then **Integrations**.
3. Click **Actions** ⋮, then select **Schedule**.

   The Schedule and future runs page is displayed.
4. Click **Actions** ⋮, then select **Unlock schedule**.
5. Click **Unlock** when prompted. Note that unlocking an integration schedule may cause data loss.

Convert a Schedule Integration to a REST Adapter-Triggered, Application Integration

You can convert a schedule integration to a REST Adapter-triggered, application integration. This eliminates the need to recreate the integration if you need to change the integration pattern from schedule to application.

Note the following restrictions:

- You cannot convert a schedule integration to a REST Adapter-triggered integration with multiple verbs.
- You cannot convert a schedule integration with assignments that have the fault data type.
- You cannot convert a schedule integration that is using a parallel for-each action. After conversion, the integration fails with the following error during activation:

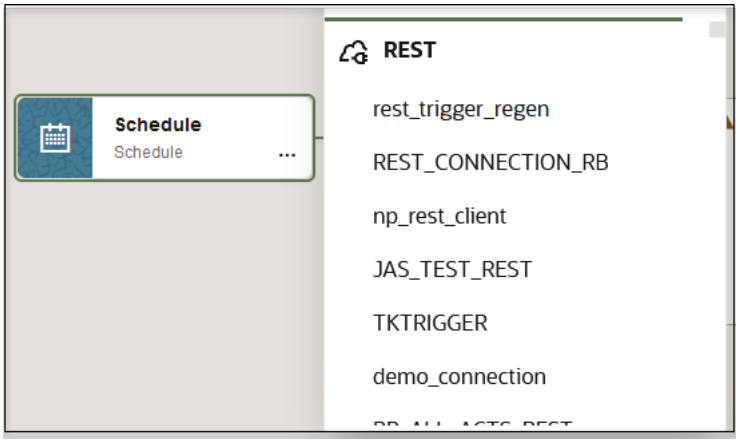
  ```
  Parallel For-Each inside non-scheduled orchestration is not supported.
  ```

1. In the navigation pane, click **Design**, then **Integrations**.
2. Open the schedule integration you want to convert.

   The integration canvas is displayed.
3. Click the Schedule icon, then click **Actions** ⋮, and then select **Replace**.

   A list of available REST Adapter triggers is displayed.

Copyright © 2022, 2026, Oracle and/or its affiliates.



4. Select the REST Adapter to use as the trigger connection in this integration.

   This invokes the Adapter Endpoint Configuration Wizard.

5. Configure the REST Adapter in the wizard.

> **Note**
>
> Do *not* select the **Select to configure multiple resources or verbs** check box on the Basic Info page. This feature is not supported when converting schedule integrations to application integrations.

6. When wizard configuration is complete, click **Done**.

   A message is displayed.

```
Clicking done will perform conversion to REST trigger
and will clear tracking information, delete any assignment variables
with fault data types and update any impacted actions. Errors and
warnings may occur which will require manual correction. Are you
sure you want to continue?
```

7. Review the message, then click **Convert**.

   A conversion progress bar is displayed. When complete, the converted integration is displayed in the canvas with the REST Adapter trigger connection.

8. Investigate any errors and manually correct the impacted actions or invokes. For example, any scheduled parameters in the schedule integration are deleted during conversion to an application integration. This results in errors in any actions that reference those schedule parameters (for example, an assign or logger action).

Delete Scheduled or Ad-Hoc 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:

- Queued
- Ready

Copyright © 2022, 2026, Oracle and/or its affiliates.

- Running
- Completed
- Blocked
- Hold
- Paused
- Schedule Ended
- Error Auto Retry
- Error Manual Recovery

1. In the navigation pane, click **Design**, then **Integrations**.
2. Hover over the integration on which the scheduled run is defined.
3. Click **Actions** ... and select **Schedule**.

   The Schedule and future runs page is displayed. The state of each run is displayed.
4. Click **Schedule menu** ☰, then select **Delete schedule**.
5. Click **Yes** when prompted to confirm your selection.

   The scheduled run is removed from the Schedule and future runs page.

Monitor Schedule Integration Runs

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

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

   For schedule integrations, the  icon is displayed.
2. If you want to resubmit a run of a schedule integration, click **Actions** ..., then select **Run**.
3. If you want to pause scheduled runs, click **Actions** ..., then select **Pause schedule**.

Understand Schedule Integration Behavior When Using the Local Time Zone

Understand how schedule integrations are impacted by use of the local time zone. This section provides several use cases.

Daylight Savings Time Begins

Assume daylight savings begins in your environment on March 1 (time is moved forward from 1 AM to 2 AM) and your integration is scheduled to run at 1:30 AM. In this scenario, the time range between 1 AM and 1:59 AM does *not* exist and the schedule integration does *not* run on March 1 at 1:30 AM.

Daylight Savings Time Ends

Assume daylight savings time ends in your environment on November 1 (the time is rolled back from 2 AM to 1 AM) and your integration is scheduled to run at 1 AM. The schedule

Copyright © 2022, 2026, Oracle and/or its affiliates.

integration runs when expected at 1 AM. When 2 AM arrives and the time is rolled back to 1 AM, the schedule integration does *not* run again.

Copyright © 2022, 2026, Oracle and/or its affiliates.

9

Create Event Integrations

You can create events that you publish and subscribe to in integrations in Oracle Integration. You can also subscribe to system events, which Oracle Integration creates for you.

Topics:

- About Publishing and Subscribing to Events
- Create Events to Publish and Subscribe to in Integrations
- Publish Events in an Integration
- Subscribe to Events in an Integration
- Define Custom Header Filters to Trigger Integrations
- Share Events Across Projects

You can monitor integrations that have subscribed to events at runtime. See Monitor Integrations That Subscribe to Events.

> **Note**
>
> You can also create events to publish and subscribe to in integrations in a project. See Create, Publish, and Subscribe to Events in a Project.

Want to see a video demo of publishing and subscribing to events in integrations?

Video

About Publishing and Subscribing to Events

Two types of events are available: published events and system events.

- **Published events**

  Create and select any events that you need to publish and subscribe to in an integration. You can also define custom header-based filters in events.

  Define the events in JSON-formatted or XML schema files.

- **System events**

  You don't have to create a system event because Oracle Integration creates it for you. Similarly, you can't publish a system event. However, a system event is always available to be subscribed to, and only in a project. System events are available for specific activities.

Available System Events

When an activity raises a system event, you can subscribe to the event without having to create or publish it. You can subscribe to a system event only within a project.

Copyright © 2022, 2026, Oracle and/or its affiliates.

System event	Activity that raises the event
File created	Creating a file in File Server
File deleted	Deleting a file in File Server
File downloaded	Downloading a file **Note:** A system event currently isn't raised when you download a file using the File server action. All other methods of downloading a file do raise an event.
File moved	Moving or renaming a file
Folder created	Creating a folder in File Server
Folder deleted	Deleting a folder in File Server
Folder moved	Moving or renaming a folder

Watch a video to learn more:

Video

Life Cycle for Published Events

The life cycle for published events consists of three high-level steps. These steps are described using an event named Patient Moved as an example. References are provided to more specific documentation for performing these steps.

The publish and subscribe feature enables you to decouple producers and subscribers. This decoupling enables you to define an event and start building your subscriber for the event before the event is published. You can create an event type (can be a producer, subscriber, or someone else). Creating an event type defines a contract, meaning that there's a contract to produce and subscribe.

Step	Description	For Specific Details...

1 - Create an event	Create a JSON-formatted or XML schema event on the Events page of Oracle Integration. For this example, the Patient Moved event contains the following JSON-formatted contents:
	
	`{`
	`"patientId": "value1",`
	`"fromDept": "value1",`
	`"toDept": "value1"`
	`}`
	
	You can also define custom headers to which to add filters later when creating a subscribing integration. See Define Custom Header Filters to Trigger Integrations.
		See Create Events to Publish and Subscribe to in Integrations.

Copyright © 2022, 2026, Oracle and/or its affiliates.

Step	Description	For Specific Details...
2 - Create an event publishing integration	Create a publishing integration to publish the event: 1. Create an integration. 2. Drag a **Publish event** action into the integration canvas. 3. Select **Patient Moved** as the event to publish in the Choose event panel.	See Publish Events in an Integration.
3 - Create an event subscription integration	Create a subscription integration to subscribe to the event: 1. Select the **Event** integration pattern in the Create integration panel. This option creates an event-based integration pattern. 2. Select **Patient Moved** in the Choose event panel. 3. Run the publishing integration described in Step 2. This action triggers the event to be subscribed to by the subscription integration.	See Subscribe to Events in an Integration.

Restrictions

A maximum of 50 integrations can subscribe to events per service instance.

Create Events to Publish and Subscribe to in Integrations

You can create events that you publish in an integration and subscribe to in another integration. The contents of the event must be in JSON or XML schema format.

If you need to subscribe to a system event, which Oracle Integration creates for you, you don't need to create or publish the event first. You can subscribe to any system event at any time, but only within a project. See About Publishing and Subscribing to Events.

1. Decide where to start:
   - Work in a project (see why working with projects is preferred).
     a. In the navigation pane, click **Projects**.
     b. Select the project name.
     c. Click **Integrations**.
     d. In the **Events** section, click **Add** if no connections currently exist or **+** if connections already exist. The Define new event panel opens.
   - Work outside a project.
     a. In the navigation pane, click **Design**, then **Connections**.
     b. Click **Create**. The Define new event panel opens.
2. Enter the information that describes event (for this example, then click **Continue**.

Copyright © 2022, 2026, Oracle and/or its affiliates.

Element	Description
Name	Enter a meaningful name to help others find your event.
Identifier	Automatically displays the name in capital letters that you entered in the **Name** field. If you modify the identifier name, don't include blank spaces (for example, `MOVE PATIENT`).
Description	Enter an optional description of the event.
Share with other projects	**Note**: This field only appears if you are creating an event in a project. Select to make this event publicly available in other projects. Event sharing eliminates the need to create and maintain separate events in different projects. Events created in standalone environments (that is, outside of projects) *cannot* be shared in projects. See Share Events Across Projects.

The Define event structure panel opens.

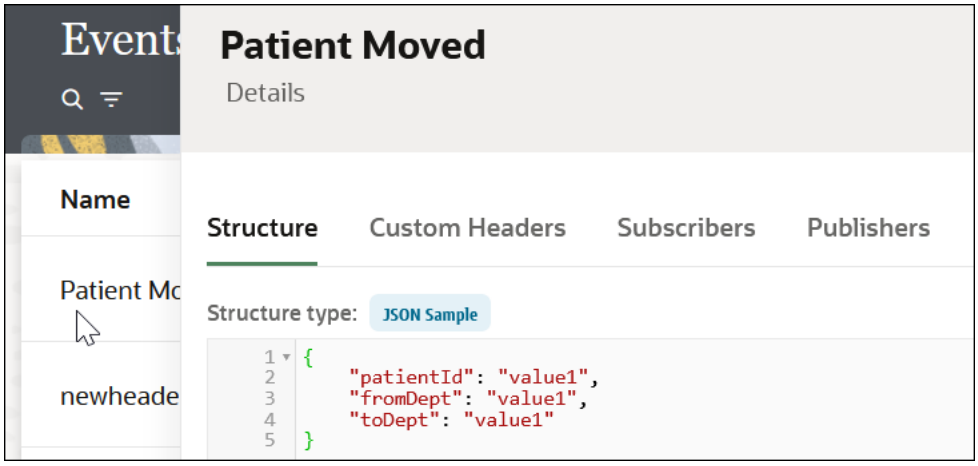
3. From the **Structure type** list, select an option:
   a. Select **JSON Sample** to upload a JSON file or drag and drop a JSON file into the box, then click **Create**. You can also manually edit the JSON sample provided to create your file contents.
   b. Select **XML Schema** to upload an XML schema file or drag and drop an XML schema file into the box, then click **Next**.
4. Click **Next**.
5. Click + to define a custom header name and description.
6. Click **Edit** to change the icon to a **Check Mark**✓. This action enables the row to be edited.
7. Enter a name and optional description, then click the **Check Mark** ✓ when complete.

   The event is displayed on the Events page.

> **Note**
>
> The **Search** feature on the Events page only works on exact matches of names and is case sensitive.

8. Hover your cursor over the event, and click **View details** to view the file contents (for this example, a JSON-formatted file). The **Custom Headers** tab enables you to view any custom headers defined for this event. The **Subscribers** and **Publishers** tabs enable you to view the integrations that have subscribed to the event and published the event.

Copyright © 2022, 2026, Oracle and/or its affiliates.



9. Perform other tasks, as necessary:

   - Click **Edit** to edit the event. You can change the name or replace the contents of the file of the event.
   - Click **Actions** to access options for cloning or deleting the event.

   The event is available for publishing in an integration. See Publish Events in an Integration.

Publish Events in an Integration

You can create an integration that publishes an event. Running the publishing integration enables subscribing integrations to subscribe to this event. This section provides an example of how to create a publishing integration.

You don't need to publish system events. You can subscribe to a system event at any time, without having to create or publish it.

1. Decide where to start.
   - Work in a project (see why working with projects is preferred).
     a. In the navigation pane, click **Projects**.
     b. Select the project name.
     c. Click **Integrations**.
     d. In the **Integrations** section, click **Add** if no integrations currently exist or **+** if integrations already exist.

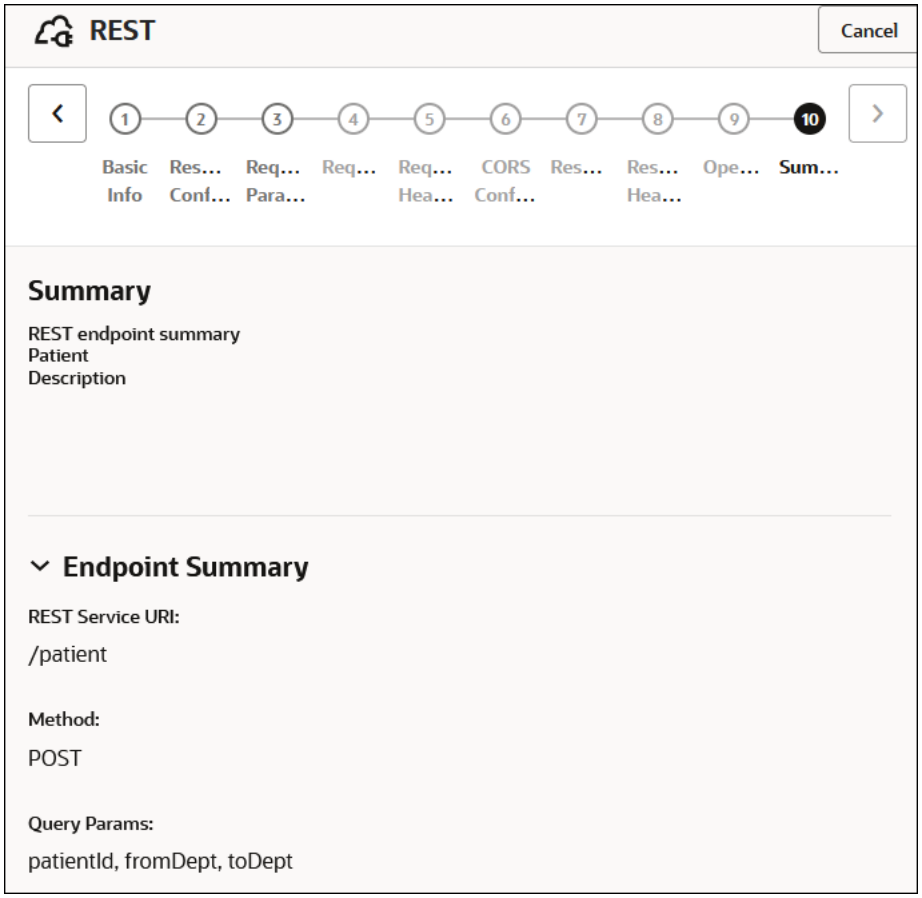
        The Add integration panel opens.
     e. Click **Create**.
   - Work outside a project.
     a. In the navigation pane, click **Design**, then **Integrations**.
     b. On the Integrations page, click **Create**.

        The Create integration panel appears.
2. Select the integration pattern to use.

Copyright © 2022, 2026, Oracle and/or its affiliates.

> **Note**
>
> Do *not* select **Event** as the integration pattern to publish the event. That pattern is for the integration that subscribes to the event. For this example, **Application** is selected to create an application integration.

3. Add a trigger connection to the integration. For this example, a REST Adapter is selected and configured with the following details shown on the Summary page of the Adapter Endpoint Configuration Wizard:
   - Endpoints relative resource URI: **/patient**
   - Action to perform on the endpoint: **POST**
   - Query parameters specified: **patientId**, **fromDept**, and **toDept** (these are the parameters defined in the event file)

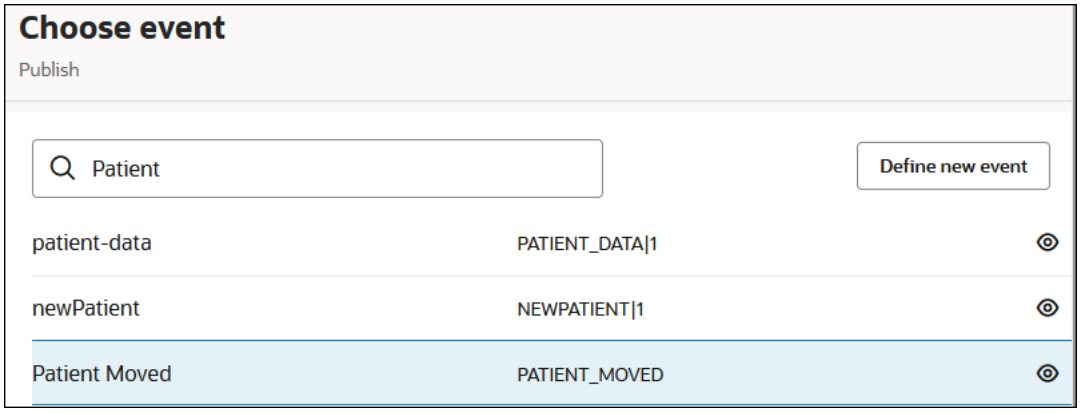


4. Drag a **Publish event** action into the integration canvas. See Publish Events in an Integration with a Publish Event Action.

   The Choose event panel opens.

5. Select the event to publish, and click **Choose**. For this example, **Patient Moved** is selected. You can also create a new event from this panel by selecting **Define new event**.

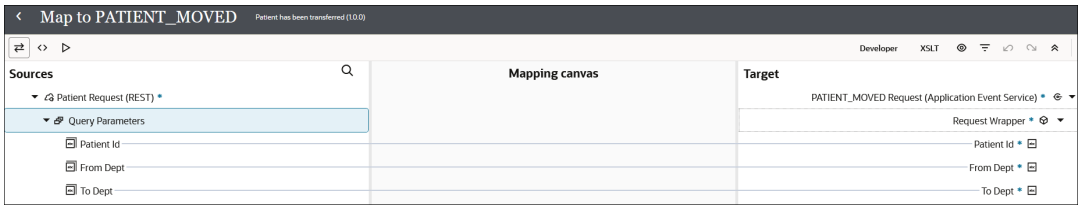
Copyright © 2022, 2026, Oracle and/or its affiliates.



The contents of the event are visible under the **View details** icon:

```
{
    "patientId": "value1",
    "fromDept": "value1",
    "toDept": "value1"
}
```

6. Click **Choose**, then click **Finish**.
7. Open the **Map** action that is automatically added above the **Publish event** action.
8. Map the source elements to the target elements. For this example, the three query parameters specified in the REST Adapter are mapped to the JSON event: **Patient Id**, **From Dept**, and **To Dept**.



9. Save and activate the integration.

   You are now ready to create the integration that subscribes to this event.

Subscribe to Events in an Integration

You can create an integration that subscribes to either a published event or a system event. The trigger for the integration depends upon the type of event.

- **Published event:** The subscribing integration runs when the integration that publishes the event is run.

  Multiple integrations can subscribe to the same published event. This section provides an example of how to create a subscription integration.

- **System event:** The subscribing integration runs when a specific activity occurs. Only specific activities raise system events, and the events are raised only when an integration is subscribed to the events. See Available System Events.

Copyright © 2022, 2026, Oracle and/or its affiliates.

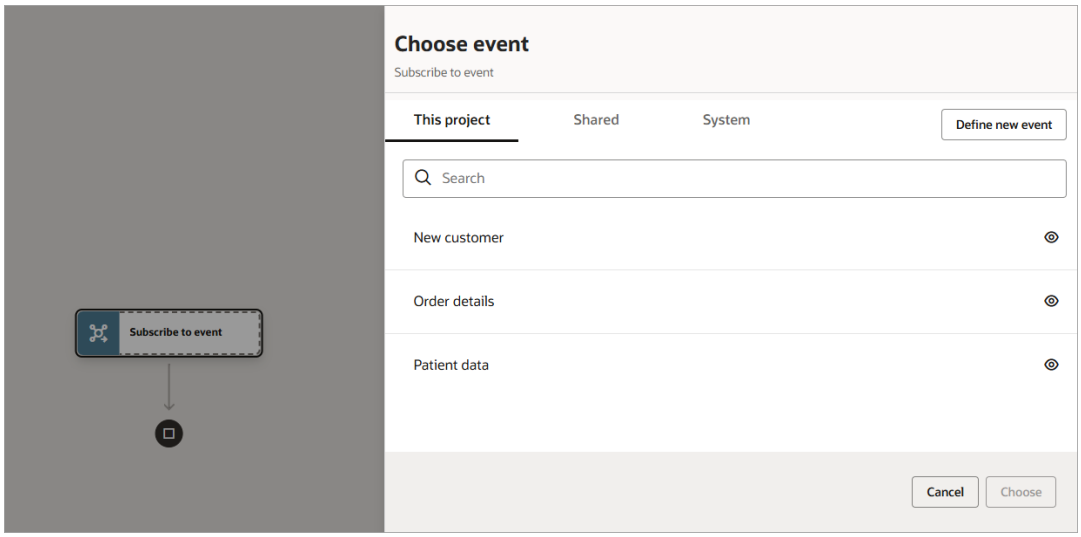
1. Decide where to start.
   - Work in a project (see why working with projects is preferred). System events are available only in a project.
     a. In the navigation pane, click **Projects**.
     b. Select the project name.
     c. Click **Integrations**.
     d. In the **Integrations** section, click **Add** if no integrations currently exist or **+** if integrations already exist.

        The Add integration panel opens.
     e. Click **Create**.
     f. Select **Event**.
   - Work outside a project.
     a. In the navigation pane, click **Design**, then **Integrations**.
     b. On the Integrations page, click **Create**.

        The Create integration panel appears.
     c. Select **Event**.
2. Complete the fields in the Create integration panel to create the event-based integration. See Create an Integration.

   A **Subscribe to event** trigger is created and the Choose event panel opens for you to select the event to which to subscribe.

> **Note**
>
> The selected event becomes the trigger for this integration. You do *not* drag an adapter connection into the integration canvas in event-based integrations to use as a trigger.

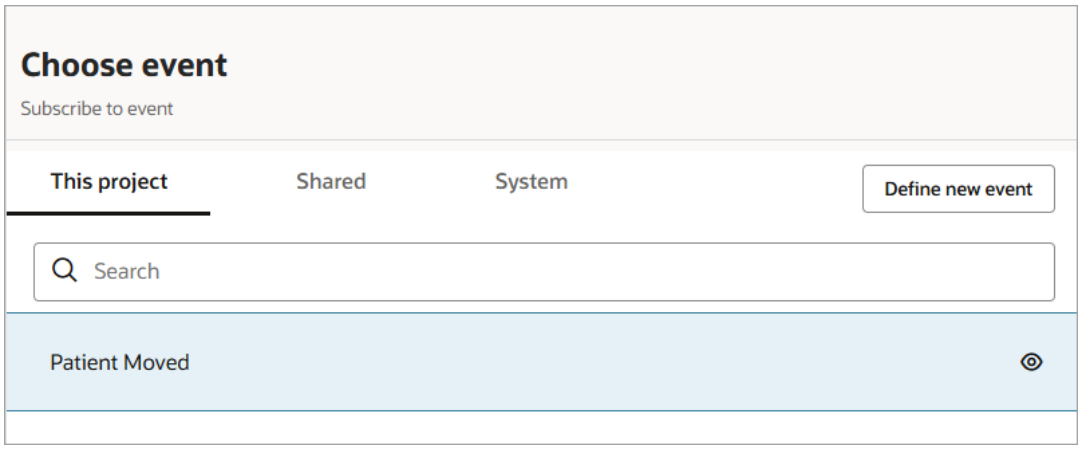


Copyright © 2022, 2026, Oracle and/or its affiliates.

3. Select the event.

   Shared events are on the **Shared** tab, and system events are on the **System** tab. You can also define a new event in this panel by selecting **Define new event**.

   For this example, **Patient Moved** is selected.



4. Click **View details** to view the JSON contents of the event. For this example:

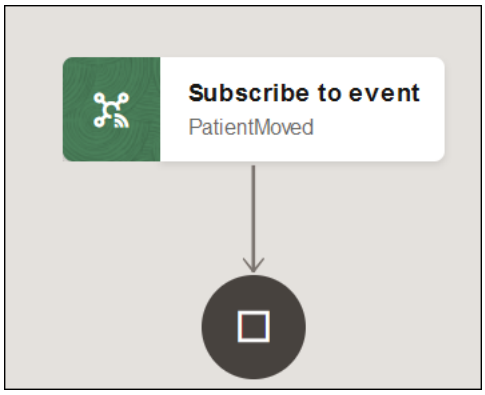
```
{
    "patientId": "value1",
    "fromDept": "value1",
    "toDept": "value1"
}
```

5. Click **Choose this event** or click **<**, then click **Choose**.

   You can also define filters on any custom headers in the event. See Define Custom Header Filters to Trigger Integrations.

6. Click **Finish**.

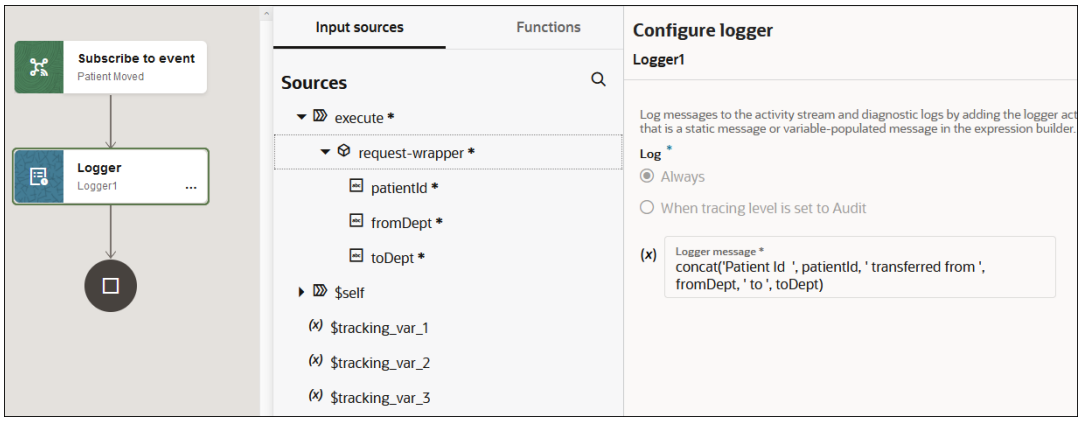
   The **Subscribe to event** trigger is updated to include the event that you selected.



Copyright © 2022, 2026, Oracle and/or its affiliates.

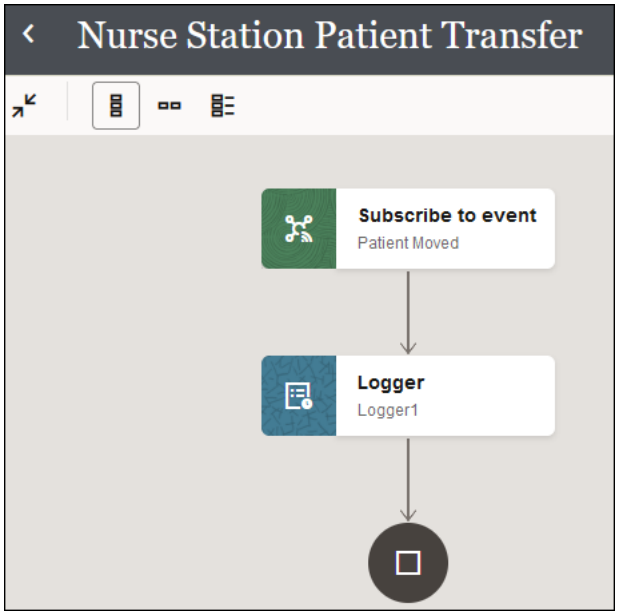
7. Design the remaining parts of the integration. For this example, a logger action is added to the integration canvas and the message is configured to include the parameters defined in the event.

```
concat('Patient Id  ', patientId, ' transferred from ', fromDept, ' to ',
toDept)
```



8. Define a business identifier for the integration.

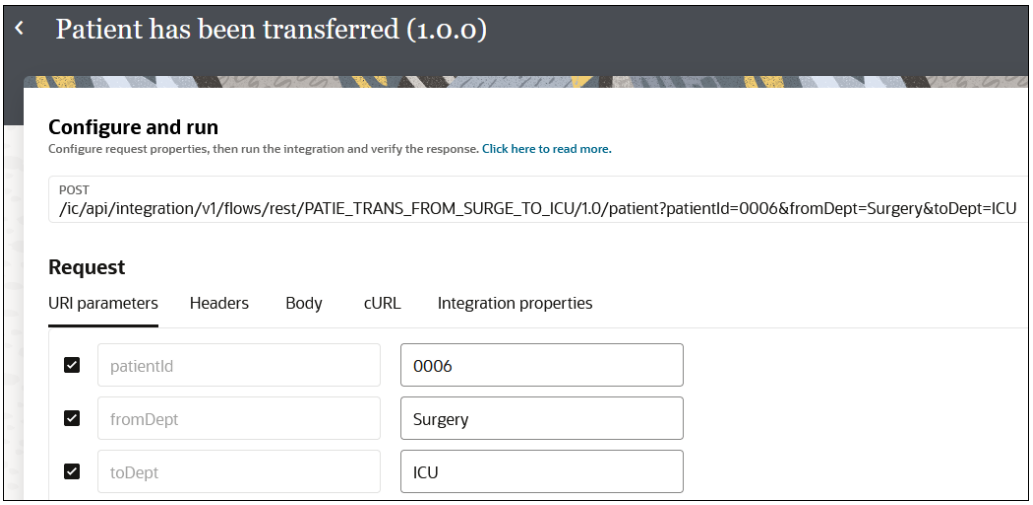
   The integration looks as follows:



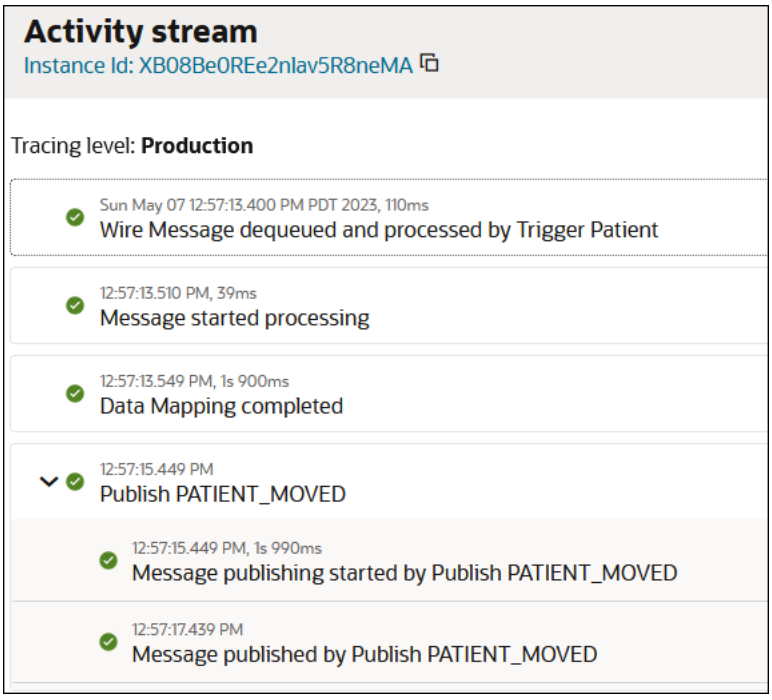
9. Save and activate the integration. This integration is now ready to subscribe to the **Patient Moved** event.
10. Run the integration that publishes the **Patient Moved** event. Once this integration is run, the event in the subscribing integration is triggered.
    a. In the navigation pane, click **Design**, then **Integrations**.
    b. Hover over the publishing integration, and select **Actions** ⋮, and then **Run**.

Copyright © 2022, 2026, Oracle and/or its affiliates.

c. Specify values for the three parameters in the event file, and click **Run**.



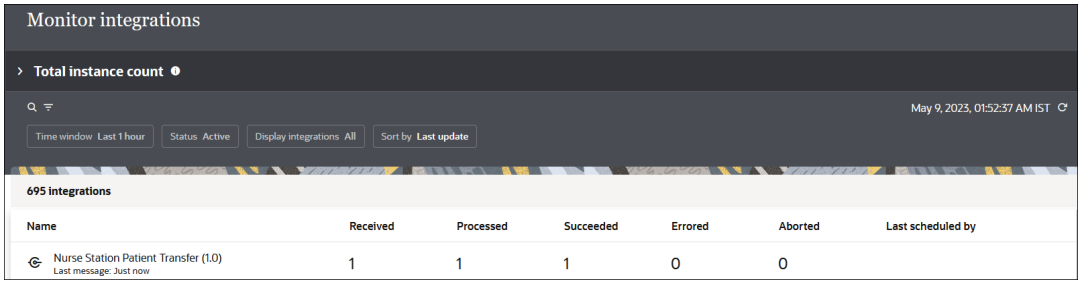
The activity stream opens and indicates that the **Patient Moved** event was published.



11. In the navigation pane, click **Observability**, then **Integrations**.

    The integration that subscribed to the event was triggered and an instance was created and processed successfully.

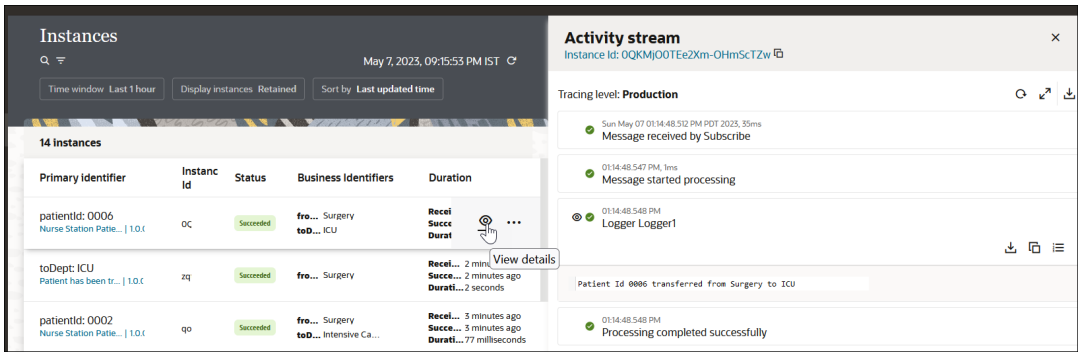
Copyright © 2022, 2026, Oracle and/or its affiliates.



12. In the navigation pane, click **Observability**, then **Instances**.
13. Hover over the business identifier for the integration instance and click **View details** to open the Activity stream panel.

    The subscribing integration successfully subscribed to the event. The logger action message in the activity stream shows that the patient transfer occurred.

    ```
    Patient ID 0006 was transferred from surgery to ICU.
    ```



Retain an Event Subscription During Integration Deactivation

You can retain an event subscription when deactivating an integration that subscribes to published events. This action enables you to begin subscribing to published events again when you reactivate the integration.

> **Note**
>
> For subscribing integrations created in release 23.10 and later, the **Delete event subscription** check box is visible. For subscribing integrations created prior to release 23.10, there is no check box and you do not have the ability to retain events when deactivating an integration.

1. In the navigation pane, click **Design**, then **Integrations**.
2. Go to the row of the subscribing integration you want to deactivate.
3. Click **Deactivate** to deactivate the integration.

   You can select whether or not to delete the event subscription.

Copyright © 2022, 2026, Oracle and/or its affiliates.

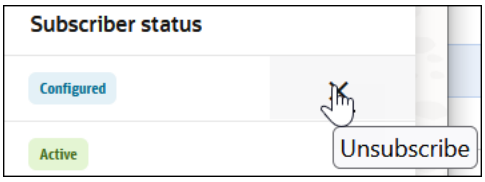


Event subscription behavior is based on what you do with the **Delete event subscription** check box.

4. If you do not select **Delete event subscription**, and click **Deactivate**.

> **Note**
>
> Events expire after 24 hours. Therefore, only events published within one day of the next activation are delivered.

   a. In the navigation pane, click **Observability**, then **Subscriptions**.
   b. Note that the deactivated integration appears with the words **Configured** in the **Subscriber status** column. If you reactivate the integration, it begins subscribing to events again.
   c. If you want to unsubscribe, hover over the row of the integration, and click **X**.



5. If you select **Delete event subscription**, and click **Deactivate**.
   a. In the navigation pane, click **Observability**, then **Subscriptions**.
   b. Note that the deactivated integration does not appear. If you reactivate the integration, it does *not* subscribe to events again.

Define Custom Header Filters to Trigger Integrations

You can define custom headers in events, select those events and map request payload elements to the custom headers when creating a publishing integration, and then define filters on those headers when creating a subscribing integration. If the filter condition defined is

Copyright © 2022, 2026, Oracle and/or its affiliates.

satisfied at runtime, the subscribing integration is triggered. If the filter condition is not satisfied, the integration is not triggered.

- Capabilities
- Restrictions
- Use Case Overview
- Create an Event with a Custom Header in a Project
- Create an Integration to Publish the Event in a Project
- Create an Integration with No Filter to Subscribe to the Event in a Project
- Create an Integration with a Name Filter to Subscribe to the Event
- Create an Integration with an Age Filter to Subscribe to the Event
- Run the Publishing Integration and Track the Created Instances
- Run the Publishing Integration Again and Track the Created Instances

Capabilities

- You define custom headers when creating an event.
- You define custom header values in the mapping of a publish event action. You can define the mapping from payload data to a custom header to enable payload data to participate in filtering.
- You define filters on the custom headers when creating a subscribing integration. Subscribers can filter events based on fields within the payload if those fields are mapped to the custom headers. If the filter condition you define is satisfied during runtime, the subscribing integration is triggered.
- You define filters in two ways:
  - JQ expressions (through the **Code** tab):

    For example:

    ```
    "filter-def": ".customheader1==\"someValue\""
    ```

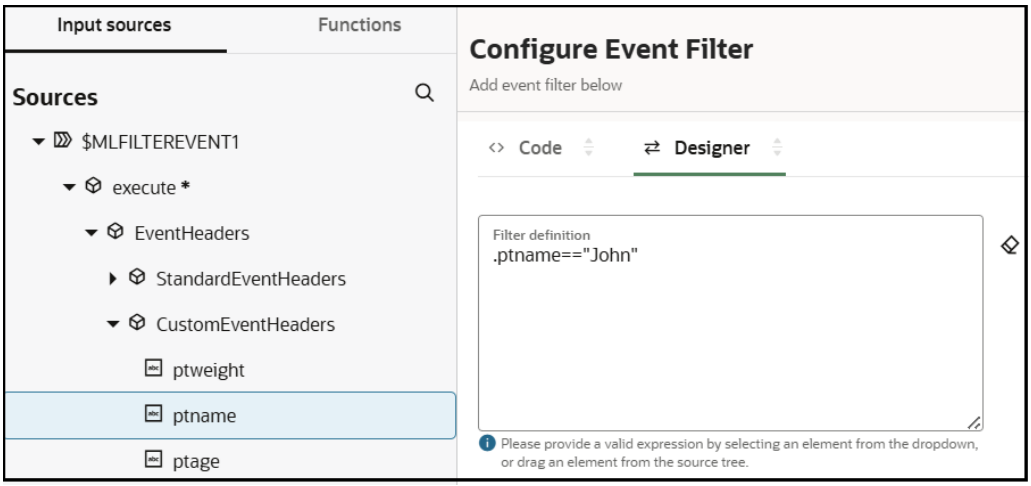


Copyright © 2022, 2026, Oracle and/or its affiliates.

JQ is a lightweight and flexible command line JSON processor primarily used for extracting and manipulating JSON data. JQ takes an input and produces an output. See jq 1.7 Manual and JQ Expression Syntax for Creating Filters.

- Expression builder (through the **Designer** tab):

  This option enables you to build an expression by dragging an element or function from the source tree, then defining it further in the designer. For example:



Restrictions

- You can define 10 custom headers per event.
- If defining a filter in the code editor, enter all functions in lower case.
- Only JQ expression filters are supported.
- The following JQ function calls are supported:
  - `any`
  - `all`
  - `ascii_downcase`
  - `ascii_upcase`
  - `tonumber`
  - `tostring`
  - `min`
  - `max`
  - `contains`
  - `startswith`
  - `endswith`
  - `ltrimstr`
  - `rtrimstr`
  - `length`
  - `now`

Copyright © 2022, 2026, Oracle and/or its affiliates.

  - `split`
  - `fromdateiso8601`
  - `floor`
  - `ceil`
- The following JQ keywords/filter expressions are not supported:
  - '.' (A single dot alone is not supported for security reasons.)
  - `while`
  - `for`
  - `repeat`
  - `input/inputs/stderr`
  - `debug`
  - `input_filename`
  - `input_line_number`
  - `recurse`
  - `foreach`
  - `until`
- The following date and time JQ filters are supported:

  JQ provides a set of date helpers that make it simple to parse, compare, and format ISO-8601 timestamps inside a JSON-streaming pipeline. The most frequently used helpers are:

  - `fromdateiso8601`: Parses an ISO-8601 string: Unix-epoch seconds.
  - `now`: Shows the current epoch seconds at evaluation time.
  - `split`: Extracts calendar pieces with pure string operations:
    * ISO-8601 stamp: split at `T` to drop the clock and split at `-` to access `Y-M-D` elements.

Use Case	Example

Determine whether a date is in the past or future.	`# Is the schedule date still ahead?`
	`.data.scheduleDate | fromdateiso8601 >  now`
	
	`# Has the schedule date already passed?`
	`.data.scheduleDate | fromdateiso8601 <  now`

Copyright © 2022, 2026, Oracle and/or its affiliates.

Use Case	Example

Check the relative age (hours/days/weeks before now).	Seconds/constants:
	– 1 hour = 3600
	– 1 day = 86,400
	– 1 week = 604,800
	
	`# 1 hour ago`
	`(now - (.data.eventTime | fromdateiso8601)) >= 1 * 3600`
	
	`# < 1 hour ago`
	`(now - (.data.eventTime | fromdateiso8601)) < 1 * 3600`
	
	`# 3 days ago`
	`(now - (.data.eventTime | fromdateiso8601)) >= 3 * 86400`
	
	`# 2 weeks ago`
	`(now - (.data.eventTime | fromdateiso8601)) >= 2 * 604800`

Compare against a static date.	`$staticDate = "2024-01-01T00:00:00Z";`
	
	`# Is the event 10 days before the anchor?`
	`( ($staticDate | fromdateiso8601) - (.data.eventTime | fromdateiso8601) ) >= 10*86400`
	
	`# Is the event 10 days after the anchor?`
	`( (.data.eventTime | fromdateiso8601) - ($staticDate | fromdateiso8601) ) >= 10*86400`

Filter on a specific date.	Day-of-month (returns string `"05"`)
	
	`(.data.eventTime | split("T")[0] | split("-")[2]) == "05"`
	
	Month-of-year (returns string `"11"`)
	
	`("2021-11-05T03:56:24Z" | split("T")[0] | split("-")[1]) == "11"`
	`Birthday check (string comparison, no date helpers) $mmdd constant holding today’s month-day in “MM-DD” form.`
	
	Example: Check birthday using a simple string comparison. Convert both the stored birthday and today’s date to an `MM-DD` fragment and compare.
	
	`(.data.birthDate | split("T")[0] | split("-") | .[1] + "-" + .[2]) == $mmdd`

Copyright © 2022, 2026, Oracle and/or its affiliates.

Use Case	Example

Filter on age (given birth date).	Exact calendar age arithmetic without dedicated date functions. Break the anchor date (`today`) and the date-of-birth into numeric arrays, subtract the years, then subtract one more if the birthday for the current year has not yet occurred. This returns the completed number of years.
	
	`(ageYears(.data.birthDate; "2025-10-30") >= 18)`
	
	`// Exact calendar age Helper converts two “YYYY-MM-DD” strings (DOB and anchor date) to numbers and applies the human calendar rule.`
	`def ageYears($dob; $today): ($today | split("-") | map(tonumber)) as $t # [$CY,$CM,$CD] | ($dob | split("T")[0] | split("-") | map(tonumber)) as $b # [$BY,$BM,$BD] | ($t[0] - $b[0]) - (($t[1] < $b[1]) or ($t[1] == $b[1] and $t[2] < $b[2])) | if . then 1 else 0 end);`

Handle time zone	The client is required to pass time zones in the UTC / GMT time zone as the `fromdateiso8601` defaults to local time if no time zone is passed. Highlight potential issues with local time and emphasize passing standardized UTC / GMT time zones by client rather than handing them by server. The client should convert the date time to the UTC time zone if they're dealing in a different time zone.

- JQ-based Lambda function definitions and their executions as part of a filter definition are not supported.

Use Case Overview

The use case in this section walks you through creating an event with custom headers, selecting the event and mapping the custom headers in a publishing integration, and adding different JSON-based filter conditions to the headers in several subscribing integrations. In this use case:

1. Event **MLFilterEvent** is created with the following event and custom header content:

Event Structure	Custom Headers

`{`
`"patientIdentifier" : "12345679",`
`"illness" : "1",`
`"name": "min",`
`"age":30,`
`"weight":100`
`}`
	`ptname`
	`ptage`
	`ptweight`

2. Publishing integration **MLFilterPub** is designed to publish the event **MLFilterEvent**. This integration consists of the following connection and actions:

Copyright © 2022, 2026, Oracle and/or its affiliates.

- A REST Adapter trigger is configured with the following JSON-formatted request payload:

```
{
  "patientIdentifier" : "12345679",
  "illness" : "1",
  "name" : "min",
  "age" : 30,
  "weight" : 100
}
```

- A mapper is configured to map patient details elements in the source request payload to target patient details elements, target standard header elements, and target custom header elements.
- A log action is configured to log standard and custom header details in the activity stream.

3. The following subscribing integrations are designed to subscribe to the **MLFilterEvent** event:
   a. Subscribing integration **MLFilterSubNoFilter** is designed *without* adding any filters.
   b. Subscribing integration **MLFilterNameSub** is designed to be triggered only if the following custom header filter condition is satisfied.

```
{
    "type": "jq_filter",
    "filter-def": ".ptname==\"John\""
}
```

   c. Subscribing integration **MLFilterAgeSub** is designed to be triggered only if the following custom header filter condition is satisfied.

```
{
    "type": "jq_filter",
    "filter-def": ".ptage|length!=0 and tonumber >= 18"
}
```

4. The three subscription integrations are activated: **MLFilterSubNoFilter**, **MLFilterNameSub**, and **MLFilterAgeSub**.
5. When publishing integration **MLFilterPub** is activated, the request payload body values are specified, and the integration is run, the following occurs:
   - Subscribing integration **MLFilterSubNoFilter** is always triggered and consumes the event. Because no custom header filters are defined on this subscribing integration, there are no restrictions on it being triggered.
   - Subscribing integration **MLFilterNameSub** is only triggered if `"name": "John"` is set in the request payload body.

```
{
    "patientIdentifier" : "123456790",
    "illness" : "1",
    "name": "John",
    "age":30,
```


Copyright © 2022, 2026, Oracle and/or its affiliates.

```
    "weight":100
  }
```

- Subscribing integration **MLFilterAgeSub** is only triggered if `age` is greater than or equal to `18`. For this example, `"age": "30"` is set in the request payload body.

```
{
    "patientIdentifier" : "123456790",
    "illness" : "1",
    "name": "John",
    "age":30,
    "weight":100
  }
```

Let's now design and run the use case.

Create an Event with a Custom Header in a Project

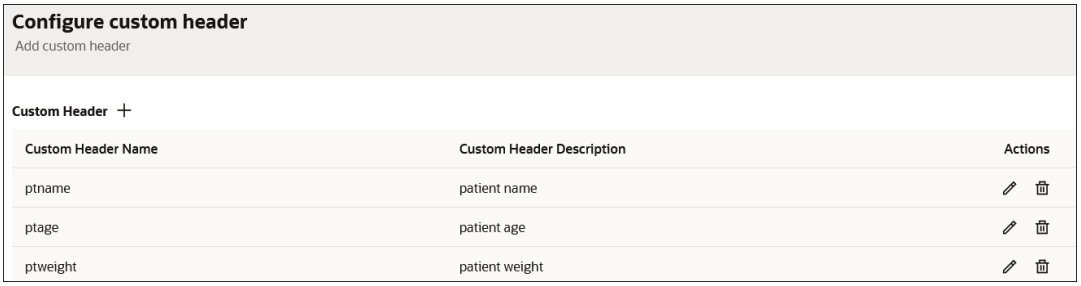
1. In the navigation pane, click **Projects**.
2. Create a project.
3. Scroll down to the **Events** section.
4. Click **Add** if no events currently exist or **+** if events already exist.
5. Enter a name (for this example, named **MLFilterEvent**) and description, and click **Continue**.
6. Define the event structure, and click **Continue**. For this event, the following JSON-formatted event is defined:

```
{
    "patientIdentifier" : "12345679",
    "illness" : "1",
    "name": "min",
    "age":10,
    "weight":100
  }
```

7. Click **+** to add a custom header to the event.
8. Click **Edit**  to change the icon to a **Check Mark** ✓. This action enables the row to be edited.
9. Enter a name and optional description, then click the **Check Mark** ✓ when complete. Note the following custom header naming conventions:
   - Can be lower-case letters (a-z) and numbers (0-9).
   - Cannot be longer than 20 characters.
   - Cannot be a duplicate of another custom header.
   - Must not be a reserved cloud events attribute (for example, data). If you use a reserved name, it does not pass validation.

   For this example, the following custom headers are added.

Copyright © 2022, 2026, Oracle and/or its affiliates.



Create an Integration to Publish the Event in a Project

1. Return to the project details page.
2. Go to the **Integrations** section and add an integration.
3. Select an application integration.
4. Add a trigger connection. For this example, a REST Adapter connection is defined to include the following JSON-formatted request payload:

```
{
  "patientIdentifier" : "12345679",
  "illness" : "1",
  "name" : "min",
  "age" : 30,
  "weight" : 100
}
```

5. Click **+** below the REST Adapter and drag a **Publish event** action into the integration.
6. Select the event in which you defined the custom headers (**MLFilterEvent**), and click **Finish**.



7. Open the mapper for the **Publish event** action and define the mappings:
   a. Map the **Request Wrapper** source elements to the **Request Wrapper** target elements. For this example:
      - **Patient Identifier** to **Patient Identifier**
      - **Illness** to **Illness**
      - **Name** to **Name**
      - **Age** to **Age**
      - **Weight** to **Weight**
   b. Expand the target **Event Headers** section to show **Standard Event Headers** and **Custom Event Headers**.

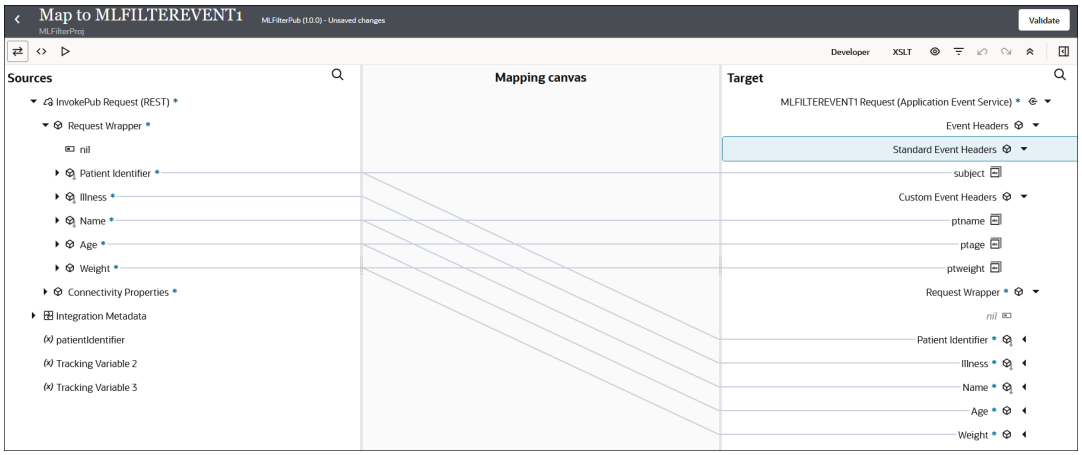
Copyright © 2022, 2026, Oracle and/or its affiliates.

c. Expand the target **Standard Event Headers** and map the source **Patient Identifier** to the target **Subject**.

d. Expand the target **Custom Event Headers** and map the following request payload elements to the target custom headers you created:

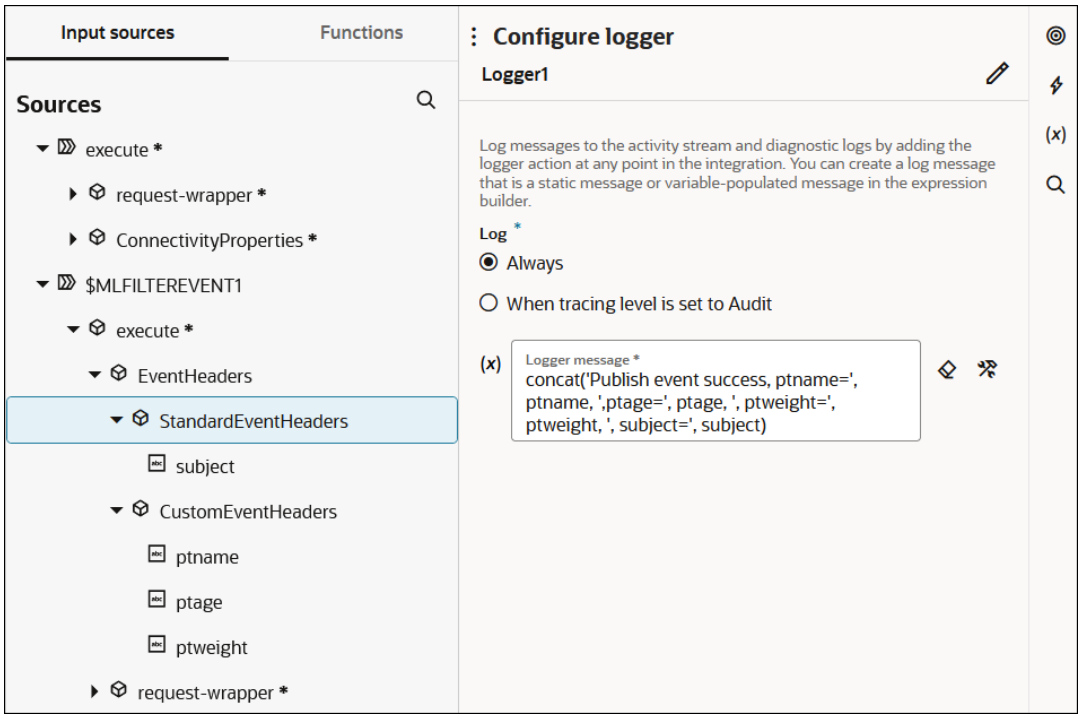
- **Name** to **ptname**
- **Age** to **ptage**
- **Weight** to **ptweight**

When complete, the mappings look as follows:



8. Validate the mapper and return to the integration canvas.

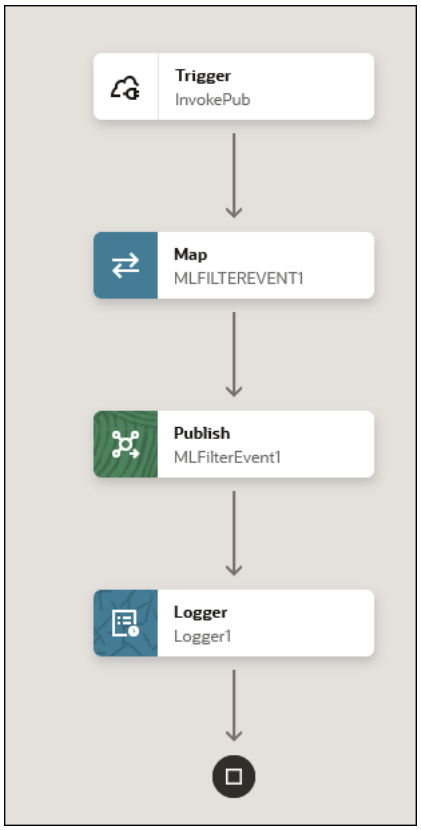
9. Add a log action to log the standard and custom headers.



10. Save the log action.

Copyright © 2022, 2026, Oracle and/or its affiliates.

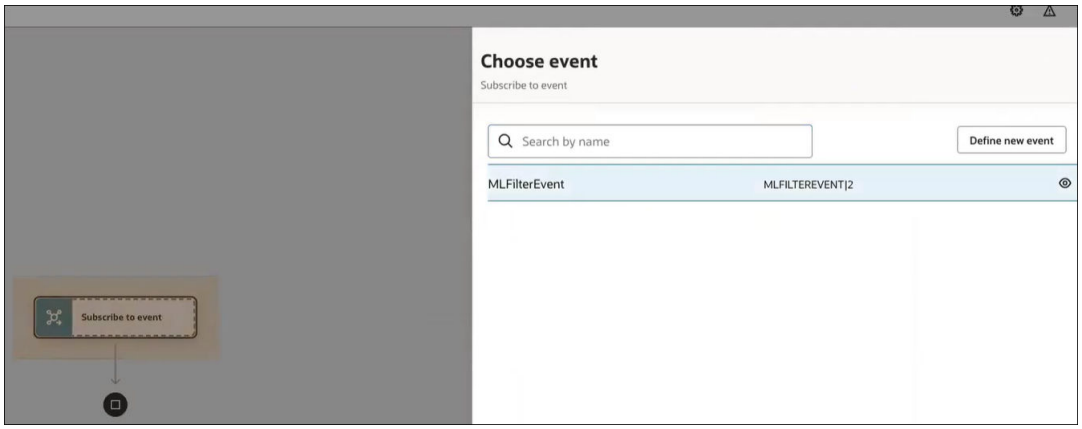
The completed integration looks as follows:



Let's now create integrations to subscribe to the event and demonstrate how filters work.

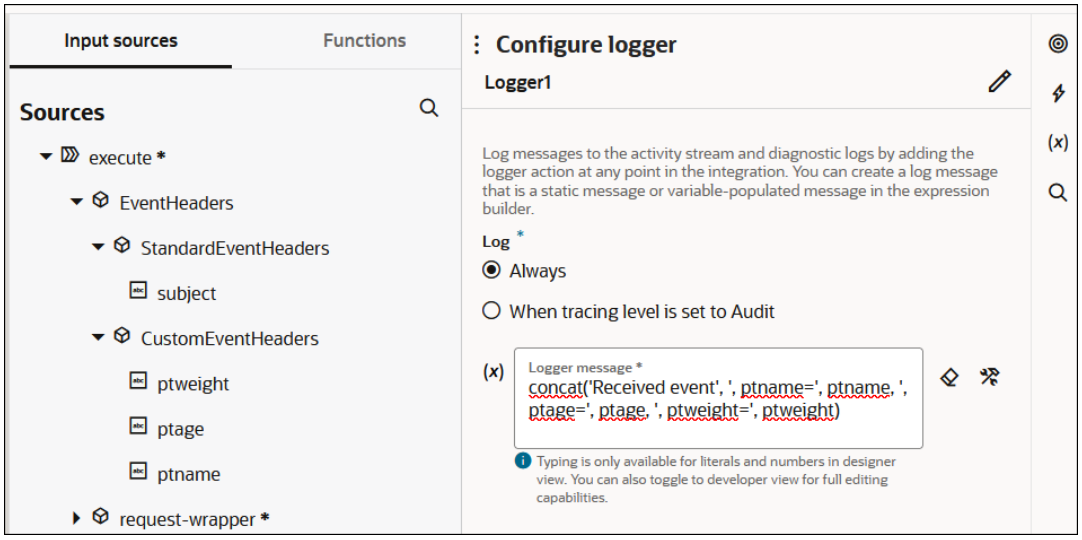
Create an Integration with No Filter to Subscribe to the Event in a Project

1. Create an event integration (for this example, named **MLFilterSubNoFilter**).
2. Select the **MLFilterEvent** event, and click **Choose**.

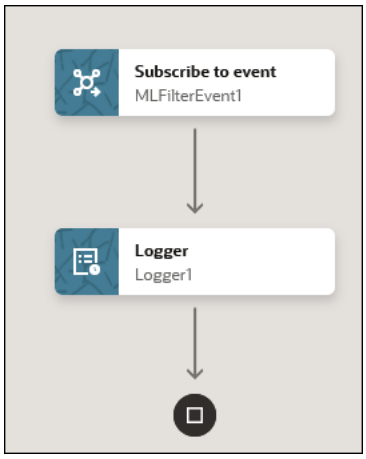


3. Click **Finish**.
4. Add a log action to log the standard and custom header details.

Copyright © 2022, 2026, Oracle and/or its affiliates.



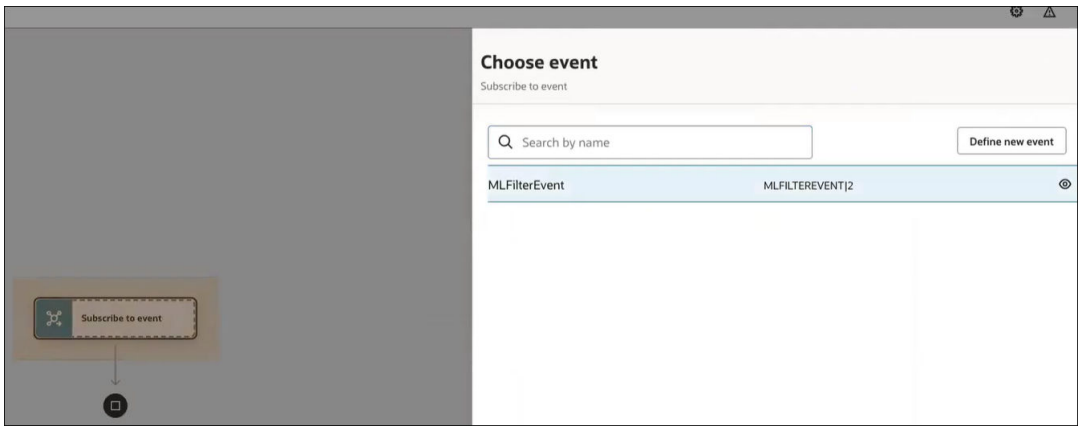
The completed integration looks as follows:



5. Activate the integration.

Create an Integration with a Name Filter to Subscribe to the Event

1. Create an event integration (for this example, named **MLFilterNameSub**).
2. Select the **MLFilterEvent** event, and click **Choose**.



Copyright © 2022, 2026, Oracle and/or its affiliates.

3. Click **Add Filter**.



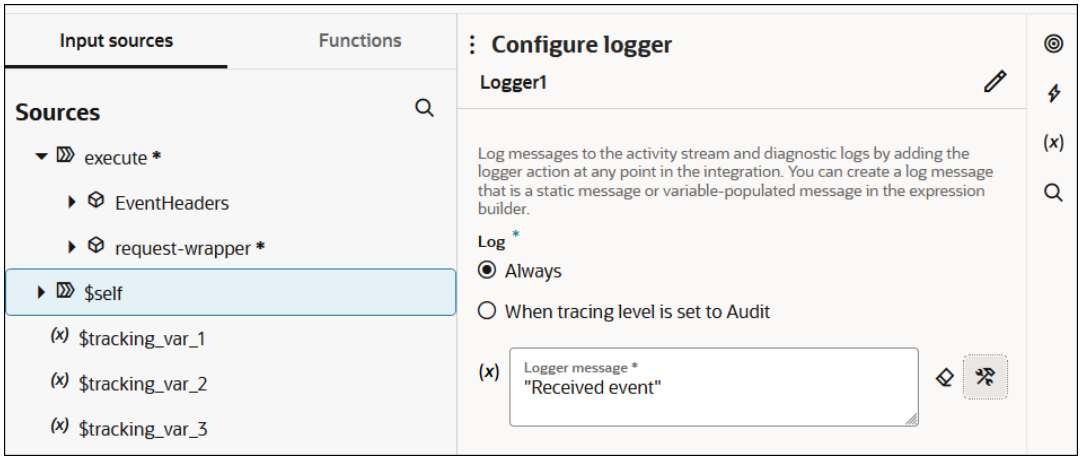
4. Manually enter the filter condition in the code editor. The only supported filter type is `jq_filter`. For this example, `ptname` (patient name) is set to `John`. If this filter condition is satisfied at runtime, this integration is triggered.

```
{
    "type": "jq_filter",
    "filter-def": ".ptname==\"John\""
}
```

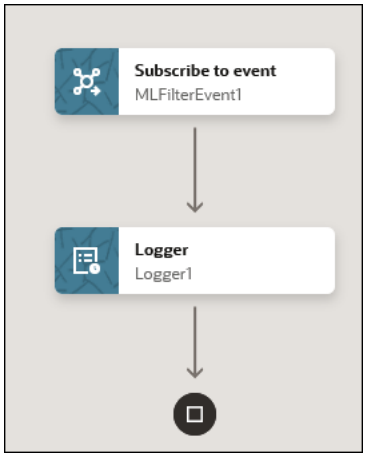


5. Click **Validate**, then click **Finish**.
6. Add a log action with a message.

Copyright © 2022, 2026, Oracle and/or its affiliates.



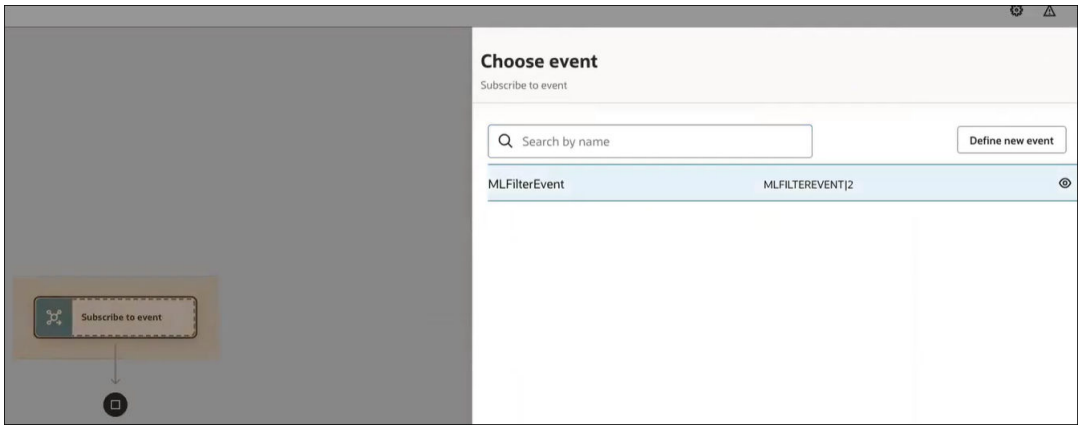
The completed integration looks as follows:



7. Activate the integration.

Create an Integration with an Age Filter to Subscribe to the Event

1. Create an event integration (for this example, named **MLFilterAgeSub**).
2. Select the **MLFilterEvent** event, and click **Choose**.



Copyright © 2022, 2026, Oracle and/or its affiliates.

3. Click **Add Filter**.



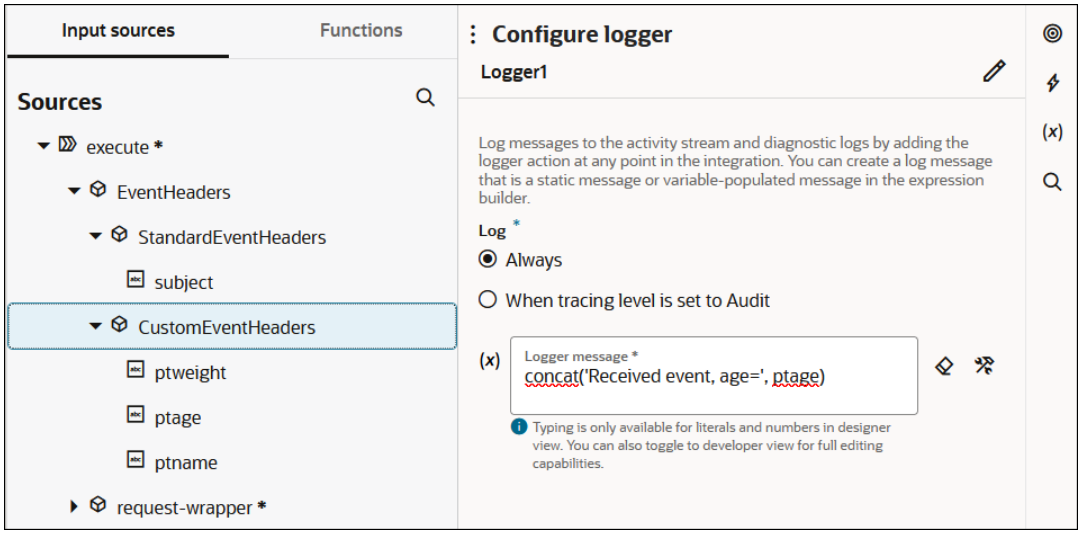
4. Manually enter the filter in the code editor. For this example, `ptage` (patient age) is set to 18 or older. If this filter condition is satisfied at runtime, this integration is triggered.

```
{
    "type": "jq_filter",
    "filter-def": ".ptage|length!=0 and tonumber >= 18"
}
```

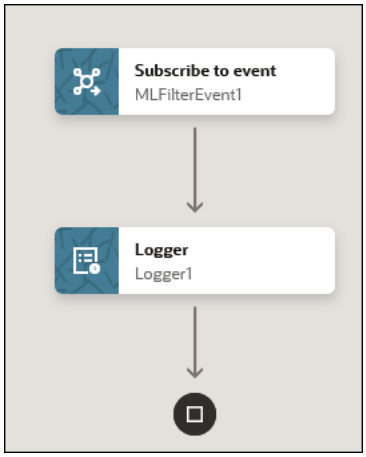


5. Click **Validate**, then click **Finish**.
6. Add a log action to log age details.

Copyright © 2022, 2026, Oracle and/or its affiliates.



The completed integration looks as follows:



7. Activate the integration.

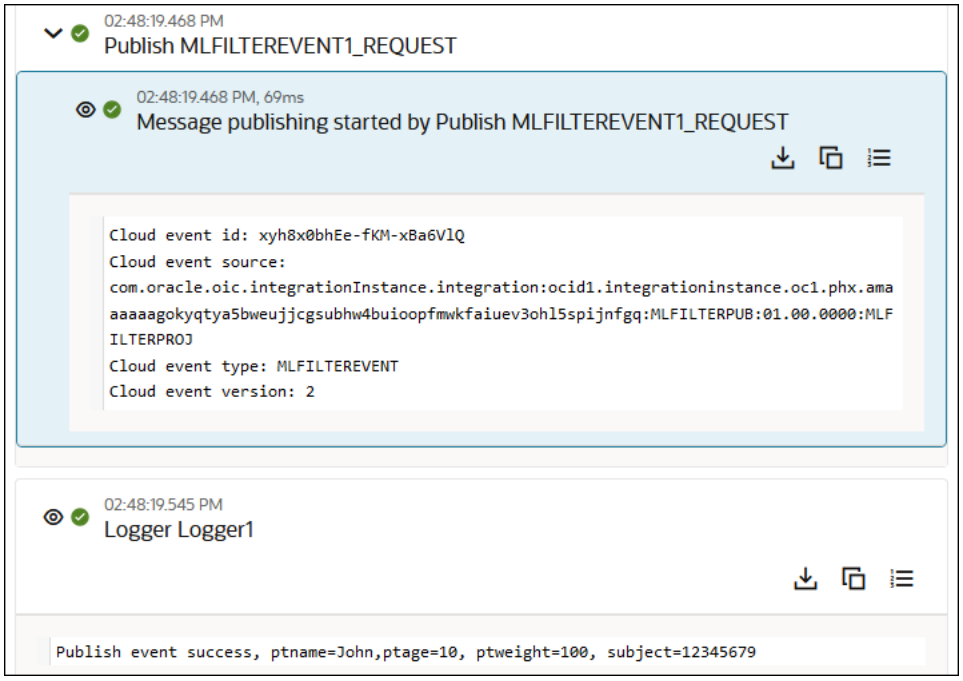
Run the Publishing Integration and Track the Created Instances

1. Activate the publishing integration.
2. Hover over the publishing integration, then click ⋯ and select **Run**. Under **Body**, the request payload is displayed.
3. Change `name` to `John` and `age` to `10`, then click **Run**.

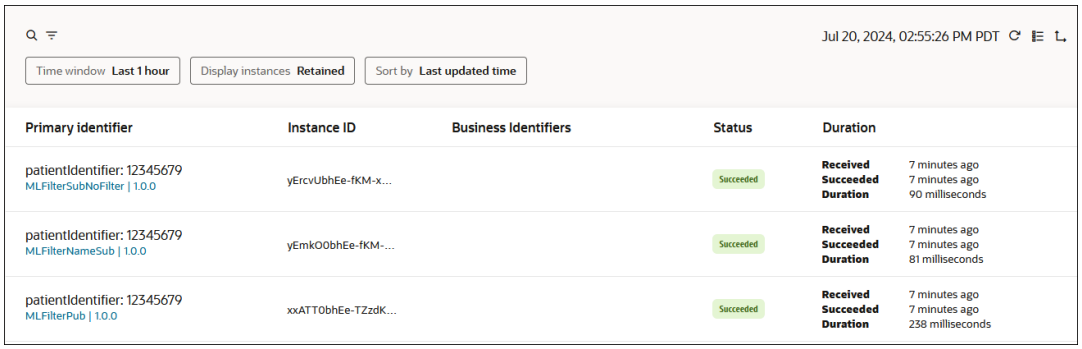
```
{
  "patientIdentifier": "12345679",
  "name": "John",
  "weight": 100,
  "age": 10,
  "illness": "1"
}
```

4. Expand key milestones in the activity stream for the publishing integration instance to see details.

Copyright © 2022, 2026, Oracle and/or its affiliates.

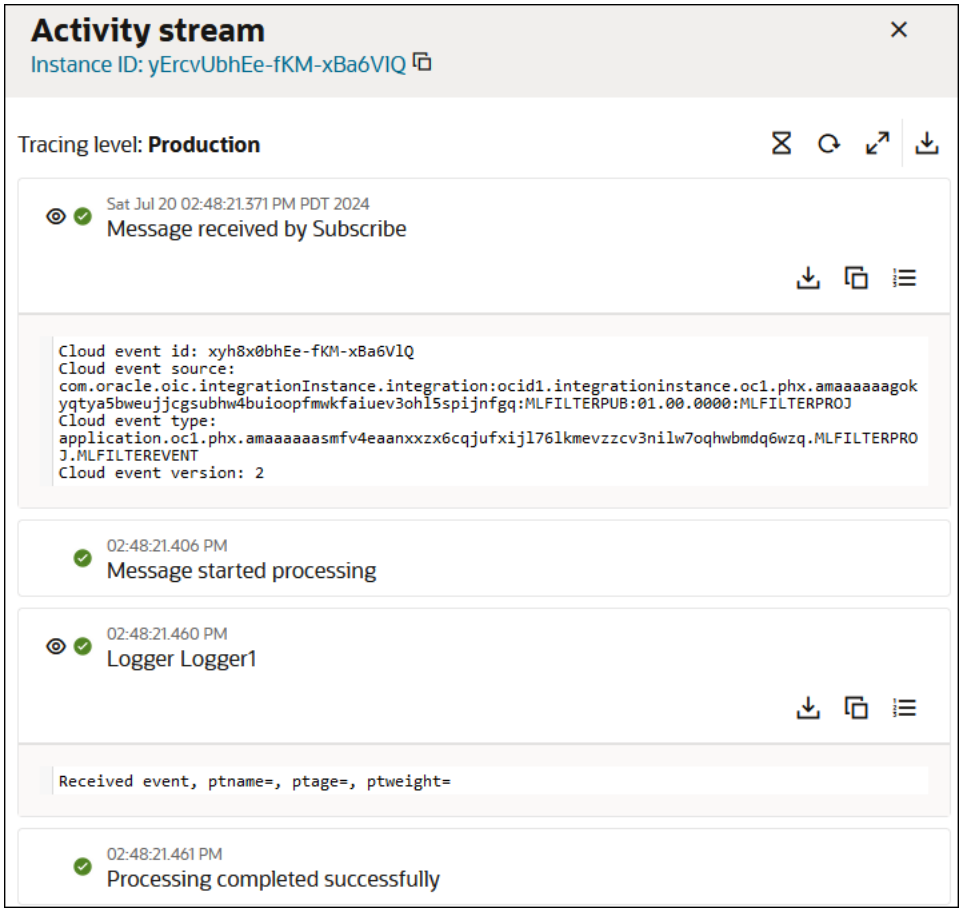


5. Return to the project details page.
6. Click **Observe**, then **Instances**.
   The following integration instances are displayed.
   - **MLFilterSubNoFilter**: The subscribing integration with no filter. This integration is triggered because no filter is set.
   - **MLFilterNameSub**: The subscribing integration with the filter of `name` set to `John`. This integration was triggered because the filtering condition was satisfied.
   - **MLFilterPub**: The publishing integration for the event.



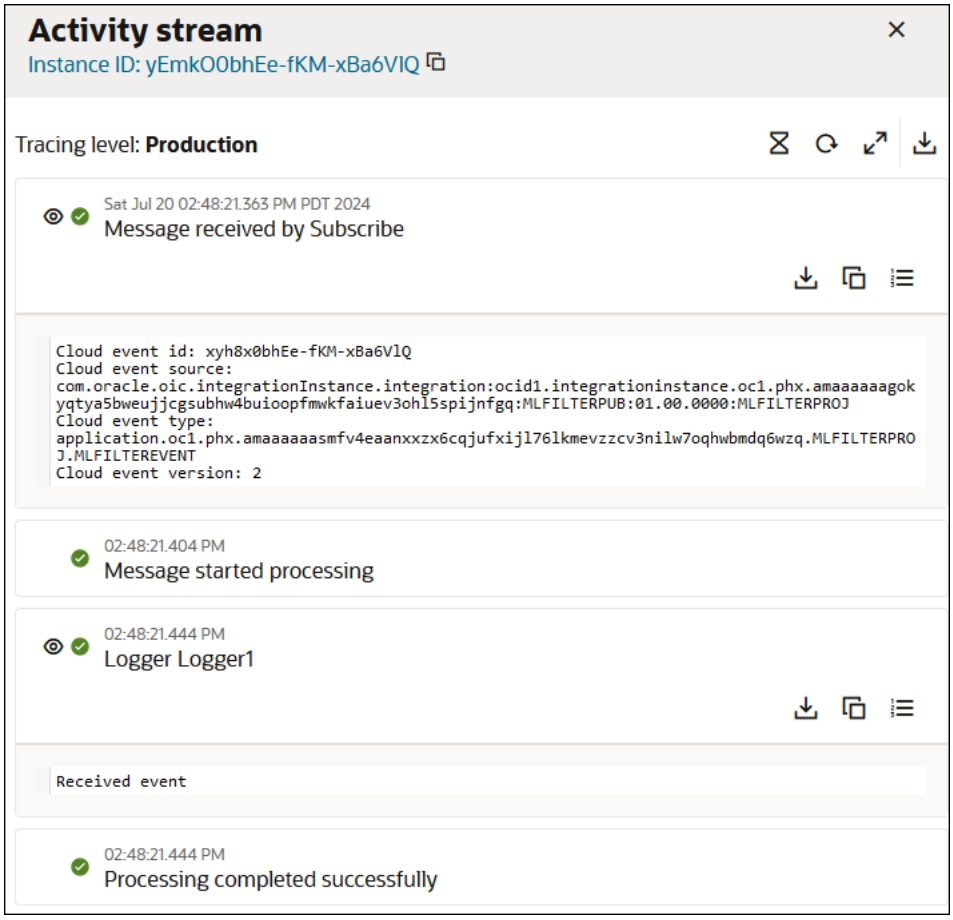
   The subscribing integration **MLFilterAgeSub** was not triggered because the filtering condition was not satisfied.
7. Open the activity stream and expand key milestones for **MLFilterSubNoFilter**.

Copyright © 2022, 2026, Oracle and/or its affiliates.



8. Open the activity stream and expand key milestones for **MLFilterNameSub**.

Copyright © 2022, 2026, Oracle and/or its affiliates.



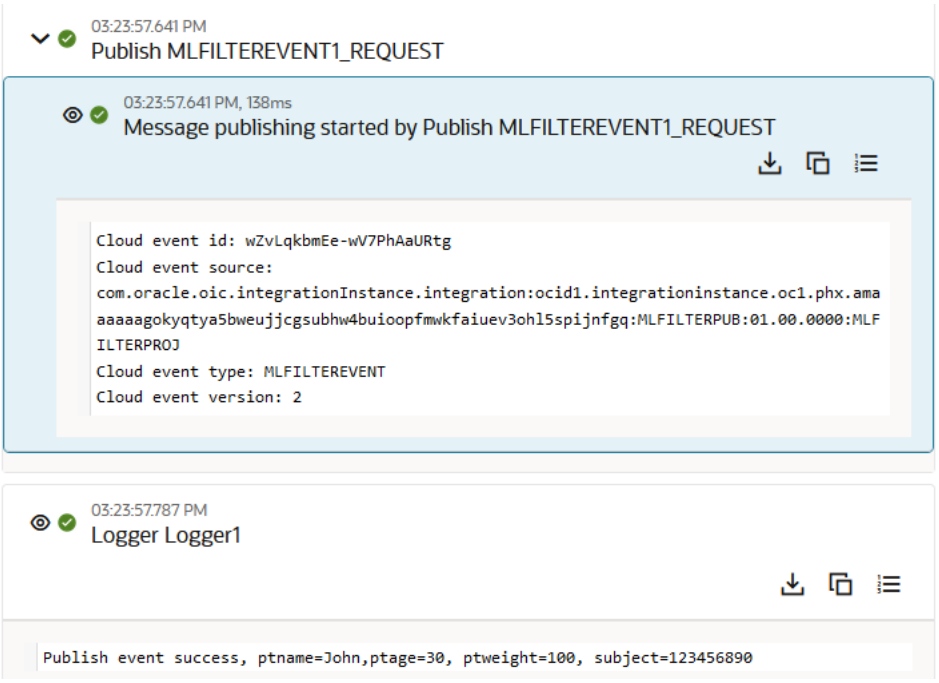
Run the Publishing Integration Again and Track the Created Instances

1. Return to the project details page.
2. Hover over the publishing integration, then click ⋯ and select **Run**.
   Under **Body**, the request payload is displayed:
3. Leave `name` set to `John`, change `age` to `30`, optionally change the `patientIdentifier` to another value, and then click **Run**.

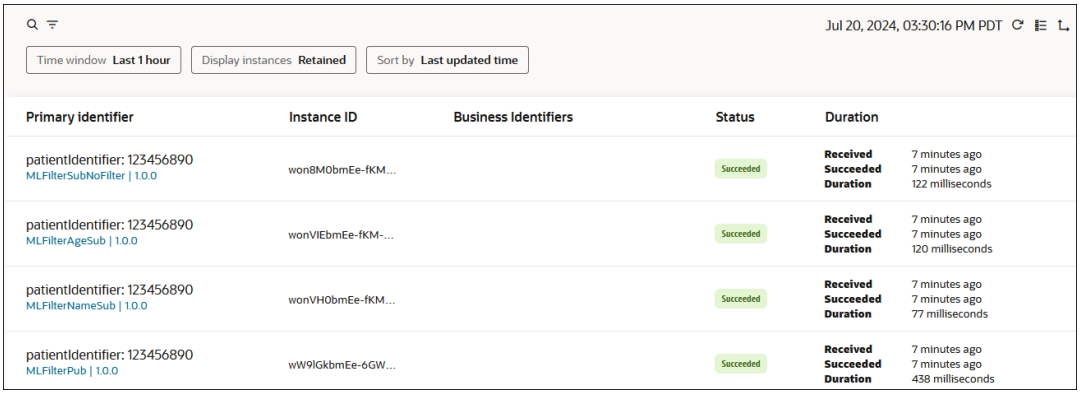
```
{
  "patientIdentifier": "123456890",
  "name": "John",
  "weight": 100,
  "age": 30,
  "illness": "1"
}
```

4. Expand key milestones in the activity stream for the publishing integration instance to see details.

Copyright © 2022, 2026, Oracle and/or its affiliates.

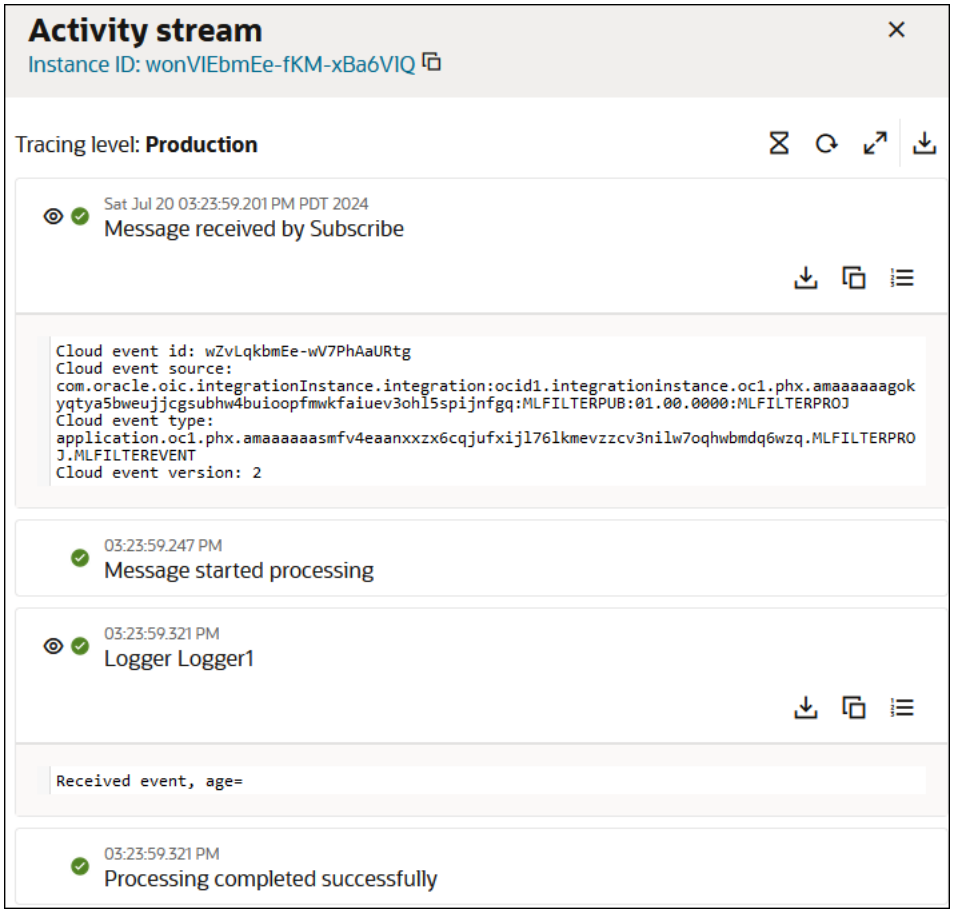


5. Return to the project details page.
6. Click **Observe**, then **Instances**.
   The following integration instances are displayed.
   - **MLFilterSubNoFilter**: The subscribing integration with no filter. This integration is triggered because no filter is set.
   - **MLFilterAgeSub**: The subscribing integration with the filter of `age` set to `30`. This integration was triggered this time because the filtering condition was satisfied.
   - **MLFilterNameSub**: The subscribing integration with the filter of `name` set to `John`. This integration was triggered again because the filtering condition was satisfied.
   - **MLFilterPub**: The publishing integration for the event.



7. Open the activity stream and expand key milestones for each instance. For this example, **MLFilterAgeSub** is expanded.

Copyright © 2022, 2026, Oracle and/or its affiliates.



JQ Expression Syntax for Creating Filters

This section describes common JQ syntax elements you can use to create filter conditions.

JQ Filter	Description	JSON Sample	Result
`[.[]]`	Selects all elements of an array.	`[1, 2, 3, 4, 5]`	`[1, 2, 3, 4, 5]`
`[.[] | .field]`	Selects a specific field from each object in an array.	`[{ "field": "value1" }, { "field": "value2" }]`	`["value1", "value2"]`
`[.[] | select(.field == "value")]`	Filters elements where a specific field equals a value.	`[{ "field": "value1" }, { "field": "value2" }]`	`[{ "field": "value1" }]`

Copyright © 2022, 2026, Oracle and/or its affiliates.

JQ Filter	Description	JSON Sample	Result
`map(has("field"))`	Extracts the results of a specific function applied.	`[{ "field": "value1" }, { "field": "value2" }]`	`[true, true]`
`[.[] | .field | select(. == "value")]`	Filters elements of an array where a specific field equals a value.	`[{ "field": "value" }, { "field": "value1" }]`	`["value"]`
`[.[] | select(length > 5)]`	Filters elements with a length greater than 5.	`["apple", "banana", "orange", "grape"]`	`["banana", "orange"]`
`[.[] | select(.field1 == "value" and .field2 > 10)]`	Selects elements where both conditions are true.	`[{"field1": "value", "field2": 15}, {"field1": "value", "field2": 20}]`	`[{"field1": "value", "field2": 15}, {"field1": "value", "field2": 20}]`
`[.[] | select(.field1 == "value" or .field2 > 10)]`	Selects elements where at least one of the conditions is true.	`[{"field1": "value", "field2": 5}, {"field1": "other_value", "field2": 15}]`	`[{"field1": "value", "field2": 5}, {"field1": "other_value", "field2": 15}]`
`[.[] | select(.field1 == "value" and .field2 < 10)]`	Selects elements where both conditions are true.	`[{ "field1": "value", "field2": 5 }, { "field1": "other_value", "field2": 15 }]`	`[{"field1": "value", "field2": 5}]`

Copyright © 2022, 2026, Oracle and/or its affiliates.

JQ Filter	Description	JSON Sample	Result
`[.[] | select(.field1 == "value" and (.field2 > 10 or .field3 == "include"))]`	Combines `and` and `or` conditions to filter elements where the first condition is true and either the second condition is true or the third condition is `include`.	`[{"field1": "value", "field2": 5, "field3": "include"}]`	`[{"field1": "value", "field2": 5, "field3": "include"}]`
`[.[] | select(.field1 != "value")]`	Selects elements where the `field1` is not equal to `value`.	`[{ "field1": "other_value", "field2": 15 }, { "field1": "value", "field2": 5 }]`	`[{"field1": "other_value", "field2": 15}]`
`[.[] | select(.field1 == "value" and (.field2 > 10 and .field3 == "include"))]`	Combines `and` conditions to filter elements where the first condition is true and both the second and third conditions are true.	`[{"field1": "value", "field2": 15, "field3": "include" }]`	`[{"field1": "value", "field2": 15, "field3": "include"}]`
`[.[] | select(.field1 == "value" or (.field2 > 10 and .field3 == "include"))]`	Combines `or` conditions to filter elements where either the first condition is true or both the second and third conditions are true.	`[{"field1": "other_value", "field2": 15, "field3": "include" }, { "field1": "value", "field2": 5, "field3": "include" }]`	`[{"field1": "other_value", "field2": 15, "field3": "include"}], [{"field1": "value", "field2": 5, "field3": "include"}]`
`[.[] | select(.field1 == "value" and .field2 > 10 and .field3 != "exclude")]`	Combines multiple `and` conditions to filter elements where all conditions are true.	`[{ "field1": "value", "field2": 15, "field3": "include" }]`	`[{"field1": "value", "field2": 15, "field3": "include"}]`

Copyright © 2022, 2026, Oracle and/or its affiliates.

JQ Filter	Description	JSON Sample	Result
`[.[] | select(.field1 != "value" and (.field2 <= 10 or .field3 == "exclude"))]`	Combines `not` and `or` conditions to filter elements where `field1` is not equal to `value` and either `field2` is less than or equal to `10` or `field3` is `exclude`.	`[{"field1": "other_value", "field2": 5, "field3": "exclude"}]`	`[{"field1": "other_value", "field2": 5, "field3": "exclude"}]`
`.[]|.foo>30 and .bar== "less interesting data"|.==true`	Returns exactly true while executing rather than returning the values.	`[{"foo": 42, "bar": "less interesting data"}]`	`true`

Share Events Across Projects

You can select to make events available across projects. This selection allows you to share the event in one project with other projects. This feature also eliminates the need to create separate events in each project.

- Guidelines
- Share an Event Across Projects

Guidelines

- Event sharing is allowed in the following use cases:
  - The publishing integration and shared event are in the same project, while the subscribing integration is in a separate project.
  - The publishing integration, shared event, and subscribing integration are all in separate projects.
  - The publishing integration and subscribing integration are in the same project while the shared event is in a separate project. This use case is described in Share an Event Across Projects.
- You can export and then import projects into another instance that share events. Any event sharing between the projects is retained.

> **Note**
>
> The order in which you import event sharing projects into another instance matters. For example, assume project A includes publishing and subscribing integrations and project B includes an event being shared (referenced) by project A. You must first import project B with the shared event into the instance before project A can be successfully imported.

- You cannot delete an event in a project that is shared with integrations in other projects.
- You cannot delete a project that includes an event that is shared with integrations in other projects.

Copyright © 2022, 2026, Oracle and/or its affiliates.

- When a shared event is updated, a newer version of the event is created. Any active integrations continue to work with the existing event version. To use the newer event version, you must deactivate the integration and edit the publish event where you can select to use the newer version.

Share an Event Across Projects

This use case provides an overview of how to use event sharing across projects. This use case consists of two projects with the following resources:

Project l2devevt	Project l2devflow
Project **l2devevt** contains the following resources: • Event **public** (shared between projects) • Event **pvtevt** (not shared between projects) There are no integrations defined in this project.	Project **l2devflow** contains the following resources: • Publishing integration **pubflow** • Subscribing integration **sub** There are no events defined in this project.

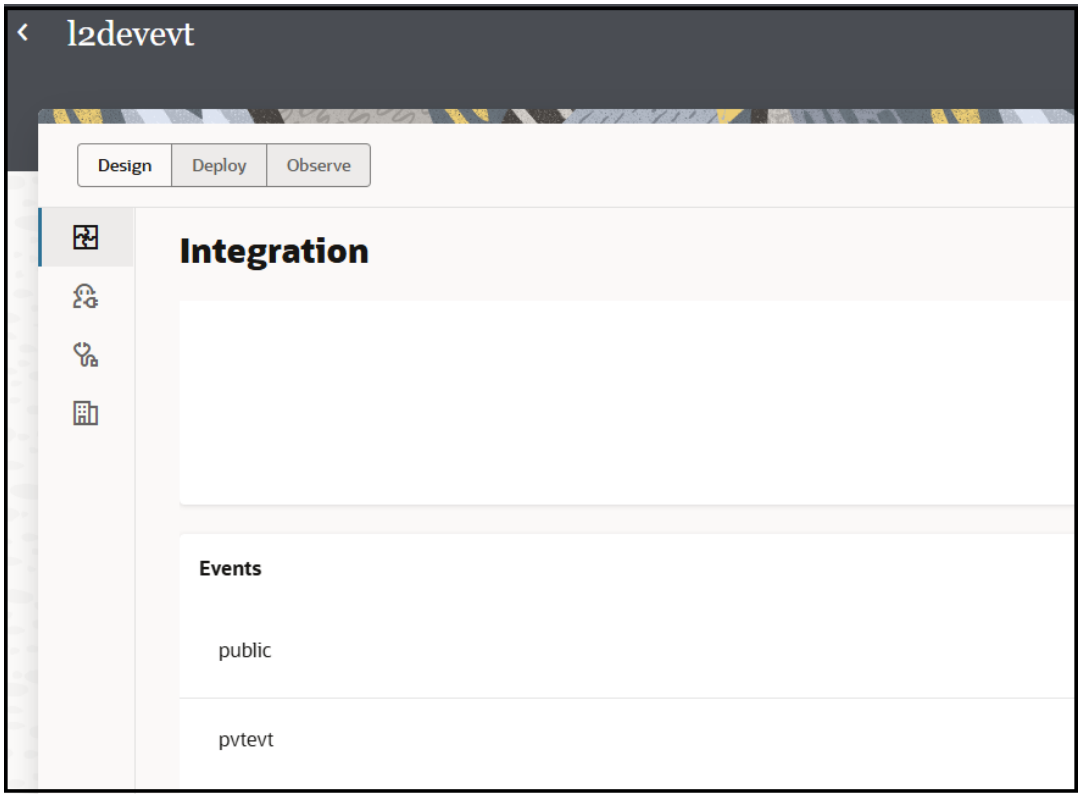
1. In the navigation pane, click **Projects**.
2. Create a project (for this example, **l2devevt** is created).
3. In the **Events** section, create an event.
4. Enter a name and an optional description (for this example, **public** is created).
5. Click **Share with other projects**. This option enables you to share the event across projects.
6. Complete the remaining steps of the wizard to define a JSON-based event.

```
{
    "input1": "value",

    "input3": false
}
```

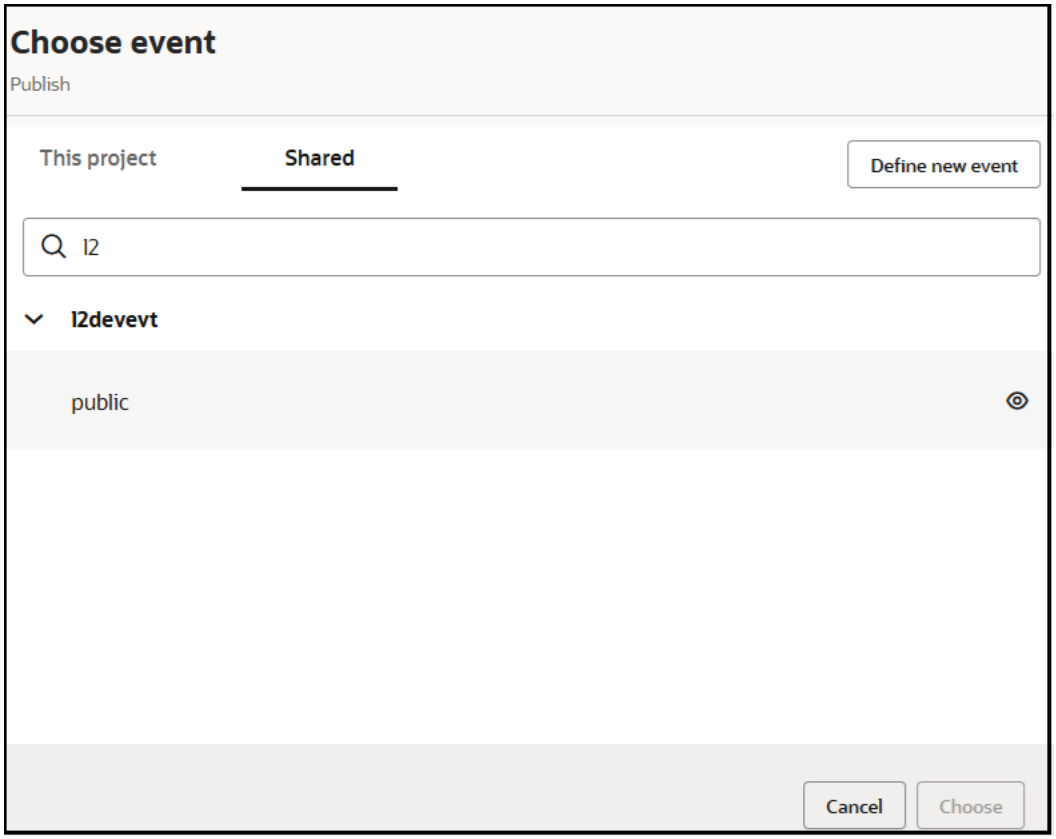
7. Create a second event, but do *not* select to share this event with other projects (for this example, **pvtevt** is created). This event can only be used in the **l2devevt** project.

Copyright © 2022, 2026, Oracle and/or its affiliates.



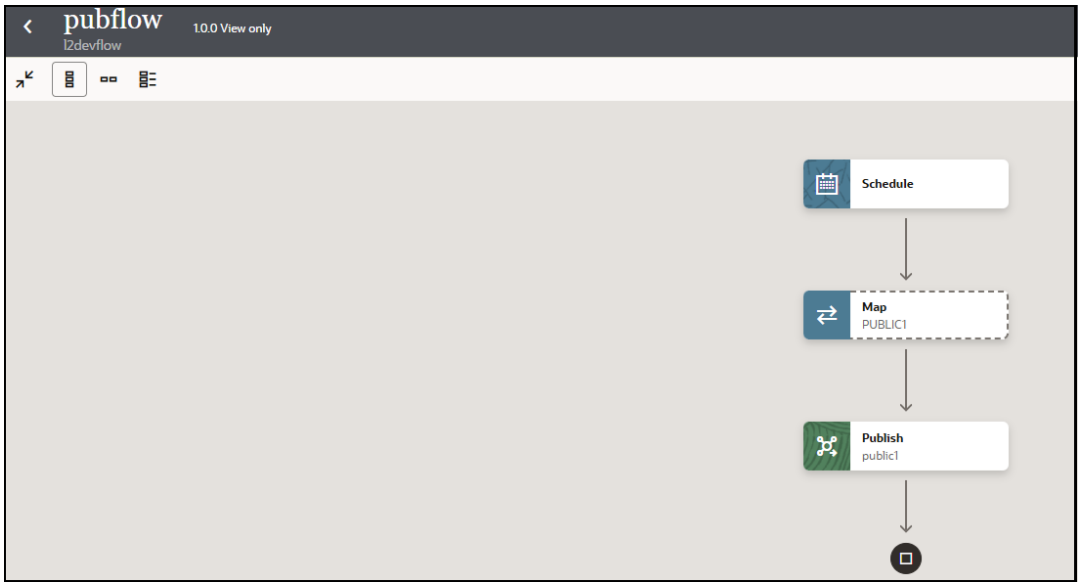
8. Create a second project (for this example, **l2devflow** is created).
9. Create a scheduled integration in this project to publish the shared event (for this example, **pubflow** is created).
10. Drag a **Publish event** integration into the integration canvas.
    The Choose event wizard is displayed.
11. Click the **Shared** tab.
12. Search for and expand the **l2devevt** project, then select **public**.

Copyright © 2022, 2026, Oracle and/or its affiliates.



Note that the **pvtevt** is not available for selection because you did *not* select to share that event with other projects.

13. Complete the remaining steps of the Choose event wizard.
When you return to the canvas, note that the word **Public** appears on the map action. This indicates that this integration is using a shared event.



14. Open the map action, and map the source elements to the target elements.

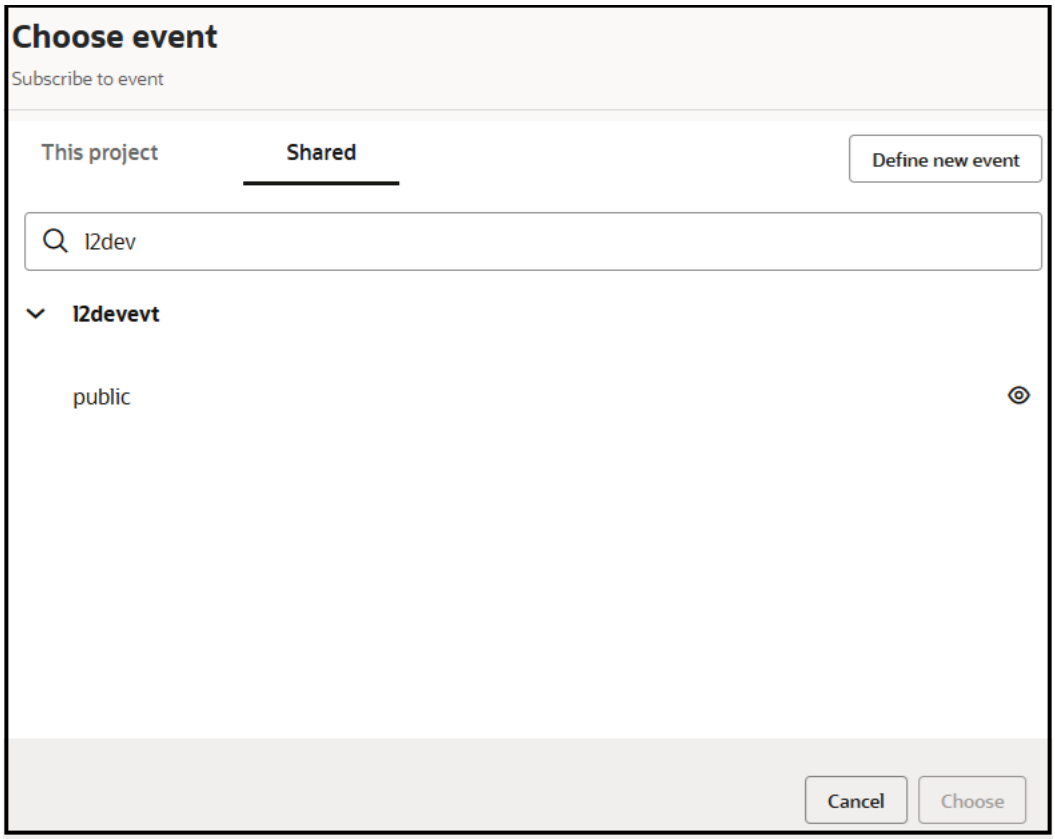
Copyright © 2022, 2026, Oracle and/or its affiliates.

15. Create an event integration to subscribe to the event (for this example, **sub** is created). A **Subscribe to event** trigger is displayed in the integration canvas.

    The Choose event wizard is displayed.

16. Click the **Shared** tab.

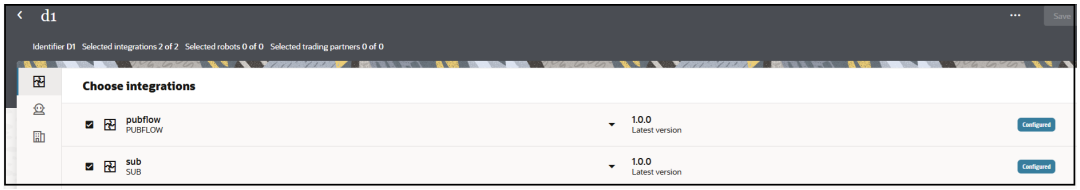
17. Select to subscribe to the same event that you published in Step 12 (**public**).



18. Define a business identifier for tracking in the subscribing integration.

19. Click **Deploy**.

20. Create a deployment that includes the publishing and subscribing integrations in **l2devevt**.

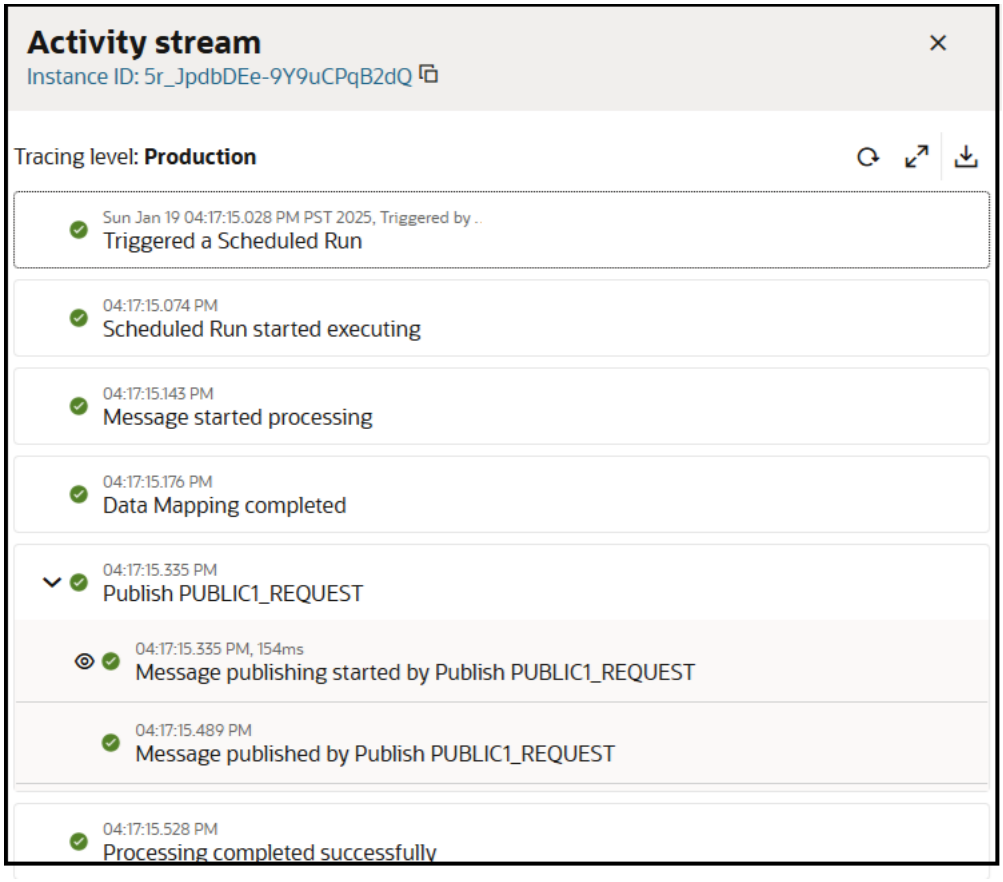


21. Activate the deployment, which activates the integrations.

22. Run the **pubflow** integration by selecting **Run** from the **Actions** ⋮ menu.

23. Click **Run**.
    View the results in the activity stream. The subscribing integration successfully subscribed to the shared event.

Copyright © 2022, 2026, Oracle and/or its affiliates.



24. Expand **Publish PUBLIC1_REQUEST**, then expand the payload to see the shared event type listed.

    ```
    Cloud event type: PUBLIC
    ```

25. Click **Observe** to view additional runtime details. For example, the **Subscription** tab shows the **sub** integration that subscribed to the published event.



Copyright © 2022, 2026, Oracle and/or its affiliates.

10

Map Data and Create Lookups

A visual mapper 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 different vocabularies or systems.

Topics:

- Map Data
- Manage Lookups

Map Data

Use the mapper to map data between the source data structure and the target data structure.

Topics:

- Access the Mapper
- Encode and Decode File Attachment Content
- Import Map Files

See Map Data of *Using the Oracle Mapper with Oracle Integration 3*.

Access the Mapper

To create mappings in an integration, you must first access the mapper.

See Access the Mapper in *Using the Oracle Mapper with Oracle Integration 3*.

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.

Copyright © 2022, 2026, Oracle and/or its affiliates.

Two XPath functions are provided to perform these tasks. These functions work with any adapter.

- `encodeReferenceToBase64(String reference)`: Accepts the VFS's file reference as input and returns the base64–encoded content of the file as the return value. This function has a file size limit of 10 MB. If a file is larger than 10 MB, an exception message of `Maximum file size supported is 10 MB` is displayed.
- `decodeBase64ToReference(String base64String)`: Accepts the base64–encoded content as input, decodes it, stores the base64–decoded value in a file in the VFS, and returns the reference to this file. There is no size limit because the content is already in memory.

  The `77u/` text at the start of an encoded base64 string is related to the byte order mark and indicates the encoding scheme used (UTF-8). This is not needed for decoding of a base64 string to a binary.

  This is needed for representing binary data in a string. This text must be extracted and removed from the original base64 string to ensure it doesn't cause decoding issues. If this is not removed from the encoded string, the decoded data, if converted to a string, has additional space.

  You can extract and remove this from the base64-encoded string with the help of standard mapper functions for strings. This information is not needed to decode the base64-encoded string to binary data. This is needed for reading the binary data using UTF-8 encoding.

The location is the relative path (reference) of the file stored in Oracle Integration. The relative path is one of the following elements:

- `fileReference`
- `attachmentReference`
- `streamReference`

The two XPath functions are available for use in Oracle Integration:

- Expression Builder, when configuring the following actions in an integration:
  - Notification
  - Logging
  - Switch
  - Assign
- Mapper (visible after selecting  > **Functions** > **Advanced**)

The attachments are not restricted to document file types. For example, an image can be base64–encoded and later decoded back to the original file.

When an attachment is stored in the VFS, a key is generated to retrieve the attachment at a later time. The key is shown in the mapper as `attachmentReference/fileReference/streamReference`. This key is propagated within Oracle Integration as part of the payload. The attachment is claimed only when needed. The names `attachmentReference`, `fileReference`, and `streamReference` are based on the adapter type. For example, in the REST Adapter, `streamReference` is used. The data type of the reference is a string.

With a multipart feature, the HTTP request payload has multiple parts separated by boundaries. Each of the individual parts are considered an attachment. For raw bytes, `streamReference` is used. FTP uses `fileReference`.

Copyright © 2022, 2026, Oracle and/or its affiliates.

Sometimes the endpoints accept only base64–encoded values. In these cases, the reference is passed as input to `encodeReferenceToBase64` to get the base64–encoded content of the file. Again, the base64–encoded value can be passed as input to `decodeBase64ToReference` to get the reference (location) to a file that contains the decoded content.

Import Map Files

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

> **Note**
>
> As an alternative, you can directly edit XSLT code in the mapper. This eliminates the need to export your XSLT code from Oracle Integration, edit the code manually in a text editor or in a separate graphical tool such as Eclipse or Oracle JDeveloper, and then re-import the code into the mapper in Oracle Integration. See Edit the XSLT Code in the Mapper in *Using the Oracle Mapper with Oracle Integration 3*.

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**.

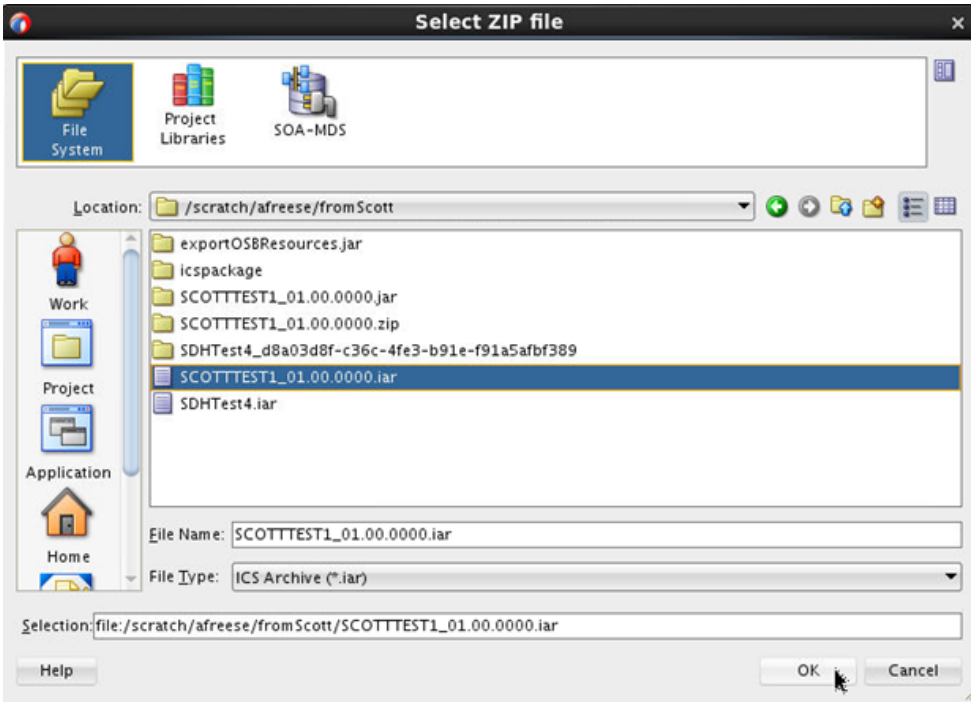
   The Import dialog is displayed.
4. Select **Service Bus Resources**, and click **OK**.

   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)**.

Copyright © 2022, 2026, Oracle and/or its affiliates.

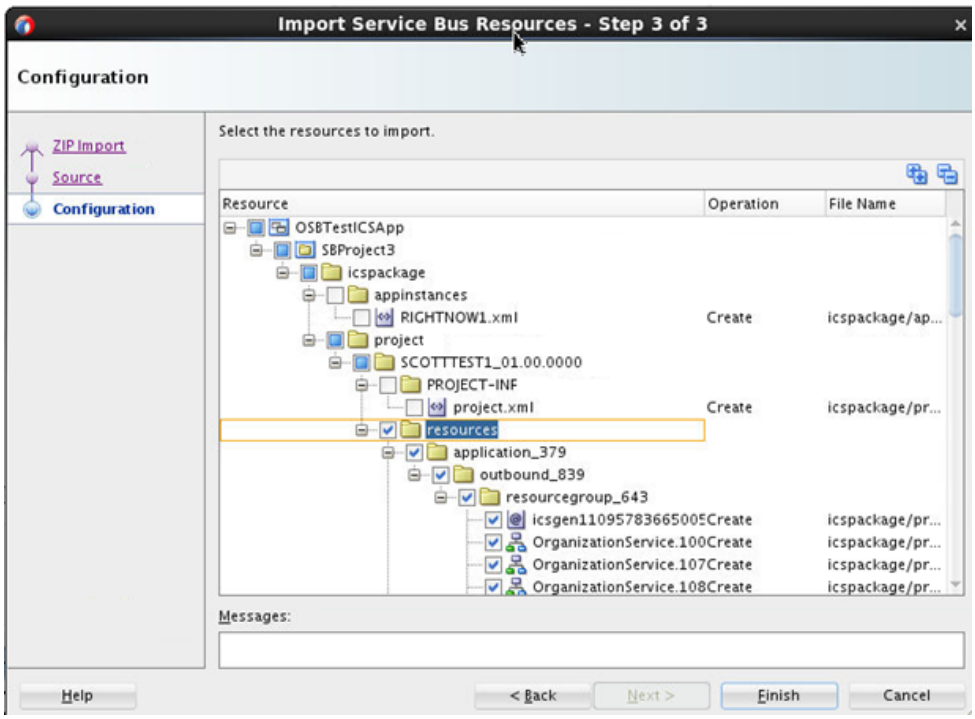
b. Browse for and select the Oracle Integration IAR archive file that you previously exported.



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**.

Copyright © 2022, 2026, Oracle and/or its affiliates.



The resources are imported into the Oracle Service Bus project. You can now open the map file for editing with the XSLT Map Editor in Oracle JDeveloper.

Import a Map File into Oracle Integration

There may be scenarios in which you need to perform an advanced XSLT task (create variables, use templates, and so on) that you cannot perform in the Oracle Integration mapper. For these cases, you can export the integration, import the integration into Oracle JDeveloper, perform these advanced tasks in the map file in the XSLT Map Editor in Oracle JDeveloper, and then save and re-import the map file into Oracle Integration. The map file must be from an Oracle Service Project in Oracle JDeveloper.

1. In the navigation pane, click **Design**, then **Integrations**.
2. Click the specific integration in which to import the map file.
3. Click the mapper icon, then select ⋮, then **Import**.
4. 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.
5. Click **Import**.

Manage Lookups

A lookup associates values used by one application for a specific field with the values used by other applications for the same field. This capability enables you to map values across

Copyright © 2022, 2026, Oracle and/or its affiliates.

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
- Clone a Lookup
- Delete a Lookup
- Edit a Lookup
- Update Lookups in Active Integrations
- Export a Lookup
- Import a Lookup
- Create an AI-Powered Lookup Description
- Create the lookupValue Function

Create a Lookup

Create a lookup to map values between applications.

1. Decide where to start:
   - Work in a project (see why working with projects is preferred).
     a. In the navigation pane, click **Projects**.
     b. Select the project name.
     c. Click **Integrations**.
     d. In the **Lookups** section, click **Add** if no lookups currently exist or **+** if lookups already exist. The Add lookup panel opens.
     e. Click **Create**. The Create lookup panel opens.
   - Work outside a project.
     a. In the navigation pane, click **Design**, then **Lookups**.
     b. In the banner, click **Create**. The Create lookup panel opens.
2. Enter the following information.

Field	Description
Name	Provide a meaningful name so that others can understand the lookup. You can include English alphabetic characters, numbers, underscores, and dashes in the identifier.
Keywords	Enter keywords (tags) to identify the lookup. You can search for lookups using keywords.

Copyright © 2022, 2026, Oracle and/or its affiliates.

Field	Description
Description	Provide a meaningful description so that others can understand the lookup.
Share with other projects	**Note**: This field only appears if you are creating a lookup in a project. Select to make this lookup publicly available in other projects. Lookup sharing eliminates the need to create and maintain separate lookups in different projects. After creation, the **Shared with other projects** field is displayed as **On** at the top of the details page of the lookup. You can reference that lookup in other projects to inherit its resources. See Add and Share a Lookup Across a Project.

3. Click **Create**.

   The lookup is stored in encrypted format.

Add Adapters or Domain Names to a Lookup

Add adapters or domain names to a lookup to map values between connections.

1. Click **Domain_Name_1**.

   A menu with options is displayed.

2. Select the type to create:

   - **Replace with adapter**: This selection invokes the Select adapter dialog for choosing the adapter to use. You can change your selection:
     - If you want to change your adapter selection, click the selected adapter type, then select a different adapter.
     - If you want to change from an adapter to a domain name, click the selected adapter type and change to a domain name.
   - **Edit 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 select **Replace with adapter**.
     - If you want to change the domain name, click the name, then select **Edit domain name**.

     The adapter or domain name is displayed.

3. Enter a value in the field after 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 after the adapter or domain name. To add more rows for additional values, click the **+** sign.

Copyright © 2022, 2026, Oracle and/or its affiliates.



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 in Oracle Integration. Cloning provides a quick way to create a new lookup with similar information. Lookups with adapters and domain names can be cloned.

1. Decide where to start:
   - Work in a project.
     a. In the navigation pane, click **Projects**.
     b. Select the project name.
     c. Click **Integrations**.
     d. In the **Lookups** section, hover over the lookup to clone.
   - Work outside a project.
     a. In the navigation pane, click **Design**, then **Lookups**.
     b. Hover over the lookup to clone.
2. Click **Actions** ⋮, then select **Clone**.

   The Clone lookup panel opens.
3. Enter a name and an optional keyword and description.
4. Click **Clone**.

Delete a Lookup

You can delete lookups in Oracle Integration that are not currently in use by integrations. Attempting to delete a lookup in use would invalidate the integration and cause an active integration to stop running. You must first remove the lookup from the integration and then delete the lookup.

1. Decide where to start:
   - Work in a project.
     a. In the navigation pane, click **Projects**.
     b. Select the project name.
     c. Click **Integrations**.
     d. In the **Lookups** section, hover over the lookup to delete.
   - Work outside a project.
     a. In the navigation pane, click **Design**, then **Lookups**.

Copyright © 2022, 2026, Oracle and/or its affiliates.

b. Hover over the lookup to delete.

2. Click **Actions** ⋮, then select **Delete**.

   The Delete lookup dialog is displayed.

3. Click **Delete** when prompted to confirm.

Edit a Lookup

You can edit a lookup.

1. Decide where to start:
   - Work in a project.
     a. In the navigation pane, click **Projects**.
     b. Select the project name.
     c. Click **Integrations**.
     d. In the **Lookups** section, find the lookup to edit.
   - Work outside a project.
     a. In the navigation pane, click **Design**, then **Lookups**.
     b. Hover over the lookup to edit.
2. Click **Actions** ⋮ and select **Edit**.

Update Lookups in Active Integrations

Lookups are not deployed as part of integration activation. Therefore, changes that you make to lookups already used in active integrations typically take effect immediately. There is *no* need to re-activate integrations using a changed lookup for the new lookup value to take effect.

There can be instances in which updates to the runtime environment are slow because it is not in real-time and queuing can occur at runtime. For those cases, updates can occur in a couple of minutes.

Export a Lookup

Once you create a lookup, you can export that lookup for use in other Oracle Integration environments.

1. Decide where to start:
   - Work in a project.
     a. In the navigation pane, click **Projects**.
     b. Select the project name.
     c. Click **Integrations**.
     d. In the **Lookups** section, click the name of the lookup to export.
   - Work outside a project.
     a. In the navigation pane, click **Design**, then **Lookups**.
     b. Hover over the lookup to export.

Copyright © 2022, 2026, Oracle and/or its affiliates.

2. Click **Actions** ⋮, then select **Export to CVS**.

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
```

1. Locate the CSV file containing the lookup table you want to import.
2. Decide where to start:
   - Work in a project.
     a. In the navigation pane, click **Projects**.
     b. Select the project name.
     c. Click **Integrations**.
     d. In the **Lookups** section, click **Add** if no lookups currently exist or **+** if lookups already exist. The Add lookup panel opens.
     e. Click **Import**. The Import lookup panel opens.
   - Work outside a project.
     a. In the navigation pane, click **Design**, then **Lookups**.
     b. In the banner, click **Import**. The Import lookup panel opens.
3. Click **Drag and Drop** to select a lookup or drag one to this field.
4. 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. You can customize or activate it, if it is ready.

Create an AI-Powered Lookup Description

You can use AI to generate a new description or replace an existing description of a lookup with a single click.

See AI Innovation and Oracle Integration.

1. Decide where to start:
   - Work in a project.
     a. In the navigation pane, click **Projects**.
     b. Select the project name.
     c. Click **Integrations**.
     d. In the **Lookups** section, hover over the lookup for which to create an AI-generated description.

Copyright © 2022, 2026, Oracle and/or its affiliates.

e. Click **Actions** ⋮, then select **Edit**.

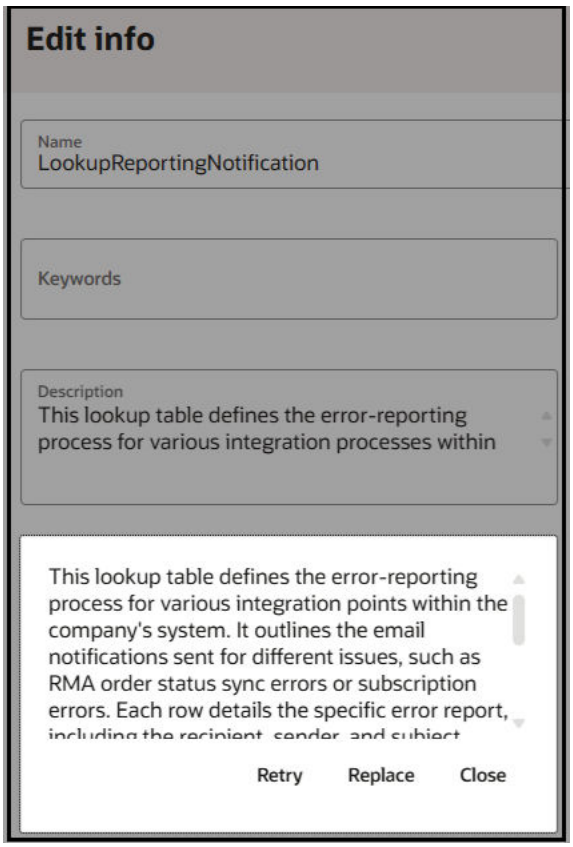
- Work outside a project.
  a. In the navigation pane, click **Design**, then **Lookups**.
  b. Hover over the lookup for which to create an AI-generated description.
  c. Click **Actions** ⋮, then select **Edit**.

2. In the upper right corner, click **Actions** ⋮, then select **Edit info**.
3. Click **Generate**.

> **Note**
>
> - If you have not configured an AI engine, an error message is displayed when you click **Generate**.
> - If you have reached the limit on your number of API keys, you receive an unauthorized key error.

The **Description** field can be empty or already include a description.

- If the **Description** field is empty, the generated lookup description is added to the field.
- If the **Description** field already includes a lookup description, the updated description is displayed as a popup. For this example, a description already existed.



4. If the **Description** field already includes a description, perform one of the following steps:

Copyright © 2022, 2026, Oracle and/or its affiliates.

a. Click **Retry** to regenerate the description. If you retry, the newer description varies slightly in terms of the words used, but the meaning remains the same.

b. Click **Replace** to replace the current description in the **Description** section. **Replace** is only selectable if the lookup is editable.

5. Click outside the Edit info panel.
6. Click **Save**.

Create the lookupValue Function

You can create the parameter values for the `lookupValue` function with the Map Lookup Value 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
- Select the Source and Target Columns
- Specify the Default Value
- Review Your Lookup Table Selections

Access the Build Lookup Function Wizard

The Map Lookup Value Wizard for creating the `lookupValue` function parameter values is accessible from the mapper or the Expression Builder in actions that support functions in Oracle Integration.

s

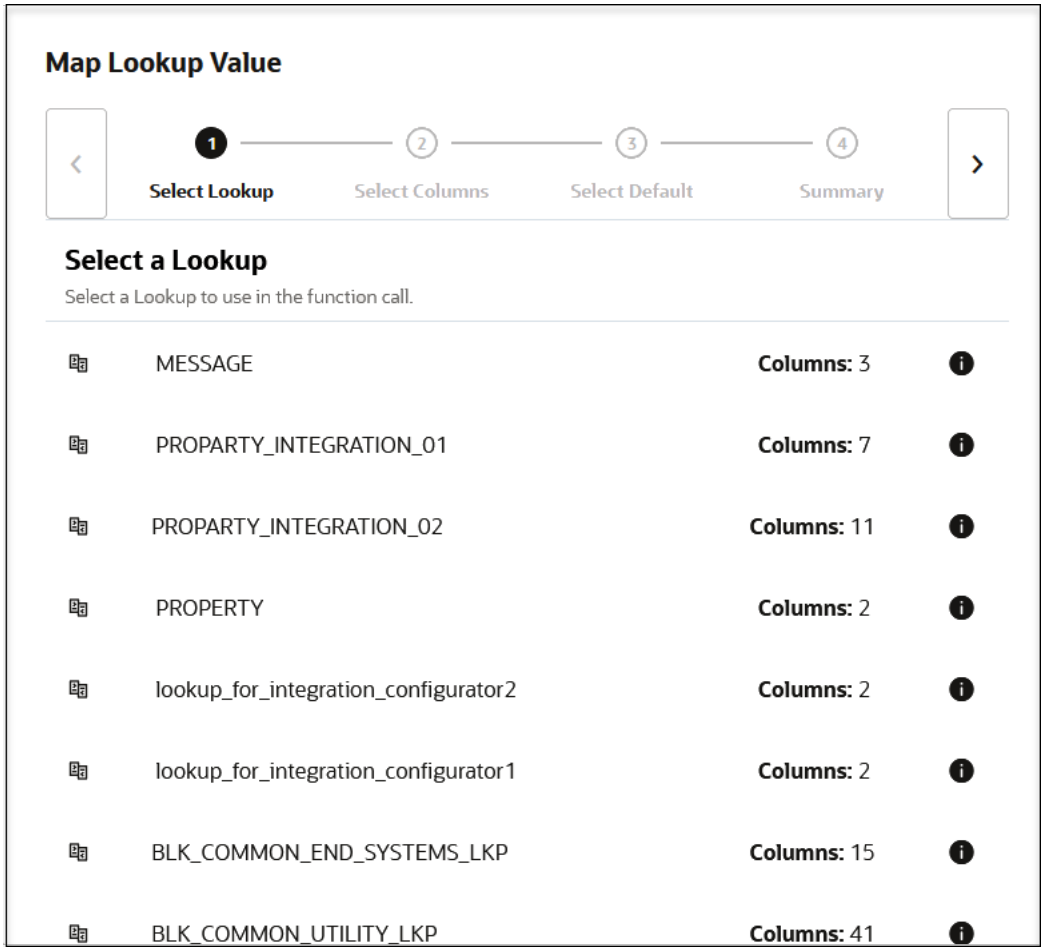
> **Note**
>
> You must already have created lookups to use this wizard.

- From the Mapper
- From the Expression Builder In an Action that Supports Functions

From the Mapper

1. In the upper right corner, click **Toggle functions**  to launch the **Components** panel.
2. Type `lookupValue` in the **Search** field, and click **Search**.
3. Drag the function onto the target element node.
   The Map Lookup Value Wizard is displayed.

Copyright © 2022, 2026, Oracle and/or its affiliates.



From the Expression Builder in an Action that Supports Functions

You can access the Map Lookup Value Wizard in the Expression Builder in an action that supports functions, such as a switch, stitch, or others.

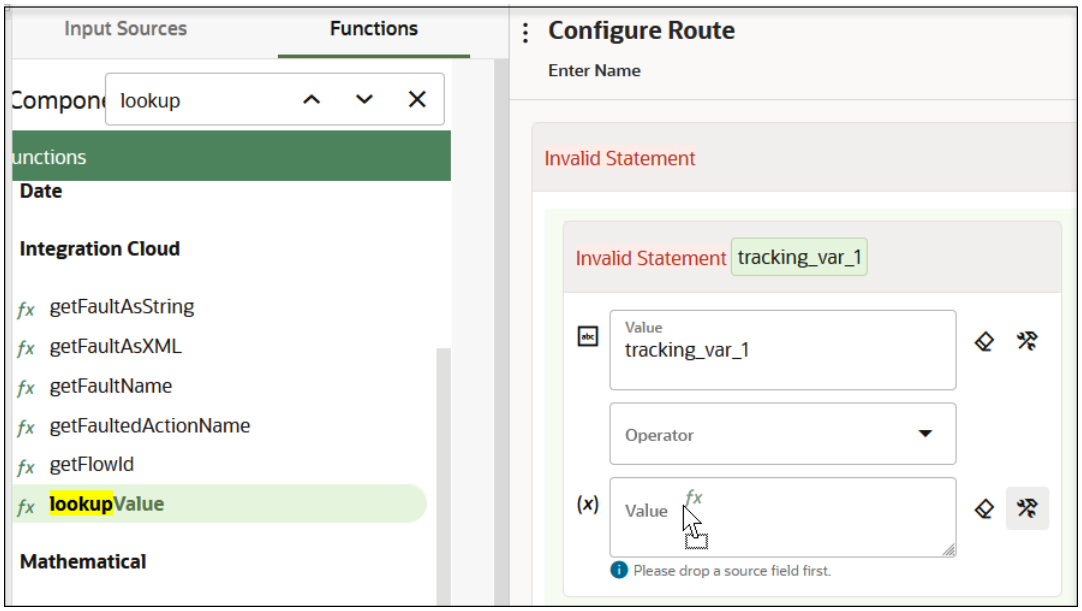
1. Open an integration for editing.
2. Open an action that supports functions in the integration (for this example, a switch action is shown, but other actions such as a stitch can be used).

   The Configure route panel opens.
3. Manually enter a value, select a value from the drop-down list, or drag a value from the **Sources** tree to the first **Value** field. For this example, **tracking_var_1** is selected.
4. Select an operator in the **Operator** field.
5. Click **Functions**, and browse or search for `lookupValue`.
6. Click `lookupValue`.
7. View the description after the function for information about the supported parameters. You define values for these parameters in the Map Lookup Value Wizard.
   - `dvm-location`
   - `src-column`
   - `src-value`

Copyright © 2022, 2026, Oracle and/or its affiliates.

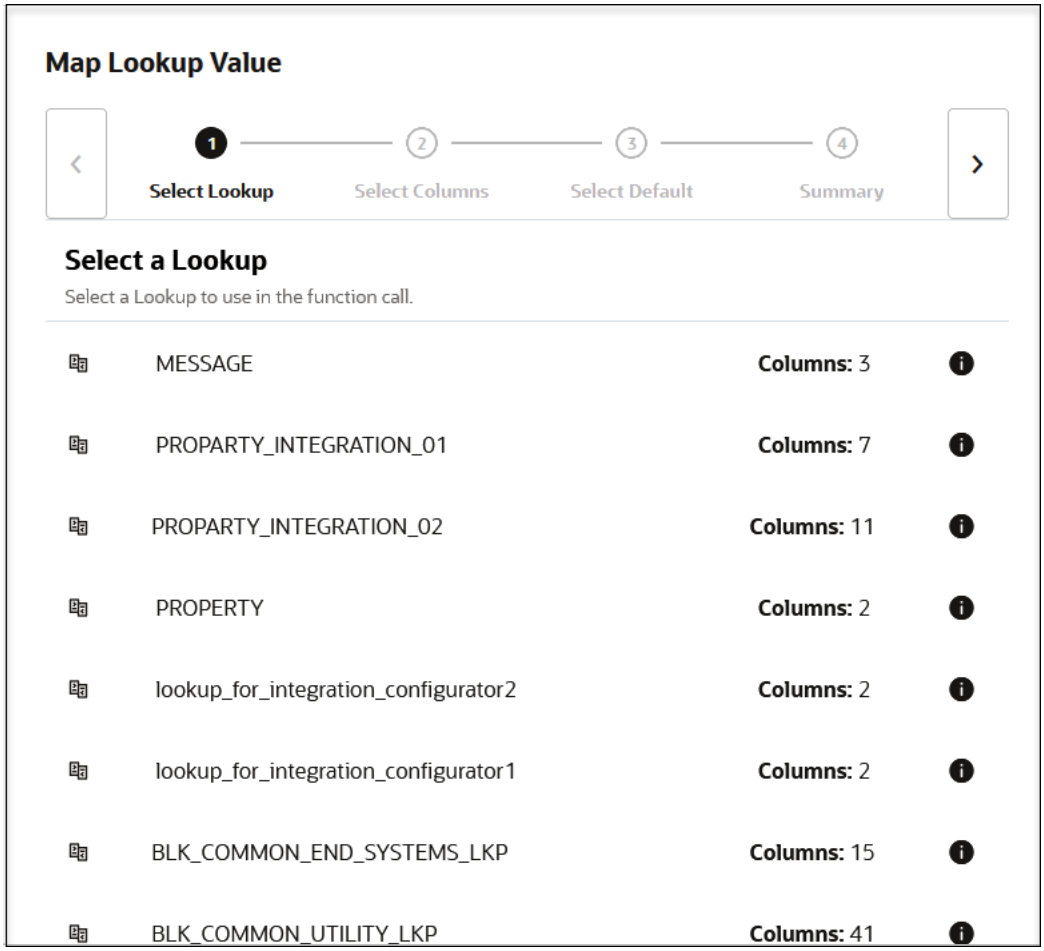
- `target-column`
- `default-value`

8. Click **Switch to Developer View** in the second **Value** field.
9. Drag the `lookupValue` function into the second **Value** field.



The Map Lookup Value Wizard is displayed.

Copyright © 2022, 2026, Oracle and/or its affiliates.



Create the `lookupValue` function parameter values.

Select the Lookup

You can select the lookup to use in the `lookupValue` function. You must already have created a lookup. Otherwise, no lookups are displayed for selection.

1. Select the lookup to use in the function. You can view the lookup description by clicking the information icon in the row. This can guide you in selecting the required lookup. The number of columns defined in the lookup is also displayed.
2. Click **Next >**.

Select the Source and Target Columns

You can 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 the source and target columns, the values available with the columns are displayed.

1. Select the source and target columns to use in the `lookupValue` function.

Copyright © 2022, 2026, Oracle and/or its affiliates.

Element	Description
Select Source	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	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.

2. Click **Next >**.

Specify the Default Value

You must specify a default value to use.

1. Enter the default value to use in the `lookupValue` function 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 (for example, an actual default value to use or an error message such as `No Value Found`).
2. Click **Next >**.

Review Your Lookup Table Selections

You can review the lookup table values to use in the `lookupValue` function on the Summary page.

1. Review the lookup table values. The Summary page is the final wizard page after you complete your configuration.

Element	Description
Parameter and Value Table	Displays a summary of the parameters and values you defined on previous pages of the wizard. To return to a previous page to update any values, click the appropriate tab in the left panel or click **Back**  .

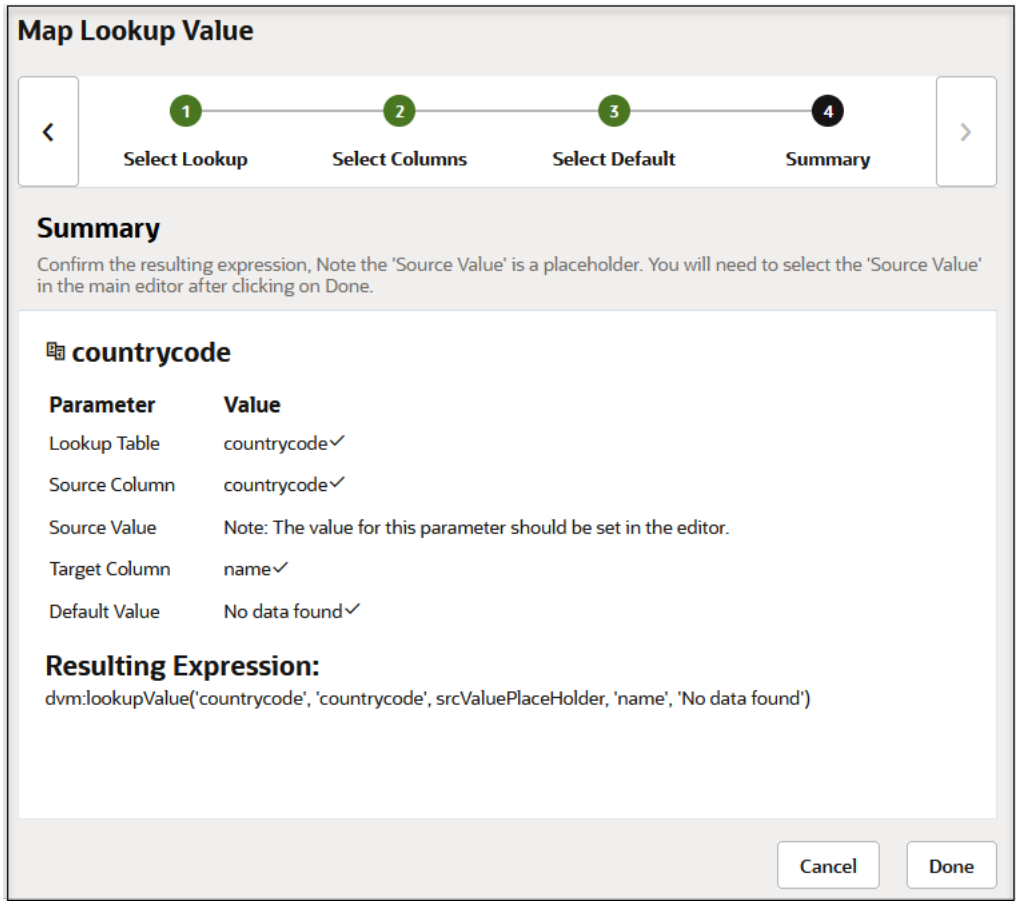
Copyright © 2022, 2026, Oracle and/or its affiliates.

Element	Description

Resulting Expression	Displays the expression you defined on the previous pages of the wizard. The `lookupValue` function takes the following format:
	
	`lookupValue(dvm-location, src-column, src-value, target-column, default-value)`
	
	Where:
	• **Lookup Table**: The lookup table selected on the Select lookup page.
	• **Source Column**: The source column selected on the Select columns page.
	• **Source Value**: The source value you enter in the mapper or Expression Builder after completing this wizard. Complete this wizard, then define the `srcValue` parameter value.
	• **Target Column**: The target column selected on the Select columns page.
	• **Default Value**: The default value entered on the Default value page.
	For example, a defined `lookupValue` function after you have completed the wizard and defined the `srcValue` parameter value in the Expression Builder can look as follows:
	
	`dvm:lookupValue('countrycode', 'countrycode', srcValuePlaceHolder, 'name', 'No data found')`

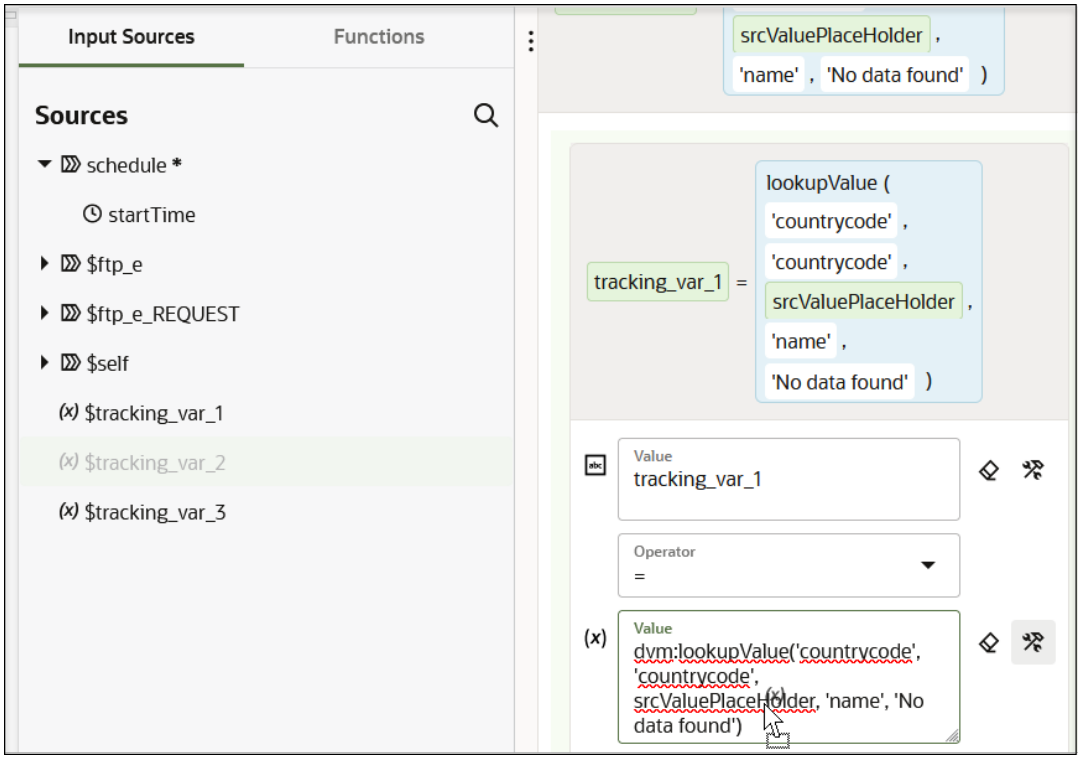
For example:

Copyright © 2022, 2026, Oracle and/or its affiliates.



2. Click **Done**.
3. In the **Sources** tree, drag a value to replace **srcValuePlaceHolder** in the **Value** field.

Copyright © 2022, 2026, Oracle and/or its affiliates.



4. Click **Save**.
You can use the configured function in the mapper. See Add Functions, Operators, and XSLT Statements in *Using the Oracle Mapper with Oracle Integration 3*.

Copyright © 2022, 2026, Oracle and/or its affiliates.

11

Use JavaScript Libraries in Integrations

You can add JavaScript libraries into integrations.

Topics:

- Use Libraries to Manage Integration Functions
- View Library Details
- Edit Library Function Values
- Determine Which Integrations Use a Library Function
- Delete a Library
- Import a Library File
- Create an AI-Powered Library Description
- Export a Library File
- Update JavaScript Files
- Use Library Functions in Integrations

Use Libraries to Manage Integration Functions

You can include JavaScript libraries in integrations. JavaScript libraries provide a means to create and organize JavaScript for use in integrations. Library functions are available for you to drag into your integrations.

You can use JavaScript library functionality to:

- Create a new JavaScript library
- See a list of all created JavaScript libraries
- View the details of a JavaScript library
- Edit the details of a JavaScript library
- Determine which integrations are using library functions
- Delete a JavaScript library
- Import functions
- Download a library source file to your desktop
- Add library functions in an integration

JavaScript libraries are not provided out of the box. You create your own libraries. However, the Expression Builder and the mapper automatically provide built-in functions. See Add Functions, Operators, and XSLT Statements in *Using the Oracle Mapper with Oracle Integration 3*.

Copyright © 2022, 2026, Oracle and/or its affiliates.

View Library Details

You can view details about a library without changing any information.

1. In the navigation pane, click **Design**, then **Libraries**.
2. Hover your cursor over the library to view, then click **View** .

   The files and their constituent functions are displayed. You can expand and contract the list of functions and search and filter the list. To view information about a function, click the function. To view the source code, click the file name and download the code to your desktop. You can also click the file name to annotate the entry with your comments.

Edit Library Function Values

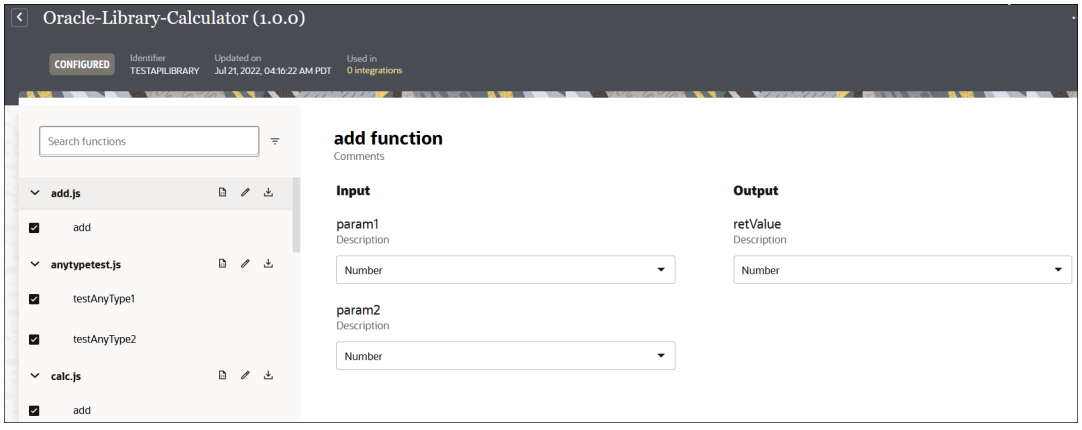
You can edit JavaScript library functions and files in Oracle Integration. The JavaScript library can consist of a single file or multiple JavaScript files in a JAR file.

1. In the navigation pane, click **Design**, then **Libraries**.
2. Either click the library name or click **Edit** .

   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.
3. To edit information about the input and output values of a function, click the function.

   If the function is in use in an integration, you are warned about it and the function is presented in view-only mode.

   If the function is available for editing, you can set the data types of the function's input and output values, defining them as **Boolean**, **Number**, or **String**.



4. When done, click **Save**.

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 **Design**, then **Libraries**.

Copyright © 2022, 2026, Oracle and/or its affiliates.

2. Go to the row of the library to check.
3. In the **Usage** column, click the number. This value indicates the number of integrations that use functions in this library.

   When you click the number, the Library is in use dialog is displayed. It lists the names of the integrations that use the library's functions. This dialog is also invoked if you attempt to edit or delete a library that is currently in use.

> **Note**
>
> If you attempt to activate an integration that includes a function that is not completely configured, an error message is displayed in the banner. You must complete configuration of this function before you can activate the integration. See Activate an Integration.

Delete a Library

You can delete a library not currently in use by an integration. If any of the library's functions are used by an integration, you are prevented from deleting the library. You must remove the functions from the integration before you can delete the library.

1. Decide where to start:
   - Work in a project.
     a. In the navigation pane, click **Projects**.
     b. Select the project name.
     c. Click **Integrations**.
     d. In the **Libraries** section, hover over the library to delete.
   - Work outside a project.
     a. In the navigation pane, click **Design**, then **Libraries**.
     b. Hover over the library to delete.
2. Click **Actions**, then select **Delete**.

   If any of the library's functions are used by an integration, the Library is in use dialog is displayed and you are prevented from deleting the library. You must remove the functions from the integration before you can delete the library.

Import a Library File

You can import a `.zip`, `.jar`, or `.js` library file into Oracle Integration. The ZIP file must contain only the library JAR and metadata XML file.

1. Decide where to start:
   - Work in a project.
     a. In the navigation pane, click **Projects**.
     b. Select the project name.
     c. Click **Integrations**.

Copyright © 2022, 2026, Oracle and/or its affiliates.

d. In the **Libraries** section, click **Add** if no libraries currently exist or **+** if libraries already exist.
The Import panel opens. Only those libraries within the project can be used by JavaScript callouts.

- Work outside a project.

a. In the navigation pane, click **Design**, then **Libraries**.

b. Click **Import**.
The Import panel opens.

2. Click **Drag and Drop** to select a file or drag one to this field.
3. Make updates, when necessary, then click **Save**.

Write JavaScript

JavaScript provides an alternative to using the mapper and assign action to enrich data, but not to develop application logic.

Refer to your JavaScript document when writing or copying JavaScript code to run in a development (sandbox) environment. See JavaScript. This documentation provides basic syntax and available constructs. Because the script runs in a development environment, network, file, and Java APIs are not available.

Code Format

Name function return parameters as shown in the following example. You must configure the `return` parameter type in the metadata user interface only if the `return` parameters are named. Consider the following code:

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

Although the above syntax is valid, the `return` parameter does not appear in the metadata user interface for configuration. For that to occur, the code must return a named parameter. Create a new variable and assign return values before the variable is returned. For example:

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

Function Within Another Function

Functions defined within another function do not appear in the metadata user interface for configuration. You may need to refactor the code if you need to access the function from the callout or mapper.

In the following code, `foo` is defined within `parseDate` and does not appear in the metadata user interface for configuration.

```
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
```


Copyright © 2022, 2026, Oracle and/or its affiliates.

```
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;
}
```

Extensions

Oracle Integration provides a set of extension functions organized under the following namespace:

Function	Example

`uuid` functions:
 	
`oic.uuid`
`uuidv1`
`uuidv4`
	`function uuidv1() {`
	`  var uuidv1_result = oic.uuid.uuidv1();`
	`  return uuidv1_result;`
	`}`
	
	`function uuidv4() {`
	`  var uuidv4_result = oic.uuid.uuidv4();`
	`  return uuidv4_result;`
	`}`

`checksum` functions:
 	
`oic.checksum`
`sha`
`sha256`
`sha512`
`md5`
	`function checksum_sha(inputStr) {`
	`  var sha_result = oic.checksum.sha(inputStr,"sha");`
	`  return sha_result;`
	`}`
	
	`function checksum_sha256(inputStr) {`
	`  var sha256_result =`
	`oic.checksum.sha256(inputStr,"sha-256");`
	`  return sha256_result;`
	`}`
	
	`function checksum_sha512(inputStr) {`
	`  var sha512_result =`
	`oic.checksum.sha512(inputStr,"sha-512");`
	`  return sha512_result;`
	`}`
	
	`function checksum_md5(inputStr) {`
	`  var md5_result = oic.checksum.md5(inputStr,"md5");`
	`  return md5_result;`
	`}`
	`}`

Copyright © 2022, 2026, Oracle and/or its affiliates.

Function	Example

base64 functions:
 	
oic.base64
encode
decode
	function base64_encode(input_encode_str) {
	var encode_result =
	oic.base64.encode(input_encode_str);
	return encode_result;
	}
	function base64_decode(input_decode_str) {
	var encode_result =
	oic.base64.decode(input_decode_str);
	return encode_result;
	}

crypto functions:
 	
oic.crypto
hmacsha256
	function generate_hashCode(hmacString_value,
	hmacKey_value) {
	var hashCode_value = oic.crypto.hmacsha256
	(hmacString_value, hmacKey_value);
	return hashCode_value;
	}

Copyright © 2022, 2026, Oracle and/or its affiliates.

Function	Example

`oic.date`
• `zonedDateTime`: Computes the zoned time from one zone to another and returns the formatted time zone in ISO date time format without the time zone data in the target time zone. For the same date and time zone data, this function returns the same results. However, because of daylight savings time (DST) rules, the result may be different depending on the date given. For example, DST starts earlier in the USA than in Europe and for a period of two weeks the offset is one hour less.
• `timeZoneOffset`: Computes the time zone offset given a date, a source time zone, and a target time zone and returns the time offset in milliseconds between the time zones on the given date and time. For the same date and time zones, this function returns the same results. However, because of DST rules, the time offset may be different depending on the date given. For example, DST starts earlier in the USA than in Europe and for a period of two weeks the offset is one hour less. The offset returned may be negative or positive, which implies that a target time zone is either behind or ahead of the source time zone.
• `lookupTimeZone`: Looks up the time zone by name. If not specified, the default time zone is returned, which is geographically dependent.
	`function oic.date.zonedDateTime (source_datetime, source_timezone_id, target_timezone_id) {`
	`  var formattedDateTime = oic.date.zonedDateTime (source_datetime, source_timezone_id, target_timezone_id)`
	`  return formattedDateTime;`
	
	Where:
	• `source_datetime`: A string in ISO format representing date and time (without the time zone information), a JavaScript date object, a numeric value representing the number of milliseconds since the epoch, or an undefined or null value, in which case the current time is used.
	• `source_timezone_id`: The source time zone name. If unspecified, it defaults to the current time zone (geographically dependent).
	• `target_timezone_id`: The target time zone name. If unspecified, it defaults to the current time zone (geographically dependent).
	
	`function oic.date.timeZoneOffset (source_datetime, source_timezone_id, target_timezone_id) {`
	`return CXDate.timeZoneOffset (date, source time zone, target time zone);`
	`}`
	
	Where:
	• `source_datetime`: A string in ISO format representing date and time (without the time zone information), a JavaScript date object, a numeric value representing the number of milliseconds since the epoch, or an undefined or null value, in which case the current time is used.
	• `source_timezone_id`: The source time zone name. If unspecified, it defaults to the current time zone (geographically dependent).
	• `target_timezone_id`: The target time zone name. If unspecified, it defaults to the current time zone (geographically dependent).
	
	`function oic.date.lookupTimeZone (time_zone, option)`
	`{`
	`return CXDate.lookupTimeZone (time zone, option);`
	
	Where:
	• `time_zone`: The time zone name.
	• `option`: How to respond.
	
	`function oic.date.timeZones () {`
	`return CXDate.timeZones ();`
	
	Examples of `oic.date.timeZoneOffset`:

Copyright © 2022, 2026, Oracle and/or its affiliates.

Function	Example

The name of the time zone must be the Internet Assigned Numbers Authority (IANA) time zone identifier (time zone data name).
If the time zone is unknown, this function returns a JavaScript fault.
If an option of `check` is passed, the function returns a null value instead of a JavaScript fault on a nonexisting time zone. This option can be used to validate the time zones prior to calling any other functions that perform time zone calculations.
This function returns the following JSON data about the time zone:
`"name": "Pacific Standard Time", "id": "America/Los_Angeles", "dst": 3600000, "observesDST": true, "rawOffset": -28800000`
• `timeZones`: This function returns a JSON array of available time zone data. The array objects are of the format returned by `lookupTimeZone()`.
	• This input:
	`oic.date.timeZoneOffset("2025-02-26T10:00:00","America/Los_Angeles","Europe/Warsaw")`
	Produces this output:
	`32400000 (9hrs, ahead)`
	• This input:
	`oic.date.timeZoneOffset("2025-02-26T10:00:00","Europe/Warsaw","America/Los_Angeles")`
	Produces this output:
	`-32400000 (-9hrs, behind)`
	• This input:
	`oic.date.timeZoneOffset("2025-02-26T10:00:00","Europe/Warsaw","Europe/Warsaw")`
	Produces this output:
	`0`
	(same time zone)
	• This input:
	`oic.date.timeZoneOffset("2025-07-26T10:00:00")`
	Produces this output:
	`0`
	(same time zone or implied local time zone)
	• This input:
	`oic.date.timeZoneOffset()`
	Produces this output:
	`0`
	(current time, same time zone, or implied local time zone)

Copyright © 2022, 2026, Oracle and/or its affiliates.

Function	Example

	• This input:
	`oic.date.timeZoneOffset(new Date())`
	Produces this output:
	`0`
	(current time as JavaScript date, same time zone, or implied local time zone)
	• This input:
	`oic.date.timeZoneOffset(1758907817048)`
	Produces this output:
	`0`
	(current time as milliseconds since the epoch, same time zone, or implied local time zone)
	Examples of `oic.date.zonedDateTime`:
	• This input:
	`oic.date.zonedDateTime ("2025-04-09T16:32:32","America/Los_Angeles", "Asia/Kolkata")`
	Produces this output:
	`"2025-04-10T05:02:32"`
	• This input:
	`oic.date.zonedDateTime ("2025-02-02T16:32:32","America/Los_Angeles", "Asia/Kolkata")`
	Produces this output:
	`"2025-02-03T06:02:32"`
	• This input:
	`oic.date.zonedDateTime ("2025-02-02T14:59:59","America/Los_Angeles", "Europe/Warsaw")`

Copyright © 2022, 2026, Oracle and/or its affiliates.

Function	Example

	Produces this output:
	
	`"2025-02-02T23:59:59"`
	
	• This input:
	
	`oic.date.zonedDateTime ("2025-03-10T14:59:59","America/Los_Angeles", "Europe/Warsaw")`
	
	Produces this output:
	
	`"2025-03-10T22:59:59"`
	
	• This input:
	
	`oic.date.zonedDateTime ("2025-03-10T14:59:59")`
	
	Produces this output:
	
	`"2025-03-10T14:59:59"`

Create an AI-Powered Library Description

You can use AI to generate a new description or replace an existing description of a library with a single click.

> **Note**
>
> AI features are available in select regions. See AI Feature Availability in *Provisioning and Administering Oracle Integration 3*.

1. Decide where to start:
   - Work in a project.
     a. In the navigation pane, click **Projects**.
     b. Select the project name.
     c. Click **Integration**.
     d. In the **Libraries** section, hover over the library for which to create an AI-generated description.
     e. Click **Actions**, then select **Edit**.
   - Work outside a project.
     a. In the navigation pane, click **Design**, then **Libraries**.
     b. Hover over the library for which to create an AI-generated description.

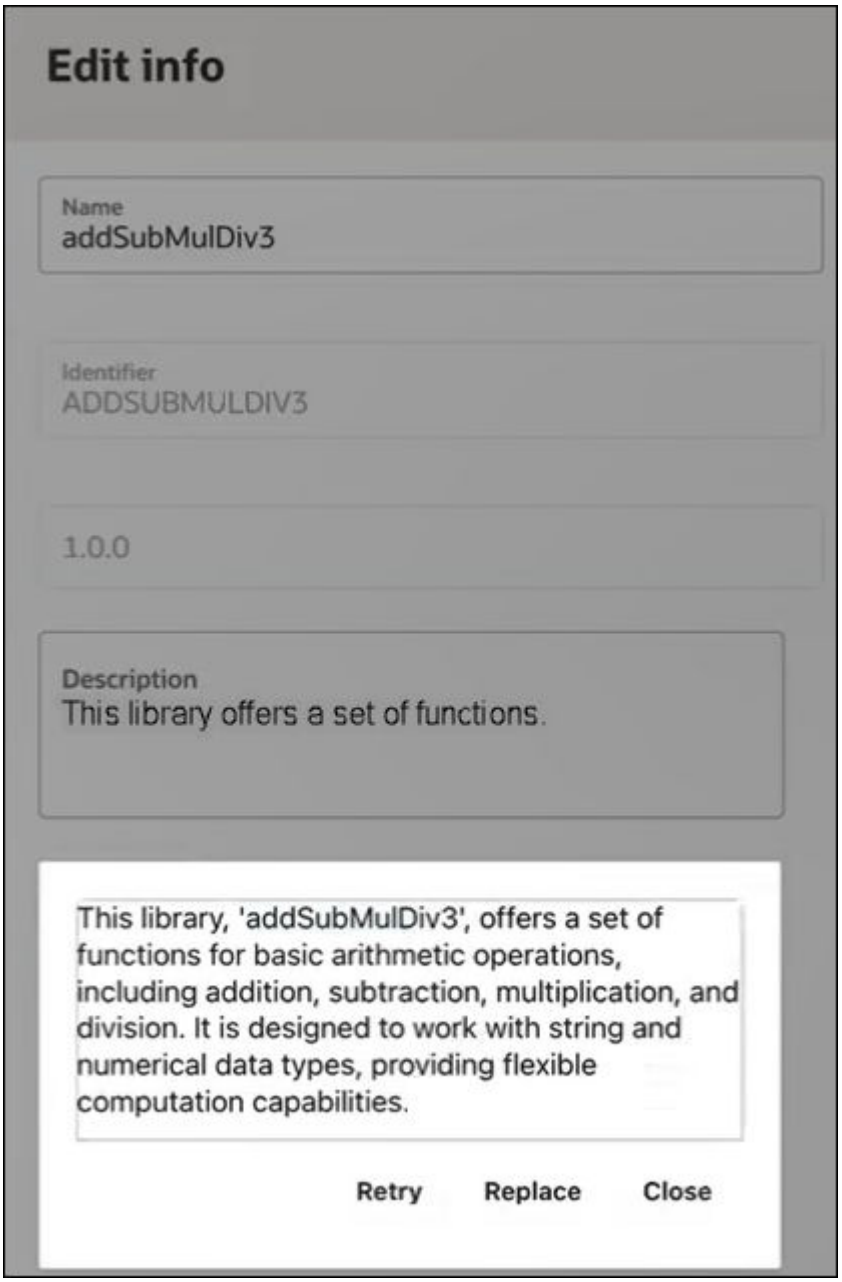
Copyright © 2022, 2026, Oracle and/or its affiliates.

c. Click **Edit** .

2. In the upper right corner, click **Actions** , then select **Edit info**.
3. Click **Generate**.

   The **Description** field can be empty or already include a description.

   - Click **Retry** to regenerate the description. If you retry, the newer description varies slightly in terms of the words used, but the meaning remains the same.
   - Click **Replace** to add the generated library description to the **Description** field. If the **Description** field is empty, the generated library description is added to the field. If the **Description** field already includes a library description, the current description is replaced by the generated description. For this example, a description already existed. **Replace** is only selectable if the library is editable.



Copyright © 2022, 2026, Oracle and/or its affiliates.

4. Click **Save**.

Export a Library File

You can export a library to a source file (`.jar` or `.js`) on your desktop.

- Decide where to start:
  - Work in a project.
    1. In the navigation pane, click **Projects**.
    2. Select the project name.
    3. Click **Integrations**.
    4. In the **Libraries** section, click the library to export.
    5. Click  to download the library.
    6. Click **Keep** when prompted to download the file.
  - Work outside a project.
    1. In the navigation pane, click **Design**, then **Libraries**.
    2. Hover over the library to export.
    3. Click **Actions**, then select **Export**.
    4. Click **Keep** when prompted to download the file.

Update JavaScript Files

You can update a JavaScript file directly in Oracle Integration or in a different editor. Updating the JavaScript file directly in Oracle Integration eliminates the need to export the file, edit it in a different editor, and then re-import it into Oracle Integration. The JavaScript library can consist of a single file or multiple JavaScript files in a JAR file.

- Guidelines
- Update a JavaScript File Directly in Oracle Integration
- Update a JavaScript File in a Different Editor

Guidelines

Note the following guidelines for updating a JavaScript file:

- Functions (within a library) used in integrations must be available in the updated JavaScript file.
- The signatures (parameters and function name) for these functions must remain intact to prevent runtime issues.
- The addition of new functions is allowed.
- Active integrations using new or updated functions must be reactivated.
- You can remove functions used in integrations.

The following changes make an updated JavaScript file invalid:

- Changing the name of a function used in an integration.

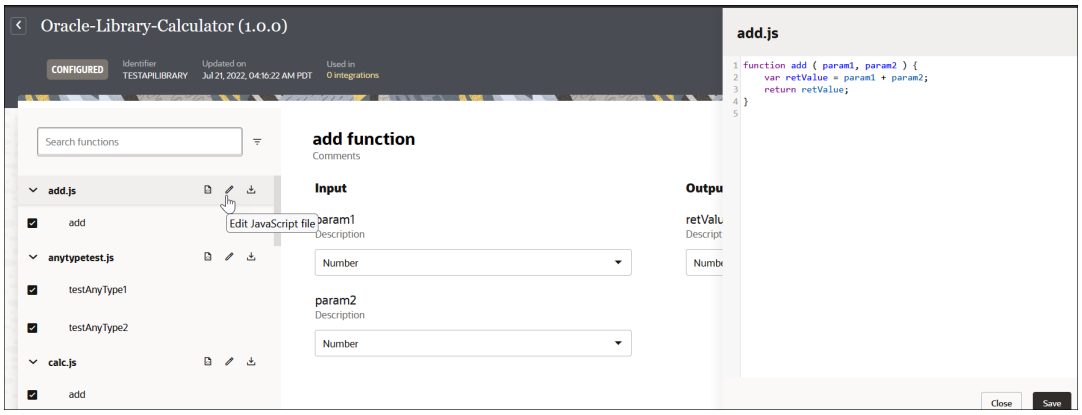
Copyright © 2022, 2026, Oracle and/or its affiliates.

- Changing the parameter definitions of functions used in an integration.
- Structural issues with the JavaScript file.
- The number of functions exceeds 500.

Update a JavaScript File Directly in Oracle Integration

You can also edit the JavaScript file code. Options are also provided for viewing the file contents in read-only mode or downloading the JavaScript file.

1. Decide where to start:
   - Work in a project.
     a. In the navigation pane, click **Projects**.
     b. Select the project name.
     c. Click **Integrations**.
     d. In the **Libraries** section, hover over the library to edit.
   - Work outside a project.
     a. In the navigation pane, click **Design**, then **Libraries**.
     b. Hover over the library to edit.
2. If you want to edit the JavaScript file code, click **Actions** ⋮, then select **Edit**.
   An editor slides opens on the right to display the content of the selected JavaScript file.



3. Add, remove, and update content, when necessary. For example, add a second function (for this example, named `add_xy`).

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

function add_xy ( x, y ) {
    var retValue = x + y;
    return retValue;
}
```

4. Click **Save** when complete. This action adds a second function called `add_xy` in the left pane. The editor performs basic validation, but is not a complete JavaScript editor.

Copyright © 2022, 2026, Oracle and/or its affiliates.

Update a JavaScript File in a Different Editor

You can still update a JavaScript file in a different editor.

1. Decide where to start:
   - Work in a project (see why working with projects is preferred).
     a. In the navigation pane, click **Projects**.
     b. Select the project name.
     c. Click **Integrations**.
     d. In the **Libraries** section, hover over the library to update.
   - Work outside a project.
     a. In the navigation pane, click **Design**, then **Libraries**.
     b. Hover over the library to update.
2. Click **Actions**, then select **Update**.
   The Update library panel appears.

   For example, assume the library to update includes four functions, one of which (`multiply`) is currently used in an integration. The content of the library is as follows:

```
function add ( param1, param2 ) {
     var retValue = param1 + param2;
     return retValue;
}
function subtract ( param1, param2 ) {
     var retValue = param1 - param2;
     return retValue;
}
function multiply ( param1, param2 ) {
     var retValue = param1 * param2;
     return retValue;
}
function divide ( param1, param2 ) {
     var retValue = param1 / param2;
     return retValue;
}
```

3. Click **Browse** to select the JavaScript file that includes the function updates, then click **Update**.
   For this example, the JavaScript file consists of the following updates:
   - Two new functions: `add_three` and `multiply_three`
   - One removed function not used by any integrations: `subtract`
   - A code change to the `multiply` function that is currently used in an integration

```
function add ( param1, param2 ) {
     var retValue = param1 + param2;
     return retValue;
}
function multiply ( param1, param2 ) {
     var retValue = param1 * param2 * 2;
     return retValue;
```


Copyright © 2022, 2026, Oracle and/or its affiliates.

```
}
function divide ( param1, param2 ) {
     var retValue = param1 / param2;
     return retValue;
}
function add_three ( param1, param2, param3 ) {
     var retValue = param1 + param2 + param3;
     return retValue;
}
function multiply_three ( param1, param2, param3 ) {
     var retValue = param1 * param2 * param3;
     return retValue;
}
```

All these updates are valid and the library is successfully updated. The library edit page is displayed with the updates. Any active integrations using the `multiply` function must be reactivated.

An example of an invalid JavaScript file that *cannot* be successfully uploaded is if the `multiply` function included a parameter definition that changed (for example, `param2` to `param3`).

```
function add ( param1, param2 ) {
     var retValue = param1 + param2;
     return retValue;
}
function multiply ( param1, param3 ) {
     var retValue = param1 * param2 * 2;
     return retValue;
}
function divide ( param1, param3 ) {
     var retValue = param1 / param2;
     return retValue;
}
```

Use Library Functions in Integrations

You can add library functions to integrations.

After you create and edit your library functions, they are available from the palette or the canvas inside an integration. See Add JavaScript Functions to an Integration with a JavaScript Action.

Copyright © 2022, 2026, Oracle and/or its affiliates.

12

Manage Integrations

When integration design is complete, you can perform management tasks such as integration activation, integration testing, and other responsibilities.

Topics:

- Activate and Deactivate Integrations
- Test Integrations in Oracle Integration
- Generate an AI-Powered Integration Description
- Edit and Replace Dependent Resources in an Integration
- Invoke a Child Integration from a Parent Integration
- Manage Packages
- Import and Export Integrations
- Clone the Design-Time Metadata of an Entire Service Instance
- Refresh Endpoints for Integrations
- Set the Inactive Session Timeout Limit and Audit Logging Levels
- Modify an Integration
- View the Actions and Connections in an Integration
- Clone an Integration
- Create a Draft Version of an Integration
- Delete an Integration
- Recover Unsaved Integration Changes
- Override Design-Time Properties in an Integration
- Use Metadata in Integrations
- Cancel a Running Integration Instance
- View the Contents of an Incomplete Integration
- Edit an Integration with Incomplete Connections
- Change the Time Zone

Activate and Deactivate Integrations

After integration design is complete, you can activate and set the tracing level for your integration. You can also deactivate integrations.

Topics:

- Activate an Integration
- Reactivate Integrations after a Connection Update

Copyright © 2022, 2026, Oracle and/or its affiliates.

- Deactivate an Integration
- Manage Tracing Levels on Integrations

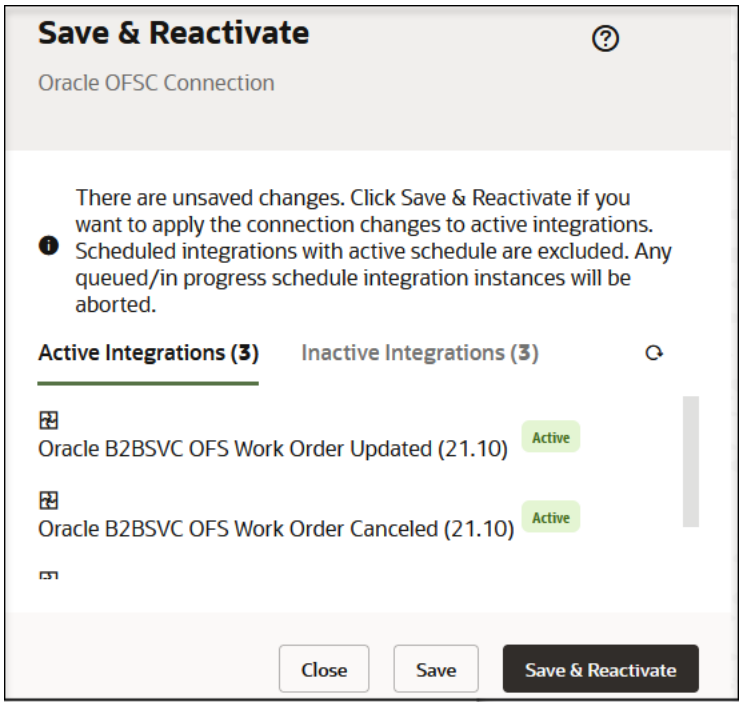
Reactivate Integrations after a Connection Update

You can reactivate all active integrations together that use the same adapter connection after a connection update. This eliminates the need to manually deactivate and reactivate each integration individually to reflect the updated adapter connection. This functionality works for all adapters.

> **Note**
>
> A maximum of 100 integrations can be reactivated after a connection update. Contact Oracle Support if you need to increase this limit.

1. In the navigation pane, click **Design**, then **Connections**.
2. Find and update the necessary connection. The number of integrations that use that connection is displayed.
3. Click **Test**, then click **Save**.

   The list of active and inactive integrations that use the updated connection is displayed on the right.



4. Click **Save & Reactivate** to apply the connection changes to all active integrations. Note the following:
   - You cannot select individual integrations to reactivate; all are reactivated.

Copyright © 2022, 2026, Oracle and/or its affiliates.

- Any active schedule integrations are temporarily paused and enter a failed state. The schedule restarts after reactivation completes. However, schedule integrations that are currently in-flight enter a failed state and are discarded.
- Any integrations using this connection that are currently listed as inactivate remain in that state.
- If you click **Save**, your connection changes are saved, but no integrations are reactivated.
- Deactivation of an integration that subscribes to business events does not unsubscribe the business events.

The page is refreshed to show integration activation progress (for example, the number of integrations that have been reactivated and the number of integrations remaining to be reactivated).

5. Click **Refresh** periodically to view progress.

   When complete, details about the reactivated integrations are displayed.
6. Click **Cancel** to close the page.
7. If any integration reactivations failed, go to the Integrations page to investigate.

Deactivate an Integration

You can deactivate an integration in Oracle Integration to stop it from processing any new messages. You can also individually deactivate an integration in a project. If you want to modify an active integration, you must deactivate it first. If the integration contains a business event subscription, a message appears asking if you want to delete the event subscription.

Deactivation is equivalent to undeployment of a project, which means all existing history, monitoring, and runtime data are lost. Oracle Integration does not support the notion of starting and stopping projects. With asynchronous patterns, the queue for the deactivated project is deleted and all messages associated with this queue are also deleted. Therefore, if there are pending requests unprocessed, they are lost after deactivation. The previous version is deactivated and all existing history, monitoring, and runtime data is lost.

To view only active integrations, select **Active** in the **Status** field of the **Filter** filter. You can also filter by integration name, time last updated, integration creator, status (all, draft, configured, active, activating, and failed), type (all, developed, accelerator type, and recipe), or style (all, application, schedule, and event) to narrow down the list.

1. Decide where to start:
   - Work in a project.
     a. In the navigation pane, click **Projects**.
     b. Select the project name.
     c. Click **Integrations**.
     d. In the **Integrations** section, find the specific integration to deactivate.
     e. Click **Actions**, and select **Deactivate**.
        The Deactivate integration panel opens.
   - Work outside a project.
     a. In the navigation pane, click **Design**, then **Integrations**.
     b. Find the specific integration to deactivate.

Copyright © 2022, 2026, Oracle and/or its affiliates.

c. Hover over the integration to deactivate, then click **Deactivate**.
The Deactivate integration panel opens.

2. Click **Deactivate** in the dialog that appears.

> **Note**
>
> Deactivation of an active, schedule integration is not allowed. You must stop the schedule, then retry.

Integration deactivation can take time to complete after you initiate the request. During this time period, there may still be in-flight or queued requests being processed. These processing instances fail when the corresponding integration artifacts and resources are removed and undeployed.

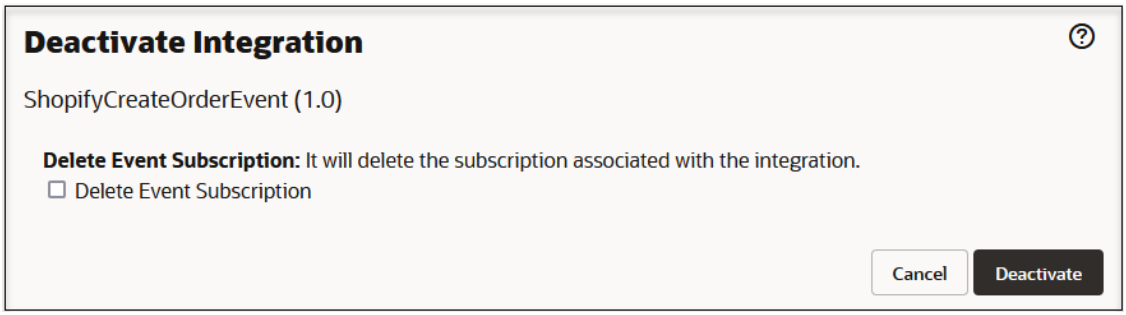
You can observe different processing errors for the failed instances depending on the point of processing during the deactivation. You can also observe the failed instances occurring after the deactivation request.

If deactivation fails and the integration state is displayed as `Deactivation Failed`, retry the deactivation. An integration in this state cannot be edited or activated.

When an integration is successfully deactivated, all its data get undeployed from runtime. This results in nonterminal instances (which are visible on the Instances and Errors pages) getting aborted. Instances in queued, in progress, in error (recoverable type), and other states all move to the aborted state. The existing terminal instances (succeeded, aborted, and in error (nonrecoverable type) remain unaffected.

Deactivate an Integration with Business Events

If the integration to deactivate contains a business event subscription, a message is displayed asking if you want to delete the event subscription. If you select to delete the event subscription, the integration does not receive any events after it is reactivated.



If you do not want to delete the event subscription, the events in this integration are resent if the integration is activated within six hours.

Copyright © 2022, 2026, Oracle and/or its affiliates.

Manage Tracing Levels on Integrations

You can change the tracing level on an active integration without having to deactivate and reactivate it. You can also set the tracing level globally for all integrations.

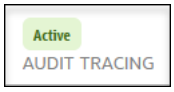
> ⚠ **Caution**
>
> Changing tracing to the audit or debug level impacts performance. Oracle strongly suggests that you don't enable the debug or audit option in production environments. Including the payload in tracing impacts the performance of your system and may violate your company's data retention rules by storing sensitive information in the payload trace. Understand the impact of the various options before making a selection. See Activate an Integration.

- Change the Tracing Level on an Active Integration
- Set the Tracing Level Globally for All Integrations

Change the Tracing Level on an Active Integration

1. In the navigation pane, click **Design**, then **Integrations**.
2. Hover over the active integration on which to change the tracing level.
3. Click **Actions** ⋮, then select **Configure activation**.
4. Configure the tracing level.
5. Click **Save**.
   The tracing level is changed in the active integration.

   The message under **Active** in the **Status** column changes for this integration. For example, if you changed the tracing level from **Production** to **Audit**, the message **Audit Tracing** is displayed after **Active** once integration reactivation completes.



Set the Tracing Level Globally for All Integrations

1. In the navigation pane, click **Settings**, then **Tracing**.
2. Select an option to configure the tracing level, and click **Save**. The tracing level has no impact on billing.

Element	Description
Integration Level	Select to enable or disable tracing at the individual integration level on the Integrations page. This enables you to select **Actions** ⋮ and then **Tracing** for an individual integration on the Integrations page and select to disable tracing or enable tracing and include the payload in the integration. This is the default selection.

Copyright © 2022, 2026, Oracle and/or its affiliates.

Element	Description
Global tracing on at audit level	Select to globally enable tracing at the audit level 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. Two other changes occur: • When you access the Integrations page, a message is displayed indicating that tracing is enabled at the audit level for all active integrations and you cannot enable or disable it at the individual integration level. • If you select **Actions** ⋮, there is no **Tracing** option available for selection. To enable or disable tracing at the individual integration level again, you must select **Integration level**, then **Save**.
Global tracing on at debug level	Select to globally enable tracing at the debug level 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. Two other changes occur: • When you access the Integrations page, a message is displayed indicating that tracing is enabled at the debug level for all active integrations and you cannot enable or disable it at the individual integration level. • If you select **Actions** ⋮, there is no **Tracing** option available for selection. **Note**: Regardless whether debug tracing is set globally or on a single integration, the tracing level is automatically reset to production after 24 hours. To enable or disable tracing at the individual integration level, you must select **Integration level**, then **Save**.
Global tracing off	Select to globally disable tracing for all integrations. After selecting and saving this option, you cannot enable tracing at the individual integration level on the Integrations page. Two other changes occur: • When you access the Integrations page, a message is displayed indicating that tracing is disabled for all active integrations and you cannot enable it at the individual integration level. • If you select **Actions** ⋮, there is no **Tracing** option available for selection.

Copyright © 2022, 2026, Oracle and/or its affiliates.

Activate an Integration

Once you have completed design of your integration and resolved any errors, you can activate it to the runtime environment. You can activate an integration in Oracle Integration to begin processing messages. You can also individually activate an integration in a project.

> **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.

Understand Restrictions About the Number of Active Integrations

The number of active integrations per Oracle Integration instance cannot exceed 800. Active integrations are defined as currently active integrations and integrations whose activations are in progress.

- If you reach 90% of this limit, the following warning is displayed:

  ```
  You've number integrations which are either active or whose activation is
  in progress.
  It is more than 90% of the allowed limit.
  ```

- If you reach the limit of 800, the **Activate** or **Activate & Schedule** button to activate an integration is disabled and the following warning is displayed.

  ```
  You've reached the limit as there are 800 integrations which are either
  active
  or whose activation is in progress. Deactivate or abort the activation of
  an integration
  and try again.
  ```

  If you are nearing your limit, review and delete any older integrations that are no longer required.

Activate an Integration

1. Decide where to start:
   - Work in a project.
     a. In the navigation pane, click **Projects**.
     b. Select the project name.
     c. Click **Integrations** .
     d. In the **Integrations** section, find the specific integration to activate.
     e. Click **Actions** , and select **Activate**.
     The Activate integration panel opens.
   - Work outside a project.
     a. In the navigation pane, click **Design**, then **Integrations**.
     b. Find the specific integration to activate.

Copyright © 2022, 2026, Oracle and/or its affiliates.

c. Hover over the integration to activate, then click **Activate** .
The Activate integration panel is displayed.

2. Select the level of tracing appropriate to your integration. This level describes the amount of data to capture in the activity stream. The tracing level has no impact on billing.

> **Warning**
>
> The **Audit** and **Debug (Not recommended)** tracing levels are not recommended for a production environment. These options may slow down your system and also present a security risk because sensitive information from the payload is logged.

Element	Description
Tracing	Select the level of tracing information to log to the activity stream. • **Production**: Logs all actions with the exception of loops and invoke/logger actions inside of loops (up to 1000 iterations) to the activity stream. Data is retained in the activity stream for 32 days. **Note** Oracle Integration for Healthcare data is retained for 184 days. • **Audit**: Logs the same data to the activity stream as the **Production** option, along with wire payloads of triggers and invokes. Data is retained in the activity stream for eight days. Wire payloads are the payloads sent and received over the wire by Oracle Integration while communicating with third-party applications. These communications either happen through triggers (where Oracle Integration receives the initial triggering message) or invokes (where Oracle Integration sends and receives data). The data in these payloads is untranslated and in the same format understood by the external applications/services that send and receive these payloads. • **Debug (Not recommended)**: Logs the same data to the activity stream as the **Production** and **Audit** options along with all payloads and all actions inside loops (up to a 1000 iterations). Data is retained in the activity stream for 24 hours. After that time period elapses, all payload, activity stream, and business identifier data is purged and a minimal amount of instance information is retained for 32 days. Note the following: – After 24 hours, **Debug (Not recommended)** is automatically reset to **Production**. This reset cannot be bypassed. – Explicitly reset the tracing level to **Production** or **Audit** when debugging is complete. Do not rely on the 24-hour automatic reset. The actual reset may take longer than 24 hours if no messages are processed by a given integration. See View Minimal Details About Debug Tracing Level Instances for 32 Days. See Activity Stream Details.
Allow to run again	This check box only appears for REST Adapter-triggered and AS2 Adapter-triggered integrations. Select this check box if you want to replay an instance of this integration during runtime. See Replay Integration Instances.

Copyright © 2022, 2026, Oracle and/or its affiliates.

Element	Description
Enable payload validation	Select to validate inbound JSON payload syntax during activation of REST Adapter trigger connection-based integrations. Validation consists of checking for duplicate keys in the payload. If you select this option, it remains selected by default. For example, if you later deactivate and reactivate,the integration, the check box remains selected. You must explicitly deselect this check box if you no longer require its use.
Edit/Add Schedule	Select to add or edit a schedule for this integration now. This invokes the page for adding or editing an integration schedule.

3. If integration activation is unsuccessful, the status of the integration changes to **Failed** in the **Status** column. Click the **Open Details** icon to see the error.

   If your integration includes a function that is not completely configured, an error message is displayed in the banner. You must complete configuration of this function before you can activate the integration. Click inside the integration and note the following errors/warnings:

   - An error icon is displayed on the function call action that uses the incomplete function. The **Error** panel on the right side of the integration canvas provides specific details about the incomplete function.
   - A warning icon is displayed on the mapper that uses the inputs and outputs of this function. After completing function configuration, you must verify the input and output mappings before activating the integration.

     If activation is successful, the status of the integration changes to **Active** in the row.

4. If integration activation is successful, an **Active** message is displayed in the **Status** column.

   Additional messages can also be displayed:

   - If you selected **Audit** during activation, an **Audit Tracing** message is displayed after **Active**.
   - If you selected **Debug (Not recommended)** during activation, a **Debug Tracing** message is displayed after **Active**.

5. For REST Adapter trigger-based integrations, click **Actions** ⋮, then **Run** to display the Configure and run page to run, test, and track instances for this integration. See Test Integrations from Outside the Integration Canvas.

Behavior of Running Instances When a Minor Integration Version is Activated

When activating the minor version of an application or schedule integration (for example, 1.0.1) while the major integration version (1.0.0) is still active, all running and failed instances for major integration version 1.0.0 complete without being terminated. All new requests are then handled by integration version 1.0.1.

The following behavior occurs to ensure zero downtime:

- Integration version 1.0.0 is activated with running, in-progress instances.
- Integration version 1.0.1 is then activated.
- All new instance requests are routed to integration version 1.0.1.
- Integration version 1.0.0 remains activated, but does not receive any new requests.
- All running instances of integration version 1.0.0 complete running (no instances are terminated).

Copyright © 2022, 2026, Oracle and/or its affiliates.

> **Note**
>
> It is recommended that you not import, overwrite, or delete an integration until all running instances complete. However, if you need to explicitly perform one of these actions, you can select **Hard Deactivation** from the **Actions** ⋮ menu.

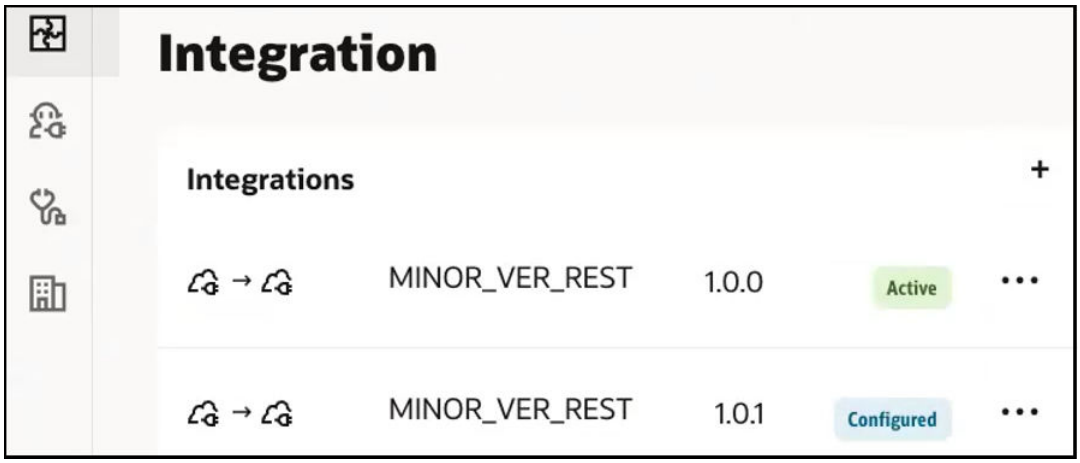
- Integration version 1.0.0 is then completely deactivated.

> **Note**
>
> For schedule integrations, all queued instances are paused and all in-progress instances are allowed to complete. No operations other than stopping or deleting a schedule are permitted. When all instances complete, the schedule integration is deactivated.

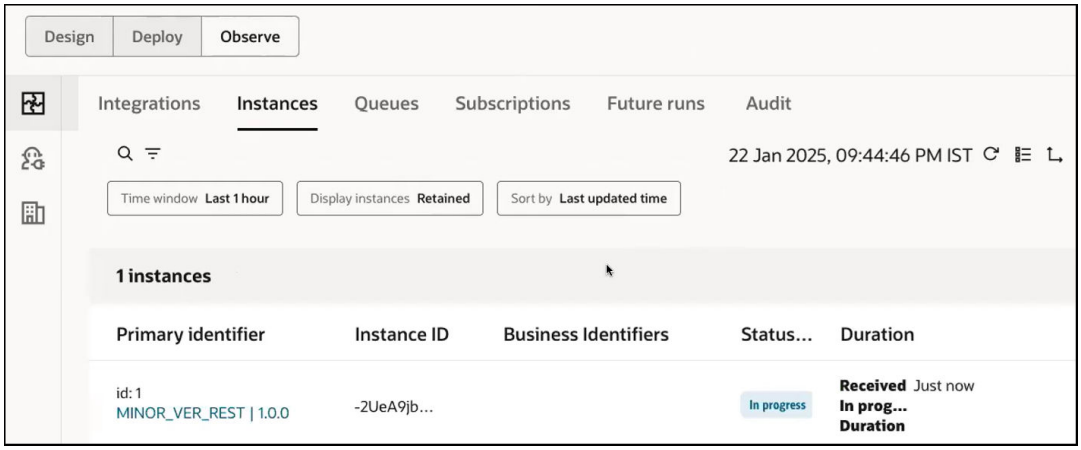
- If any instances of integration version 1.0.0 fail, integration version 1.0.1 is used for resubmitting.

Let's view a simple use case:

1. Activate an integration (for this example, **MINOR_VER_REST** version 1.0.0).

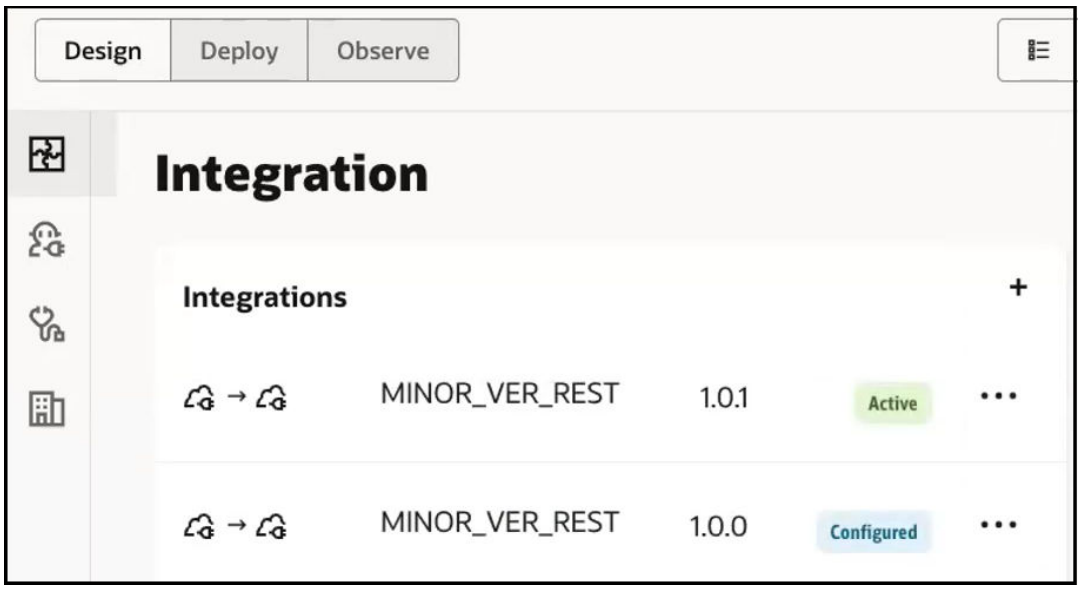


2. Run **MINOR_VER_REST** version 1.0.0, which creates an in-progress instance that's visible on the **Observe** tab.

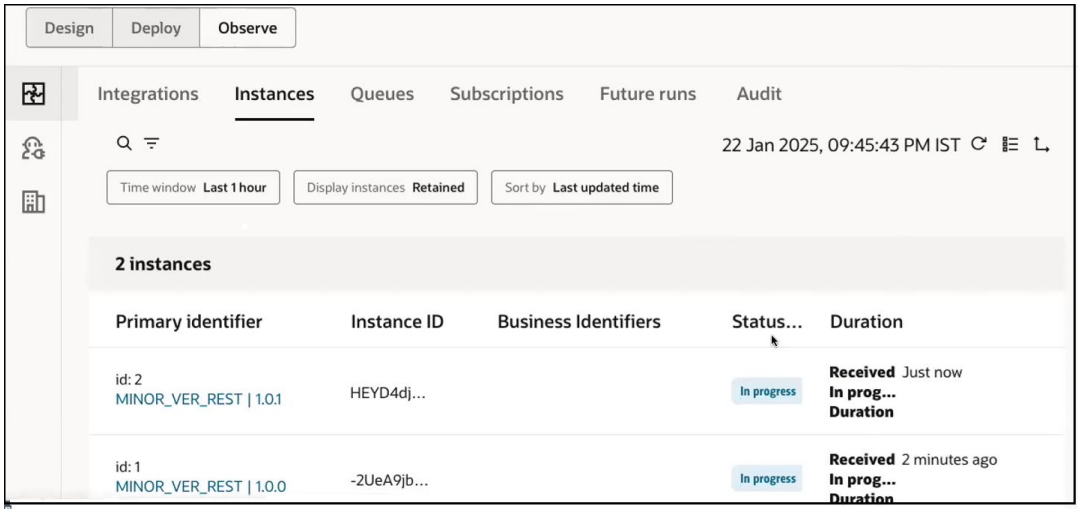


Copyright © 2022, 2026, Oracle and/or its affiliates.

3. Activate **MINOR_VER_REST** 1.0.1, which automatically deactivates integration version 1.0.0.

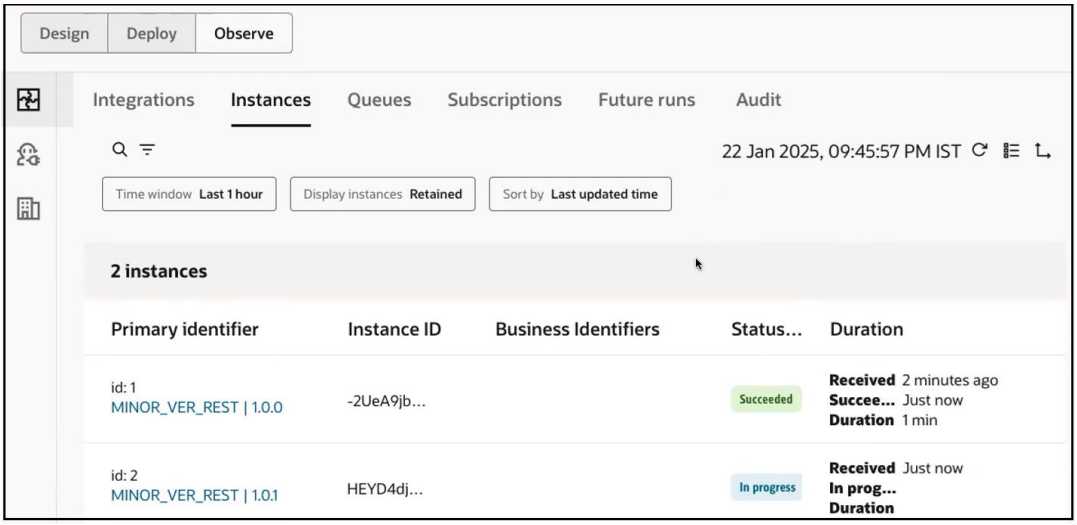


4. Run **MINOR_VER_REST** 1.0.1, which creates an in-progress instance that's visible on the **Observe** tab. The version 1.0.0 instance is not terminated. Both instances are shown as in-progress.

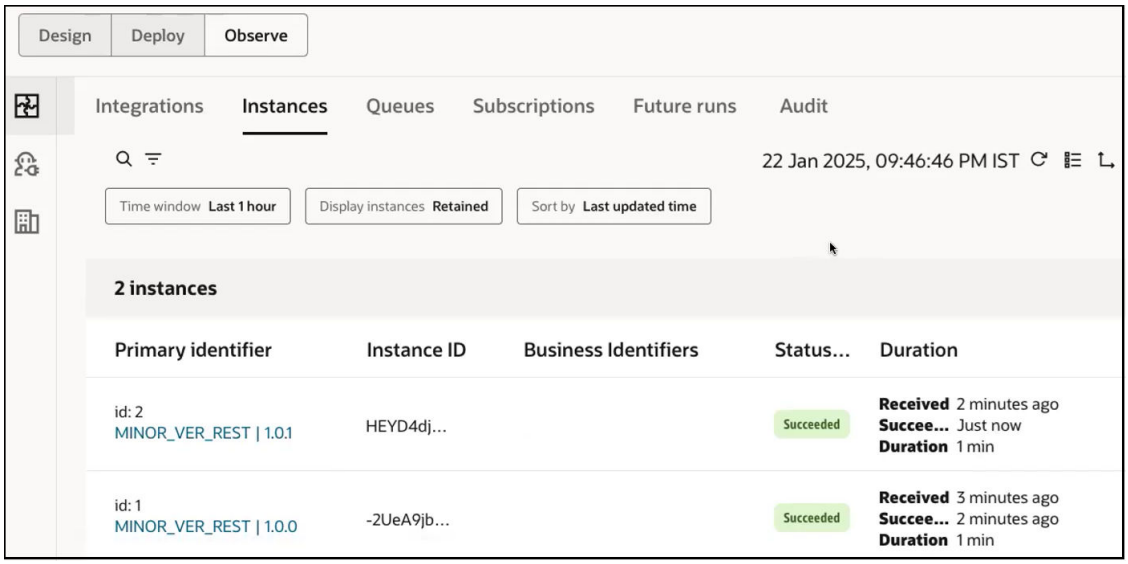


5. Refresh the **Observe** tab and note that the version 1.0.0 instance completed successfully.

Copyright © 2022, 2026, Oracle and/or its affiliates.



6. Refresh again and note that the version 1.0.1 instance also completed successfully.



Any new instance requests are handled by integration version 1.0.1.

Test Integrations in Oracle Integration

You can test REST Adapter trigger connection-based integrations and scheduled integrations in Oracle Integration. You can test the integration either from within the integration canvas or from the **Actions** menu of the integration outside of the integration canvas. This eliminates the need to use third-party software to run this type of integration.

Topics:

- Test Integrations from Inside the Integration Canvas
- Test Integrations from Outside the Integration Canvas

Copyright © 2022, 2026, Oracle and/or its affiliates.

Test Integrations from Inside the Integration Canvas

You can test REST Adapter trigger connection-based integrations and scheduled integrations from inside the integration canvas in Oracle Integration.

This capability lets you automatically activate your integration with the tracing level set to debug and specify request details from the Run integration panel inside the integration canvas.

This capability simplifies integration testing and eliminates the need to exit the canvas, manually activate the integration, set the tracing level, select **Run** from the **Actions** ⋮ menu, and specify the request details as separate steps.

1. Design your integration.

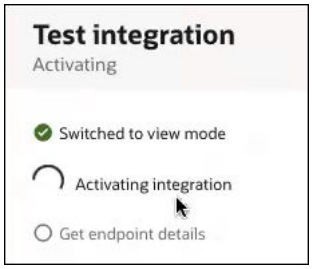
   When complete, perform the following steps to activate and run your integration.

2. Click **Run test** ▷ in the integration canvas bar.



The Test integration panel opens and the following actions are automatically performed:

- The integration switches to view mode.
- The integration is activated with the tracing level set to debug.
- The endpoint details are obtained.



While activation tasks are in progress, you can perform other actions:

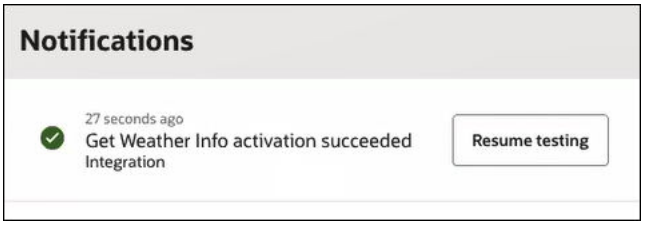
- Perform actions inside the canvas. For example, you can click **Properties** to view integration properties. When you perform an action inside the canvas, the Test integration panel is closed. To bring back the Test integration panel and continue testing, click in the toolbar.
- Exit the canvas to perform actions. Once activation completes, a message appears in the upper right part of the page.

  ```
  Integration_name activation succeeded
  Resume testing
  ```

  You can click **Resume testing** to return to the canvas where you are prompted to specify request values to run the activated integration.

Copyright © 2022, 2026, Oracle and/or its affiliates.

You can also click **Notifications** in the Oracle Integration banner, and then click **Resume testing** to return to the canvas where you are also prompted to specify request values to run the activated integration.



If activation fails, the **Try again** option is enabled in the Test integration panel.

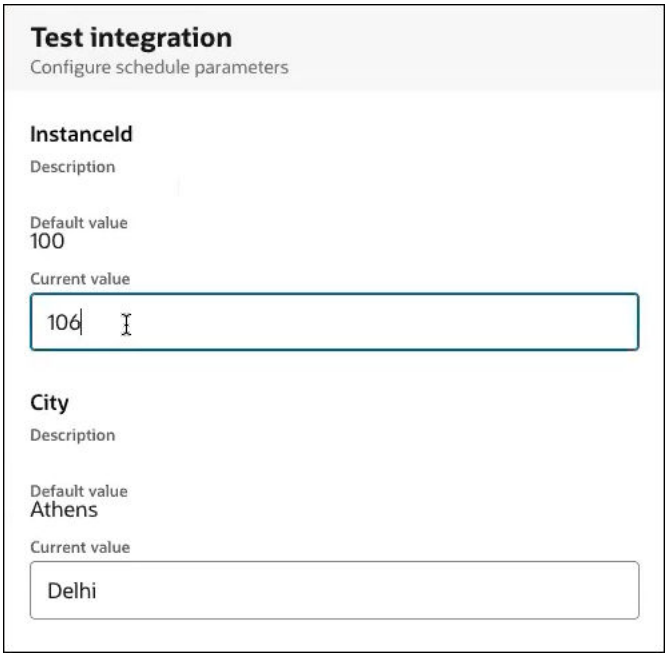
When these tasks complete, the panel is refreshed to show the following:

a. For a REST Adapter trigger connection-based integration, you are prompted to specify request parameter values:



b. For scheduled integrations with defined schedule parameters, you are prompted to specify values:

Copyright © 2022, 2026, Oracle and/or its affiliates.

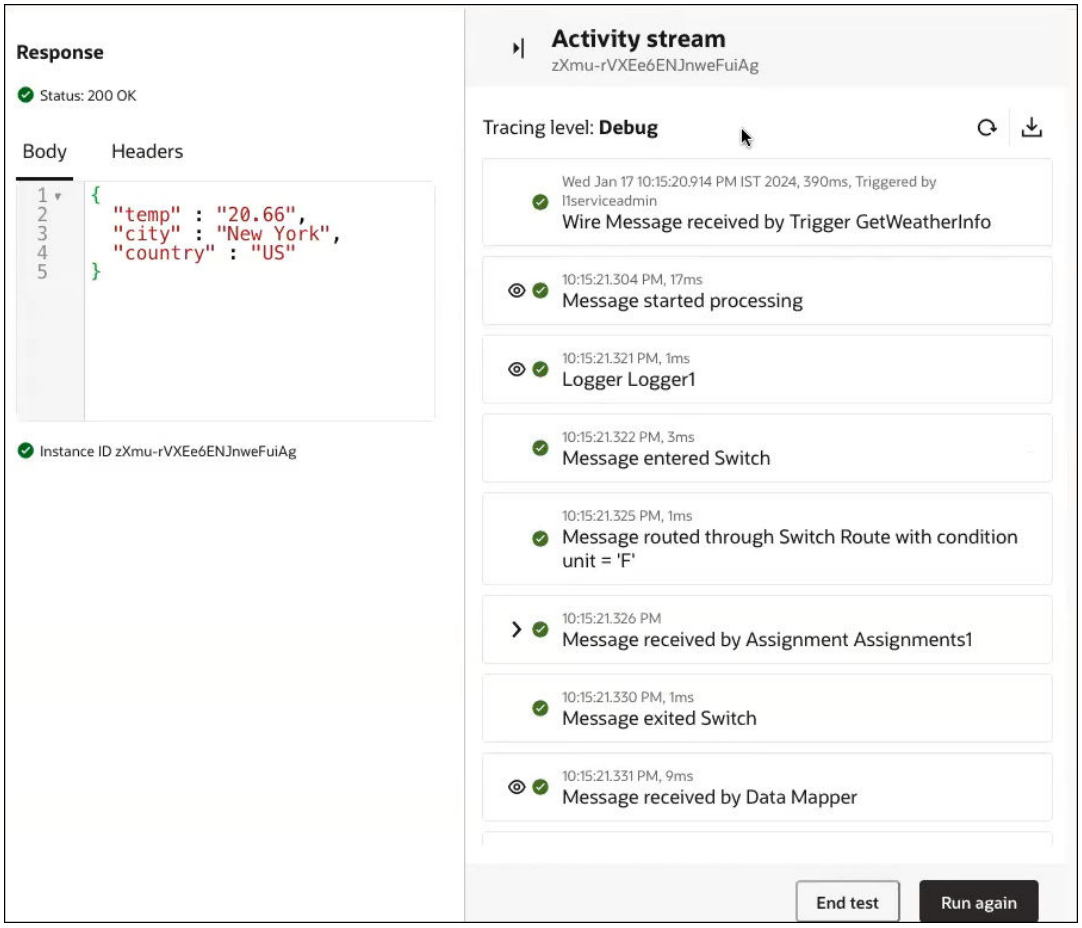


c. For scheduled integrations without defined schedule parameters, no additional input is required. The integration runs automatically after activation and the results/activity stream are displayed.

3. Specify the request or scheduled parameter details, and click **Run**.

   The response details (for REST Adapter trigger connection-based integrations) and instance ID are first displayed, followed by the activity stream. The integration canvas switches from edit mode to runtime tracking mode.

Copyright © 2022, 2026, Oracle and/or its affiliates.



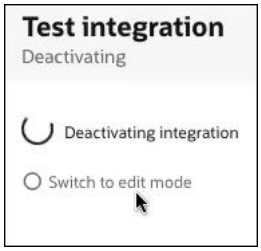
4. Perform runtime monitoring details, such as expanding sections of the activity stream, viewing the business identifiers that were passed, tracing the message delivery through the integration, and so on.
5. If you want to run the integration again, click **Run again**.
6. Specify the request or scheduled parameter details again, and click **Run**.
7. When testing is complete, click **End test** at the bottom of the activity stream or **End test** ☐ in the integration canvas bar.



The Test integration panel opens and the following actions are automatically performed:

- The integration is deactivated.
- The integration switches back to edit mode.

Copyright © 2022, 2026, Oracle and/or its affiliates.



Test Integrations from Outside the Integration Canvas

You can test application integrations designed with a REST Adapter trigger connection and scheduled integrations from the **Actions** menu of the integration in Oracle Integration.

See Run a Schedule Integration Immediately.

1. In the navigation pane, click **Design**, then **Integrations**.
2. Hover over the REST Adapter trigger connection-based integration to run, then click ⋮ and select **Run**.

   The Configure and run page is displayed with the following sections:

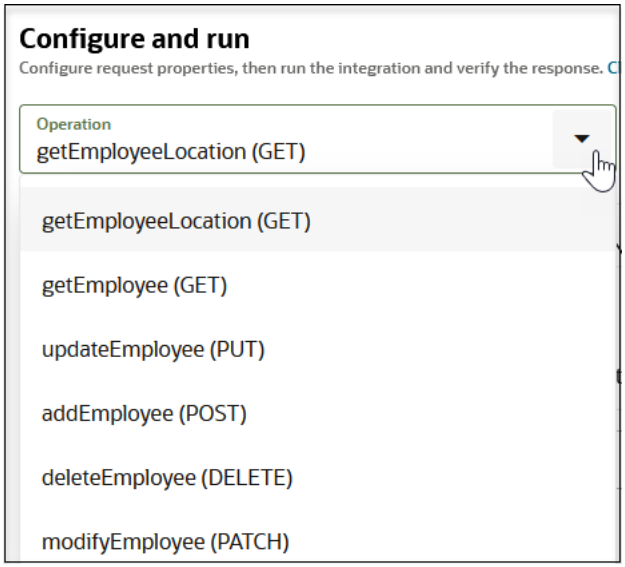
   - Operation Section
   - Request and Response Sections

   The **Operation** and **Request** sections include the endpoint's metadata.

Operation Section

If the REST Adapter trigger connection-based integration is configured with multiple operations, the **Operation** section is displayed. This section contains the **Operation** list for selecting the appropriate operation, along with the HTTP method and relative URI (for the selected operation in case of multiple operations). If your integration contains only a single operation, this list is not displayed.

1. Select the available operation to test.

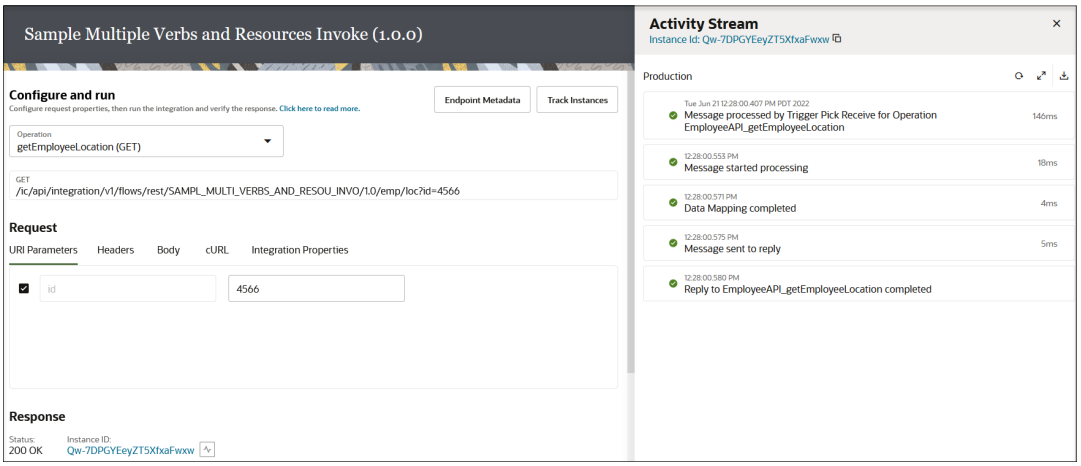


Copyright © 2022, 2026, Oracle and/or its affiliates.

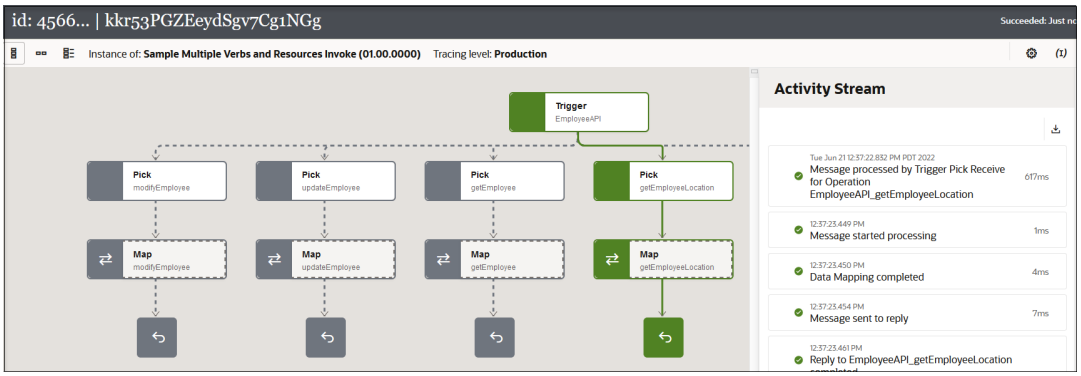
2. View the following links.
   - **Endpoint Metadata**: Displays endpoint description details, such as the Swagger path, the OpenAPI path, how to run the integration, the operation name, and the endpoint URL.
   - **Track Instances**: Displays the Instances page, which shows the details of runs of this instance.
3. Enter a value in the **Request** section and click **Run** to invoke the integration (for this example, a value is specified for **id**).

   `Integration has been invoked with instance id id_number. Please check the Response section for more details.`

   The Activity Stream panel opens and the **Response** section shows a status value of `200 OK`.



4. Click the instance ID in the **Response** section to access the Instance Details page for this instance.



Copyright © 2022, 2026, Oracle and/or its affiliates.

Request and Response Sections

The **Request** and **Response** sections enable you to trigger and view response output for your REST Adapter trigger connection-based integrations. Links are provided at the top to access the Instances page and the endpoint metadata.

The **Request** section consists of the following fields:

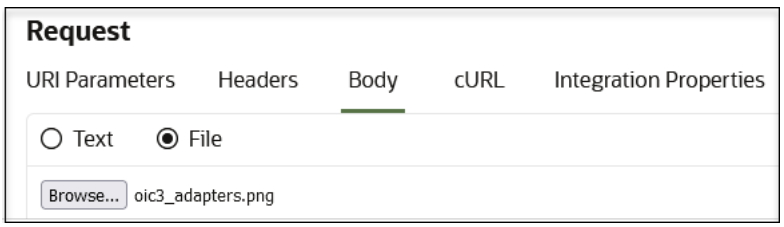
- **URI Parameters**: Shows the list of expected path (or template) and query parameters. You can enter data in this field.
- **Headers**: Shows the custom headers, including **Accept** based on the integration configuration.
- **Body**: Enables you to view the message body contents or upload a file. Support is provided for uploading multipart payloads or for uploading nontextual payloads such as PDF or image files.
  - If the request operation does not have a body, the following message is displayed.

    ```
    This request does not have a body
    ```

  - If the request operation includes a body, its contents are automatically displayed. For example:

  - If you want to browse for and upload a binary or nontextual payload (such as a PDF or image file), click **File**. For this example, an image file is uploaded.

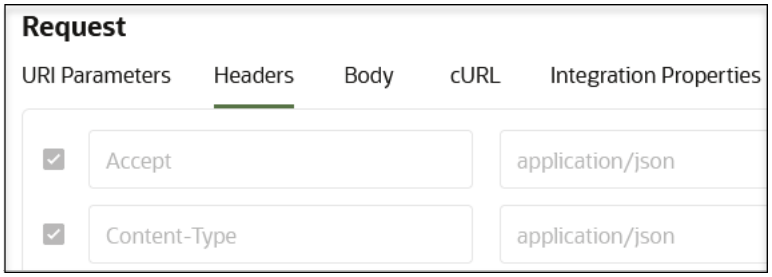
    

  - For endpoints with multipart payloads, a form is displayed under the **Body** field that can be used to provide a value for each part in the multipart body. Two multipart attachments types are supported in the request payload:
    * multipart/form-data

Copyright © 2022, 2026, Oracle and/or its affiliates.

* multipart/payload with structured data such as JSON and XML:
  * multipart/form-data
  * multipart/mixed

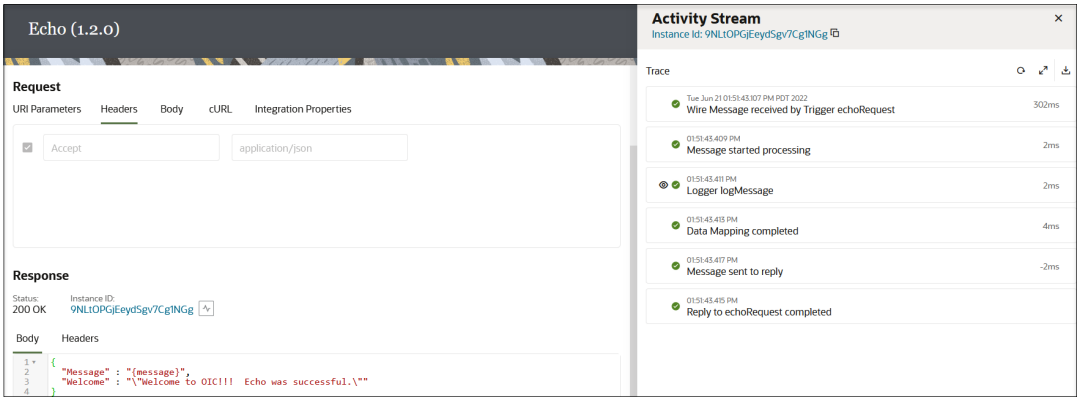
For the following example, two headers for multipart/form-data are visible under the **Headers** tab.



Use cases for both multipart attachment types are provided. See Test REST Adapter Trigger-Based Integrations with Multipart Attachments on the Configure and run page.

- **cURL**: Enables you to copy the equivalent `curl` command. The `curl` command is generated based on the endpoint's metadata and your provided input.
- **Integration Properties**: Any defined integration properties are visible. You can configure integration properties outside of the regular integration design. At runtime, these integration properties take effect and override the values you configured at design time. See Override Design-Time Properties in an Integration.

1. Click **Run** to invoke the integration.
   The Activity Stream panel for this instance is displayed on the right side of the page. You can also access the instance in the Instances page by clicking **Track Instances** above the operation.



2. Check the **Response** section for response details.
   The **Response** section consists of the HTTP status of the headers (`200 OK` for this example) and instance ID of the triggered integration that links to the Instance Details page. After this are the response body and headers included in the body.
3. Scroll to the right of **Status** and note the instance ID link. This link takes you to the Instance Details page for this instance. See Track Integration Instances.

Copyright © 2022, 2026, Oracle and/or its affiliates.

Generate an AI-Powered Integration Description

You can generate artificial intelligence (AI)-powered descriptions of integrations in Oracle Integration. This capability eliminates the need to manually enter a description, improves productivity, and provides valuable insights to other users.

An integration design can become large and complicated, making it difficult to view the integration and quickly understand the overall meaning. This feature enables you to quickly generate an overall description of the integration with a single click.

For information about Oracle Integration's embedded AI capabilities, see AI Innovation and Oracle Integration.

- Understand the Guidelines
- Generate the Integration Summary

Understand the Guidelines

Note the following guidelines:

- The integration description can be generated and replaced only if the integration is editable. For example, you cannot replace an integration description in an active integration.
- You can regenerate and replace the integration description as often as you want.
- You cannot generate and replace a description for an integration that is part of an accelerator obtained from the Integration Store.
- Any user with the ServiceAdministrator role or ServiceDeveloper role can generate and replace a description within the integration canvas.

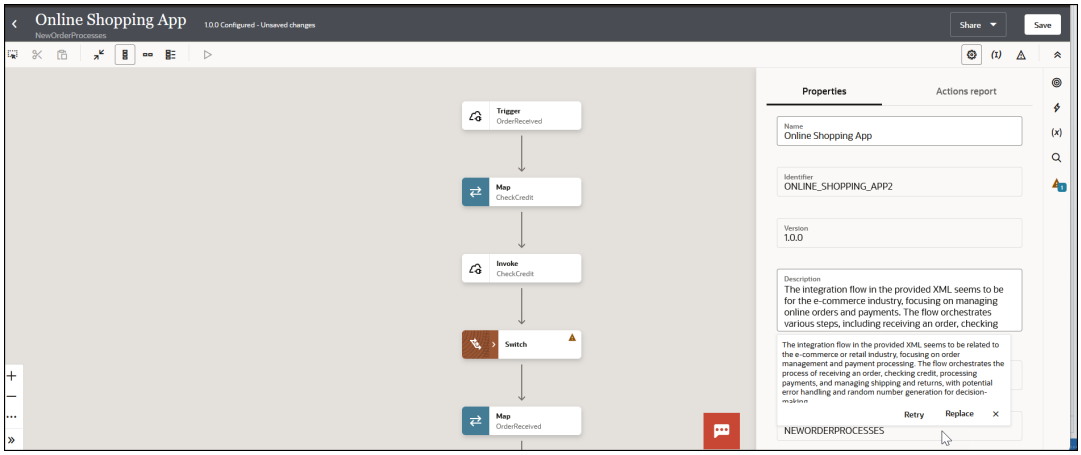
Generate the Integration Summary

1. Decide where to start:
   - Work in a project.
     a. In the navigation pane, click **Projects**.
     b. Select the project name.
     c. Click **Integrations**.
     d. In the **Integration** section, click the integration for which to generate the description.
   - Work outside a project.
     a. In the navigation pane, click **Design**, then **Integrations**.
     b. Click the integration for which to generate the description.
2. Above the integration canvas, click **Primary info**.
3. Click **Generate**.

   The **Description** field can be empty or already include a description.
   - If the **Description** field is empty, the generated integration description is added to the field.

Copyright © 2022, 2026, Oracle and/or its affiliates.

- If the **Description** field already includes a description, the updated description is displayed as a popup. For this example, a description already existed.



4. If the **Description** field already includes a description, perform one of the following steps:
   a. Click **Retry** to regenerate the description. If you retry, the newer description varies slightly in terms of the words used, but the meaning remains the same.
   b. Click **Replace** to replace the current description in the **Description** section. **Replace** is only selectable if the integration is editable.
5. Click **Save**.

Edit and Replace Dependent Resources

You can edit dependent resources (connections, libraries, lookups, and PGP keys) in a single integration or all integrations in a package with a wizard that's accessible from the Configuration Editor page. You can also replace connections and PGP keys unless they are included in business and technical accelerators or the integration is active, locked, in process of activation, or in process of deactivation.

Only the administrator user can edit PGP keys. You can also update integration property values for integrations in which properties have been defined, edit schedules in existing schedule integrations, or add new schedules to schedule integrations.

The Configuration Editor Wizard validates your selections as follows:

- When you move to the next step in the wizard (for example, from connections to certificates), validation is performed on the previous step (connections).
  If validation succeeds, a green icon with a check mark is displayed on the previous step. If validation fails, a warning icon is displayed on the previous step.
- When you access the Configuration Editor page, the connections step in the wizard is shown by default. If you jump directly to the activation step from the connections step, validation runs on all the steps.
- For a simple integration without any dependencies, there is only a single activation step in which validation runs and provides the correct status for the step.
- Validation runs to check whether *all* resources under a step are in the necessary state and not just a single resource. For example, if all the connections under the connections step are in a configured state, the step status evaluates to a green icon with a check mark. Otherwise, a warning icon is displayed.

Copyright © 2022, 2026, Oracle and/or its affiliates.

Topics:

- Edit and Replace Dependent Resources in a Package
- Edit and Replace Dependent Resources in an Integration
- Unlock Dependent Resources

Edit and Replace Dependent Resources in a Package

You can configure the dependent resources used by all integrations in a specific package.

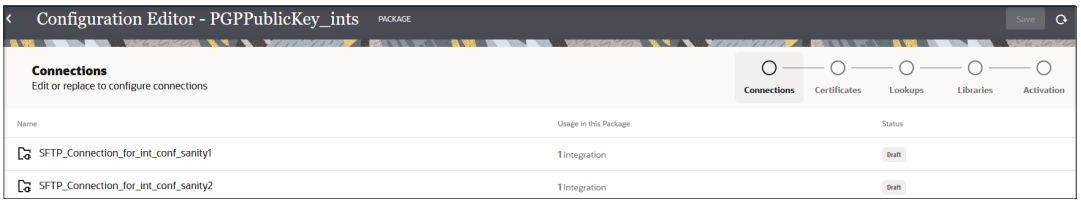
Editing a dependent resource impacts all integrations that use that resource. Replacing a dependent resource only impacts the integrations in the specific package.

1. In the navigation pane, click **Design**, then **Packages**.
2. Select the package in which to edit or replace resources in either of two ways:
   a. In the banner, click **Import**.
   b. Browse for and select the PAR file when prompted.
   c. Click **Import and Configure**.

   Or:

   a. Hover over the package.
   b. Click **Configure resources** .

   The Configuration Editor page shows the package name in the banner and a wizard that lists the steps for each resource type included in the package (connections, certificates (PGP keys), lookups, or libraries).



3. For connections, click **Edit** to edit the connection configuration and security properties or **Replace**

   to replace the connection with a different one.
   Note the following connection replacement guidelines:

   - Only connections of the same role type (trigger, invoke, or trigger and invoke) are displayed for selection. You can only replace a connection resource with another resource of the same role.
   - The connection status must be displayed as **Configured**. You cannot replace a connection with a connection whose status is **Draft**. If a compatible connection resource does not exist, a message is displayed.
   - You cannot replace connections used in integrations with the following state:

Copyright © 2022, 2026, Oracle and/or its affiliates.

  - Active
  - Activation in progress
  - Deactivation in progress
  - Locked
- If you replace a connection, a **Revert** icon is displayed in the row that enables you to change back to the previous connection.

4. Click the next resource step in the wizard (for this example, **Certificates**).
   If the connection is not in a configured state, an error icon is displayed. Hover over the icon to show information about the error.



   You must successfully configure the connection to activate the integration at the end of the wizard. If the connection is in a configured state, a green icon is displayed.

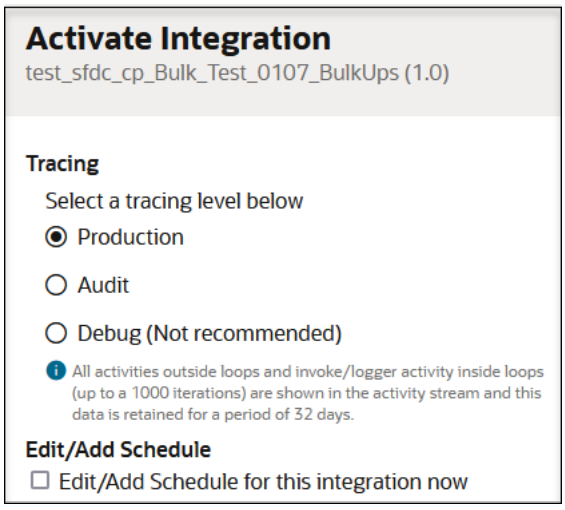


   A PGP key must be the same type of key. For example, a public PGP key can only be replaced with another public PGP key, and not a private PGP key.

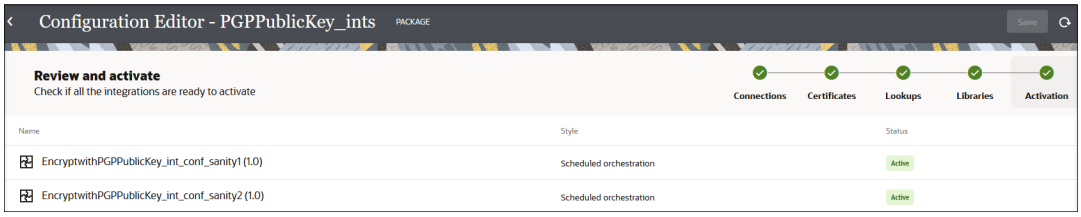
5. Click **Edit** if you want to edit the certificate or **Replace** to replace the certificate with a different one. If you replace a certificate, a **Revert** icon is displayed in the row that enables you to change back to the previous certificate.
6. Click the **Lookups** icon.
7. Click **Edit** if you want to edit the lookup.
8. Click the **Libraries** icon.
9. Click **Edit** if you want to edit the library.
   If all resources are displayed as configured with a green icon and a check mark, the integration is ready for activation.
10. Click **Save**. You can also save after you update or replace a resource in each step of the wizard.
11. Hover over the row of an integration and click **Activate**. The number of active integrations per Oracle Integration instance cannot exceed 800. Active integrations are defined as currently active integrations and integrations whose activations are in progress. If you reach this limit, the **Activate** button is disabled. See Understand Restrictions About the Number of Active Integrations.
    You can activate business accelerators, recipes, and technical accelerators in bulk. For packages that you created, each integration must be activated individually.

    The Activate Integration panel is displayed.

Copyright © 2022, 2026, Oracle and/or its affiliates.



12. Specify the following options.
    - Select a tracing level. See Activate an Integration.
    - If this is a schedule integration, you can select to edit an existing schedule or add a new schedule. See Edit an Integration Schedule or Define the Integration Schedule.
    - If there are property values defined for this integration, you can select to edit them. You can configure integration properties outside of the regular integration design. At runtime, these integration properties take effect and override the values you configured at design time. See Override Design-Time Properties in an Integration.
13. Click **Activate**.
    The package appears as follows:



Edit and Replace Dependent Resources in an Integration

You can configure dependent resources used by standalone integrations that are not part of a package.

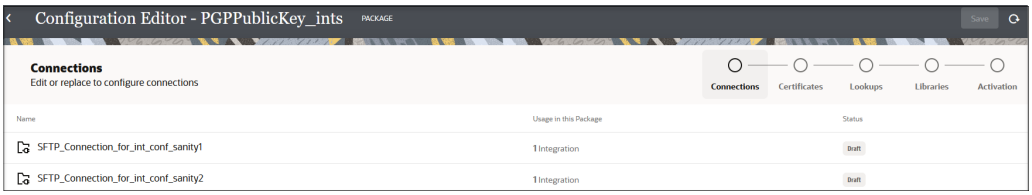
Editing a dependent resource impacts all integrations that use that resource. Replacing a dependent resource only impacts the integrations in the specific package or the standalone integration that is not part of any package.

1. In the navigation pane, click **Design**, then **Integrations**.
2. Select the integration in which to edit or replace resources in either of two ways:
    - Hover over the integration, then click ⋮ and select **Configure**.
    - Select **Import**, then browse for and select the integration to import, then select **Import and Configure**.

Copyright © 2022, 2026, Oracle and/or its affiliates.

The Configuration Editor page shows either of two views:

- If the integration you selected is part of a package, the Configuration Editor page shows the package name in the banner and a wizard that lists the steps for each resource type included in the package (connections, certificates (PGP keys), lookups, or libraries).



- If the integration you selected is not part of a package, the Configuration Editor page shows the integration name in the banner and a wizard that lists the steps for each resource type included in the integration (connections, certificates (PGP keys), lookups, and libraries).



3. For connections, hover over the row and click **Edit** to edit the connection configuration and security properties or

   to replace the connection with a different one.
   Note the following connection replacement guidelines:

   - Only connections of the same role type (trigger, invoke, or trigger and invoke) are displayed for selection. A connection resource can only be replaced by another resource of the same role.
   - The connection status must be displayed as **Configured**. You cannot replace a connection with a connection whose status is **Draft**. If a compatible connection resource does not exist, a message is displayed.
   - You cannot replace connections used in integrations with the following state:
     - Active
     - Activation in progress
     - Deactivation in progress
     - Locked
   - If you replace a connection, a **Revert** icon is displayed when you hover over the row that enables you to change back to the previous connection.

4. Click the next resource step in the wizard (for this example, **Certificates**).
   If the connection selected in the previous step is not in a configured state, an error icon is displayed. Hover over the icon to show information about the error.

Copyright © 2022, 2026, Oracle and/or its affiliates.



You must successfully configure the connection to activate the integration at the end of the wizard. If the connection is in a configured state, a green icon with a check mark is displayed.

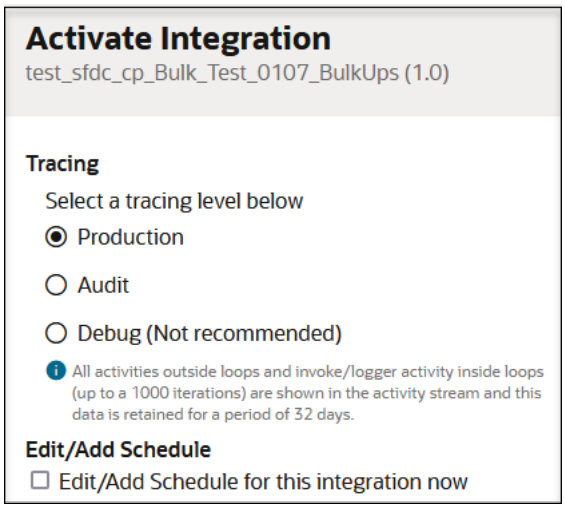


A PGP key must be the same type of key. For example, a public PGP key can only be replaced with another public PGP key, and not a private PGP key.

5. Hover over the row and click **Edit** if you want to edit the certificate or **Replace resource**

   to replace the certificate with a different one. If you replace a certificate, a **Revert** icon is displayed when you hover over the row that enables you to revert to the previous certificate.
6. Click the **Lookups** icon.
7. Hover over the row and click **Edit** if you want to edit the lookup.
8. Click the **Libraries** icon.
9. Hover over the row and click **Edit** if you want to edit the library.
   If all resources are displayed as configured with a green icon and a check mark, the integration is ready for activation.
10. Click **Save**. You can also save after you update or replace a resource in each step of the wizard.
11. Hover over the row and click **Activate**. The number of active integrations per Oracle Integration instance cannot exceed 800. Active integrations are defined as currently active integrations and integrations whose activations are in progress. If you reach this limit, the **Activate** button is disabled. See Understand Restrictions About the Number of Active Integrations.
    The Activate Integration panel is displayed.

Copyright © 2022, 2026, Oracle and/or its affiliates.

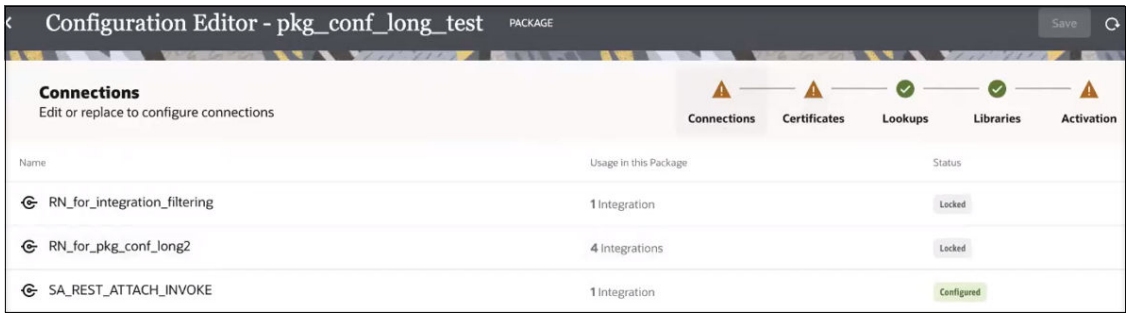


12. Specify the following options.
    - Select a tracing level. See Activate an Integration.
    - If this is a schedule integration, you can select to edit an existing schedule or add a new schedule. See Edit an Integration Schedule or Define the Integration Schedule.
    - If there are property values defined for this integration, you can select to edit them. You can configure integration properties outside of the regular integration design. At runtime, these integration properties take effect and override the values you configured at design time. See Override Design-Time Properties in an Integration.
13. Click **Activate**.

Unlock Dependent Resources

You can directly unlock any locked dependent resources in the Configuration Editor Wizard for editing instead of navigating to the specific resource page (for example, the Connections page or Lookups page). Resources can become locked because of a system or browser crash.

If any resources are listed as locked (for this example, connections are shown), you can unlock and edit them.



1. Hover over the resource.
2. Click **Actions** ..., then select **Unlock** to unlock the resource for editing.
3. Click **Unlock** to confirm when prompted.

Copyright © 2022, 2026, Oracle and/or its affiliates.

Locked integrations with unsaved changes show the words **Unsaved Changes** on the Activation step of the Configuration Editor Wizard. This example shows several integrations in the package with this status.



1. If the locked integration does not include unsaved changes:
   a. Hover over the integration.
   b. Click **Actions** ⋮, then select **Unlock** to unlock an integration with no unsaved changes.
   c. Click **Unlock** to confirm when prompted.
2. If the integration includes unsaved changes, two options are displayed:
   a. Select **Unlock & Discard** to discard all unsaved changes.
   or
   b. Select **Edit & Recover** to recover all unsaved changes.
   The canvas for the integration is displayed.
   c. Select **Save** when prompted. Otherwise, click **Discard**.
   You are returned to the Configuration Editor Wizard.

Invoke a Child Integration from a Parent Integration

You can invoke a co-located (child), active integration from the parent integration that you are designing through use of the local integration adapter. Co-located means the integration is running on the same host instance or in the same domain. Upon activation and invocation of the parent integration, it invokes and consumes the co-located integration.

- Capabilities
- Restrictions
- Best Practices for Designing Local Invoke Calls
- Invoke a Co-located Integration with a Parent Integration

Capabilities

The local integration adapter is a system adapter that is not exposed on the Connections page when you create a connection. Instead, the local integration adapter is invoked by a preseeded connection. It is created when creating an integration, if the local integration connection does not exist. This adapter has no connection properties and no security policies.

The local integration adapter works as follows:

- You select an active integration from runtime (a child) to build a new integration. Note the following guidelines:

Copyright © 2022, 2026, Oracle and/or its affiliates.

  - Active synchronous-synchronous, asynchronous with no callback, and schedule integrations are available for selection.
  - Asynchronous with callback integrations do not appear.
  - De-activated and draft integrations do not appear.
  - Schedule integrations are permitted as an invoke connection and automatically called by the **Submit Now** option.
- You can add multiple child integrations. You can also add an integration containing a child integration (embedding multiple levels).
- You can map return values downstream. This mapping functionality is the same as for any other invoke mappings in the integration.
- You can add header support to the payload of the SOAP Adapter or REST Adapter used in the child integration. See Create a Co-located Integration with Header Support.
- You can create a parent integration using a REST Adapter invoke connection configured with the OpenAPI URL connection type to pass a binary payload to a child integration. Use binary data with payloads that are unstructured and inline (for example, application/octet-stream). The file contents are preserved, but the receiver is required to determine the file type (for example, from the file name extension). The internet media type for an arbitrary byte stream is application/octet-stream.

The local integration adapter and the existing SOAP Adapter and REST Adapter differ as follows:

- You do not need to create multiple connections (SOAP/REST) to invoke the local integration service.
- Runtime communication always uses HTTP (that is, non-SSL).

This adapter is invoked internally in Oracle Integration to perform the local service invocation.

Restrictions

Note the following local integration restrictions:

- You cannot send or receive file references between parent and child integrations, in either direction, whether using local integrations or otherwise. The stage file is scoped in the context of its own flow instance and is not shared between parent and child integrations. This is by design. Instead, you can send or receive the stage file as an attachment instead of a direct reference.
- You cannot create a child integration with a WSDL having a nested anonymous schema. As a workaround, manually edit the WSDL and make the schema non-anonymous.
- SOAP-based child integrations:
  - A parent integration cannot send an attachment to a configured child integration.
- REST-based child integrations:
  - A parent integration cannot send an attachment to a configured child integration.
  - A parent integration cannot process an XML payload that is expected by the child integration. The child flow must be configured to receive a JSON payload only, and cannot be invoked locally using an XML payload.

Best Practices for Designing Local Invoke Calls

- Calls to schedule tasks or asynchronous integrations are always fine with a local invoke because they are not processed in the context of the current node.

Copyright © 2022, 2026, Oracle and/or its affiliates.

- Calls from nonparallel code sections are also always fine because the calling integration block is waiting for a response. There is currently no parallel capability in synchronous integrations. Therefore, a local invoke is always fine in a synchronous flow.
- Calls from parallel code sections (parallel for-each and parallel file processing) are problematic when using a local invoke because they may use the same node for processing. In those cases, calls do not benefit from parallelism across nodes and instead are competing for resources on a single node.

Invoke a Child Integration with a Parent Integration

The following steps provide an overview of creating an application integration in which a parent integration invokes a child integration.

> **Note**
>
> - If you need to move a parent integration to another instance, it is best to include that integration and all its child integrations in a package.
> - If the following integrations are imported from one environment to another (having different host names), then editing the local (child) integration or editing the REST Adapter in the Adapter Endpoint Configuration Wizard leads to major changes in the mapper that may require remapping.
>   - Integrations in which a child integration is invoked from a parent integration.
>   - Integrations with a REST Adapter using a Swagger-based connection.

1. Create and design an application integration.
2. Add an integration action to an integration in either of the following ways:
   - On the side of the canvas, click **Actions**  and drag the **Integration** action to the appropriate location.
   - Click  at the location where you want to add the integration action, then select **Integration**.

   The Integration Adapter Wizard is displayed.
3. Specify the following details, and click **Next**.

Element	Description
What do you want to call your local integration invocation?	Specify a name.
What does this local integration invocation do?	Specify a description.

4. Specify the following details, and click **Next**.

Copyright © 2022, 2026, Oracle and/or its affiliates.

Element	Description
Select Integration Context **Note**: This field is only displayed if the parent integration is in a project.	Select the location of the child integration to invoke. • **Project**: Select this option, then choose the specific project. Your current project is highlighted and selected by default at the top of the list. Two types of child integrations are shown for selection: – If you select the project you are already in, all integrations are shown, regardless of whether they are set as public. – If you select a different project, all integrations that are set as public in that project are shown. See Set a Project Integration as Public. • **Not in Project**: Select to invoke a child integration in the global context (that is, not available in a project). **Note**: You can invoke an integration not set as public from a runtime tool such as `Postman`.
Integration	Select the child integration to invoke. Only active integrations are displayed.
Identifier	Displays the identifier of the selected integration. This value cannot be changed.
Description	Displays the description of the selected integration.

5. Select the operation for the child integration to perform, and click **Next**. The operation depends on the child integration. Note the following:
   - If the child integration trigger connection is a REST Adapter, it can have Get, Post, Put, and Delete operations.
   - If the child integration trigger connection is a SOAP Adapter, it can have the WSDL operation name.
   - If the child integration trigger connection is a schedule, it processes a submit now configuration.
6. Review your selections, and click **Done**.
7. Complete design of the parent integration.
8. Specify appropriate business identifiers for tracking the integration at runtime.
9. Activate and invoke the integration.
10. In the navigation pane, click **Observability**, then **Instances**.
    If successful, the Instances page has two instances (one for the parent and one for the child). If unsuccessful, and the parent is unable to call the child, there is one entry (for the parent). If the parent can call the child, there are two entries. If the child fails or the child was successful, but the parent did not properly process the response, the parent may fail.

Create a Co-located Integration with Header Support

Headers are optional elements that pass extra information about your application requirements. For example, you can use the header element to specify a digital signature for password-protected services. You can configure and select the headers to send with the

Copyright © 2022, 2026, Oracle and/or its affiliates.

payload. Header selection is automatic based on the WSDL URL definition you provide on the Connections page.

Along with request and response SOAP and REST headers, you can configure standard HTTP headers with the SOAP Adapter and REST Adapter, custom HTTP headers with the SOAP Adapter and REST Adapter, and custom SOAP headers. You can view and edit these headers in the request and response mapper under the header elements for either the SOAP Adapter or REST Adapter.

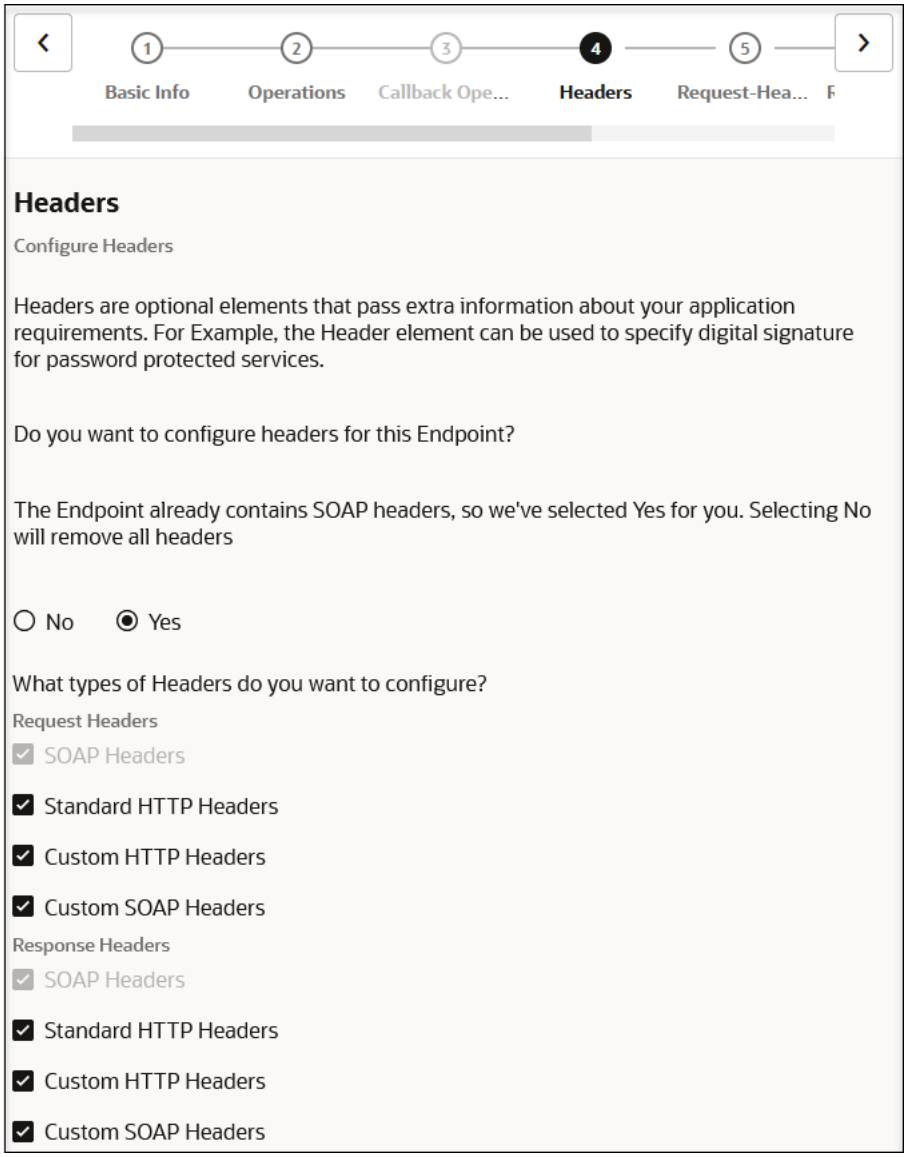
The following use case provides a high-level overview of creating a co-located integration with header support.

1. Create a SOAP Adapter connection (for this example) with a WSDL URL that includes the header definitions.



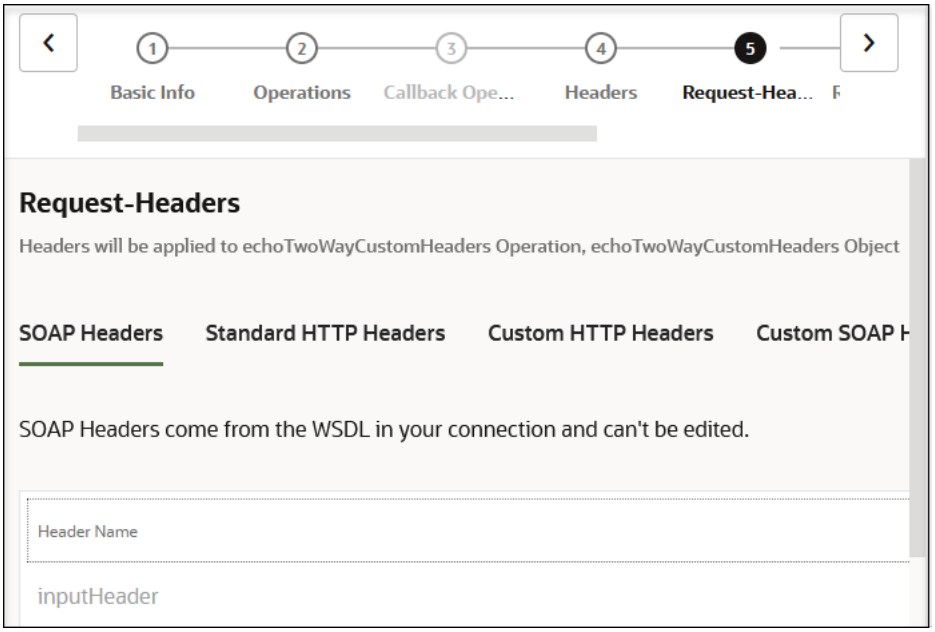
2. Complete the remaining fields on the Connections page.
3. Test and save the connection.
4. Create an integration to act as the co-located (child) integration.
5. Add the SOAP Adapter as a trigger connection in the integration.
6. Complete the pages of the Adapter Endpoint Configuration Wizard. In particular:
   a. On the Headers page, specify request and response header details.

Copyright © 2022, 2026, Oracle and/or its affiliates.

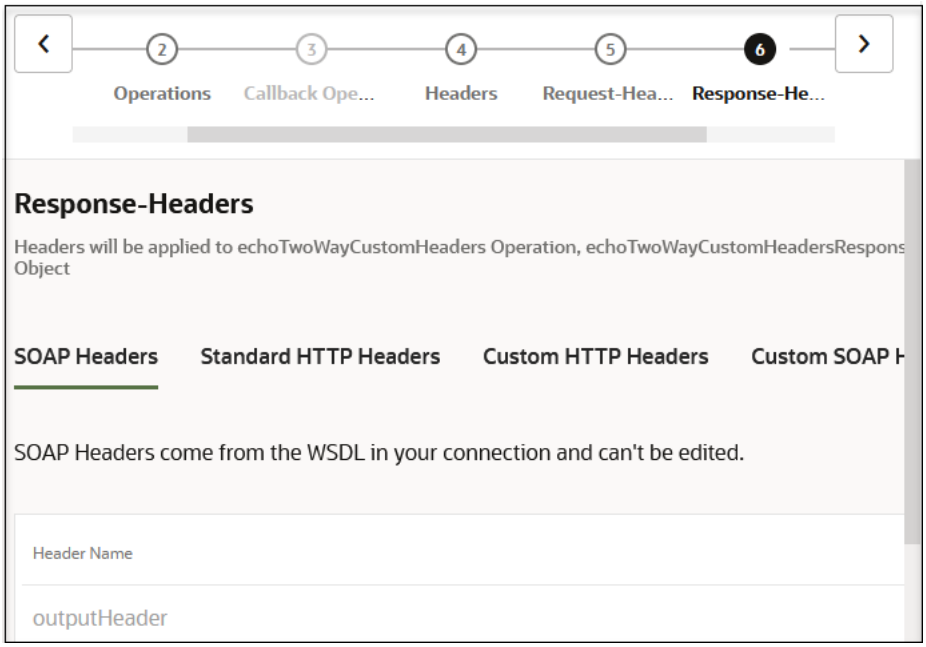


b. On the Request-Headers page, specify request header names.

Copyright © 2022, 2026, Oracle and/or its affiliates.



c. On the Response-Headers page, specify response header names.



7. In the mapper, map the request and response headers. While this example maps headers between two SOAP Adapters, you can also map request and response headers between two REST Adapters.
8. Save and activate the integration.
9. Create another integration to act as the parent integration.
10. Add the REST Adapter as a trigger connection in the integration.
11. Configure the pages in the Adapter Endpoint Configuration Wizard.
12. Add an **Integration** action to the integration.

Copyright © 2022, 2026, Oracle and/or its affiliates.

This invokes the Integration Adapter Wizard.

13. Complete the pages of the wizard. In particular, select the child integration created in Step 4 on the Selection Integration page.
14. Click the map icon between the REST Adapter trigger connection and SOAP invoke (the **Integration** action you configured) to configure the map.
15. Map the source **id** element to the target SOAP, HTTP, and custom header elements.
16. Save and activate the parent integration. The parent integration is now designed with a child integration as a SOAP invoke.

Backward Compatibility for Header Support

Note the following details when using header support in existing integrations created prior to the release of support for headers in co-located integrations.

Question	Answer
What happens if an existing integration created with a header is imported into an environment in which header support in co-located integrations is supported.	You can activate the integration without any issues. However, if you want to use header functionality, you must edit the adapter by clicking through the pages of the Adapter Endpoint Configuration Wizard, then clicking **Done** on the Summary page. In this scenario, mapping is not deleted; you must to correct it.
What happens if you perform the following steps. 1. Create a child flow with headers in a trigger connection. 2. Use this child endpoint as a SOAP connection. 3. Use the above SOAP connection as a trigger and create an integration in which you set **Do you want to configure headers for this Endpoint?** to **No** on the Headers page of the Adapter Endpoint Configuration Wizard. 4. Activate the integration. 5. Invoke this child integration in a parent integration that uses the **Integration** action. 6. Check if the integration generates a wrapper in the local invoke.	Based on the selection of **Yes** or **No**, headers are visible or invisible in the mapping.

Dynamically Invoke a Child Integration

You can dynamically invoke a (co-located) child integration at runtime. In this example, a parent integration can call either of two child integrations at runtime based on the data. The parent integration is designed to invoke child integration one. However, it can also invoke child integration two. The code and version values are passed to the appropriate child integration.

> **Note**
>
> Dynamic invocations are only supported with the REST Adapter.

1. Create and activate an initial child integration with a REST Adapter trigger connection.

Copyright © 2022, 2026, Oracle and/or its affiliates.

a. Create an integration (for this example, named `childone`).

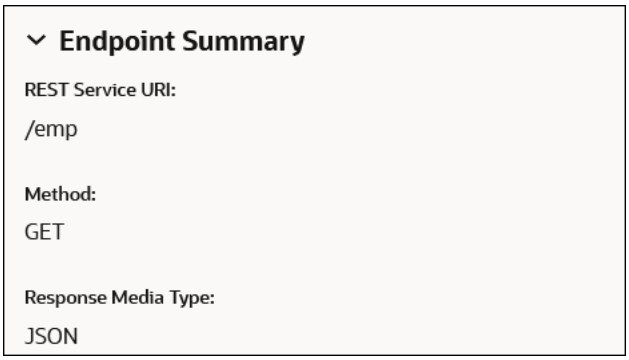
b. Add a REST Adapter trigger connection.

   This invokes the Adapter Endpoint Configuration Wizard.

c. Provide the necessary details on the pages of the wizard.

d. Review your configurations on the Summary page, then click **Done**. For this example, the following values are configured:

   - The relative resource URI is **/emp**.
   - The verb (method) is **GET**.
   - The response media is **JSON**.



e. Perform mappings in the mapper.

f. Activate the integration.

2. Create and activate a second child integration with a REST Adapter trigger connection.

   a. Create a second integration (for this example, named `childtwo`).

   b. Add a second REST Adapter trigger connection.

   c. Configure the second REST Adapter with the same values as the first REST Adapter.

   d. Perform mappings in the mapper.

   e. Activate the integration.

3. Create and activate a parent integration.

   a. Create an integration.

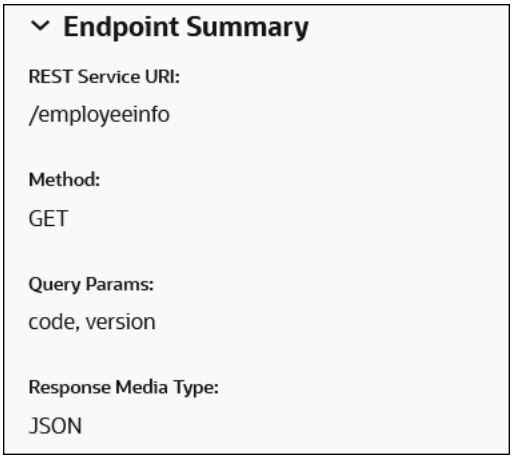
   b. Add a REST Adapter trigger connection.

   c. Provide the necessary details on the pages of the wizard, including creating the `code` and `version` query parameters on the Request Parameters page.

   d. Review your configurations on the Summary page, then click **Done**. For this example, the following values are configured:

      - The relative resource URI is **/employeeinfo**.
      - The verb (method) is **GET**.
      - The query parameters are **code** and **version**.
      - The response media is **JSON**.

Copyright © 2022, 2026, Oracle and/or its affiliates.

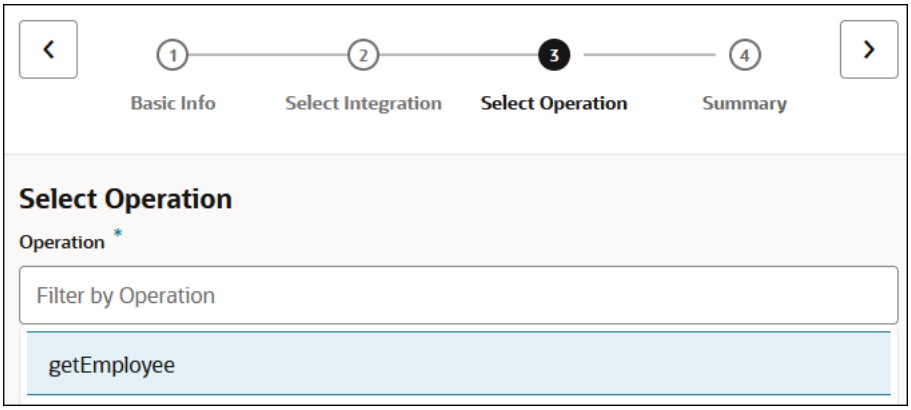


e. Add an **Integration** action to the integration.

   This invokes the Integration Adapter Wizard.

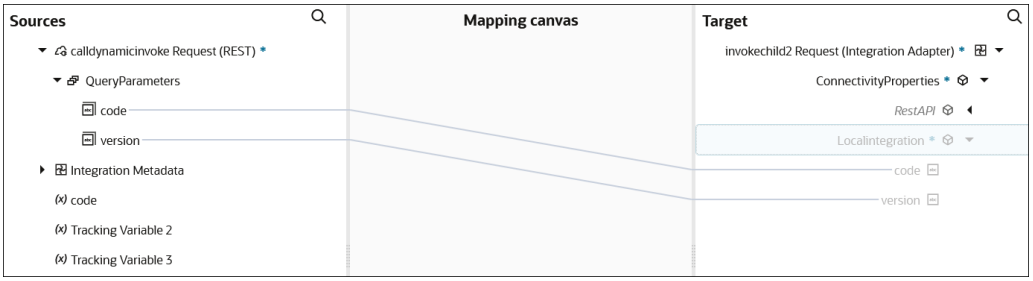
f. Select the child integration to invoke. For this example, **childone** is selected.

g. Select the operation.



h. Complete the wizard, and click **Done**.

i. Map the **code** and **version** source query parameters to the **code** and **version** target local integration elements.



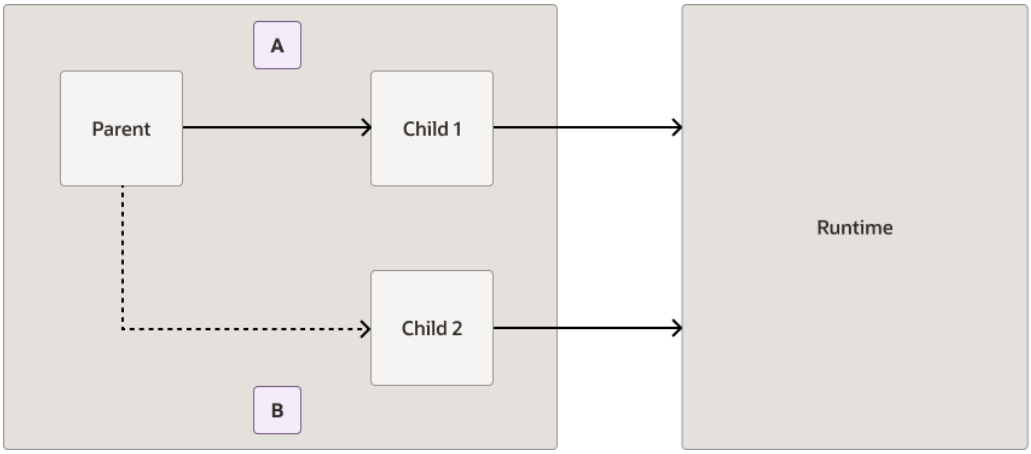
j. Complete design of any remaining actions in the integration.

k. Activate the integration.

Copyright © 2022, 2026, Oracle and/or its affiliates.

Even though the parent integration was designed to invoke only child integration one, you can now invoke either child integration by replacing the code and version of that child integration in the endpoint below.

```
http://host_name:port/ic/api/integration/v1/flows/rest/
DYNAMIC_INVOKE/1.0/employeeinfo?
code=[code-value]&version=[version-value]
```



Where:

- A - `http://host_name:port/ic/api/integration/v1/flows/rest/DYNAMIC_INVOKE/1.0/employeeinfo? code=CHILDONE&version=1.0`
- B - `http://host_name:port/ic/api/integration/v1/flows/rest/DYNAMIC_INVOKE/1.0/employeeinfo? code=CHILDTWO&version=2.0`

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 the Package for an Integration
- Delete a Package
- Convert a Package to a Project

When you create an integration, you can also create a package or select an existing package in which to include the integration. See Create an Integration.

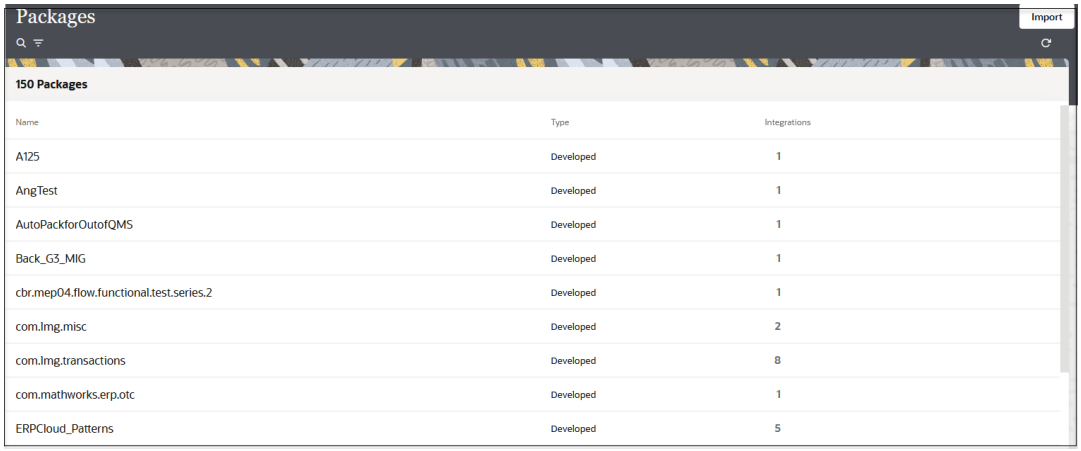
View the Integrations in a Package

You can view the integrations included in a package.

Copyright © 2022, 2026, Oracle and/or its affiliates.

1. In the navigation pane, click **Design**, then **Packages**.

   The Packages page is displayed. The package names and the number of integrations included in each package are displayed. If you have not yet created or imported a package into Oracle Integration, this page is empty.



2. Use **Search** or **Filter** to find the package you want, then click the package name.

   The integrations included in the selected package and their current states are displayed (for example, active or inactive).

3. Click **Close**.

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 Cloud Marketplace.

> **Note**
>
> You cannot import a package if it contains activated integrations. Ensure that all integrations are deactivated before including them in a package archive (PAR) file.

1. In the navigation pane, click **Design**, then **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 prebuilt packages from Oracle Cloud Marketplace. See Import a Prebuilt Integration and Get Familiar with the Home Page in *Getting Started with Oracle Integration 3*.

Copyright © 2022, 2026, Oracle and/or its affiliates.

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 **Design**, then **Packages**.
2. Hover over the package to export.
3. Click **Actions** ⋮, then select **Export**.
4. Click **Export** when prompted to confirm your selection.
5. Save the package (PAR) file of integrations to a file system location when prompted. The individual integrations inside the PAR file are exported as integration archive (IAR) files.

Update the Package for an Integration

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. In the navigation pane, click **Design**, then **Integrations**.
2. Hover over the integration of the package that you want to update, then click **Edit** .

> **Note**
>
> The integration must not be active.

3. Click **Primary Info** .
4. In the **Package** field, enter a new package name or search for 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**.
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 **Design**, then **Packages**.
2. Hover over the package to delete.
3. Click **Actions** ⋮, then select **Delete**.
4. Click **Delete** when prompted to confirm your selection. The package and *all* of its integrations are deleted.

Copyright © 2022, 2026, Oracle and/or its affiliates.

> **Note**
>
> 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 the Package for an Integration.

Convert a Package to a Project

You can convert a user-developed package into a project in Oracle Integration. This action creates a project that includes all resources in the package (integrations, connections, lookups, and JavaScript libraries). The existing package remains untouched.

1. In the navigation pane, click **Design**, then **Packages**.
2. Hover over the user-developed package to convert to a project. Accelerator and recipe packages cannot be converted to projects.
3. Click **Actions** ⋮, then select **Create Project**.
4. Enter a name and optional keywords and descriptions for the project.
5. Review the following options to determine if you want to select the **Anyone can edit, view, and monitor** check box.
   - Select this option to allow anyone to edit, view, and monitor resources in the project. You can always restrict access at a later time.
   - Don't select this option if you want to restrict access to editing, viewing, and monitoring resources in this project. After project creation, you can select the specific groups and users to which to provide access to this project. See Control Who Can Edit, View, and Monitor in a Project.
6. Click **Create**.

   A message is displayed.

   ```
   Project project_name has been submitted for creation. Click here to see
   the newly created project.
   ```

7. Click the link to access the Projects page.
8. Open the project and note that the resources available in the package (integrations, connections, lookups, and JavaScript libraries) are now available in the project.

Import and Export Integrations

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

See the following topics:

- Import an Integration
- Import a Prebuilt Integration
- Export an Integration

Copyright © 2022, 2026, Oracle and/or its affiliates.

Import an Integration

You can import integrations that were previously exported as an IAR file from Oracle Integration. You must update your connection properties after the import.

> **Note**
>
> - 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.
> - You cannot import an Oracle Integration 3 integration into Oracle Integration Generation 2.
> - You can export an integration from a higher service instance version and import it into a lower service instance version *as long as* no new features from the higher version are used (for example, create an integration on a 25.02 development instance, then export and import it to a 24.10 production instance).
> - If you import a basic routing integration into Oracle Integration 3, it is converted to an application integration.

To import an integration:

1. Decide where to start:
   - Work in a project.
     a. In the navigation pane, click **Projects**.
     b. Select the project name.
     c. Click **Integrations**.
     d. In the **Integrations** section, click **Add** if no integrations currently exist or **+** if integrations already exist.
     The Add integration panel opens.
     e. Click **Import**.
     The Import integration panel appears.
   - Work outside a project.
     a. In the navigation pane, click **Design**, then **Integrations**.
     b. In the banner, click **Import**.
     The Import integration panel appears.
2. Click **Drag and Drop** to select a file or drag one to this field.
3. Select one of the following options.
   - **Import**: Imports the integration for you to configure your connection properties at a later time.
   - **Import and Configure**: Imports the integration and shows the Configuration Editor page for configuring your connection properties now.

Copyright © 2022, 2026, Oracle and/or its affiliates.

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 for you to configure or activate.

Import a Prebuilt Integration

You can import prebuilt integrations into your Oracle Integration environment.

There are two types of prebuilt integrations:

- User-created integrations. These are integrations that you or another user created. See Import an Integration.
- Oracle-created integrations from the Oracle Integration Home page or from the Oracle Cloud Marketplace as part of a package. Integrations from the Home page are designated with a **RECIPE** message that is displayed next to the integration name on the Integrations page. See Get Familiar with the Home Page in *Getting Started with Oracle Integration 3* and the Oracle Cloud Marketplace.

Export an Integration

Once you create an integration, you can export that integration as an IAR file for use in other Oracle Integration environments. You can export a locked integration. You can also export integrations that publish and subscribe to events and import them into a project.

To export an integration:

1. In the navigation pane, click **Design**, then **Integrations**.
2. Hover over the integration to export.
3. Click **Actions** ⋮, then select **Export**.

   A message is displayed indicating that the integration has been downloaded. 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.

Clone the Design-Time Metadata of an Entire Service Instance

You can clone (create) archives of integration design-time metadata to export and import between Oracle Integration 3 instances. This feature can be useful if you want to take a snapshot of a production environment and try to recreate a problem in a different environment,

Copyright © 2022, 2026, Oracle and/or its affiliates.

move metadata from an instance in one region to an instance in another region, perform manual archives, or automate your environment to archive daily to a repository such as Git.

> **Note**
>
> You cannot export Oracle Integration Generation 2 archives and import them into Oracle Integration 3 or export Oracle Integration 3 archives and import them into Oracle Integration Generation 2.

Topics:

- Step 1: Create an Object Storage Bucket and Construct the Storage URL
- Step 2: Configure the Instance Object Storage Bucket
- Step 3: Export and Import Design-Time Metadata Between Instances
- Step 4: Perform Post-Processing Steps

Step 1: Create an Object Storage Bucket and Construct the Storage URL

You must create an object storage bucket (if one does not exist) and construct the storage URL in the Oracle Cloud Console in order to export and import design-time metadata into an Oracle Integration instance. You specify this URL during object storage bucket configuration on the Instance storage page.

1. Log in to the Oracle Cloud Console with the user account that your administrator added to the group for exporting design-time metadata. You must have the admin role to create an object storage bucket. See Manage Access and Assign Roles in *Provisioning and Administering Oracle Integration 3*.
2. In the navigation pane, select **Storage**.
3. Click **Buckets**.
4. Select the compartment in which to create the storage bucket.
5. Click **Create Bucket**.
6. Enter the following information, then click **Create**.

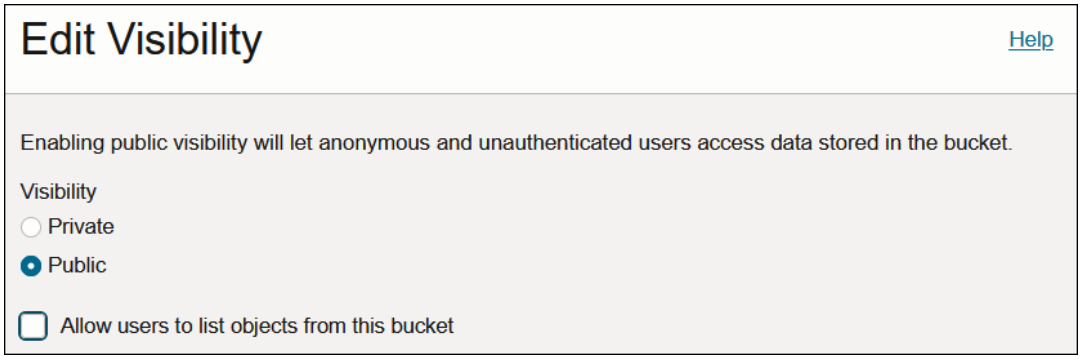
Element	Description
Bucket Name	Enter a name.
Default Storage Tier	Select **Standard**.
Enable Auto-Tiering	Automatically move infrequently accessed objects from the Standard tier to less expensive storage. See Managing Buckets.
Enable Object Versioning	Create an object version when a new object is uploaded, an existing object is overwritten, or an object is deleted. See Using Object Versioning.
Emit Object Events	Create automation based on object state changes using the Events Service. See Overview of Events.
Uncommitted Multipart Uploads Cleanup	Create a life cycle rule to automatically delete uncommitted multipart uploads older than seven days. See Using Object Lifecycle Management.

Copyright © 2022, 2026, Oracle and/or its affiliates.

Element	Description
Encryption	Select an encryption method. • **Encrypt using Oracle managed keys** Leaves all encryption-related matters to Oracle. • **Encrypt using customer-managed keys** Requires a valid key from a vault that you can access. See Overview.
Resource logging	Click to enable resource logging to allow resource tracking, troubleshooting, and data insights. By default, resource logging is disabled.
Tags	Add tags to organize your resources.

The new bucket is displayed in the list.

7. To view bucket details, click the name.
8. Click **Edit Visibility**.
9. Click **Public** and deselect the **Allow users to list objects from this bucket** check box. This action ensures that the storage bucket you create is publicly visible. Otherwise, the export utility cannot see the storage.



10. In the navigation pane, select **Identity & Security**.
11. Under **Identity**, select **Domains**.
12. Select the name of the identity domain in which you want to work. For example, under the **Profile** icon in the upper right corner, select the domain. You may also need to change the compartment to find the domain that you want.
13. Click **User management**.
14. Select the appropriate user.
15. Under **Resources**, select **Auth Tokens**.
16. Click **Generate Token**.
17. Provide a token description and click **Generate Token**.
18. Copy and save the generated token immediately because you cannot retrieve it after closing the dialog.

    For additional information, see Creating an Auth Token.

Copyright © 2022, 2026, Oracle and/or its affiliates.

The storage payload format is as follows with object storage bucket details:

```
{"storageInfo":
       {
             "storageUrl":"https://
swiftobjectstorage.region.oraclecloud.com/v1/
namespace/bucket",
             "storageUser":"my.email@oracle.com",
             "storagePassword":"generated_token"
            }
}
```

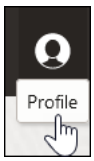
For example:

```
{
      "storageInfo":
            {
                  "storageUrl":"https://swiftobjectstorage.us-
ashburn-1.oraclecloud.
com/v1/paasdevoic/cloneRepo",
                  "storageUser":"my.email@oracle.com",
                  "storagePassword":"generated_token"
            }
}
```

19. Construct the storage URL using the following example. You must specify the URL later when configuring the object storage bucket. See Step 2: Configure the Instance Object Storage Bucket.

Step	Description	Example
1	Enter the Swift API as the base for the URL because you are dealing with object storage.	`https://swiftobjectstorage`
2	Enter a required data center. To find a data center, go to the drop-down menu in the banner (for example: `us-ashburn-1`).	`https://swiftobjectstorage.us-ashburn-1`
3	Enter the domain name. This is typically `oraclecloud.com`.	`https://swiftobjectstorage.us-ashburn-1.oraclecloud.com`
4	Enter the version: `v1`.	`https://swiftobjectstorage.us-ashburn-1.oraclecloud.com/v1`

Copyright © 2022, 2026, Oracle and/or its affiliates.

Step	Description	Example
5	Enter the automatically generated object storage namespace. a. In the upper right, click the **Profile** menu.  b. Click **Tenancy**: ***your_tenancy_name***. The namespace string is listed under **Object Storage Settings**. See Understanding Object Storage Namespaces.	`paasdevoic`
6	Enter the bucket name you created in Step 6 as the last part of the URL.	`https://swiftobjectstorage.us-ashburn-1.oraclecloud.com/v1/paasdevoic/`*`BUCKET_NAME`*

Examples of the full storage URL to use later on the Instance storage page in Step 2: Configure the Instance Object Storage Bucket are as follows:

```
https://swiftobjectstorage.us-ashburn-1.oraclecloud.com/v1/paasdevoic/
cloneRepo
```

```
https://swiftobjectstorage.us-ashburn-1.oraclecloud.com/v1/paasdevoic/
migrationartifacts_bucket-20190717-1223
```

Step 2: Configure the Instance Object Storage Bucket

You must specify the Swift URL location and user name and password credentials of your existing object storage bucket instance before you can create export and import archives of design-time metadata on the Import and Export page.

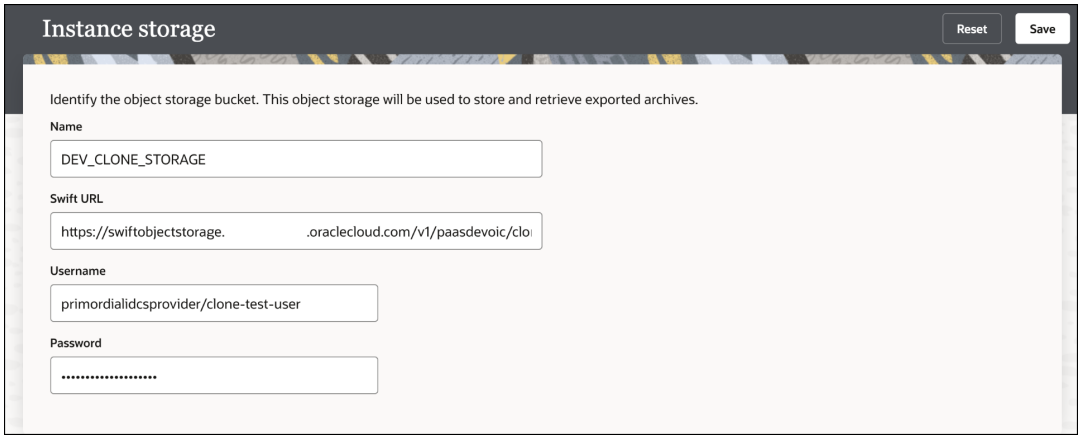
To complete the fields on this page, an object storage bucket instance must already exist. Otherwise, you must create a new instance. See Step 1: Create an Object Storage Bucket and Construct the Storage URL.

1. In the navigation pane, click **Settings**, then **Storage**.
2. Enter the following details. You specified these details when creating your object storage bucket. See Step 1: Create an Object Storage Bucket and Construct the Storage URL.

Element	Description
Name	Enter the name of the object storage bucket.

Copyright © 2022, 2026, Oracle and/or its affiliates.

Element	Description
Swift URL	Enter the object storage bucket Swift URL you constructed. For example: `https://swiftobjectstorage.us-ashburn-1.oraclecloud.com/v1/paasdevoic/cloneRepo` See Step 1: Create an Object Storage Bucket and Construct the Storage URL.
User	Specify the object storage bucket user name. The user name and password were created when you generated the token in the Oracle Cloud Console. See Step 1: Create an Object Storage Bucket and Construct the Storage URL. **Note**: Ensure that the user name is prefixed with the domain name. For example: `primordialidcsprovider/clone-test-user` You can also get this user name from your profile link in the top right corner of the Oracle Cloud Console. Not specifying the domain name can result in an error. See Object Storage Bucket Error Creation.
Password	Specify the password.



3. Click **Save**. You can now export and import archives of design-time metadata on the Import and Export page.

Step 3: Export and Import Design-Time Metadata Between Instances

You can export and import archives of integration design-time metadata between instances. This feature can be useful if you want to take a snapshot of a production environment and try to recreate a problem in a different environment, move metadata from an instance in one

Copyright © 2022, 2026, Oracle and/or its affiliates.

region to an instance in another region, perform manual archives, or automate your environment to archive daily to a repository such as Git.

> **Note**
>
> - Ensure that you first configure the Swift URL location and username and password credentials of the object storage bucket instance to which to export an archive of your design-time metadata on the Instance storage page. See Step 2: Configure the Instance Object Storage Bucket.
> - You cannot export Oracle Integration Generation 2 archives and import them into Oracle Integration 3 or export Oracle Integration 3 archives and import them into Oracle Integration Generation 2.
> - This feature cannot be used to export specific integrations. All integrations are exported.
> - The target instance must be completely empty prior to importing the exported archive job.

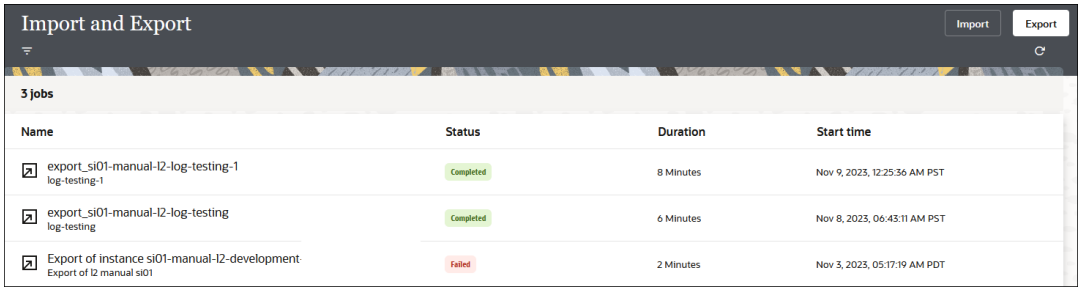
- Create an Export Job
- Create an Import Job

Create an Export Job

You create an export job that consists of an archive file of design-time metadata that you want to export to the object storage bucket you configured on the Instance storage page.

1. In the navigation pane, click **Settings**, then **Import and export**.

   The Import and Export page is displayed with the status of any import and export jobs.

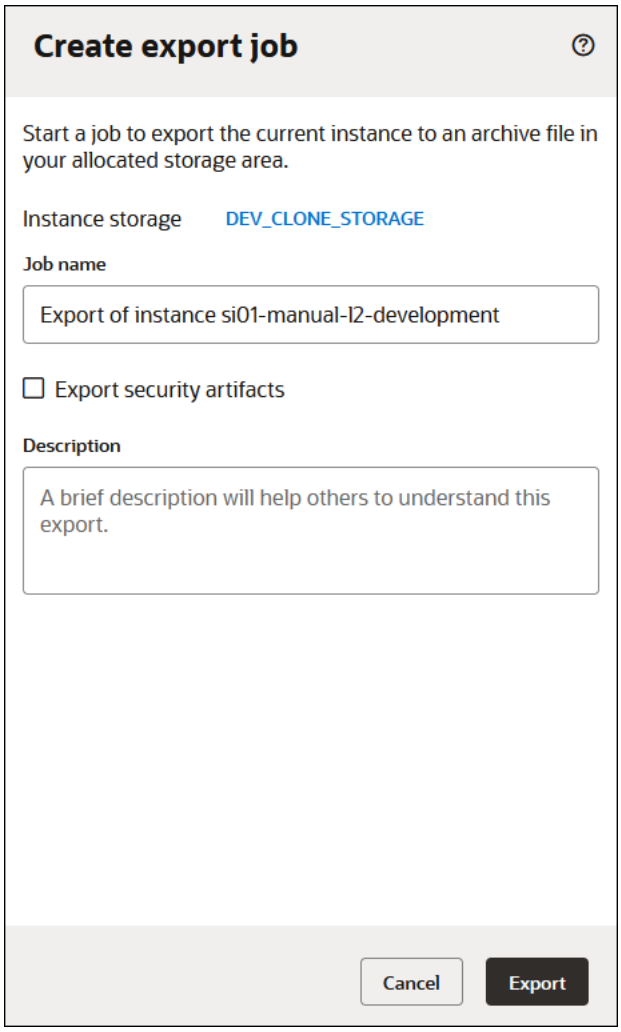


2. Click **Export** to create a job. A job consists of an archive file of design-time metadata that you want to export to the object storage bucket. If you have not configured an object storage bucket, you are prompted to click **Configure now**.
3. Complete the following fields.

Element	Description
Job Name	Enter a unique job name or accept the default value.

Copyright © 2022, 2026, Oracle and/or its affiliates.

Element	Description
Export security artifacts	Select the check box to export the following security artifacts with your job: • Security policies • Security credentials (for connections) • Customer certificates
Description	Enter an optional description that describes the export job.



4. Click **Export**.

   A message is displayed in the banner at the top of the page.

   ```
   Export job has been successfully started.
   ```

5. View the status of export job creation and click **Refresh** periodically to view progress.

   When the export job completes successfully, **Completed** is displayed in the **Status** field.

6. Click **Details** to view details about the job.

Copyright © 2022, 2026, Oracle and/or its affiliates.

7. If export archive creation does not complete successfully, click **Download detailed report** to download a report about the export job.

Create an Import Job

You create a job to import the exported archive job from the object storage bucket instance into the new instance.

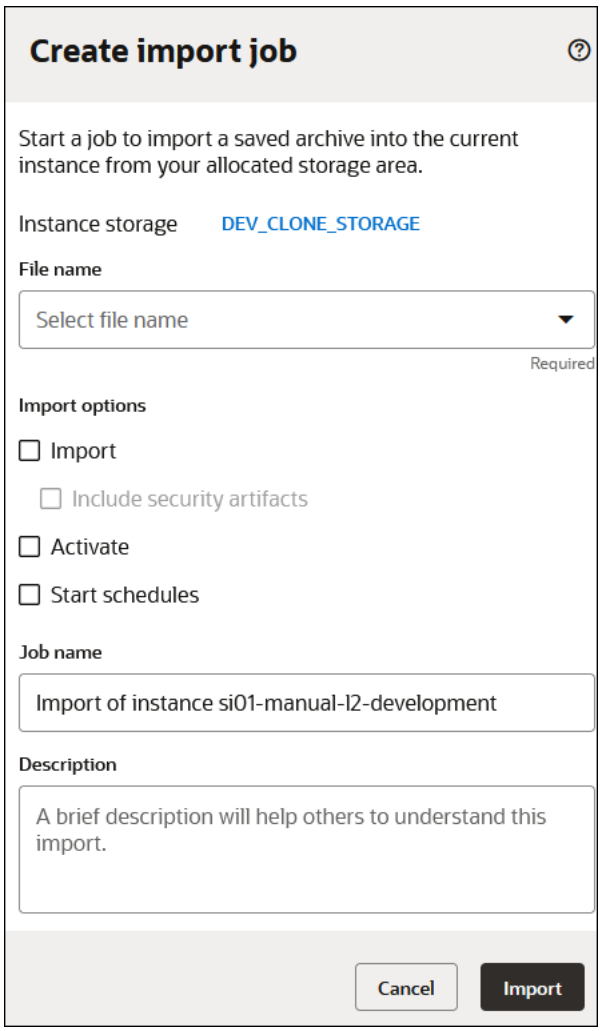
1. Sign in to the instance in which to import the exported archive of design-time metadata.

> **Note**
>
> The target instance must be completely empty prior to importing the exported archive job.

2. In the navigation pane, click **Settings**, then **Import and export**.
3. Click **Import** to create a job to import the exported archive job from the object storage bucket instance into the new instance.
4. Complete the following fields.

Element	Description
File name	Select the archive from the list to import into the instance.
Import options	Select the import mode: • **Import**: Imports all integrations in the archive. You can also select **Activate** and **Start schedules** to activate all integrations and start all schedules during this same import session or during a separate session at a later time. Selecting those options separately enables you to first update any configuration properties in the imported integrations (for example, modify any necessary configuration or security properties on the Connections page for each integration). • **Activate**: Activates all integrations imported with the **Import** option during the same session or during a separate session. You can also select **Start schedules** to start any integration schedules. • **Start schedules**: Starts integration schedules during the same session in which you selected **Import** and **Activate** or during a separate session.
Import security artifacts	Select this check box if you previously selected **Export security artifacts** when creating your export archive job. You can also show this option by clicking **Import**.
Job name	Enter a unique job name or accept the default value.
Description	Enter an optional description that describes the import job.

Copyright © 2022, 2026, Oracle and/or its affiliates.



5. Click **Import** to start the job to import the archive from the object storage bucket instance into the new Oracle Integration instance.

   A message is displayed in the banner at the top of the page.

   ```
   Import job has been successfully started.
   ```

6. View the status of import job creation and click **Refresh** periodically to view progress.

   When the import job completes successfully, **Completed** is displayed in the **Status** field.

7. Click **Details** to view details about the job.

8. If the import archive is not successful, click **Download detailed report** to download a report about the import job.

9. Browse the pages and note that the design-time metadata you exported is now visible. For example, for Integrations, look for integrations, connections, lookups, and more.

Copyright © 2022, 2026, Oracle and/or its affiliates.

> **Note**
>
> Your archive file resides in the object storage bucket until you delete it.

Step 4: Perform Post-Processing Steps

If you selected the **Import security artifacts** checkbox and **Activate** under the **Import options** section when creating your import job, you must manually perform the following tasks after importing the job.

The following types of integrations were not automatically activated when importing the job into the new instance and require a manual update to their connections before activation.

1. Reconfigure the connectivity agents on the new instance.
   a. Test the connections associated with the connectivity agents.
   b. Activate the integrations using these agent-based connections that were active on the source instance.
2. If you had connections using the following security policies on the source instance, reconfigure and test the connections, then activate the integrations:
   - Using Three-legged OAuth.
   - Using OAuth on the current Oracle Integration instance (source) to access the APIs described in the *Developer API for Oracle Integration 3*.
   - Using OAuth on the current Oracle Integration instance (source) to access the integration endpoints.
3. Start the schedules related to the above integrations that were active on the source instance.

Refresh Endpoints 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, user name, 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. For custom fields or updated fields to appear in the mapper, you must refresh the endpoints in Oracle Integration.

As an example, the WSDLs for one of your connections (for example, a Salesforce connection) change frequently and you must uptake the latest WSDLs into your integrations. By refreshing the WSDL file, the custom fields of imported mappings are not deleted, and are available for editing, when needed. This eliminates the need to remap source and target elements completely from scratch in the mapper.
Endpoint refreshing is an asynchronous operation. This reduces the chance of timeouts occurring. The refresh can scale from a single action to an entire integration.

Restrictions

- Integrations cannot be refreshed when they are active, being activated, or being deactivated.
- Integrations are locked during the refresh. No users can modify the integration during this period.

Copyright © 2022, 2026, Oracle and/or its affiliates.

- Integration activation is blocked during a refresh.
- During endpoint refreshing, you can view or exit the integration canvas. If you are still on the canvas when endpoint refreshing completes, the canvas refreshes and you can resume editing. If you leave the canvas, you receive an alert that the refresh has completed, and the lock is released.
- The Connections page shows a warning message if you attempt to edit during integration refreshing.
- You cannot unlock an integration until refreshing completes.
- Once refreshing completes successfully, you cannot roll back to the previous state.
- Minimize your use of refreshing. Only use this feature when absolutely necessary.
- There cannot be root level differences between the old and new WSDLs. If there are differences, WSDL refreshing 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 Oracle Service Cloud (RightNow) Adapter WSDL version 1.2 is used and try to refresh it with Oracle Service Cloud (RightNow) Adapter WSDL version 1.3, a mapper error occurs.
- JCA artifacts are not refreshed.
- 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 is *not* 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 refresh 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` must not change.
  - The old root type must 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 refresh 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 is not imported or included in the final refreshed WSDL.

Copyright © 2022, 2026, Oracle and/or its affiliates.

> **Note**
>
> If you receive an error during the refresh, do not save the endpoint/mapping updates. Instead, close and discard the integration without saving, reopen the integration for editing, and try refreshing again.

1. Decide where to start:
   - Work in a project.
     a. In the navigation pane, click **Projects**.
     b. Select the project name.
     c. Click **Integrations**.
     d. In the **Integrations** section, find the integration to refresh.
   - Work outside a project.
     a. In the navigation pane, click **Design**, then **Integrations**.
     b. Find the integration to refresh.

   You can refresh the endpoints for an individual endpoint or all endpoints in an integration.
2. Refresh the endpoints in either of two ways:
   a. Click **Actions**.
   b. Select **Refresh endpoints** for the appropriate endpoint.

   Or:

   a. Click the name of the integration in which to refresh the endpoints.

      The integration canvas is displayed.
   b. Click the icon in the palette above the integration to refresh the endpoints.

      The Refresh endpoints panel opens.
3. Select the connections from the list for which to refresh artifacts. Refreshing endpoints may impact the business identifiers and maps.
4. Click **Refresh**.

   The refresh of the impacted connection endpoints is displayed as in progress. On the Integrations page, the state of the overall integration refresh is also displayed as in progress.

   This process refreshes the endpoint and any dependent artifacts to enable any custom elements to 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.

Copyright © 2022, 2026, Oracle and/or its affiliates.

5. If you need to cancel in-progress endpoint refreshing, click **Stop and rollback**.

See Map Data in *Using the Oracle Mapper with Oracle Integration 3*.

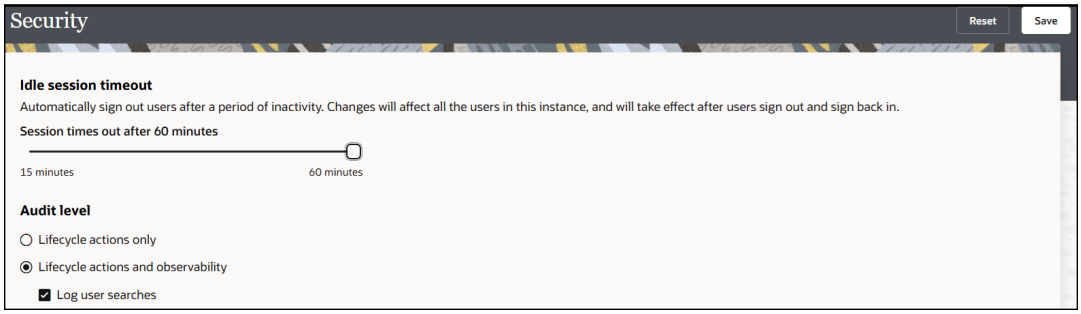
Set the Inactive Session Timeout Limit and Audit Logging Levels

You can set the time limit for inactive sessions in an Oracle Integration instance. This setting automatically signs out a currently authenticated user after a specified period of inactivity. This setting impacts all users of the instance and takes effect after a user signs out and signs back in. You can also select to capture lifestyle and observability actions in the audit logs.

Note the following guidelines for using the options on this page:

To Set the Timeout for Inactive Sessions	To Set the Audit Levels
• Only a user with the ServiceAdministrator role can view the Security page and set the time limit for an inactive session in an Oracle Integration instance. • The default value for an inactive session in an instance is 60 minutes, unless a user with the ServiceAdministrator role sets it to another value. • The minimum and maximum values for session inactivity timeouts are 15 minutes and 60 minutes, respectively. • Assume you have multiple open tabs in your browser for the same integration instance. – You are actively working in one tab. – The other tab remains idle. When the session inactivity timeout value for the integration instance is reached for the idle tab, you are not logged out of either tab. • Assume user A is signed in when the session inactivity timeout is set to 45 minutes. A user with the ServiceAdministrator role then changes the setting to 30 minutes. The previous setting of 45 minutes remains in effect for user A as long as they are signed in. Once they sign out and sign back in, the setting of 30 minutes takes effect.	• Only a user with the ServiceAdministrator role can activate or deactivate observability auditing. This change is always logged to the lifecycle audit log.

1. In the navigation pane, click **Settings**, then **Security**.



2. To set the timeout limit for inactive sessions:
   a. Drag the slider to the applicable session inactivity value.

Copyright © 2022, 2026, Oracle and/or its affiliates.

b. Click **Save**.

c. If you want to revert to the previous session inactivity value, click **Revert**.

3. To set the audit level for lifecycle actions and observability.

a. Select the level of auditing to perform.

Element	Description
Lifecycle actions only	Logs lifecycle actions to the audit logs. This option is enabled by default. Lifecyle actions include the following: • Integration activation and deactivation • Connection creation, deletion, and update • Integration unlocking, importing, and exporting
Lifecycle actions and observability	Select to log both lifecycle actions and observability details to the audit logs (such as runtime messages). The observability audit captures the following actions: • Changing the tracing level of an integration. • Accessing any Observability page or API that displays business identifiers or any payload data, specifically: – Viewing errors by selecting **Observability**, and then **Errors**. – Accessing instances by selecting **Observability**, and then **Instances**. – Accessing the activity stream by selecting **Observability**, and then **Instances**. This is an additional audit event on top of accessing the Instances page that logs the specific integration that was viewed. – B2B tracking. • Viewing or downloading logs from the activity stream. • Viewing or downloading audit logs that contain any payload data or business identifier. Selecting **Lifecycle actions and observability** enables the following option to be displayed: • **Log user searches**: Select to log the search text you entered. Because this is an asynchronous operation, performance impact is minimal. For example, if you go to the **Instances** page under **Observability**, then enter `OrderOpportunity` as the text for which to search, the audit log captures the text `OrderOpportunity`. If **Log user searches** is not selected, the audit log records that a search was conducted, but does not record the actual text that was entered.

The retention periods for these audit logs are the same as other audit logs.

See Check the Design Time Audit History for an Integration or Other Component.

b. Click **Save**.

c. If you want to revert to the previous audit level values, click **Revert**.

Copyright © 2022, 2026, Oracle and/or its affiliates.

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. The integration must not be active. See Deactivate an Integration for instructions.

1. In the navigation pane, click **Design**, then **Integrations**.
2. Hover over the integration to change, then click **Edit** .
3. To modify the trigger or invoke adapter connection, select the connection and click **Actions** , then click **Edit** .
4. Modify any necessary fields in the pages of the Adapter Endpoint Configuration Wizard.
5. To assign a new invoke connection, select the connection, and click **Actions** , then click **Delete** . You cannot delete a trigger connection in an integration.
6. Add the new adapter to the canvas.
7. Configure the new connection.
8. To modify a data mapping, click the appropriate map icon and update the mappings. See Mapping Data of *Using the Oracle Mapper with Oracle Integration 3*.
9. When you are done making changes, click **Save**.

> **Note**
>
> If your integration is locked and stuck in Draft mode, see Recover Unsaved Integration Changes.

Impact on the Mapper of Editing the Endpoint Information in an Integration

You can edit the endpoint information in an integration that is not active. The changes that you make can impact your mappings. For example, minor edits such as changing the endpoint description do not delete the existing mappings. Major edits such as changing the selected business objects or operations delete the mappings. In these cases, you must recreate your mappings. Before you save your updates, you are prompted to confirm your changes.

The impact of major and minor endpoint changes on an integration are as follows:

- If a minor change is detected, for example:
  - If a map is using either the request or response of the application as a primary input, output, or secondary input, the map is validated.
  - If a map is using a fault of the application as a primary input or output, the map is deleted.
  - If a map is only using a fault of the application as a secondary input, the secondary input is removed.
- If a major change is detected, for example:

Copyright © 2022, 2026, Oracle and/or its affiliates.

  - 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 **Design**, then **Integrations**.
2. Hover your cursor over the integration to edit, then click **Edit** .

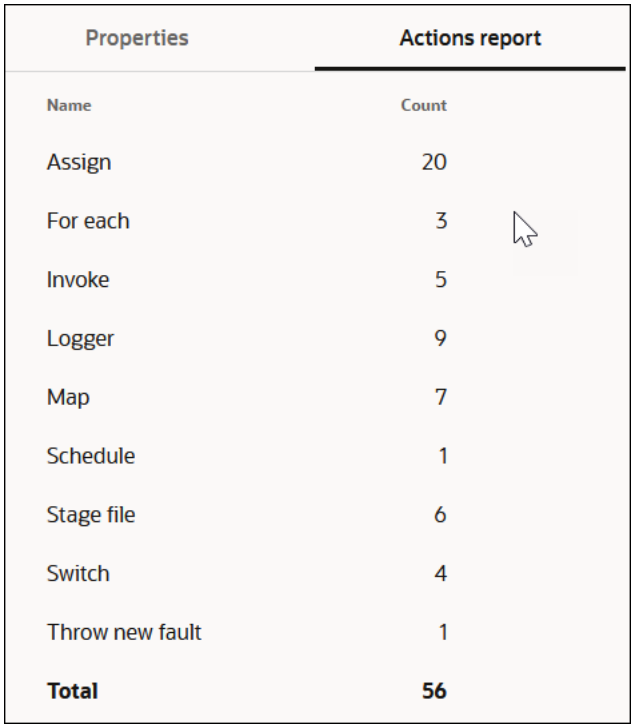
   The integration must not be active.
3. In the integration canvas, select the trigger or invoke endpoint to edit, then click **Edit** .
4. Make appropriate changes in the Adapter Endpoint Configuration Wizard, then click **Done**.
5. Select to confirm your changes when prompted. Minor edits do not delete your mappings. Major edits delete your mappings.

View the Actions and Connections in an Integration

You can view the total number of actions and invoke connections in an integration.

1. In the navigation pane, click **Design**, then **Integrations**.
2. Click the integration to view.
3. Click **Primary Info** .
4. Click **Actions report**.
   A summary of each action and invoke connection in the integration is displayed.

Copyright © 2022, 2026, Oracle and/or its affiliates.



Properties | Actions report

Name	Count
Assign	20
For each	3
Invoke	5
Logger	9
Map	7
Schedule	1
Stage file	6
Switch	4
Throw new fault	1
Total	**56**

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 `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 an Integration.

1. In the navigation pane, click **Design**, then **Integrations**.
2. Hover over the integration to clone.
3. Click **Actions** ⋯, then select **Clone**.

   The Clone integration panel opens.
4. Enter a name, unique identifier, version number, documentation URL, keywords, 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`.

Copyright © 2022, 2026, Oracle and/or its affiliates.

5. Click **Clone**.
6. You can modify the clone in any of the ways described in Modify an Integration.

Create a Draft Version of an Integration

Creating a draft version 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. Unlike a cloned integration, you *cannot* update the integration name or identifier. You can also create a draft of 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 `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.

You cannot create a draft of a recipe or accelerator.

1. Decide where to start:
   - Work in a project.
     a. In the navigation pane, click **Projects**.
     b. Select the project name.
     c. Click **Integrations**.
     d. In the **Integrations** section, hover over the integration for which to create a draft version.
   - Work outside a project.
     a. In the navigation pane, click **Design**, then **Integrations**.
     b. Hover over the integration for which to create a draft version.
2. Click **Actions**, then select **Create new version**.

   The Create new version panel opens.
3. Modify the name, unique identifier, version number, keywords, and optionally the documentation URL, keyword, package, and description. Not modifying the version number results in the following error:

   ```
   The version of Integration "integration_name" you are trying to create
   already exists in the system.
   ```
4. Click **Create version**.

Copyright © 2022, 2026, Oracle and/or its affiliates.

Delete an Integration

You can delete an integration that is no longer required.

Make sure the integration you want to delete is not active. To deactivate the integration, see Deactivating an Integration.

1. In the navigation pane, click **Design**, then **Integrations**.
2. Hover over the integration to delete.
3. Click **Actions** ⋮, then select **Delete**.
4. Click **Delete** when prompted.

   Recipe and accelerator integrations cannot be deleted individually. You must delete the package containing these integrations.

Recover Unsaved Integration Changes

You can recover unsaved changes in an integration that crashes during design time because of a browser crash, loss of network connectivity, a server going down, and so on.

When you design an integration, a backup copy is created whenever you complete a task such as configuring an action, deleting an invoke connection, configuring business identifier tracking, or defining global faults. There is no time limit in which you need to log back in and recover your changes. The unsaved changes remain as long as the integration is not unlocked by an administrator.

The following data in an integration cannot be recovered after a crash:

- Changes made while editing an action (for example, you are designing a stage file action when your integration crashes)
- Changes made while working inside the mapper
- Layout changes such as repositioning an action
- Connections page edits

Recovery behavior is different based on the roles assigned to the user. For example, assume you are designing the following integration when your browser crashes.

1. Design an integration as a user with the *Service*Developer role. Assume you have not saved your changes when your browser crashes.
2. If you log back in as that same user with the *Service*Developer role, perform the following tasks to recover your unsaved changes.
3. Decide where to start:
   - Work in a project.
     a. In the navigation pane, click **Projects**.
     b. Select the project name.
     c. Click **Integrations** ⇄.
     d. View the word **Locked** next to the impacted integration.
   - Work outside a project.
     a. In the navigation pane, click **Design**, then **Integrations**.

Copyright © 2022, 2026, Oracle and/or its affiliates.

b. View the word **Locked** next to the impacted integration.

4. Select to resume editing and recover your unsaved integration changes in any of the following ways:

   a. Click the integration name in the **Name** field or select **Edit** from the **Actions** menu.
   Your integration design is displayed. A warning message is displayed indicating that unsaved changes have been recovered and are displayed in the canvas, but not saved yet.

   b. Click **Save**.
   A message is displayed indicating that the integration was successfully saved.

   Or:

   a. Select **Unlock** from the **Actions** menu.
   A warning message is displayed indicating that you can edit and recover your changes have been recovered and are displayed in the canvas, but not saved yet.

   b. If you want to save your changes, select **Edit & recover**, then click **Save** to save your changes and continue with integration design.

   c. If you do not want to recover your changes, select **Unlock & discard**.

Override Design-Time Properties in an Integration

You can configure integration properties outside of the regular integration design. At runtime, these integration properties take effect and override the values you configured at design time. For example, you can specify an email address to use in the **To** field of a notification action during runtime that overrides the email address specified at design-time. This feature enables you to configure the value without the need to edit the integration design or pass the value at runtime.

You can configure all types of properties, including file path, database properties, server properties, credentials, and any constant property that changes at regular intervals.

When you can edit the integration properties is based on the type of integration pattern you are using.

Integration Pattern	Editing Behavior
Schedule	Runtime configuration is supported. You can update the integration property values when the integration is active. You can view, but not edit, the properties when the integration is deactivated. When the integration is invoked: • If the integration property value is not updated, the default value is used. • If the integration property value is updated with a new value, the new value becomes the current value and is used.

Copyright © 2022, 2026, Oracle and/or its affiliates.

Integration Pattern	Editing Behavior
Application	Runtime configuration is not supported. You can update the integration property values when the integration is *not* active. The configured property values takes effect after integration activation. • If the integration property value is not updated, the default value is used. • If the integration property value is updated with a new value, the new value becomes the current value and is used.

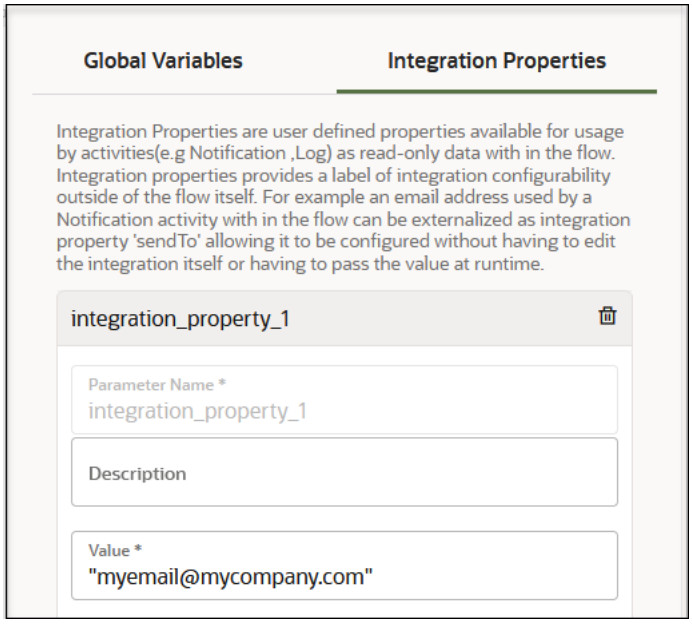
Note the following behavior regarding integration properties and major versions (for example, `01.00.0000`) and minor versions (for example, `01.10.0000`) of an integration.

- The current integration property values are shared by all minor version integrations with the same major version and identifier.
- Updating the current value for an integration property is reflected in all other minor version integrations if they have an integration property with the same name.
- Deleting an integration property from an integration does not remove its current value if there are other minor version integrations with the same major version and identifier.

See Create an Integration for information about versions and identifiers.

1. Decide where to start:
   - Work in a project.
     a. In the navigation pane, click **Projects**.
     b. Select the project name.
     c. Click **Integrations**.
     d. In the **Integrations** section, click the specific integration to update.
   - Work outside a project.
     a. In the navigation pane, click **Design**, then **Integrations**.
     b. Click the specific integration to update.
2. Click **Global Variables** (x).
3. Click the **Integration Properties** tab.
4. Click **+** to add a name, description, and value for the property. For example, enter an email address as the value. You can add a maximum of 10 integration properties.

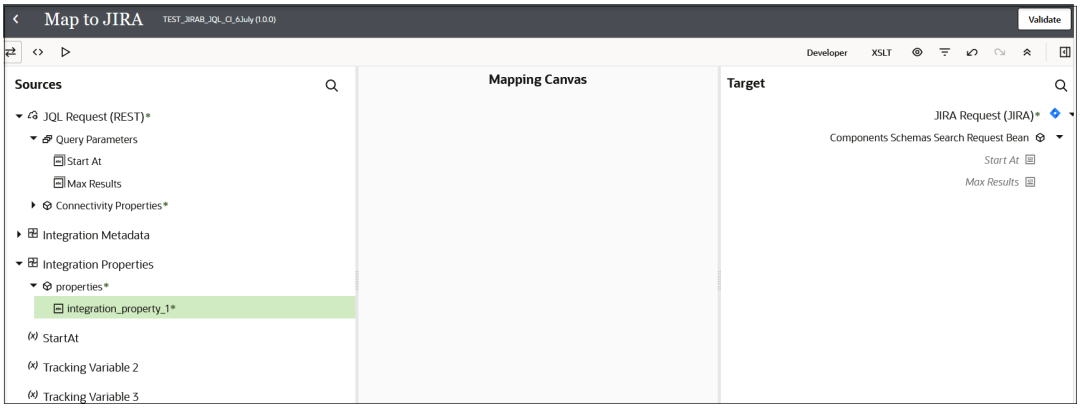
Copyright © 2022, 2026, Oracle and/or its affiliates.



5. When complete, click **Save** to save your changes and return to the integration canvas.

> **Note**
>
> The integration property value can't be updated within the integration. For example, you cannot update the integration property in an assign action.

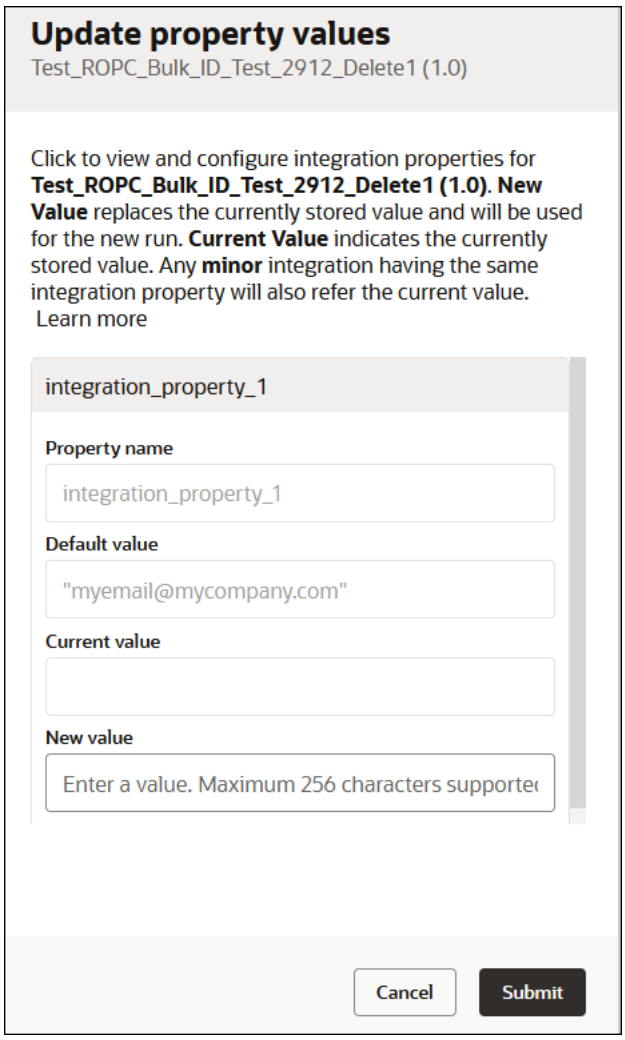
6. In the mapper, map the integration property (named **integration_property_1** for this example) visible under **Integration Properties** in the **Sources** element tree to an appropriate element in the **Target** tree.



7. Go to the Integrations page.
8. Hover over the integration whose properties you want to update.
9. Click **Actions** ⋮, then select **Update property values**.

   The Update property values panel open on the right side of the page.
10. Click the integration property to update, then enter a value in the **New Value** field and click **Submit**.

Copyright © 2022, 2026, Oracle and/or its affiliates.



A confirmation message indicates that the property was updated.

The new value is used for a new run of the integration. For this example, the email address you specified in the **New Value** field is used as the **To** field email address of the notification action in the integration.

If you select **Update property values** again to reopen the Update property values panel, the value in the **New Value** field now appears in the **Current Value** field.

You can also view previously configured integration properties.

Use Metadata in Integrations

You can access the integration name, identifier, and version instead of hard coding those values in integrations. You can also access runtime data and environment data inside the integration.

The following metadata is exposed in integrations. You can use this metadata in the mapper and actions that include the Expression Builder such as assign, log, notification, and others.

- integration
  - name

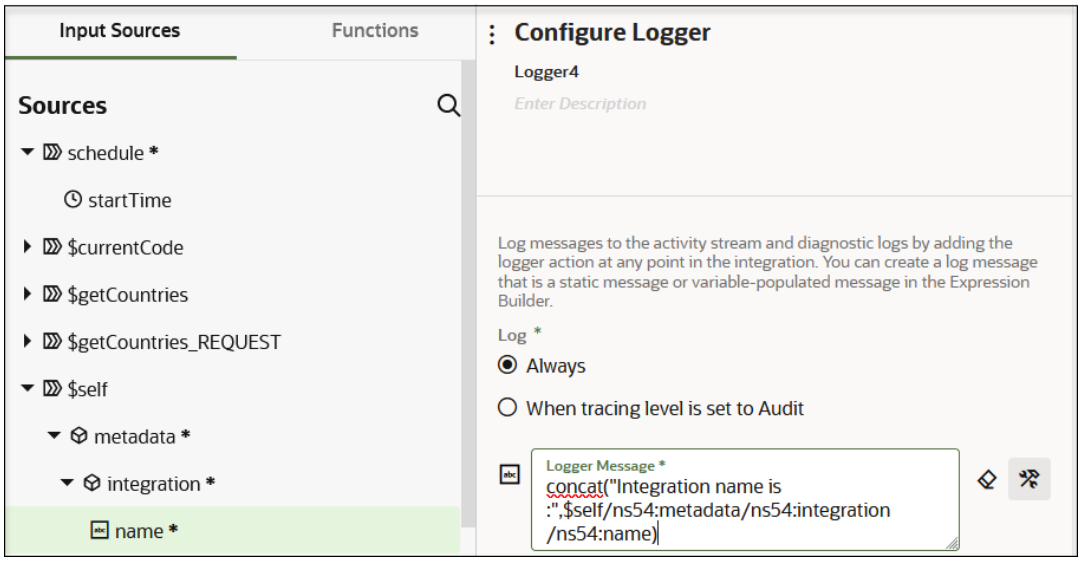
Copyright © 2022, 2026, Oracle and/or its affiliates.

  - identifier
  - version
- runtime
  - instanceId
  - invokedBy
- environment
  - serviceInstanceName
  - baseURL

The following use case provides a high-level overview of using metadata in an integration.

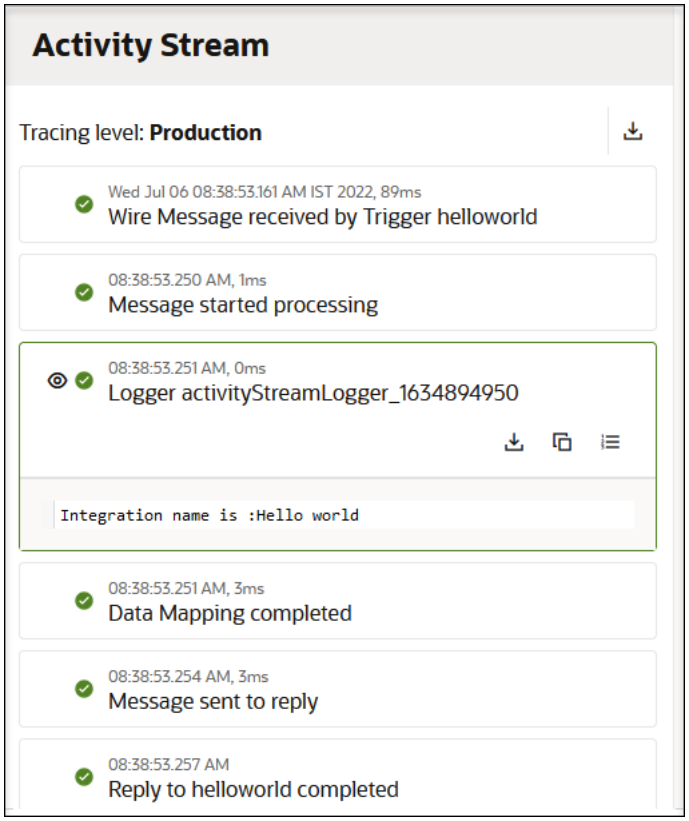
1. Create and design an application integration.
2. Add an action that includes an Expression Builder.

   For this example, a log action is added.
3. In the source tree, find the integration metadata.
4. Add the required metadata to the **Logger Message** field. For this example, a **concat** function is used to include the integration **name** metadata.



5. Complete design of the integration.
6. Activate and run the integration.
7. Go to the Instances page.
8. Click the business identifier of the integration.
9. Scroll down and note that the log message in the activity stream shows the integration name (for this example, **Hello World**).

Copyright © 2022, 2026, Oracle and/or its affiliates.



Cancel a Running Integration Instance

You can cancel an integration instance that is not running 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 **Observability**, then **Instances**.
2. Hover over the integration to cancel, then select **Abort** .

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

View the Contents of an Incomplete Integration

You can open an integration with incomplete connection details on the Integrations page and view its contents. These integrations can only be viewed, and not edited.

1. In the navigation pane, click **Design**, then **Integrations**.
2. Open an integration with incomplete connection details (for example, an integration imported from another instance or from the Oracle Cloud Marketplace).

Copyright © 2022, 2026, Oracle and/or its affiliates.

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, and other components in the integration for viewing.
5. Go to the Connections page for the adapter connection requiring configuration.
6. Update the necessary credential and security details.
7. 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. In the navigation pane, click **Design**, then **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.
3. Edit any completed connections, actions, and maps, as necessary.
4. To complete the integration, go to the Connections page for the incomplete connection and complete the configuration.
5. Return to the integration and edit the newly completed connection in the Adapter Endpoint Configuration Wizard, if necessary.
6. Save and exit the integration, then activate it.

Change the Time Zone

You can change the time zone that is displayed in Oracle Integration.

1. In the upper corner, click the ***user name*** icon, then select **Preferences**.
2. From the **Time Zone** list, select the time zone you want to use.
3. Click **Save**.
4. Go to the Integrations page and note that the time zone is changed in the message after 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 Instances page).

Copyright © 2022, 2026, Oracle and/or its affiliates.

13

Monitor Integrations During Runtime

You can monitor and manage your integrations in the runtime environment. You can also perform runtime tasks, such as tracking the status of your integration instances, viewing and diagnosing errors, and resubmitting and discarding integration failed instances.

Topics:

- Get Started with Observability
- View the Dashboard
- View Notification Alert Announcements and the Status of Asynchronous Operations
- Monitor the Message Processing Status of Integrations
- Monitor Agents
- Learn About the Payloads in the Activity Stream
- Measure an Integration's Performance
- Find Integrations That Experienced Errors
- Find Integrations That Timed Out or Were Aborted
- Manage Errors
- Track Integration Instances
- Monitor Integrations That Subscribe to Events
- View Minimal Details About Debug Tracing Level Instances for 32 Days
- View the Calendar of Schedule Integration Runs
- Diagnose and Manage Event-Based Oracle Fusion Applications Integrations
- Send System Status Reports with Notification Emails

> **Note**
>
> Oracle Integration includes several preconfigured logging analytics dashboards in the Oracle Cloud Console to help you monitor resources, diagnostics, and key metrics for your tenancy. See Monitor Oracle Integration 3 Instances in *Provisioning and Administering Oracle Integration 3*.

Get Started with Observability

Use the observability features to monitor your automation solutions. Browse the following popular topics to learn more.

Copyright © 2022, 2026, Oracle and/or its affiliates.

Explore Oracle Integration

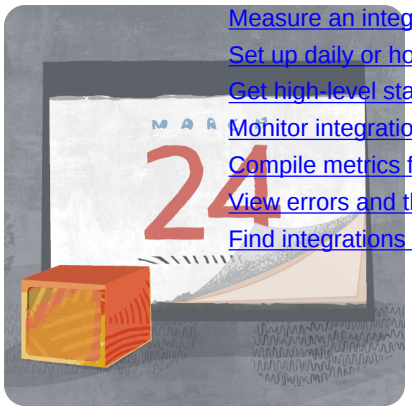
Image	Links
	Quick introduction to Oracle Integration 10 things to know about Oracle Integration Learn about observability and the Observability pages Build your own observability pages Develop an observability strategy

Explore Observability

Image	Links
	Recommended initial monitoring tasks Recommended daily monitoring tasks Recommended occasional monitoring tasks

Copyright © 2022, 2026, Oracle and/or its affiliates.

Key Observability Tasks

Image	Links
	Measure an integration's performance Set up daily or hourly notification emails Get high-level status information on the dashboard Monitor integration instances Compile metrics for your integration components View errors and the audit trail Find integrations that experienced errors

View the Dashboard

You can view information about integration performance on the Integration dashboards page, including the error rate; the total number of instances, instances that succeeded, and instances in error; the number of active integrations; the status of connectivity agents; and design-time metrics.

You can view information about how your integrations are performing. The main page of the integration shows a snapshot of successful and failed integrations.
The Integration dashboards page reflects historical data. The number of all types of instance states (successful, failed, and so on) are collected hourly and displayed on the Integration dashboards 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)

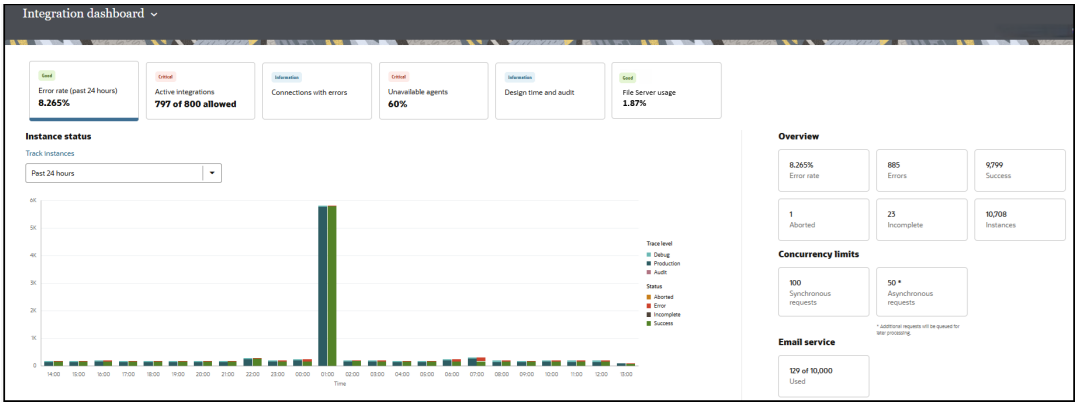
If instance state information is collected between time 1 and 2, the Integration dashboards page shows some failed instances. The history graphs on the Integration dashboards page also show the hour/day during which that was the instance state. If the recovery of failed instances occurs before any new data is collected, the Integration dashboards page shows those instances as successful.

1. In the navigation pane, click **Observability**, then **Dashboards**.

   The Integration dashboards page is displayed by default. From the drop-down list, you can also select **B2B** to view the overall status of B2B transactions. See Monitor the Overall Health of B2B Transactions in *Using B2B for Oracle Integration 3*.

2. Click a section for dashboard details.

Copyright © 2022, 2026, Oracle and/or its affiliates.



Section	Description	Actions
Error rate (at least 24 hours) (This section is displayed by default.)	The instance status chart shows two bars per hour. • The left bar shows the number of instances classified by their tracing level. This is important because the tracing level determines the life cycle of instances (for example, debug level tracing instances are purged in 24 hours) and the life cycle of the instance payload (for example, payloads captured for instances set to audit level tracing are purged in seven days). • The right bar shows the number of instances classified by their states. The **Incomplete** state contains all instances that have not yet completed processing. These instances may be in the **Queued**, **In Progress**, **Resubmitted**, or **Blocked** states. Data outside the gray box (usually older historical data) does not retain or contain any information about instances set to debug level tracing. The following graphical and numerical information is also displayed: • Error rate percentage; the number of failed, successful, terminated, and incomplete integration instances; and the total number of integration instances. • Integration instances generated and processed at different tracing levels.	a. Select the time period for viewing integration instance status from the **Instance Status** list (24 hours (the default), the past week, the past month, or since the Oracle Integration instance was provisioned). The bar graphs and **Overview** section are refreshed to reflect information based on your selection. b. Click **Track Instances** to access the Instances page. See Track Integration Instances.
Active integrations	Provides statistics on the number of scheduled, application, and event integrations, including the number in projects, packages, and standalone (globally available). The number of active integrations per Oracle Integration instance cannot exceed 800. Active integrations are defined as currently active integrations and integrations whose activations are in progress. If this number is: • above 90% of the limit, a warning message is displayed. • equal to or above the limit, a critical message is displayed.	a. Click **Monitor integrations** to access the Monitor integrations page. See Monitor the Message Processing Status of Integrations.

Copyright © 2022, 2026, Oracle and/or its affiliates.

Section	Description	Actions
Connections with errors	Displays a graph of connections with the most errors. The connection names appear to the right of the graph. The **Overview** section on the right shows the type of adapters with the most errors (for example, Oracle ERP Cloud Adapter, Salesforce Adapter, and others).	a. Click a section in the graph to access the Errors page, which lists the integration instances that use this connection. The connection name appears in the filtered search at the top of the Errors page. b. Click **View more** under **Top connections with errors** to go to the Errors page. The connection names and the total instance error counts for each are displayed with the **Error by connection** filter automatically selected. Error count totals are displayed in descending order.
Unavailable agents	Displays the percentage of unavailable connectivity agents versus the number of currently running connectivity agents. For example, if you have five connectivity agents, and all are running and available and no connectivity agents are unavailable, then 0% is displayed for unavailable agents. The graph and **Overview** section provide statistics about the number of running and unavailable connectivity agents.	a. Click **Monitor agents** to access the Monitor Agents page. See Monitor Agents.
Design time and audit	Displays a graphical view of the state of integration components. • Active, configured, and draft integrations • Active, configured, draft, and in-use connections • In-use and configured lookups • User-developed and recipe packages • Connectivity agent groups • Preinstalled adapters • Configured and draft libraries By default, the **Integrations** section shows the number of integration patterns and adapters in use.	a. Hover over each chart to see the number of components in each state. b. Click the **Integrations** graph to show the integration patterns (application or schedule) and adapters in use. c. Click the **Connections** graph to show the number of each type of adapter in use. d. Click the **Agents** graph to show the agent groups in use. e. Click **Audit** to access the Design Time Audit page. See Check the Design Time Audit History for an Integration or Other Component.

Copyright © 2022, 2026, Oracle and/or its affiliates.

Section	Description	Actions
Overview	Displays the status of messages for the rolling 24-hour window, including the percentage of failed messages and the total number of failed messages, successful messages, terminated messages, and incomplete messages. The number of instances that have run in the rolling 24-hour window is also displayed.	View the message details.
Concurrency limit	Displays the enforced concurrency limits for the following requests. The values displayed are based on the number of message packs you are consuming. The actual number of requests consumed is not displayed. • **100 Synchronous requests** (if consuming one message pack.) • **50 Asynchronous requests** (if consuming one message pack) Create a custom metrics chart to monitor your requests and set up an alarm to notify you when you approach the threshold. See Monitor Concurrent Synchronous Requests in *Provisioning and Administering Oracle Integration 3*. See Service Limits in *Provisioning and Administering Oracle Integration 3*.	View the concurrency limits.
Email service	If you are using the default email method provided by Oracle Integration, your email usage is displayed based on the 10,000 email message limit for a rolling 24-hour window. If you are using your own customer email tenancy, no usage metrics are displayed. See Monitor the Number of Emails Sent from Oracle Integration.	View the usage metrics for the default email method.
File Server usage	Displays the File Server usage details. **Storage** • Good: 0-80% • Almost full: >80-95% • Full >95% **Overview**: Displays the Used and Total storage values.	a. Click **Runtime audit** to access the Runtime audit page and view the real-time audit history of runtime file and folder operations in File Server. See Check the Runtime Audit History for File Server.

Monitor the Number of Emails Sent from Oracle Integration

You can send up to 10,000 emails in a rolling 24-hour window from Oracle Integration. However, if you use your own customer tenancy, you can increase your limit to the number allowed by the Oracle Cloud Infrastructure Email Delivery Service. When your limit is exceeded, delivery occurs successfully one last time. For the next delivery, emails stop being sent and a `Daily outbound email limit crossed` error message is written to the activity stream for the impacted integration instance.

You monitor the current email count from the **Email service** section of the Integrations dashboard page. The count includes successful and failed messages. See View the Dashboard.

- How the Rolling 24 Hour Window is Calculated

Copyright © 2022, 2026, Oracle and/or its affiliates.

- Increase the Limit on the Number of Emails to Send

How the Rolling 24 Hour Window is Calculated

- The email quota is built on a 24-hour rolling window. At any particular point in time, the last 24 hours of outbound emails are counted.
- When you exceed your quota in only one hour (same hour), you are blocked from sending emails for the next 24 hours.

To handle this type of scenario in your integration, you can also include the notification action in a scope action so that you can emit log messages in a fault handler and then resume instance processing, if required.

Increase the Limit on the Number of Emails to Send

The default method manages email delivery in Oracle Cloud Infrastructure. This tenancy limits the number of emails you can send in a rolling 24-hour window to 10,000 and the size of attachments to 2 MB.

If you instead use your own customer tenancy for sending emails, you receive the following benefits:

- The email limit is increased to the number allowed by the Oracle Cloud Infrastructure Email Delivery Service. The customer tenancy uses SMTP credentials to send email through the Oracle Cloud Infrastructure Email Delivery Service.
- You manage your approved senders and suppression list from your own customer tenancy instead of the Notifications page in Oracle Integration.

See Configure Notification Emails.

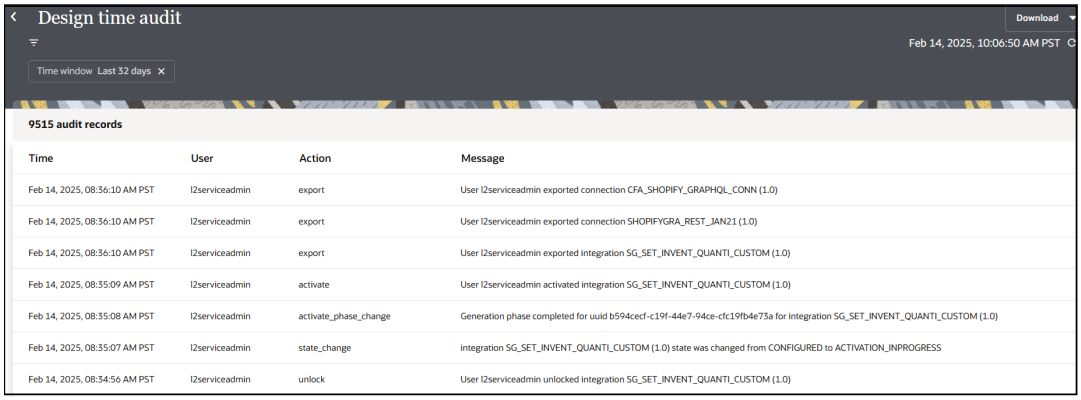
Check the Design Time Audit History for an Integration or Other Component

The Design time audit page shows the audit history of integrations and other components, including administrative updates made using the File Server user interface.

Learn who updated a component, what changes they made, and when they made the update. This information is helpful for various scenarios, including troubleshooting unplanned changes. Audit history is retained for the retention period of the instance.

1. In the navigation pane, click **Observability**, then **Dashboards**.
2. Click the **Design time** section.
3. Click **Audit**.

   The Design Time Audit page shows details about the audit for all invoked integrations.



Copyright © 2022, 2026, Oracle and/or its affiliates.

4. Click **Filter** to filter data by time window, project name, integration name, user name, or actions such as activate and deactivate. You can click the **Date range** option to specify a date range of up to 365 days. The maximum date range is six months. If you are using projects, the project identifier is also included in the audit logs. Use this identifier to audit instance and error details.
5. Click **Download**, and select the following:
   - **Download all**: Downloads a maximum of 10,000 audit records.
   - **Download filtered**: Downloads records for the currently-selected filters.

   For example:

```
[2022-06-18 00:57:56.372] [userId: l2gen3serviceadmin]
[l2gen3serviceadmin,UNLOCK,ICS_ProjectV2,
TEST_DEMO_FAULT_AGGR,TEST_DEMO_FAULT_AGGR,02.00.0000] User
l2gen3serviceadmin unlocked Integration
TEST_DEMO_FAULT_AGGR 02.00.0000
[2022-06-18 00:38:37.468] [userId: l2gen3serviceadmin]
[l2gen3serviceadmin,LOCK,ICS_ProjectV2,TEST_DEMO_FAULT_AGGR,
TEST_DEMO_FAULT_AGGR,02.00.0000] User l2gen3serviceadmin locked
Integration TEST_DEMO_FAULT_AGGR 02.00.0000
[2022-06-18 00:37:08.655] [userId: l2gen3serviceadmin]
[l2gen3serviceadmin,UNLOCK,ICS_ProjectV2,PFE_SRS_FAUL_AGGR_G3_DD_MS,
PFE_SRS_FAUL_AGGR_G3_DD_MS,02.00.0000] User l2gen3serviceadmin unlocked
Integration PFE_SRS_FAUL_AGGR_G3_DD_MS 02.00.0000
[2022-06-18 00:34:10.968] [userId: l2gen3serviceadmin]
[l2gen3serviceadmin,LOCK,ICS_ProjectV2,PFE_SRS_FAUL_AGGR_G3_DD_MS,
PFE_SRS_FAUL_AGGR_G3_DD_MS,02.00.0000] User l2gen3serviceadmin locked
Integration PFE_SRS_FAUL_AGGR_G3_DD_MS 02.00.0000
[2022-06-17 23:01:14.955] [userId: l2gen3serviceadmin]
[l2gen3serviceadmin,UNLOCK,ICS_ProjectV2,PFE_SRS_FAUL_AGGR_G3_DD_MS,
PFE_SRS_FAUL_AGGR_G3_DD_MS,02.00.0000] User l2gen3serviceadmin unlocked
Integration PFE_SRS_FAUL_AGGR_G3_DD_MS 02.00.0000
. . .
. . .
. . .
```

The following table describes some of the actions that can appear.

Action	Description	User Action
activate	Indicates that the user activated an integration.	Activate the integration.
activate_phase_change	Indicates that the user activated an integration. This action captures the intermediate state in the activation.	Activate the integration.
audit_migration	Indicates that the user cloned an instance.	Clone the instance from the Settings page, under Import and Export.
bulk_reactivation_completed	Indicates that the user performed a Save and Reactivate operation from the Connections page.	Save and reactivate the connection.

Copyright © 2022, 2026, Oracle and/or its affiliates.

Action	Description	User Action
bulk_reactivation_started	Indicates that the user performed a Save and Reactivate operation from the Connections page.	Save and reactivate the connection.
clone	Indicates that the user cloned a resource (for example, integration, connection, or lookup).	Clone the resource.
configure_integration	Indicates that the user configured an integration from the Configuration Editor page.	Configure the integration.
create	Indicates that the user created a resource.	Create the resource.
deactivate	Indicates that the user deactivated an integration.	Deactivate the integration.
delete	Indicates that the user deleted a resource.	Delete the resource.
delete certificate	Indicates that the user deleted the certificate.	Delete the certificate.
disable	Indicates that the user disabled a File Server user.	Disable a user.
import	Indicates that the user imported an integration or package.	Import the integration or package.
instance_resubmission	Indicates that the user submitted a request to resubmit an errored instance for an integration.	Resubmit.
instances_resubmission	Indicates that the user submitted a request to resubmit an errored instance for an integration.	Perform a bulk resubmit.
instances_abort	Indicates that the user submitted a request to abort an errored instance for an integration.	Abort.
instance_abort	Indicates that the user submitted a request to abort an errored instance for an integration.	Perform a bulk abort.
instance_abort	Indicates that the user submitted a request to abort a running instance for an integration.	Abort.
lock	Indicates that the user locked a resource.	Edit the resource.
paused	Indicates that the user paused a schedule for an integration.	Pause the schedule.
reactivation	Not being used.	- -
reactivation_fail	Not being used.	- -
register	Indicates that the user created an application type (that is, an adapter).	Create the adapter.
rename	Indicates that the user renamed a resource.	Rename a resource.

Copyright © 2022, 2026, Oracle and/or its affiliates.

Action	Description	User Action
resumed	Indicates that the user resumed a schedule for an integration.	Resume a schedule.
start	Indicates that the user started File Server.	Start File Server.
started	Indicates that the user started a schedule for an integration.	Start a schedule.
state_change	Indicates that the user activated an integration. This action captures the intermediate state in the activation.	Activate the integration.
stop	Indicates that the user stopped File Server.	Stop File Server.
stop_schedule_deactivation	Not being used.	- -
stopped	Indicates that the user stopped a schedule for an integration.	Stop the schedule.
unlock	Indicates that the user unlocked a resource.	Edit and save a resource.
update	Indicates that the user updated a resource.	Update the resource.
update_schedule_parameter	Indicates that the user updated a schedule parameter's value for an integration.	Update the schedule parameter.
upload_certificate	Indicates that the user uploaded the certificate.	Upload / create the certificate.

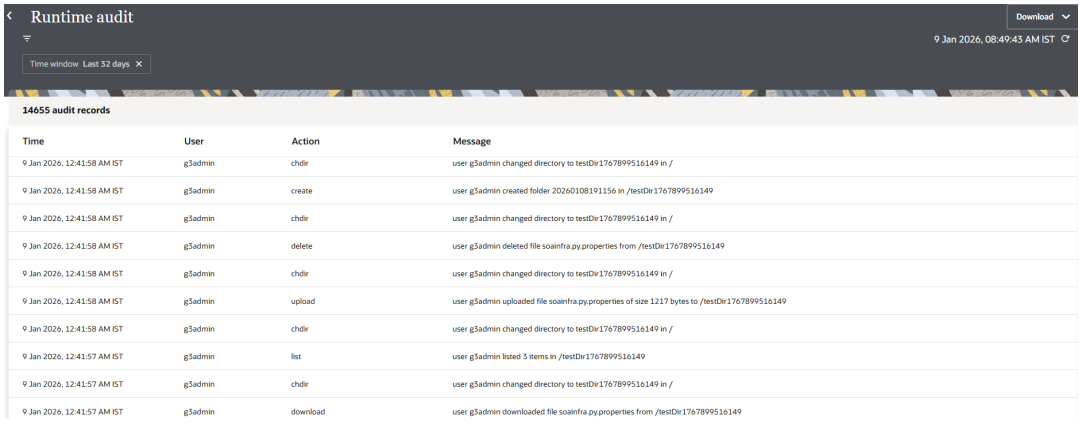
Check the Runtime Audit History for File Server

The Runtime audit page shows the real-time audit history of runtime file and folder operations in File Server.

Learn who updated a file or folder, what changes they made, and when they made the update. This information is helpful for various scenarios, including troubleshooting unplanned changes. Audit history is retained for the retention period of the instance.

1. In the navigation pane, click **Observability**, then **Dashboards**.
2. Click the **File Server usage** section.
3. Click **Runtime audit**.

   The Runtime audit page shows details about the audit for all file and folder operations.



Copyright © 2022, 2026, Oracle and/or its affiliates.

4. Click **Filter** to filter data by time window, user name, or actions such as connect and disconnect. You can click the **Date range** option to specify a date range.
5. Click **Download**, and select the following:
   - **Download all**: Downloads a maximum of 10,000 audit records.
   - **Download filtered**: Downloads records for the currently-selected filters.

   For example:

```
[2026-01-08 19:11:57.987 Z] [userID: g3admin] [g3admin,LIST,FOLDERS,,/
testDir1767899516149,,] user g3admin listed 3 items in /
testDir1767899516149
[2026-01-08 19:11:57.940 Z] [userID: g3admin] [g3admin,CHDIR,FOLDERS,,,,]
user g3admin changed directory to testDir1767899516149 in /
[2026-01-08 19:11:57.917 Z] [userID: g3admin] [g3admin,DOWNLOAD,FILES,,,,]
user g3admin downloaded file soainfra.py.properties from /
testDir1767899516149
[2026-01-08 19:11:57.657 Z] [userID: g3admin] [g3admin,CHDIR,FOLDERS,,,,]
user g3admin changed directory to testDir1767899516149 in /
[2026-01-08 19:11:57.497 Z] [userID: g3admin] [g3admin,UPLOAD,FILES,,,,]
user g3admin uploaded file soainfra.py.properties of size 1217 bytes to /
testDir1767899516149
[2026-01-08 19:11:57.057 Z] [userID: g3admin] [g3admin,CHDIR,FOLDERS,,,,]
user g3admin changed directory to testDir1767899516149 in /
[2026-01-08 19:11:57.034 Z] [userID: g3admin] [g3admin,CHDIR,FOLDERS,,,,]
user g3admin changed directory to testDir1767899516149 in /
[2026-01-08 19:11:56.958 Z] [userID: g3admin]
[g3admin,CONNECT,SESSION,,,,] user g3admin established a new connection
with fileserver instance <instance_id>
[2026-01-08 19:11:56.096 Z] [userID: g3admin]
[g3admin,CONNECT,SESSION,,,,] user g3admin established a new connection
with fileserver instance <instance_id>
[2026-01-08 19:11:55.647 Z] [userID: g3admin]
[g3admin,DISCONNECT,SESSION,,,,] user g3admin closed connection with
fileserver instance <instance_id>
[2026-01-08 19:11:55.636 Z] [userID: g3admin] [g3admin,DELETE,FOLDERS,,,,]
user g3admin deleted folder 20260108191155 from /testDir1767899512230
[2026-01-08 19:11:55.582 Z] [userID: g3admin] [g3admin,CHDIR,FOLDERS,,,,]
user g3admin changed directory to testDir1767899512230 in /
[2026-01-08 19:11:55.507 Z] [userID: g3admin] [g3admin,RENAME,FOLDERS,,/
testDir1767899512230/20260108191155,,] user g3admin renamed folder from /
testDir1767899512230/20260108191152 to /testDir1767899512230/20260108191155
[2026-01-08 19:11:55.396 Z] [userID: g3admin] [g3admin,CHDIR,FOLDERS,,,,]
user g3admin changed directory to testDir1767899512230 in /
[2026-01-08 19:11:55.175 Z] [userID: g3admin] [g3admin,CREATE,FOLDERS,,,,]
user g3admin created folder 20260108191152 in /testDir1767899512230
[2026-01-08 19:11:55.077 Z] [userID: g3admin] [g3admin,CHDIR,FOLDERS,,,,]
user g3admin changed directory to testDir1767899512230 in /
[2026-01-08 19:11:54.988 Z] [userID: g3admin] [g3admin,DELETE,FILES,,,,]
user g3admin deleted file soainfra.py.properties from /testDir1767899512230
[2026-01-08 19:11:54.711 Z] [userID: g3admin] [g3admin,CHDIR,FOLDERS,,,,]
user g3admin changed directory to testDir1767899512230 in /
[2026-01-08 19:11:54.454 Z] [userID: g3admin] [g3admin,UPLOAD,FILES,,,,]
user g3admin uploaded file soainfra.py.properties of size 1217 bytes to /
testDir1767899512230
```


Copyright © 2022, 2026, Oracle and/or its affiliates.

The following table describes some of the actions that can appear.

Activity type	Action	Description	User Action
File	create	Indicates that the user created a file.	Create the file.
	delete	Indicates that the user deleted a file.	Delete the file.
	move	Indicates that the user moved a file.	Move the file.
	read	Indicates that the user read from a file.	Read from the file.
Folder	create	Indicates that the user created a folder.	Create the folder.
	delete	Indicates that the user deleted a folder.	Delete the folder.
	update	Indicates that the user updated a folder.	Update the folder.
	list	Indicates that the user listed the number of items in the folder.	List the number of items in the folder.
Connection	open	Indicates that the user established a new connection with a File Server instance.	Open a new connection.
	closed	Indicates that the user closed the connection with a File Server instance.	Close the connection.

See the Current Status of All Integration Components

Get at-a-glance information about the number and status of your projects, integrations, connections, and more. For example, learn how many REST connections your organization has created.

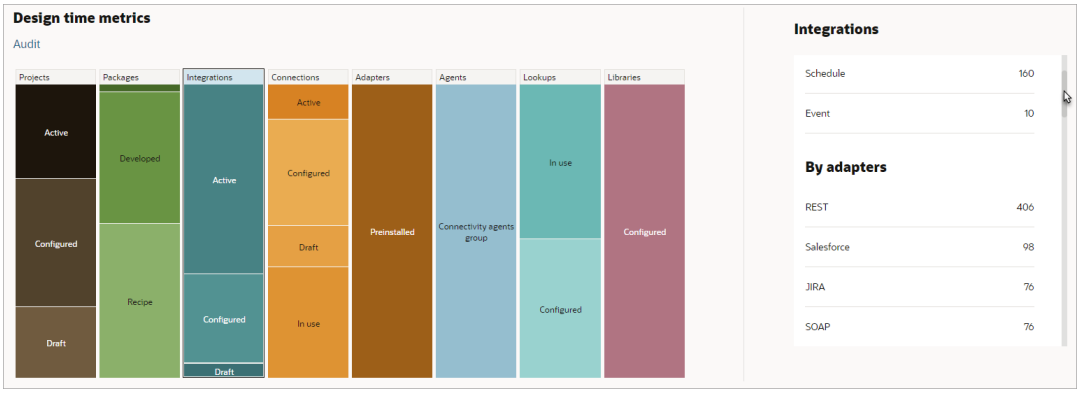
This information can be helpful for showing your team's contributions, such as when you're compiling year-end metrics for the work that you've delivered.

1. In the navigation pane, click **Observability**, then **Dashboards**.
2. Along the top, select the **Design time and audit** box.

   The Design time metrics heading appears. A chart visualizes the components in your Oracle Integration instance.
3. Obtain metrics using one or both of the following options:
   - Below the **Design time metrics** heading, point to an entry in the chart, and view a metric in the tooltip that appears.
   - On the right, below the **Integrations** heading, obtain additional metrics.

Copyright © 2022, 2026, Oracle and/or its affiliates.

> **Tip**
>
> The table of metrics on the right often contains more information than can fit on the page. Point within the table of metrics to see a scroll bar, and scroll down to see additional metrics.

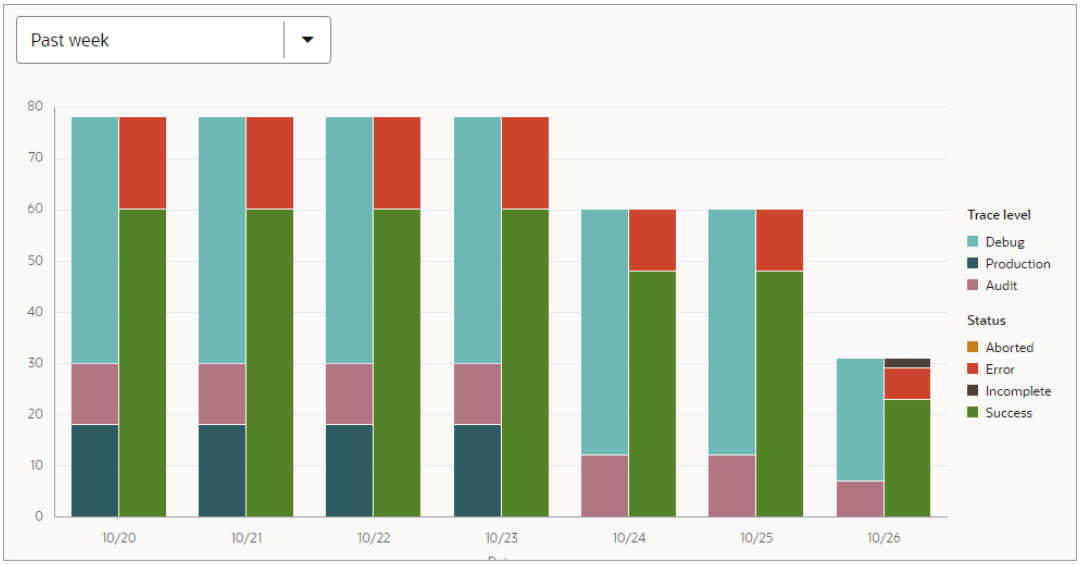


Check an Integration's Tracing Level

To ensure that your integrations are capturing the minimum required tracing information, consider routinely checking their tracing levels. For instance, Oracle recommends not using the Debug tracing level in production.

1. In the navigation pane, click **Observability**, then **Dashboards**.
2. Below the **Instance status** heading, select the appropriate time period to review from the drop-down.

   For instance, you might want to check the tracing levels for the last 24 hours or week.
3. Familiarize yourself with the chart that appears below **Instance status**.



- Each time period contains two vertical bars.

Copyright © 2022, 2026, Oracle and/or its affiliates.

- The **left bar** visualizes the **tracing levels** for the integrations that ran during the time period.
- The **right bar** visualizes the **statuses** for the integrations that ran during the time period.

4. Check the tracing levels for your integration instances for the time period.

View Notification Alert Announcements and the Status of Asynchronous Operations

You can view notification alert announcements and the status of asynchronous operations in the banner.

Topics:

- View Notification Alert Announcements
- View the Progress of Asynchronous, Design-Time Operations

View Notification Alert Announcements

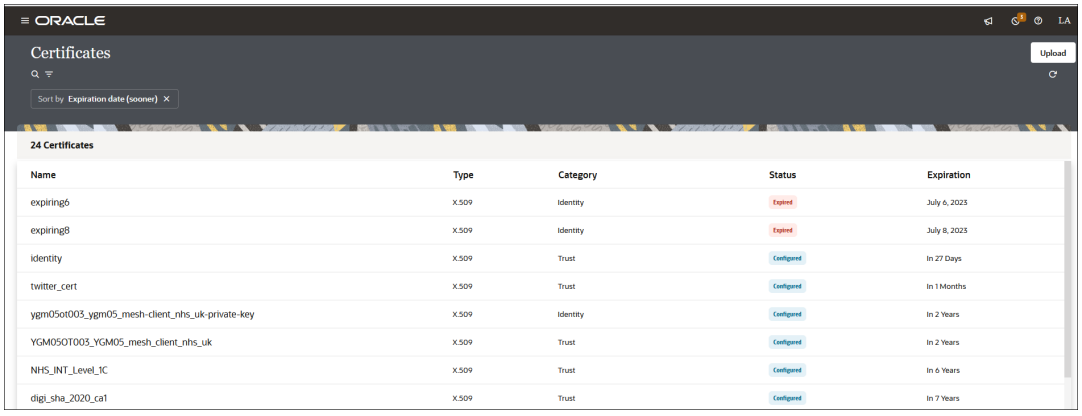
A notification alert announcement icon is displayed in the banner that provides links to the status of connectivity agents, active integrations, and certificate expiration dates.

When an alert is raised, a number is displayed in the alert icon in the banner.

1. Click **Announcements** in the banner, then select an option to view status. For this example, the following alerts have been raised.



If you select **Certificates**, the Certificate page opens with a filter set to show the soonest-to-expire certificates. In this case, two have expired.



2. If you select **Agents** or **Active integrations**, the following pages are displayed:

Copyright © 2022, 2026, Oracle and/or its affiliates.

- **Agents**: Takes you to the **Unavailable agents** card of the Integration dashboards page. The percentage of agents running and not running is displayed.
- **Active integrations**: Takes you to the **Active integrations** card of the Integration dashboards page. The number of active integrations and percentage of active scheduled, event, and application integrations are displayed.

3. Take appropriate action based on the information provided.

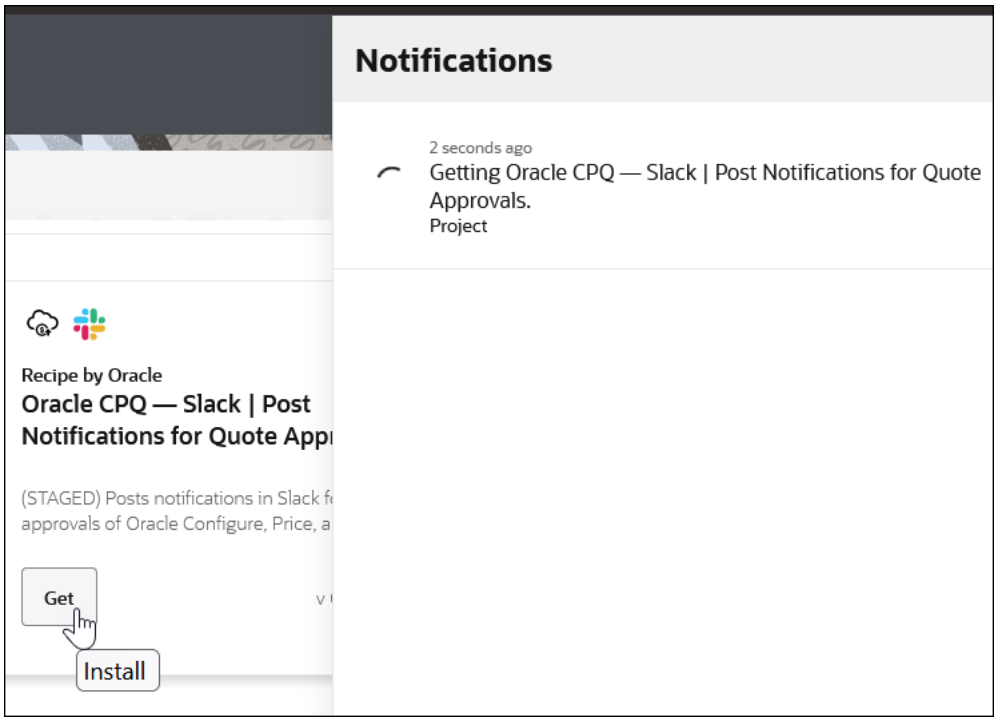
View the Progress of Asynchronous, Design-Time Operations

You can view the current progress of some asynchronous, design-time operations from the notifications center icon in the Oracle Integration banner. This view eliminates the need to constantly refresh the page to check progress.

You can view the progress for the following asynchronous operations from the notifications center:

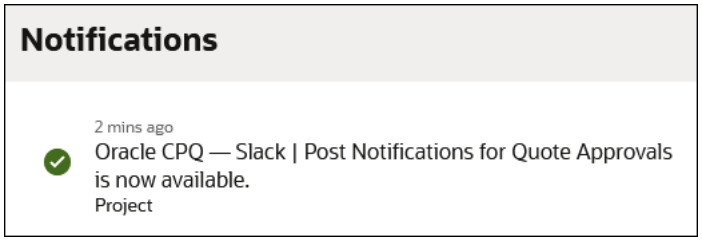
- Install accelerator projects
- Import projects
- Deploy integrations to the Oracle Cloud Infrastructure API Gateway

1. Perform one of the supported asynchronous operations.
2. Click **Notifications** in the Oracle Integration banner.
3. Follow the real-time progress. For this example, a recipe is being installed.



4. Reopen the notification center. The progress message has changed to indicate that installation has completed. The message is retained for six hours.

Copyright © 2022, 2026, Oracle and/or its affiliates.



5. Click the recipe name in the notification center to access the project.

Monitor the Message Processing Status of Integrations

The Monitor integrations page enables you to view the message processing status of your running integrations, including how many messages have been received and processed, how many successful messages and errors have occurred, and how many messages have been aborted.

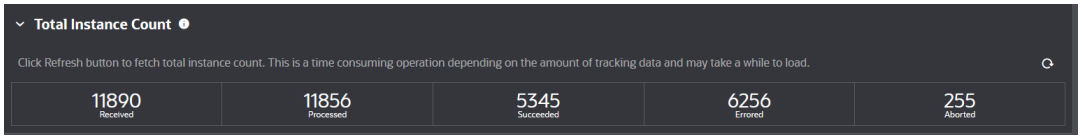
Only activated integrations are listed on this page by default. Using the appropriate filter, you can also view data for inactive integrations.

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

   A list of active integrations appears by default, along with processing information about the number of messages received, the number of messages processed, the number of successful messages, the number of failed messages, and the number of aborted messages. For schedule integrations, navigation links are provided that take you to the Schedule and Future Runs page.
   A **Total instance count** summary is displayed at the top of the page. This section is collapsed by default. Data is not immediately fetched.

2. Expand the **Total instance count** section and click the **Refresh** button to fetch and view data.

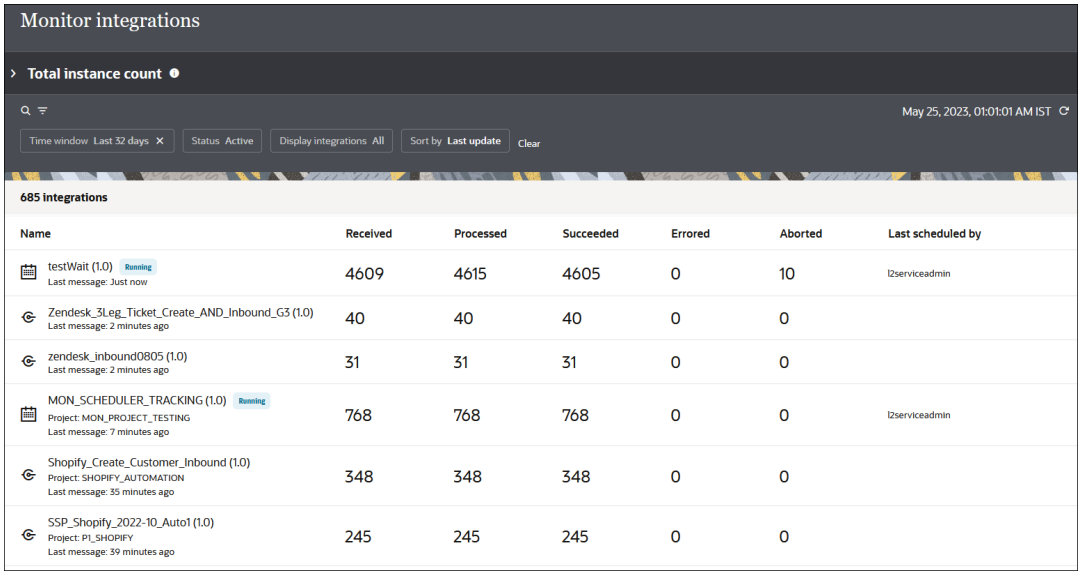
   This summary shows the count of total available messages in various states present in the tracking store. This summary is different from the metrics shown on the Integration dashboards page. While the Integration dashboards page metrics consist of historical data collected since the environment was created, the Monitor Integrations page shows only the current set of messages that have not yet been purged.



3. Click **Filter** , then select the time window, integration status (active, configured, activating, failed, and all), integration style (application, schedule, or event), or integration type (all integrations, only integrations in a project, or only integrations not in a project). If you select to show integrations in a project, you are prompted to begin entering the project name and select from the matching suggestions.

   You can also specify a custom time range to filter the search for integrations under **Date range** with the **From** and **To** buttons. This functionality also exists on other monitoring pages such as the Instances page or Errors page.

Copyright © 2022, 2026, Oracle and/or its affiliates.



The values shown in the table columns are based on the selected time window (last 1 hour, last 6 hours, and so on):

- **Received**: Shows the number of new incoming messages (or instances created). It is possible that some of these messages will not complete processing within the selected time window.
- **Processed**: Shows the sum of the **Succeeded**, **Errored**. and **Aborted** column values. This is the number of messages that completed processing in the selected time window. Some messages may have been received in a prior time window, but completed processing during the currently-selected window.
- **Succeeded**, **Errored**, and **Aborted**: The explanation is the same as the **Processed** column count. The only difference is the different terminal states.

A difference can exist between the number of messages listed in the **Received** and **Processed** columns. For example, assume there are 76 **Received** column messages and 184 **Processed** column messages. This means that for the selected time window, 76 new messages were received. In that same time window, 184 messages completed processing. Many of the 184 would have been received before the selected time window, and therefore not counted in the **Received** message total. Similarly, many of the 76 **Received** messages may not complete within the selected time window. Therefore, even though they are counted in the **Received** total, they are not counted in the **Processed** total.

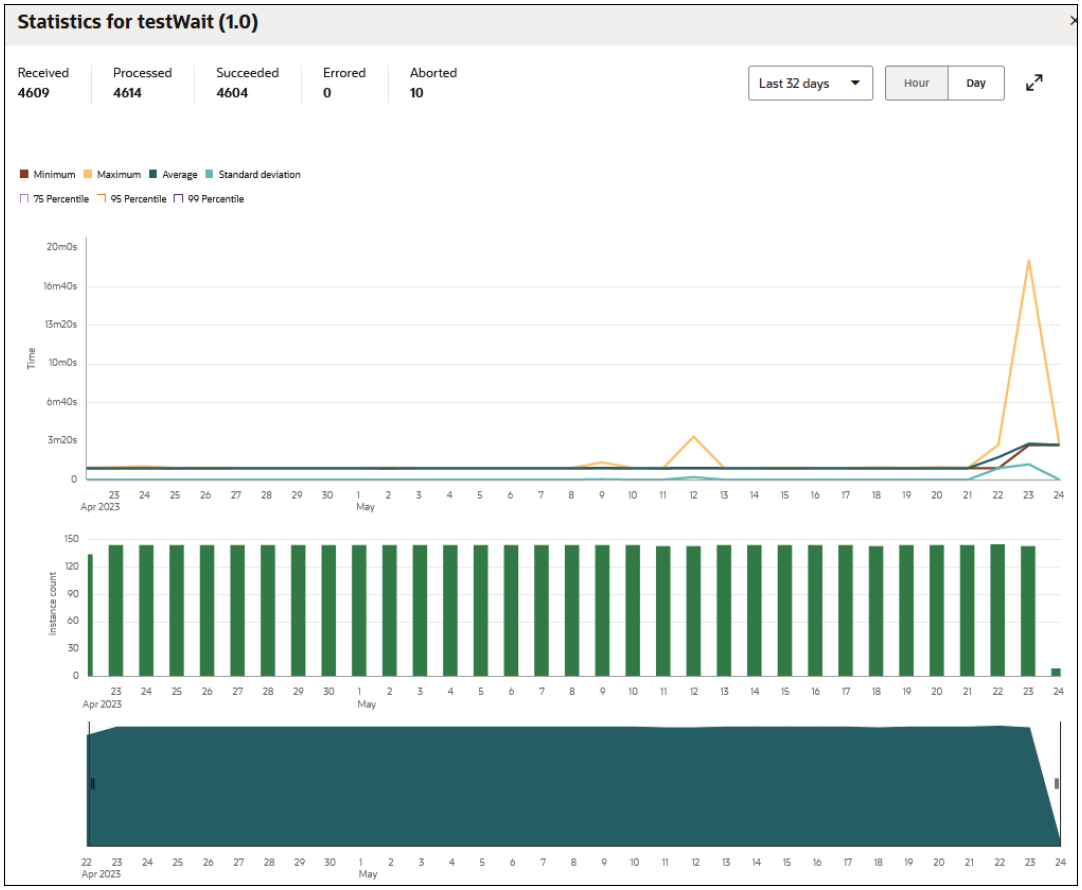
For schedule integrations (identified by ), the **Name** column includes a label identifying the state of the schedule integration:

- No schedule defined
- Paused
- Running
- Started
- Stopped

The user who last submitted the schedule of an integration is displayed in the **Last scheduled by** column. For nonschedule integrations such as application integrations, this column remains empty.

Copyright © 2022, 2026, Oracle and/or its affiliates.

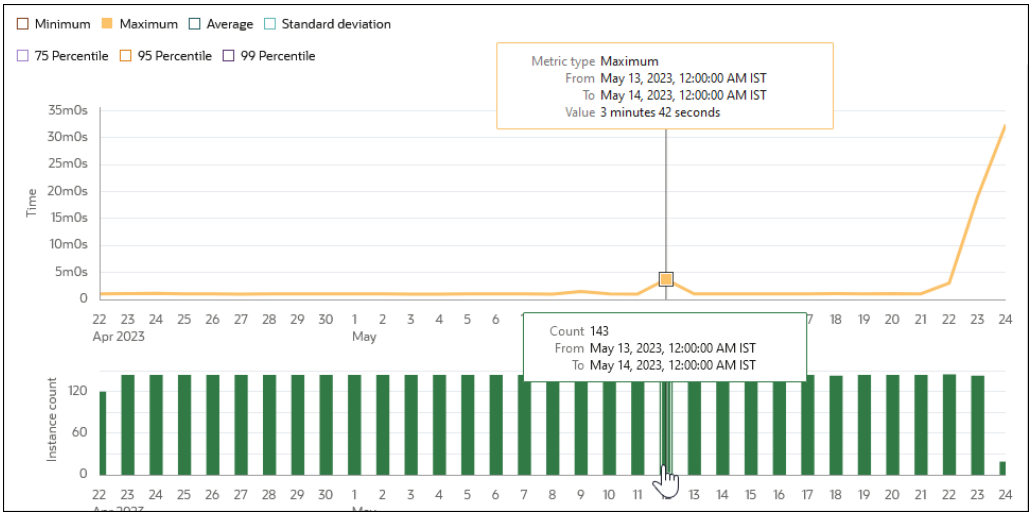
4. Click numbers in the **Received**, **Processed**, **Succeeded**, and **Aborted** columns to access the list of instances in those states on the Instances page. If you click the **Received** number, instances are displayed based on their creation date. All other clickable numbers are sorted by their last updated date.
5. Click numbers in the **Errored** column to access the list of instances in that state on the Errors page.
6. Hover over a row for an integration and click **View statistics** to show aggregated instance metrics for the selected integration. You can view the total counts and the minimum (quickest), maximum (slowest), average (mean), and standard (largest) deviation execution times of integrations over a selected time period (for example, last hour, last 32 hours, last week, and more). You can also select to view which of the execution times fall into the 75th, 95th, and 99th percentiles.



   a. Select the check boxes for which to view execution time metrics. By default, each check box is selected.
      - **Minimum**: Minimum (quickest) integration execution times.
      - **Maximum**: Maximum (slowest) integration execution times.
      - **Average**: Average (mean) integration execution times.
      - **Standard deviation**: Standard (largest) deviation execution times.
   b. If you want to know the percentile into which the selected metric falls, select a percentile.
   c. Select a time range for which to view metrics.

Copyright © 2022, 2026, Oracle and/or its affiliates.

d. Drag your cursor across the time segments in the **Instance count** section or drag the vertical bar across the **Time** section to show instance execution metrics for a particular time period. These metrics can help identify execution trends. For example, execution times are slower at one time, which may occur because of an adapter connection issue. Or there is a significant wait time for schedule integrations to run because too many schedule integrations are executing at once. You can then revise the schedule to make more optimal use of Oracle Integration.

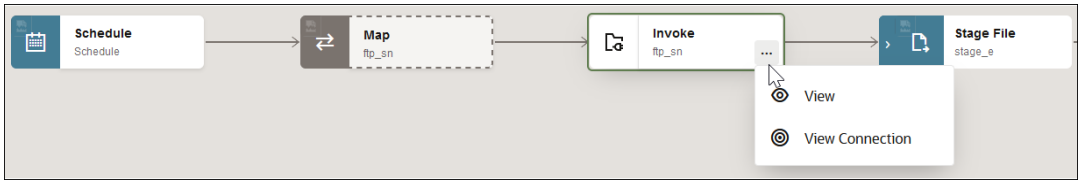


e. Click a time segment bar in the **Integration count** section to access a filtered list of the instances for that segment on the Instances page.

7. Click an integration in the **Name** column to access a graphical view of the integration flow.

   The **Share** drop-down list provides options for exporting the integration or printing a graphical view of the integration.

   The inline menus of the triggers, invokes, and actions in the integration flow provide access to additional details. For example, for an invoke connection:



8. Click **Actions** ... to view a set of options. The options that appear are based on the trigger, action, or invoke that you selected.

   a. For actions, click **View** to view configuration details.

   b. For triggers and invokes, click **View** to view Adapter Endpoint Configuration Wizard configuration details and **View connection** to view connection configuration details.

   c. For schedules, click **View** to view configured schedule parameters or **Edit schedule definition** to edit the schedule definition.

   a. For logic actions such as a scope, switch, for each, and others, click **Expand** to view all elements within the action or **Collapse** to close the logic action.

   b. For scopes, click **Fault handlers** to view any fault handlers defined in the scope.

Copyright © 2022, 2026, Oracle and/or its affiliates.

9. For schedule integrations (identified by ), hover over the row of the integration to access and click **Actions** ⋯. The tasks available for selection are based upon the state of the schedule integration.

Schedule Integration State	Actions Menu Items
No schedule defined	• **Add schedule** • **Run**
Paused	• **Run** • **Stop schedule** • **Resume schedule** • **Schedule**
Running	• **Pause schedule** • **Run** • **Schedule** • **Stop schedule** • **View future runs**
Started	• **Retry** • **Run**
Stopped	• **Edit schedule** • **Run** • **Schedule** • **Start schedule**

> **Note**
>
> If you are a user with the *Service*Administrator role, you can change the user with which to run the schedule.

10. For schedule integrations, hover over a row and select **Open Details** ⌄ to view details about the time of the last message and the time for the next scheduled run.

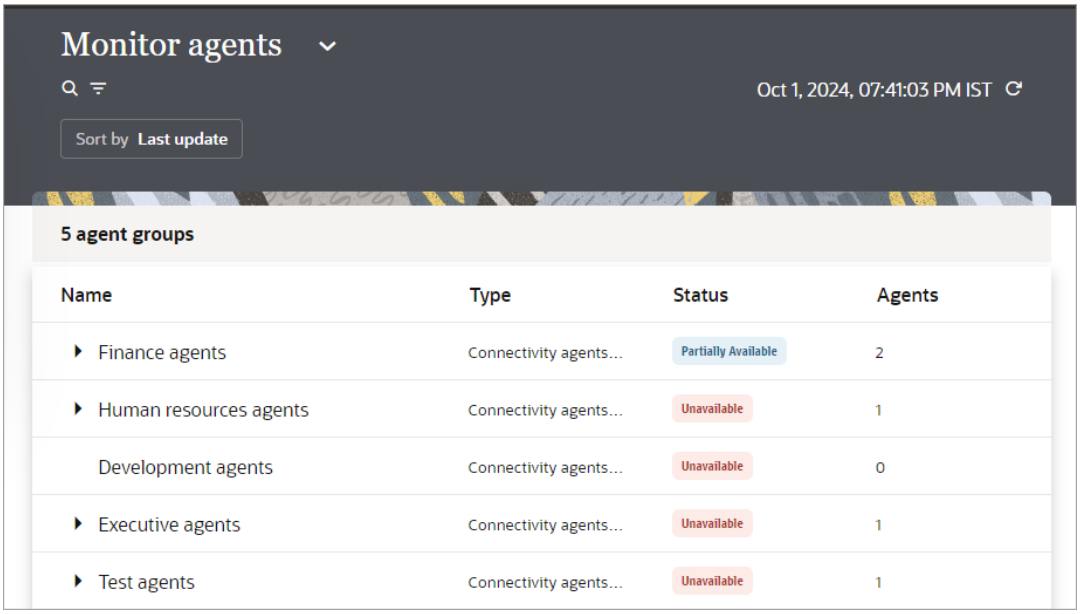
Monitor Agents

When the Dashboard tells you that one or more connectivity agents are unavailable, come to the Agents page. Here, you can determine which connectivity agents and agent groups are down.

For details about monitoring the environment that robot instances run on, see Check Whether an Environment is Available in *Using Robots in Oracle Integration 3*.

1. In the navigation pane, click **Observability**, then **Agents**.

Copyright © 2022, 2026, Oracle and/or its affiliates.



The Monitor Agents page shows details such as the time at which the agents were last updated and the connectivity agents associated with the agent groups.

- If the agent status is green, this indicates that the agent is running and able to process messages.
- If the agent status is red, you cannot select it in the Select an Agent Group dialog of the Connections page, even though the agent may be up and running.

2. Click **Expand** to view details about the agent group such as the creation date and the last startup date for the connectivity agent.

> **Note**
>
> When you shut down the connectivity agent server, several minutes pass before the agent is displayed as down on the Monitor Agents page (as indicated by a red thunderbolt).

Learn About the Payloads in the Activity Stream

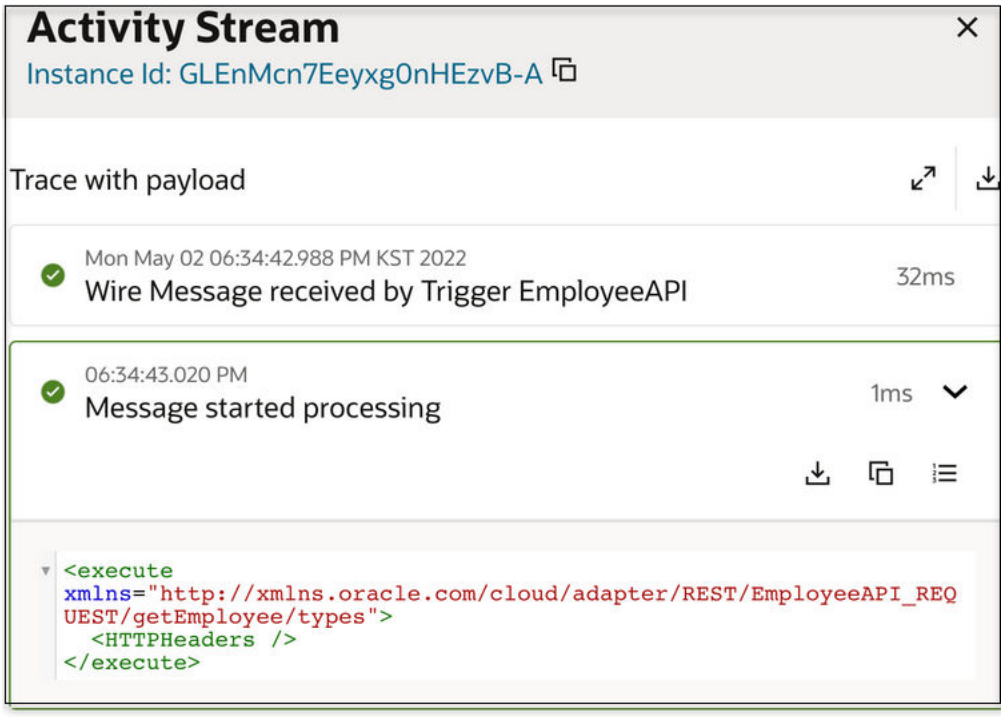
The activity stream shows details about how messages move through an integration, including whether failures occurred. The activity stream in the user interface shows only messages of certain sizes and types. However, you can see all payloads at any time by downloading the activity stream.

- Types of Payload Messages That Appear in the Activity Stream
- Payload Messages That Appear in the Activity Stream

Types of Payload Messages That Appear in the Activity Stream

The activity stream provides an inline view of some payloads. However, binary payloads, such as a zip file, cannot be viewed in the activity stream. Instead, you can directly download those payloads for viewing.

Copyright © 2022, 2026, Oracle and/or its affiliates.

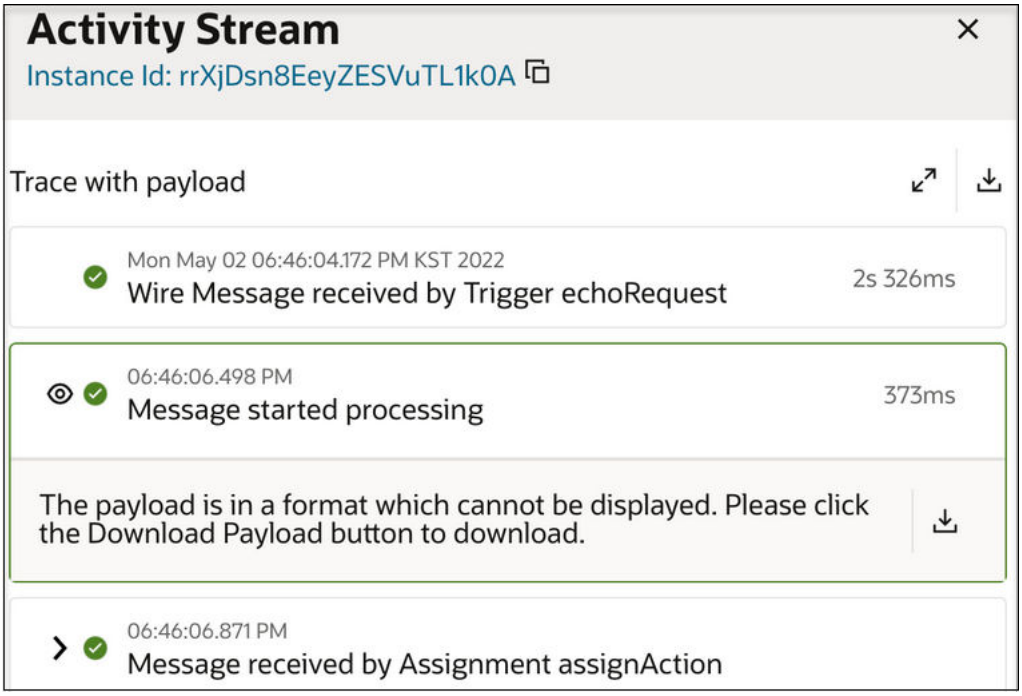


Payloads with the following media types (MIME types) can be viewed in the activity stream:

- `text/javascript`
- `text/typescript`
- `text/strings`
- `text/html`
- `text/x-yaml`
- `text/css`
- `text/x-sql`
- `text/x-mysql`
- `text/x-sh`
- `text/plain`
- `text/xml`
- `application/json`
- `application/xml`
- `application/typescript`
- `application/javascript`

For all other media types, download the payloads directly to your system by clicking **Download** at the top of the activity stream to download the entire activity stream of the instance or inline to download the payload for the selected section of the activity stream.

Copyright © 2022, 2026, Oracle and/or its affiliates.



Payload Messages That Appear in the Activity Stream

The size of the payload determines whether it can be viewed in the activity stream or must be downloaded for viewing.

Attachments are not considered part of the payload and are not visible in the activity stream. Adapters store attachments and delete them after processing completes.

Payload Sizes	Viewable in the Activity Stream
Small: Less than 32 KB	Yes (visible inline)
Large: More than 32 KB	No (can be downloaded from the activity stream)

See Track the Status of Integration Instances and View Error Messages and the Message Payload in the Activity Stream of Failed Instances.

Measure an Integration's Performance

Expect to measure an integration's performance often, including while designing the integration, before promoting the integration to production, and periodically after the integration is in production.

Perform any or all of the following tasks to measure an integration's performance. If you're troubleshooting, these steps help you determine the activities that are taking the most time and analyze how many integration instances have run for longer than expected. This information can help you determine your next steps.

What Do You Want to Do?

- Determine an Integration's Typical Processing Times
- Determine an Integration's Performance on a Given Day
- Determine Whether an Integration's Performance Is Consistent Over Time

Copyright © 2022, 2026, Oracle and/or its affiliates.

- Find Integration Instances with Increased Processing Times
- Measure the Running Times of an Integration's Actions

Determine an Integration's Typical Processing Times

You should know your integrations' typical running times so that you can more easily identify the integration instances that took longer than expected to run.

You have two options for checking processing times: review numerical times on the Instances page, or review graphical information on the Integrations page.

How Oracle Integration Calculates Duration

Oracle Integration calculates an integration instance's duration in the following ways:

- Schedule integrations: Time that passes from start to finish.
- Failed integrations: Time for all activities, including sending requests and rerunning the integration.
- All other integrations: Time that passes between the triggering event and the flow finishing.

Option 1: Review Graphical Information

1. In the navigation pane, click **Observability**, then **Integrations**.
2. Hover over the row of an integration, and select **View processing time statistics** .

   The Processing time panel opens. The panel shows the processing time for all integration instances for the integration.
3. If needed, update the time filters at the top of the panel.



4. Above the chart, select the following values to hide the lines for the values:
   - **Minimum**
   - **Maximum**
   - **Standard deviation**

   Only the Average value remains.
5. Review the average processing times for the integration over time.

Option 2: Review Numerical Times

1. In the navigation pane, click **Observability**, then **Instances**.
2. Click **Filter** , and update the filtering requirements for the page.

   For example, from the **Sort by** drop-down, select **Duration**. You might also want to update the time window filter.
3. For each integration instance, find the **Duration** column, and review the value in the **Duration** field.

Copyright © 2022, 2026, Oracle and/or its affiliates.

Determine an Integration's Performance on a Given Day

You can quickly measure an integration's performance for a given day.

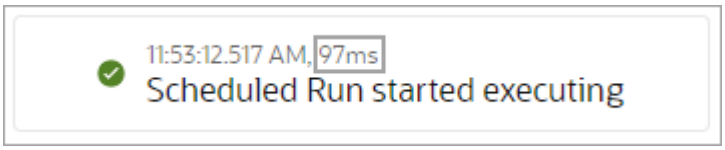
1. In the navigation pane, click **Observability**, then **Integrations**.
2. Hover over the row of an integration, and select **View processing time statistics**.

   The Processing time panel opens. The panel shows the processing time for all integration instances for the integration.
3. If needed, scroll down until you see the Instances chart, which is the second graphical chart on the page.
4. Point to the vertical line for a given date, such as a date with a spike in its processing times, and click the line.

   The Instances page appears and shows only the integration instances that ran on the selected date. The integration instances are sorted by instance duration, with the longest durations appearing first.
5. Optional: Identity the activity that required a longer-than-usual processing time.
   a. For an instance with a longer-than-expected duration, hover over the row, and select **View details**.

      The activity stream opens.
   b. Review the processing time for each part of the integration flow to determine which steps took the most time.

      The processing time appears after the timestamp, above the information about the activity.

   c. Identity the activity that required a longer than usual processing time.

Determine Whether an Integration's Performance Is Consistent Over Time

You typically verify that an integration meets your organization's performance requirements before promoting the integration to production and then periodically after the integration is in production.

1. In the navigation pane, click **Observability**, then **Integrations**.
2. Hover over the row of an integration, and select **View processing time statistics**.

   The Processing time panel opens. The panel shows the processing time for all integration instances for the integration.
3. If needed, update the time filters at the top of the panel.



Copyright © 2022, 2026, Oracle and/or its affiliates.

4. Check whether the standard deviation varies from the average performance time.

   Ideally, the values for these lines should be similar. If they're not, a number of integration instances might take longer than expected to run.

   a. Above the first chart, select the following values to hide the lines for the values:

      - **Minimum**
      - **Maximum**

      Only the **Average** and **Standard deviation** values remain.

   b. Compare the **Average** and **Standard deviation** lines.

      Similar values indicate that the integration instances have consistent durations. When the lines aren't similar, consider investigating the outliers.

5. Check for spikes in the the maximum and average performance times.

   Ideally, the values for the **Maximum** and **Standard deviation** lines should remain low over time. When these values spike, some integrations are taking longer than usual to run.

   a. Hide the **Average** line, and show the **Maximum** line.

      Only the Maximum and Standard deviation values remain.

   b. Review the **Maximum** and **Standard deviation** lines.

      Low values indicate that integration instances for this integration have similar processing times. On the other hand, spikes in the lines indicate that some instances have taken longer than usual to process. Consider investigating these spikes.

Find Integration Instances with Increased Processing Times

Charts and graphs help you quickly determine whether a slow-performing integration instance is an outlier or a trend.

1. In the navigation pane, click **Observability**, then **Integrations**.
2. Hover over the row of an integration, and select **View processing time statistics**  .

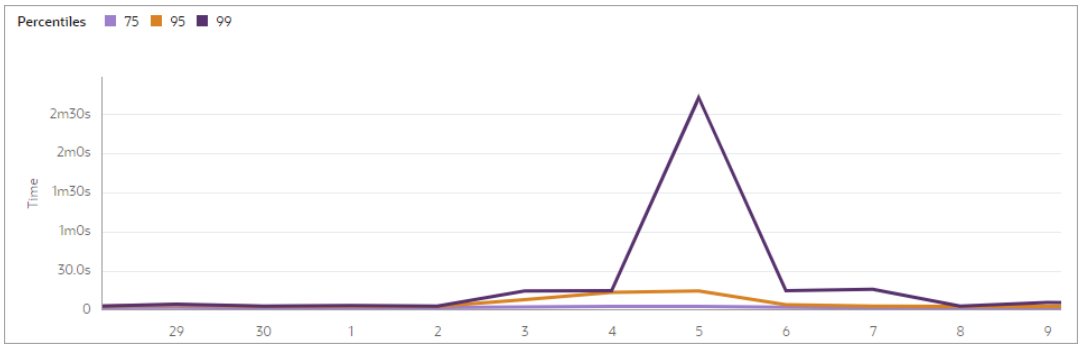
   The Processing time panel opens. The panel shows the processing time for all integration instances for the integration.
3. Above the first chart, select the following values to hide the lines for the values:

   - **Minimum**
   - **Maximum**
   - **Average**
   - **Standard deviation**
4. Above the first chart, select the following values to show the lines for the values:

   - **75**
   - **95**
   - **99**
5. Point to the spikes for the each of the percentile lines, and view their processing times in a popup.

Copyright © 2022, 2026, Oracle and/or its affiliates.

These numbers offer insights into the processing times of integrations. For example, consider the following chart.



- The 75 line indicates that 75 percent of instances of this integration finished in about 3 seconds or less.
- The 95 line indicates that 95 percent of instances of this integration finished in about 15 seconds or less.
- The 99 line indicates that 99 percent of instances of this integration finished in about 2 minutes, 30 seconds or less.

6. Determine whether the performance for the integration has remained consistent.

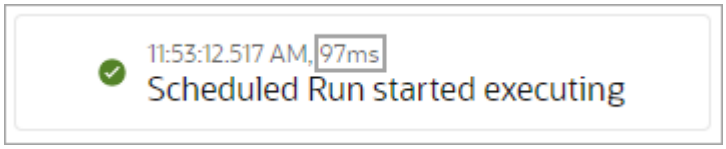
Measure the Running Times of an Integration's Actions

When an integration instance takes longer than expected to finish, determining the actions that took the most time can help you troubleshoot.

1. In the navigation pane, click **Observability**, then **Instances**.
2. Click **Filter**, and update the filtering requirements for the page.
3. Hover over the row of an integration instance, and select **View details**.

   The activity stream opens.
4. Review the duration for each part of the integration flow to determine which steps took the most time.

   The processing time appears after the timestamp, above the information about the activity.



Find Integrations That Experienced Errors

When you're responsible for monitoring integrations, part of your job involves determining whether any integrations experienced errors and then taking the appropriate next steps.

Errors in live integrations are never expected. Oracle recommends investigating every error that an integration experiences.

Copyright © 2022, 2026, Oracle and/or its affiliates.

What Do You Want to Do?

- Find Integrations with Errors
- Determine the Source of an Integration's Error
- Find the Integration Error Rate for the Last 24 Hours
- Measure Your Integration Errors Over Time
- Find the Connections with the Most Errors

Find Integrations with Errors

Oracle recommends investigating every error that occurs with your integrations.

1. In the navigation pane, click **Observability**, then **Integrations**.
2. Click **Filter**, and update the filtering requirements for the page.
3. In the **Errored** column, look for any non-zero values.

Determine the Source of an Integration's Error

Find the source of an error by checking the activity stream. You can also see the location of the error within an integration flow on the integration canvas.

1. In the navigation pane, click **Observability**, then **Integrations**.
2. Click **Filter**, and update the filtering requirements for the page.
3. In the **Errored** column, look for any non-zero values.
4. For an integration with one or more errors, determine the location of the error(s).
   a. Click the number value in the **Errored** column for the integration.

      The Errors page appears and shows only the integration instances that experienced errors.

   b. For an error, review the value in the **Fault location** column.

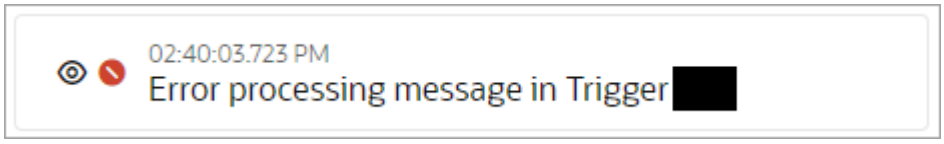
      The fault location indicates where the error occurred. For example, if errors occurred for an invoke that multiple integration instances use, the application that the invoke connection is connecting to might be unavailable.
5. Locate the source of the issue by reviewing the activity stream.

   a. Hover over an errored integration instance, and select **View details**.

      The activity stream opens.

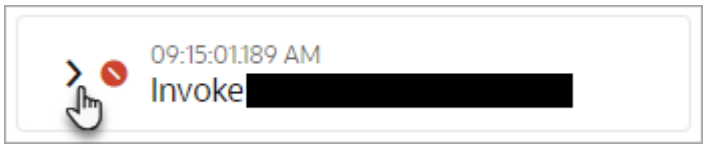
   b. In the activity stream, find the milestone with an error and open the activity stream from it using one of the following options.

      - If **View payload** appears on the milestone, click **View payload**.



Copyright © 2022, 2026, Oracle and/or its affiliates.

- If **View payload** doesn't appear on the milestone, expand the milestone by clicking the expand icon **>**, and then click **View payload** for the part of the integration flow that failed.



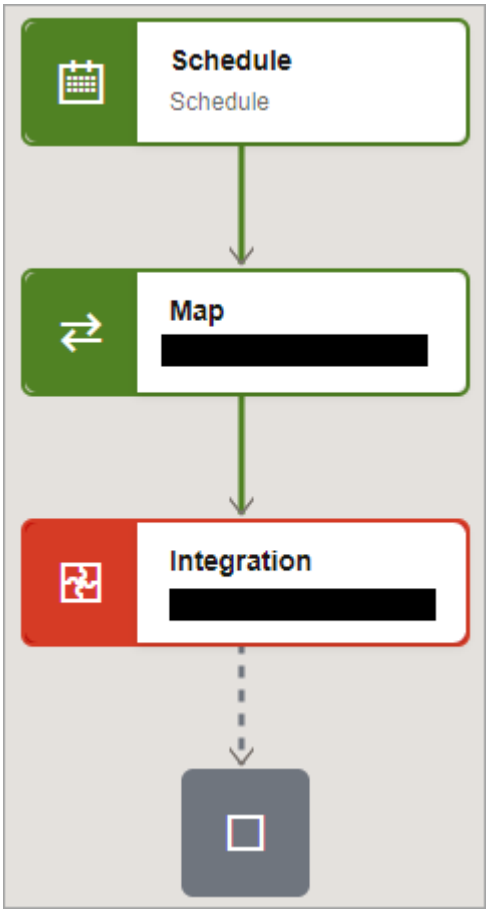
c. Review the payload and take the appropriate action.

   For example, if a 404 error occurred, check the configuration of the connection.

6. Optional: To see the error within the integration flow, open the integration canvas.

   a. On the Errors page, in the **Primary identifier** column, click the name of the primary identifier, such as **Primary: undefined**.

      The integration canvas appears. The integration component with an error is marked in red.



   b. To see the error in the activity stream: Point to the component with an error, click **Actions** ..., and select **Activity stream**.

      The activity stream opens.

Copyright © 2022, 2026, Oracle and/or its affiliates.

7. If you are not responsible for addressing the error, send the error information to the owner of the integration.

   For example, you might perform some or all of the following steps:

   - Take a screenshot of the activity stream.
   - Copy the payload text.
   - Download the logs for the activity stream.

   > **Tip**
   >
   > Click **Download Logs**  at the top of the activity stream.

Find the Integration Error Rate for the Last 24 Hours

Consider completing a daily check of your integration error rate over the last 24 hours. If the value is greater than 0, investigate the errors.

1. In the navigation pane, click **Observability**, then **Dashboards**.
2. Along the top of the page, make sure that **Error rate (past 24 hours)** is selected.
3. Review the value in the **Error rate (past 24 hours)** box.

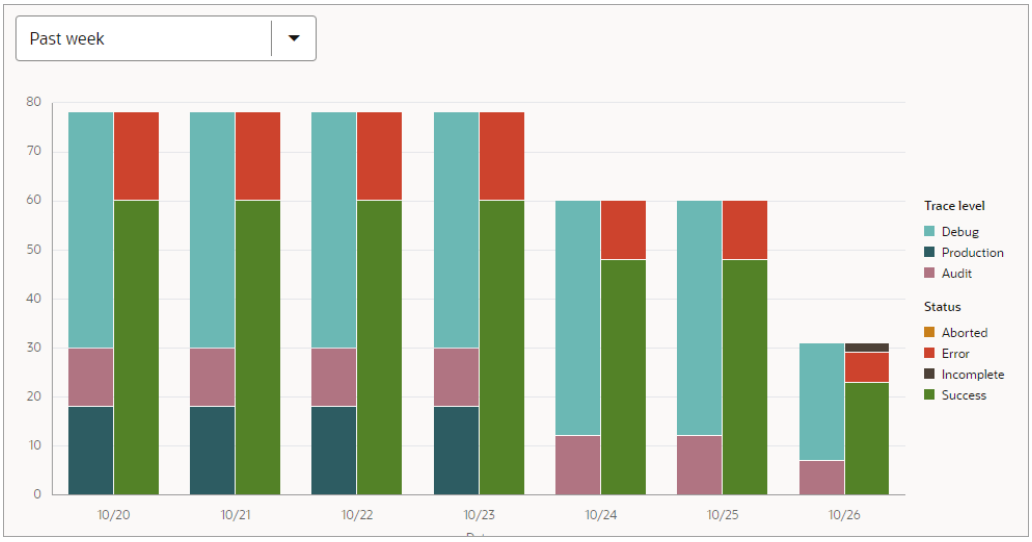
   This metric shows the percentage of integration instances that experienced errors in the last 24 hours.

Measure Your Integration Errors Over Time

Measuring your integration errors over time can help you track trends and measuring improvements.

1. In the navigation pane, click **Observability**, then **Dashboards**.
2. Below the **Instance status** header, use the drop-down to adjust the time period, if necessary.
3. Review a chart to visually track your errored integrations over time.
   a. Familiarize yourself with the chart that appears below **Instance status**.

Copyright © 2022, 2026, Oracle and/or its affiliates.



- Each time period contains two vertical bars.
- The **left bar** visualizes the **tracing levels** for the integrations that ran during the time period.
- The **right bar** visualizes the **statuses** for the integrations that ran during the time period.

b. Determine whether any integration instances experienced errors during the time period.

4. Review the metrics to understand how many integration instances have experienced errors over time.

   The error-related metrics appear to the right of the chart, below the **Overview** heading.

   - **Error rate:** Error rate for all of the integration instances during the time period.
   - **Instances:** Number of integration instances that ran during the time period.
   - **Errors:** Number of integration instances with a state of Errored during the time period.

Find the Connections with the Most Errors

Knowing the connections that result in integration instances with errors can help you quickly prioritize integration failures and start the troubleshooting process.

Why Connections Experience Errors

The typical causes of connection errors are varied and depend upon the adapters you use, the way you use the connections, and your use cases. For example:

- The integration developer configured the connection incorrectly.

  In this case, the connection always fails.

- The connection cannot complete its task.

  Consider a connection that is based on the FTP Adapter. The connection fails if someone moves or deletes the folder that the integration reads from or changes the permissions on the folder.

Copyright © 2022, 2026, Oracle and/or its affiliates.

To Find the Connections with the Most Errors:

1. In the navigation pane, click **Observability**, then **Dashboards**.
2. Along the top, select the **Connections with errors** metric.

   The **Top connections with errors** section appears, listing up to five of the connections with the most errors.
3. Below the **Top connections with errors** heading, point to the box for a connection.

   A tooltip displays the number of integration instances that use the connection and experienced an error.
4. Find the integration instances that used the connection and experienced errors.
   a. Click within one of the connections boxes.

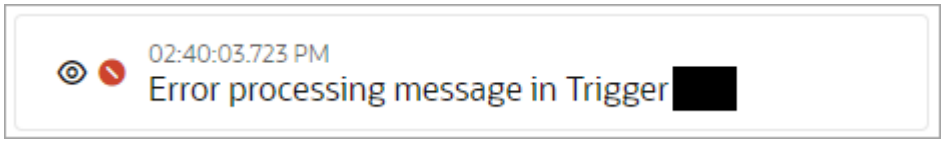
      The Errors page appears and shows only the integration instances that used the connection and experienced an error.

   b. Hover over an errored integration instance, and select **View details** .

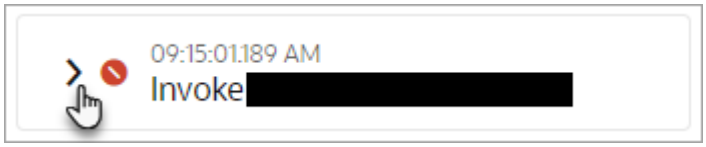
      The activity stream opens.

   c. In the activity stream, find the milestone with an error and open the activity stream from it using one of the following options.

      - If **View payload** appears on the milestone, click **View payload** .

      - If **View payload** doesn't appear on the milestone, expand the milestone by clicking the expand icon **>**, and then click **View payload** for the part of the integration flow that failed.

   d. Review the payload and take the appropriate action.

      For example, if a 404 error occurred, check the configuration of the connection.

Copyright © 2022, 2026, Oracle and/or its affiliates.

Find Integrations That Timed Out or Were Aborted

Being able to quickly find an integration that has timed out or been aborted can help you troubleshoot.

Why Integration Instances Have a Status of Aborted

Several activities can change an integration instance's status to Aborted, including the following:

- Someone canceled the integration.
- Someone stopped the scheduled of a scheduled integration.
- Someone edited a connection for a queued or in progress schedule integration.
- The integration timed out.

Time Limits for Integrations

Different integrations can run for different amounts of time. To see the time limits, see Service Limits in *Provisioning and Administering Oracle Integration 3*.

To Find All Aborted Integrations:

You can work on either the Instances or Integrations page.

- Find all aborted integrations using one of the following options.
  - Filter the Instances page so it shows only aborted integrations.
    1. In the navigation pane, click **Observability**, then **Instances**.
    2. Click **Filter**, and from the **Status** dropdown, select **Aborted**.
    3. Update the other filters as needed, and click **Apply**.

       The page shows only aborted integrations.
  - Scan the Integrations page and look for any integrations with one or more aborted instances.
    1. In the navigation pane, click **Observability**, then **Integrations**.
    2. Review the Aborted column for any non-zero values.

       An integration with a non-zero value in the Aborted column has one or more aborted integration instances.

Next, determine why an integration was aborted. See Determine Why an Integration Instance Stopped Running.

Manage Errors

You can manage errors from the Errors pages in Oracle Integration. You can resubmit failed instances, discard failed instances, view message recovery status, and view basic and detailed error messages. You can also view and download the activity stream, which provides access to error messages and message payloads in failed instances.

Topics:

- View Error Messages and the Message Payload in the Activity Stream of Failed Instances

Copyright © 2022, 2026, Oracle and/or its affiliates.

- View an AI-Powered Error Message
- Resubmit Failed Messages
- Discard Errors
- View the Status of Message Recovery
- View Errors By Integration and Connection
- View the Business Identifiers Used in Failed Integration Instances

See About Error Management.

View Error Messages and the Message Payload in the Activity Stream of Failed Instances

You can view error messages and the message payload in the activity stream of failed integration instances on the Errors page. You can also download the activity stream logs and the payload of the failed section of an integration instance.

1. In the navigation pane, click **Observability**, then **Errors**.
2. Perform the following tasks:
   - Filter the Display of Error Message Details
   - View Error Message and Payload Details on the Errors Page
   - View the Flow of the Failed Instance and Payload Errors in the Activity Stream on the Instance Details Page

Filter the Display of Error Message Details

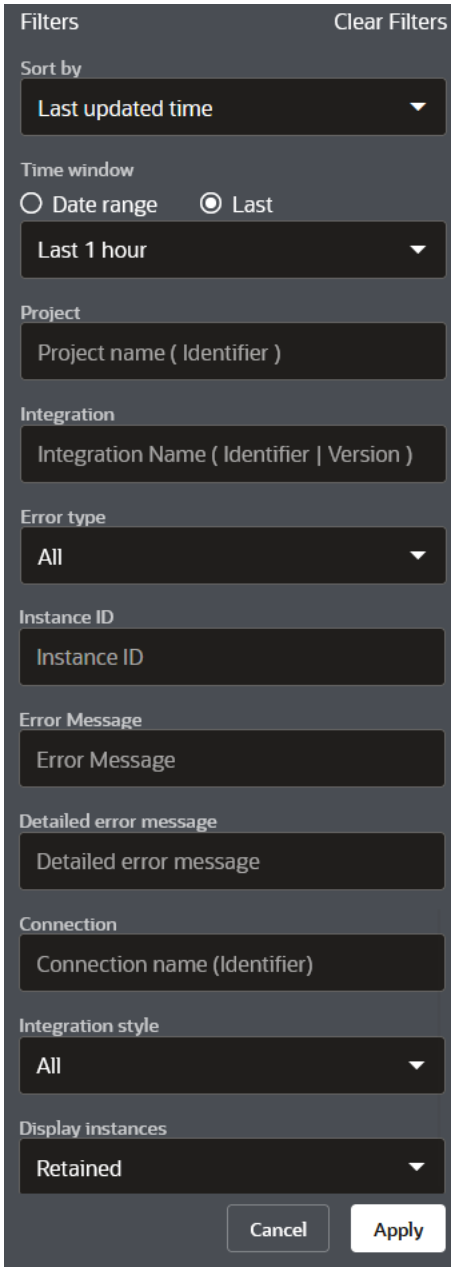
1. Click **Filter** to filter the display of errors by the following:
   - Last updated or creation time
   - Project (begin entering a name to display a list from which to select)
   - Integration
     Begin typing an integration name. Do not use copy and paste. The filter is refreshed to display fields for the defined secondary business identifiers.

     You can now begin typing the values of the business identifiers to narrow down the search. This input follows the same rules (for example, for exact matches and phrase matches) as described in Search for Primary and Secondary Business Identifier Values.
   - Error type (all, recoverable, or nonrecoverable)
   - Instance ID
   - Error message
     For example:
     - Enter part of an error message, such as `unknown`, `unauthorized`, or `endpoint`.
     - Enter an exact phrase match of an error message in double quotes, such as `"Could not activate endpoint for endpointID"`
   - Detailed error message
   - Connection name
   - Integration style (all, application, schedule, or event)

Copyright © 2022, 2026, Oracle and/or its affiliates.

- Integration state (all, retained, or purged)

When you enter an integration name, the filter is refreshed to display fields for the defined business identifiers. You can enter variable values with special characters or variable values of multiple words with white space in double quotes (for example, `&filename` or `"file tracking"`).



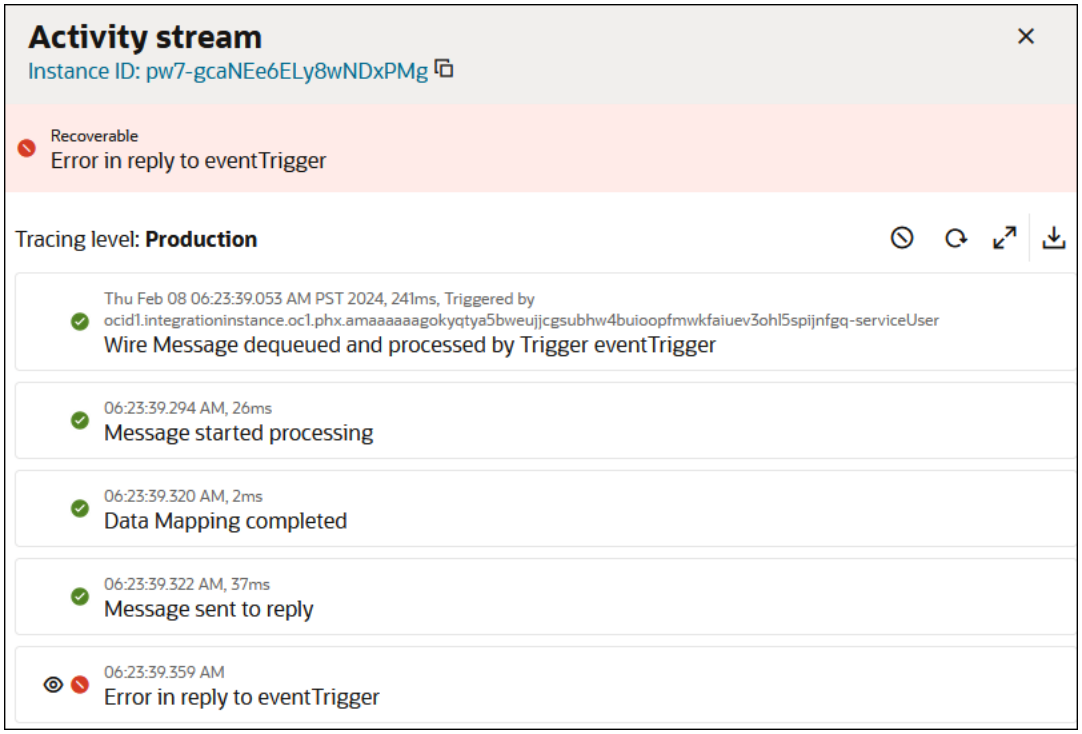
2. Click **Date Range** in the **Time Window** section.
3. Specify a custom time range with which to filter the search for errors with the **From** and **To** buttons. This functionality also exists on other monitoring pages such as the Instances page and Monitor Integrations page.

Copyright © 2022, 2026, Oracle and/or its affiliates.

View Error Message and Payload Details on the Errors Page

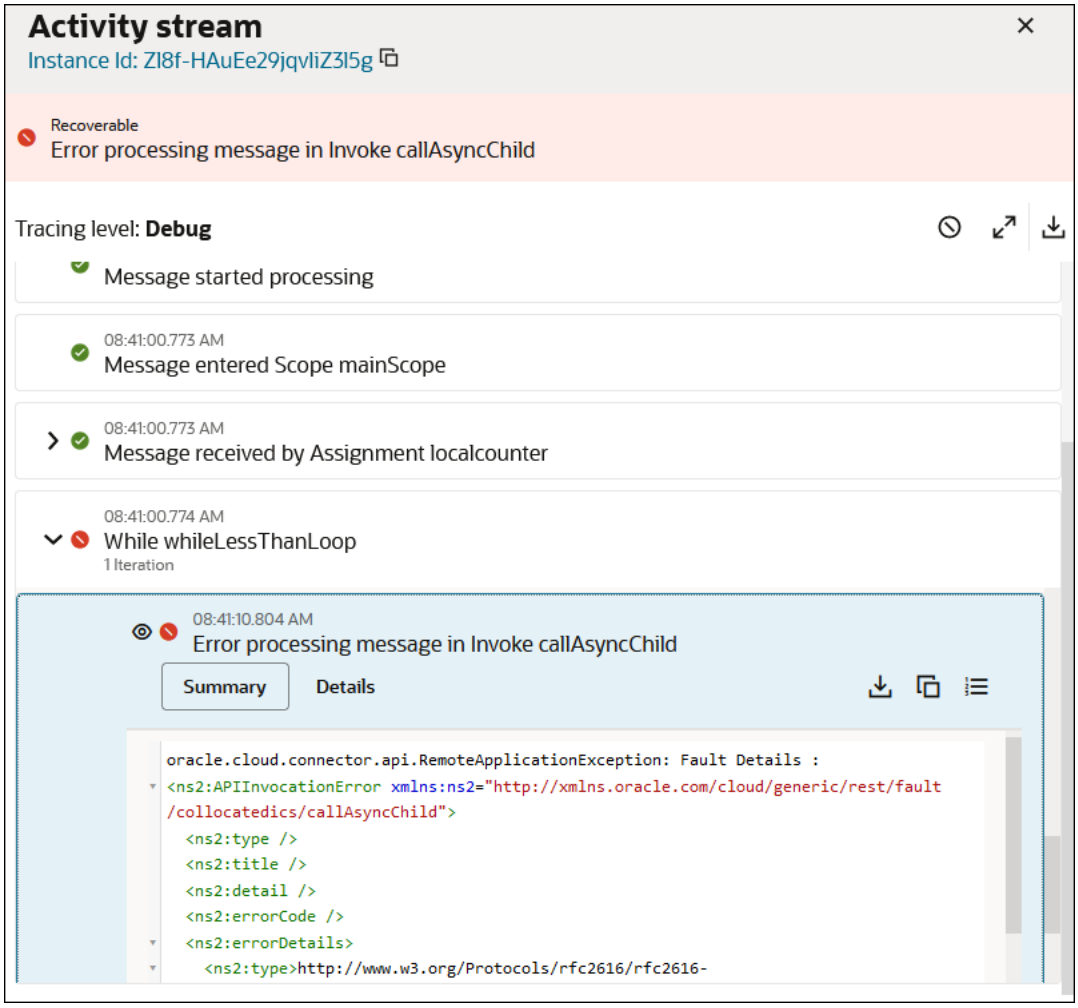
Error message details can be displayed in two parts: a summarized error message for easy understanding and resolution and, if necessary, a more specific error message if more detailed troubleshooting is required. You can view error messages on the Errors page or by clicking on a business identifier to view the entire integration flow on the Instance Details page. You can also view the payload details in the activity stream of a failed instance.

1. Hover over the row of the failed instance.
2. Click **View** . The Activity Stream panel opens. The activity stream shows details about the movement of the message through the integration, including where any failures occurred. The date and time according to your user preferences are displayed.



3. Click **Download Logs** at the top of the activity stream to download the activity stream logs for the entire instance. The Oracle Cloud identifier (OCID) value of your service instance is also included in the logs.
4. Scroll through, expand milestones, and click **View Payload** to view the payload for the part of the integration flow that failed.

Copyright © 2022, 2026, Oracle and/or its affiliates.

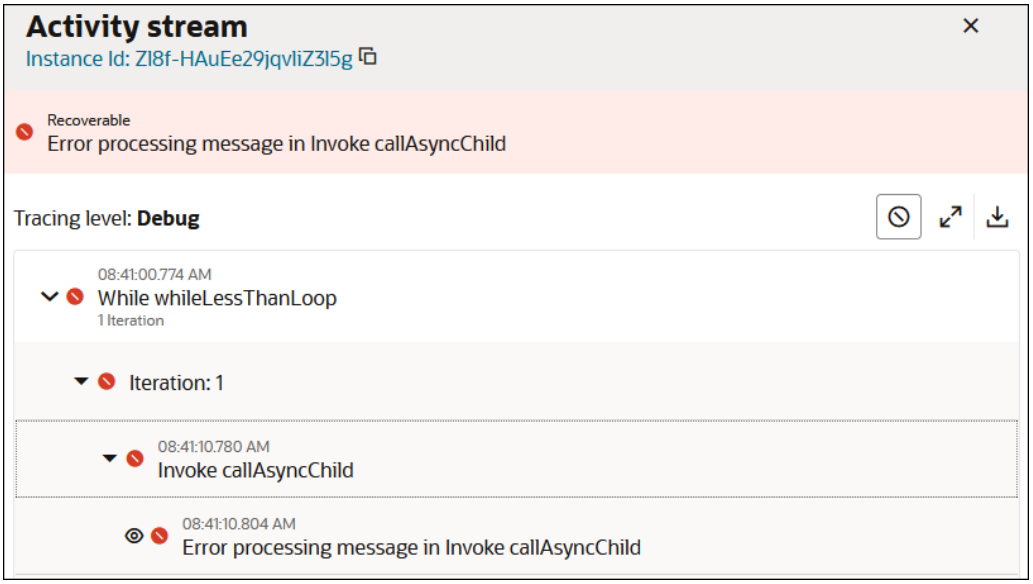


An option is provided in the activity stream for quickly showing all error content only. This option can be especially useful for instances in which a logic action such as a for-each loop or while loop with many iterations has failed. This option enables you to jump quickly to the iteration that failed instead of scrolling through each iteration of the loop to access the errors. Show errors in either of two ways:

a. Click **View Errors**

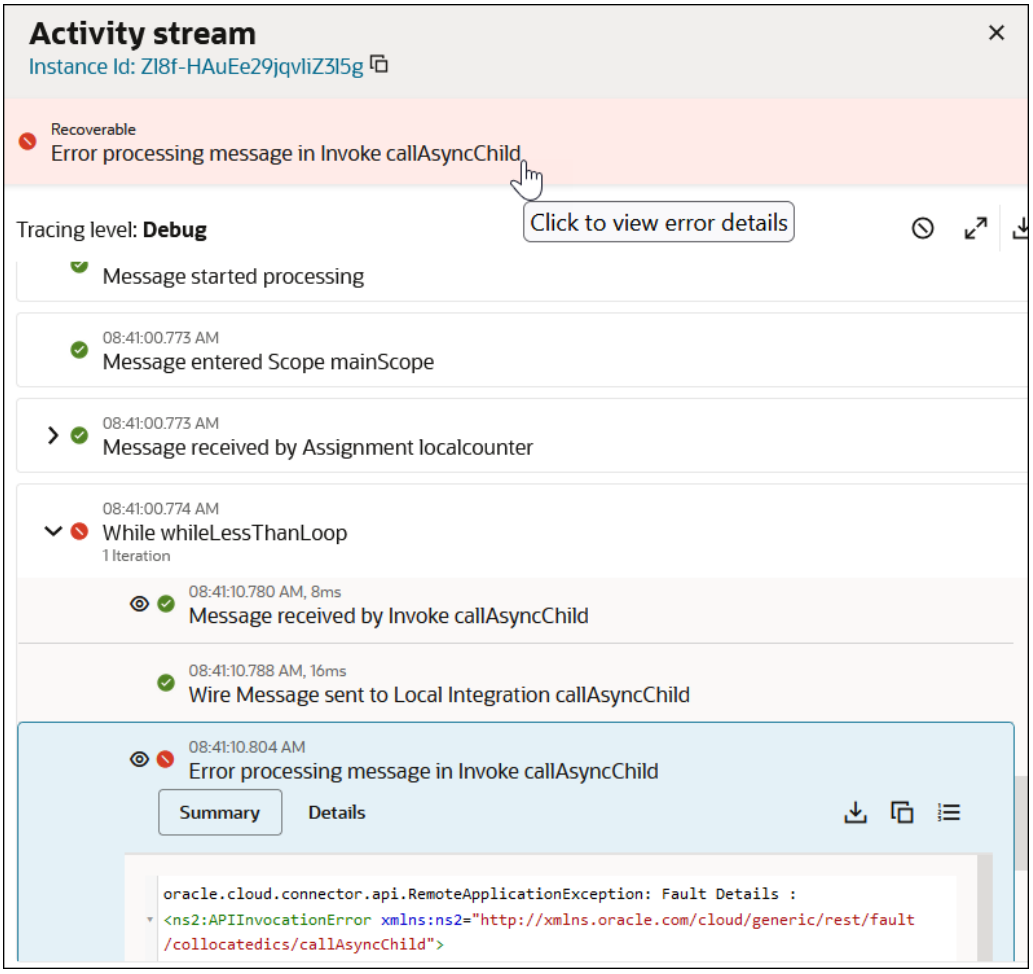
to view only the errors in the activity stream. This persists the selection of errors in the activity stream across multiple instances until you deselect the button. This button enables you to toggle between showing only errors and showing all content in the activity stream.

Copyright © 2022, 2026, Oracle and/or its affiliates.



or

b. Click the error in the activity stream banner to jump to that specific error.



5. Perform tasks within the failed part of the payload.

Copyright © 2022, 2026, Oracle and/or its affiliates.

Element	Description
Download Payload	Click to download this part of the payload to your desktop.
Copy Payload	Click to copy this part of the payload for pasting into a document.
Toggle Line Numbers	Click to show or hide line numbers in the payload.

or

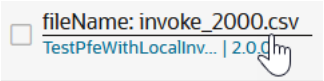
1. Hover over the row of the failed instance.
2. Click **Expand** to view a summarized version of the error message for easy understanding and a detailed version that describes how to resolve the error.



View the Flow of the Failed Instance and Payload Errors in the Activity Stream on the Instance Details Page

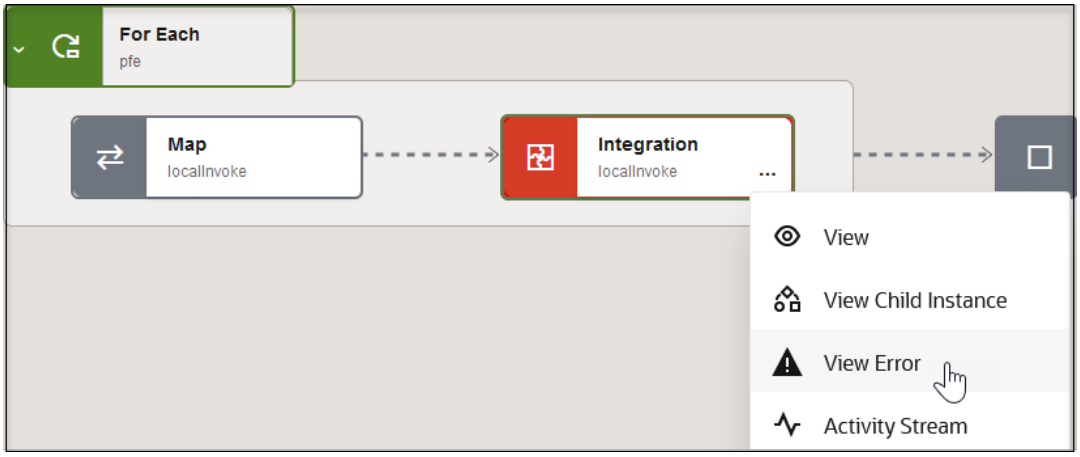
You can view the flow of the failed instance to identify any trigger, actions, or invokes in error on the Instance Details page. You can also drill into those elements to view specific error messages and payload errors.

1. Click a specific business identifier of an integration instance in the **Primary Identifier** column.

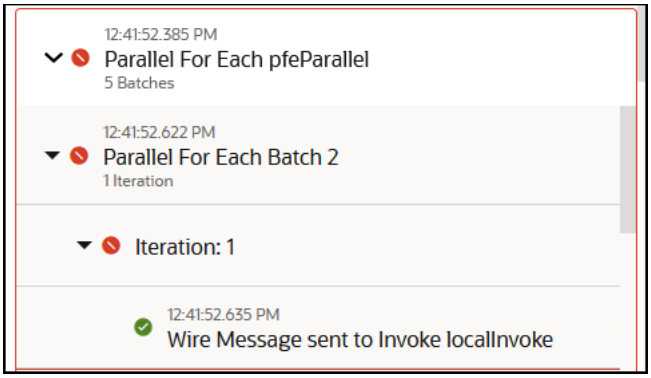


   The integration flow of the failed instance and Activity Stream panel are displayed.
2. Scroll through the flow to find the trigger, action, or invoke in error.
3. Click the element, then click **Actions** and select **View Error**.

Copyright © 2022, 2026, Oracle and/or its affiliates.



The Activity Stream panel opens to the specific section at which the error occurred. For this example, the error occurred in an integration action (parent integration calling a child integration) of a parallel for-each action. All five of the iterations of the parallel for-each are displayed for viewing in the activity stream.

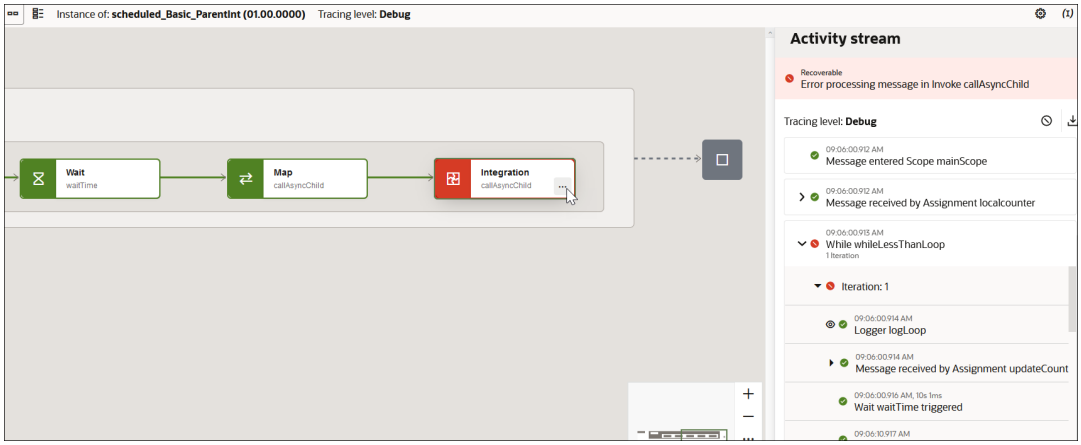


4. Scroll through and view the errors.
5. Click **Activity Stream**

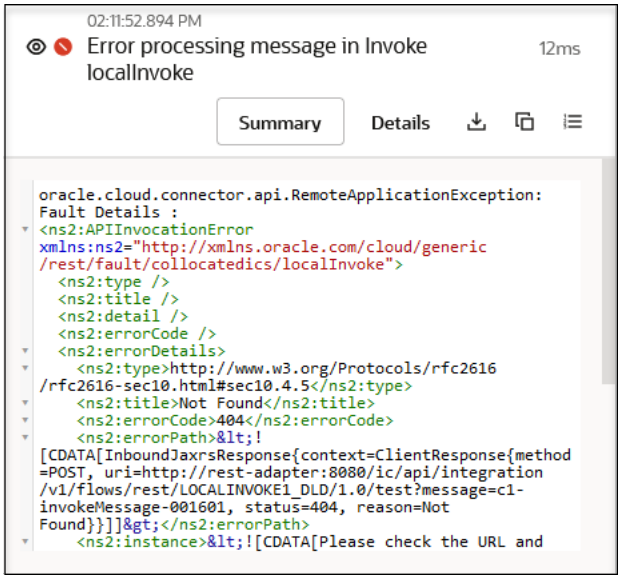
   to close the activity stream. This link operates as a toggle for closing or opening the activity stream.
6. Click the failed trigger, action, or invoke again, then click **Actions** ⋮ and select **Activity Stream**.
   This also opens the Activity Stream panel to the specific section at which the error occurred.

Copyright © 2022, 2026, Oracle and/or its affiliates.



7. Scroll through the payload and expand the sections in error to view specific details.

8. Click **View Payload** to view the payload of the section in error.



9. Perform other tasks within this section.

Element	Description
Summary	Click to view a summary of payload details. This action also highlights the section of the integration flow in error.
Details	Click to view more specific payload details. This action also highlights the section of the integration flow in error.
Download Payload	Click to download this part of the payload to your desktop.
Copy Payload	Click to copy this part of the payload for pasting into a document.

Copyright © 2022, 2026, Oracle and/or its affiliates.

Element	Description
Toggle Line Numbers	Click to show or hide line numbers in the payload.

10. Click **Activity Stream**



to close the activity stream.

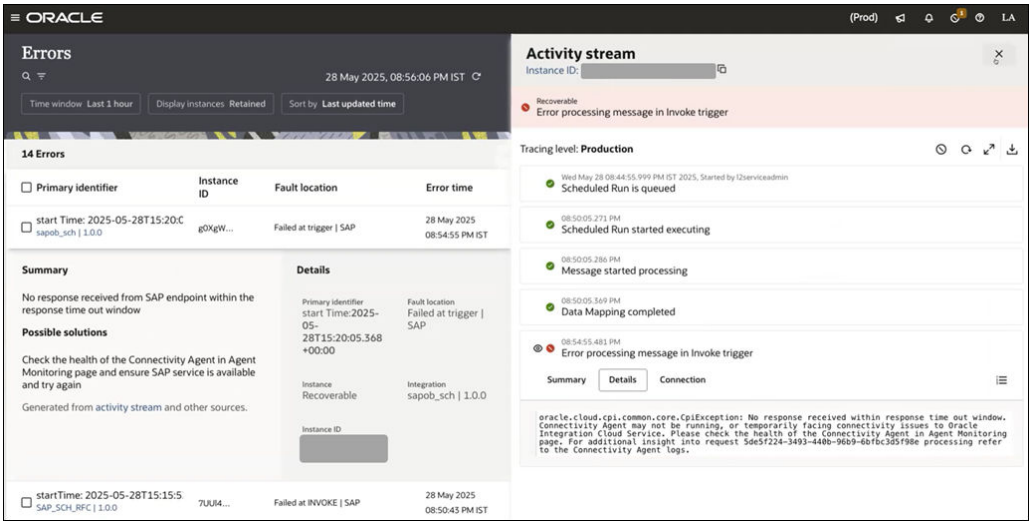
View an AI-Powered Error Message

You can view error messages generated by AI for failed integration instances. These error messages contain meaningful information that might help you identify the issues in the failed instances. They also provide possible solutions that might help you resolve the issues in the failed instances.

> **Note**
>
> AI features are available in select regions. See AI Feature Availability in *Provisioning and Administering Oracle Integration 3*.

1. To view the error message generated by AI for failed integration instances in a project:
   a. In the navigation pane, click **Projects**.
   b. Select the project name.
   c. Click **Observe**.
   d. Click **Instances**.
   e. Hover over an errored instance (instance whose status is **Errored**) row to see the actions you can perform on an errored instance.
   f. Click **Expand** to view an AI-powered summary describing the error and possible solutions that might help you resolve the error. You can optionally also click the **activity stream** link that is available below the solutions to view the information available in the activity stream. The errored activity is highlighted in the activity stream, and you can click **View** to see the summary and details of the original non-AI error messages.
2. To view the error message generated by AI for failed integration instances in a standalone environment:
   a. In the navigation pane, click **Observability**, then **Errors** or **Instances**.
   b. Hover over an errored instance (instance whose status is **Errored**) row to see the actions you can perform on an errored instance.
   c. Click **Expand** to view an AI-powered summary describing the error and possible solutions that might help you resolve the error. You can optionally also click the **activity stream** link that is available below the solutions to view the information available in the activity stream. The errored activity is highlighted in the activity stream, and you can click **View** to see the summary and details of the original non-AI error messages.

Copyright © 2022, 2026, Oracle and/or its affiliates.



Resubmit Failed Messages

You can manually resubmit failed messages. Oracle Integration does not automatically resubmit failed messages. After resubmitting an instance, you can track all resubmissions of that instance on the Instances page. You can also change the tracing level for a resubmitted message.

After resubmitting a failed message, the instance can restart from different places in the integration:

- From the beginning of the integration.
- From another location in the integration, depending upon the design. For example, if there is a wait activity in the integration prior to the recoverable fault, then resubmitting the fault starts the integration just after the wait activity.

Faulted instances in asynchronous flows in Oracle Integration are recoverable and can be resubmitted. An exception to this are scheduled flows with scheduled parameters, which are nonrecoverable. Synchronous flows also cannot be resubmitted. You can resubmit an instance a maximum of ten times. You can resubmit errors in the following ways:

- Single failed message resubmissions
- Bulk failed message resubmissions

There is a time window for recovering a failed integration instance that's recoverable. See Service Limits in *Provisioning and Administering Oracle Integration 3*.

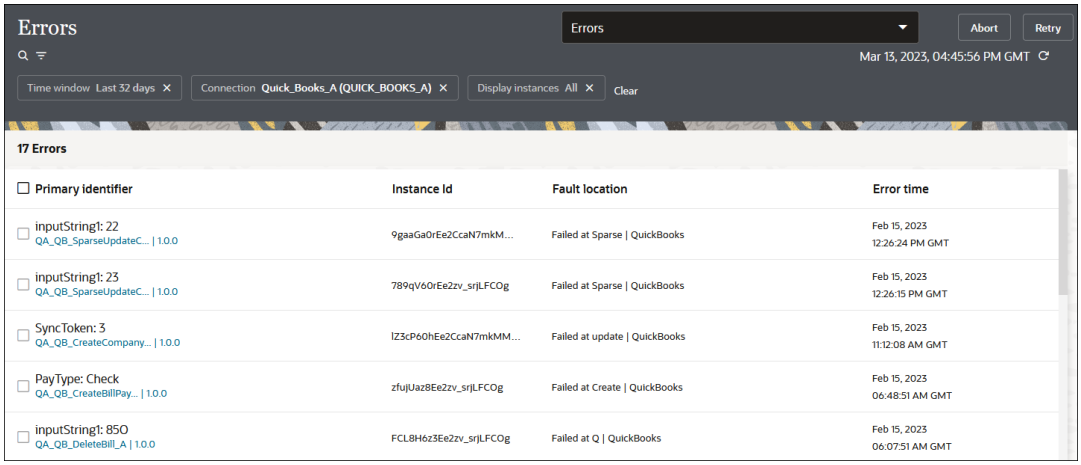
Error instances that are resubmitted and successfully resolved are removed from the error list. If an instance is resubmitted and is in progress, a state of `In Progress` is displayed in the list. During this state, additional resubmissions of this error instance are not permitted.

> **Note**
>
> Do *not* discard a message that you want to resubmit. A discarded message cannot be resubmitted.

1. In the navigation pane, click **Observability**, then **Errors**.

Copyright © 2022, 2026, Oracle and/or its affiliates.

2. Click **Filter**, then select the time period or retention period during which to search for integration errors.



3. Resubmit errors in either of two ways:

   a. Hover over the row of the specific error to resubmit.

   b. Click **Resubmit** .

      The Resubmit instance panel opens.

   c. Select the tracing level to use for the resubmitted instance. When resubmitting an errored integration instance, you can change the level of tracing (production, audit, or debug) for just this specific resubmitted instance. Increasing the tracing level (for example, setting it to **Debug**) enables you to view more specific information about errors in the activity stream and downloaded logs. This setting applies only to this instance and does *not* change the tracing level set when the integration was activated.

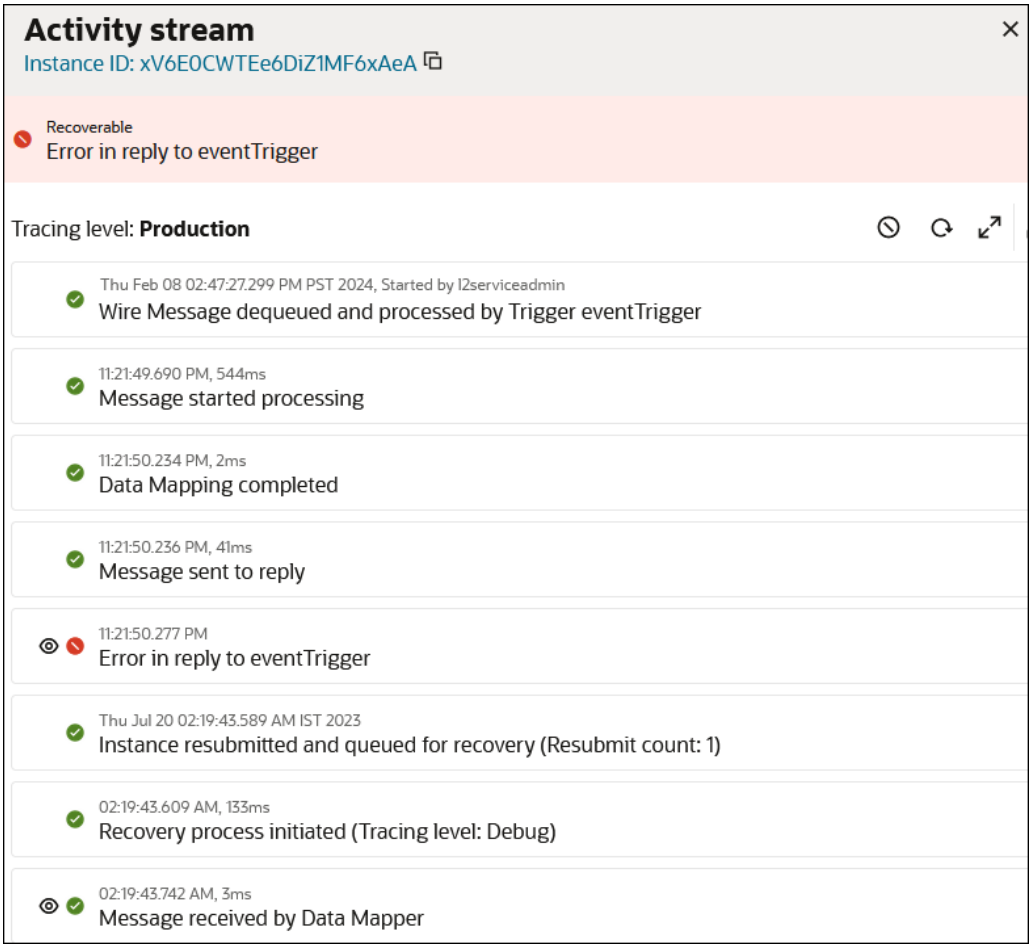
      - **Production**
      - **Audit**
      - **Debug**

      The tracing options provide different levels of information logging. See Activate an Integration.

      When the instance resubmission completes, the activity stream provides details. The tracing level shown at the top represents the level set during integration activation. To see details about the tracing level set for this specific resubmission, scroll down to this message:

      ```
      Recovery process initiated (Tracing level: Debug)
      ```


Copyright © 2022, 2026, Oracle and/or its affiliates.



Or

a. Select the check boxes of errors to resubmit together. Only instances identified by the **Resubmit** icon that is visible when you hover over a row can be resubmitted.

b. Click **Resubmit** in the upper right corner.

> **Note**
>
> This method of message resubmission does not allow you to select the tracing level to use for the resubmitted instance. Instead, the tracing level selected during integration activation is used.

4. Click **Confirm** to resubmit the instance when prompted.
A Resubmission summary panel opens with details about the instance being resubmitted for processing. Click **Refresh** to see if it still appears on the Errors page before trying another resubmission.

   Instances aren't accepted for resubmission for the following reasons:

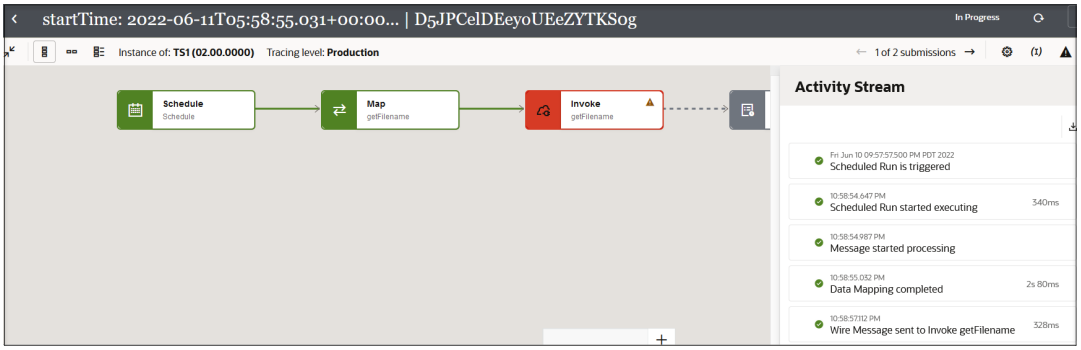
   - The instance is not recoverable.
   - The instance has reached the maximum number of retries limit.

Copyright © 2022, 2026, Oracle and/or its affiliates.

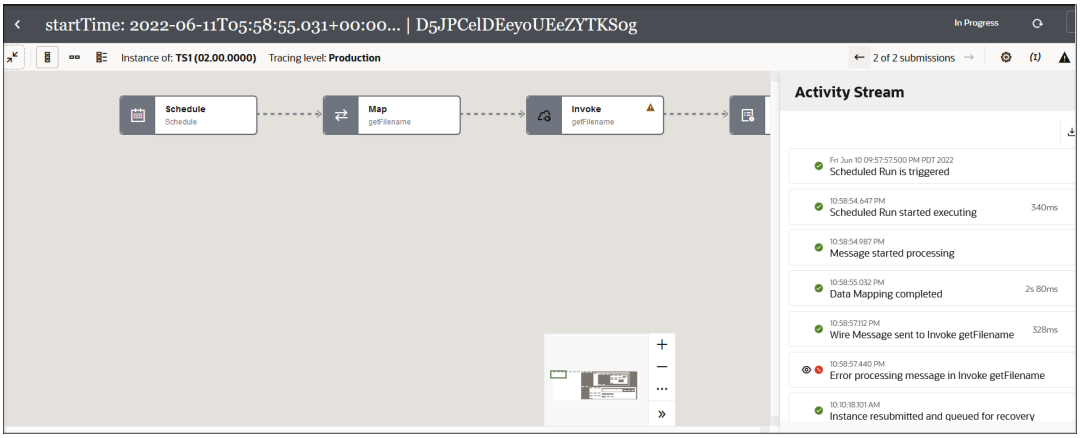
- The instance has not yet reached the deadline for recovery set internally for batch processing.

You can track all instance runs (both the initial run and all resubmissions) above the activity stream on the Instance Details page.

1. In the navigation pane, click **Observability**, then **Instances**.
2. Click the instance name.
3. Note the number of instance runs above the Activity Stream panel. The number 1 indicates that this is the first run of the integration.
   <- 1 of 2 submissions ->



4. Click **->** to access tracking details for the first resubmission of this instance (that is, the second run of this instance).
   The status for the first resubmitted instance is displayed.



5. Click the **<-** to return to the tracking details for the first run of this instance.

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 an aborted state on the Instances 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.

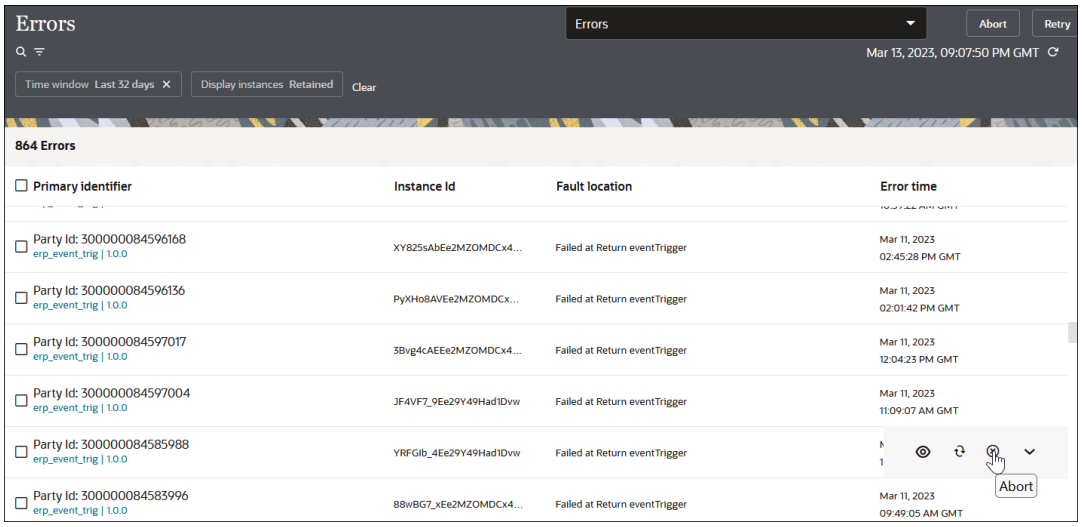
To discard errors:

1. In the navigation pane, click **Observability**, then **Errors**.

Copyright © 2022, 2026, Oracle and/or its affiliates.

2. Click **Filter**, then select the time period or retention period during which to search for integration errors.

   You can also specify a custom time range for which to search for errors.



3. Discard errors in either of two ways:
   a. Select the check boxes of individual errors to discard. Only instances identified by the **Abort** icon that is visible when you hover over a row can be discarded.
   b. Click **Abort**.

   or

   a. Go to the row of the specific error to discard.
   b. Click **Abort** .
4. Click **Confirm** to abort the instance when prompted.

   A message is displayed at the top of the page:

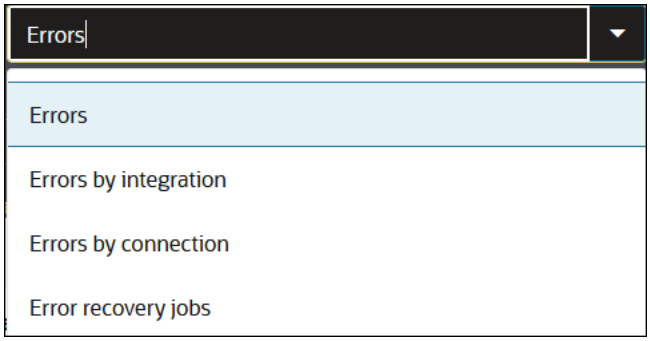
   ```
   Instance ID "number" is submitted for abort. Click on the refresh button
   to see if this instance still appears on the errors list before trying to
   abort again.
   ```

View the Status of Message Recovery

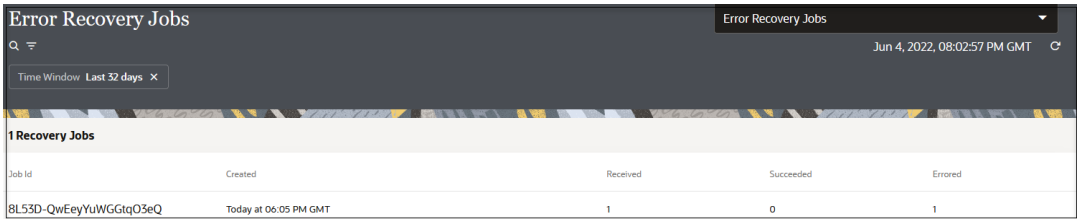
You can search for and view the status of failed messages that have been submitted for recovery on the Errors page.

1. In the navigation pane, click **Observability**, then **Errors**.
2. Submit a failed message for recovery.
3. From the drop-down list at the top right, select **Error Recovery Jobs**.

Copyright © 2022, 2026, Oracle and/or its affiliates.



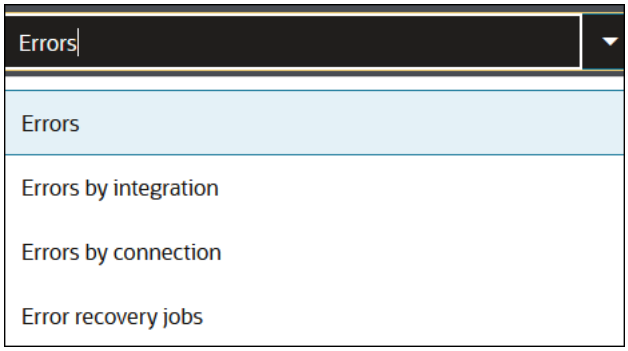
4. The Error Recovery Jobs page is displayed for viewing the status of message recovery.
   - **Received**: Recovery is still in progress.
   - **Succeeded**: Recovery succeeded.
   - **Errored**: Recovery failed.



View Errors By Integration and Connection

You can view errors by integration and connection on the Errors page.

1. In the navigation pane, click **Observability**, then **Errors**.
2. From the drop-down list at the top, select an option:
   - To view errors by integration, select **Errors by integration**.
   - To view errors by connection, select **Errors by connection**.



3. If you selected **Errors by integration** or **Errors by connection**, click **Filter** to show errors alphabetically by name or by error count in descending order.

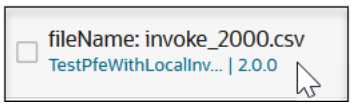
Copyright © 2022, 2026, Oracle and/or its affiliates.

4. Click the error count for an integration or connection to invoke the Errors page and view all errored-out instances for the given integration or connection.
5. Hover over an integration:
   - Click **Retry** to resubmit errors.
   - Click **Abort** to discard errors.

View the Business Identifiers Used in Failed Integration Instances

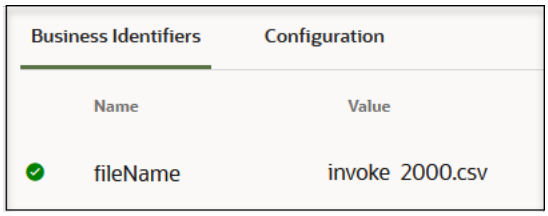
You can view the business identifiers used in failed integration instances.

1. In the navigation pane, click **Observability**, then **Errors**.
2. Click the business identifier of the failed integration instance.



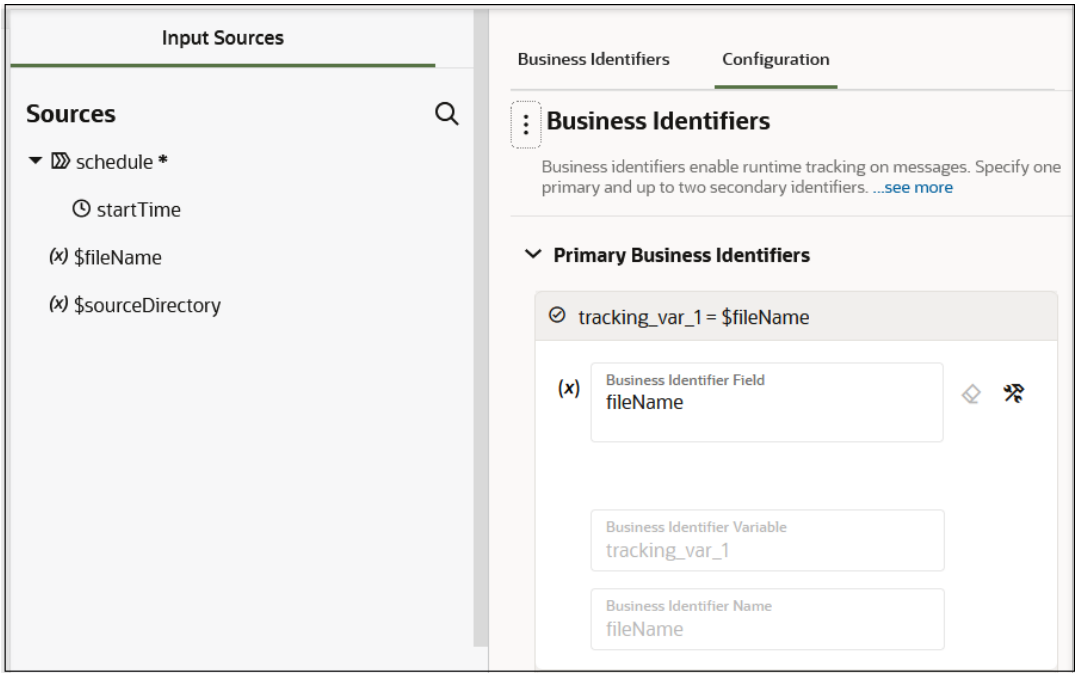
The integration instance and the Activity Stream panel are displayed.

3. Click **Business Identifiers** above the activity stream to display all the defined business identifiers and values for the integration.



4. Click **Configuration**.
5. Click **Actions** to show the input sources for the business identifiers.

Copyright © 2022, 2026, Oracle and/or its affiliates.



See Assign Business Identifiers and Track Integration Instances.

Track Integration Instances

You can filter and track the status of integration instances on the Instances page through business identifiers. You can also view the message payload of an instance you are tracking.

Topics:

- Filter the Display of Integration Instances
- Search for Business Identifier Values
- Track the Status of Integration Instances
- Determine Why an Integration Instance Stopped Running
- Replay Integration Instances

You assign business identifiers during integration design time. At least one business identifier is required. See Assign Business Identifiers for Tracking Fields in Messages.

Filter the Display of Integration Instances

You can filter the display of integration instances on the Instances page during runtime.

1. In the navigation pane, click **Observability**, then **Instances**.
2. Click **Filter** to filter the display of instances by the following:
   - Last updated or creation time
   - Time window
   - Minimum and maximum duration of instances in milliseconds
     For example, enter the minimum and maximum duration values of `200` and `400` milliseconds, respectively, to show all instances that completed in that time window. If

Copyright © 2022, 2026, Oracle and/or its affiliates.

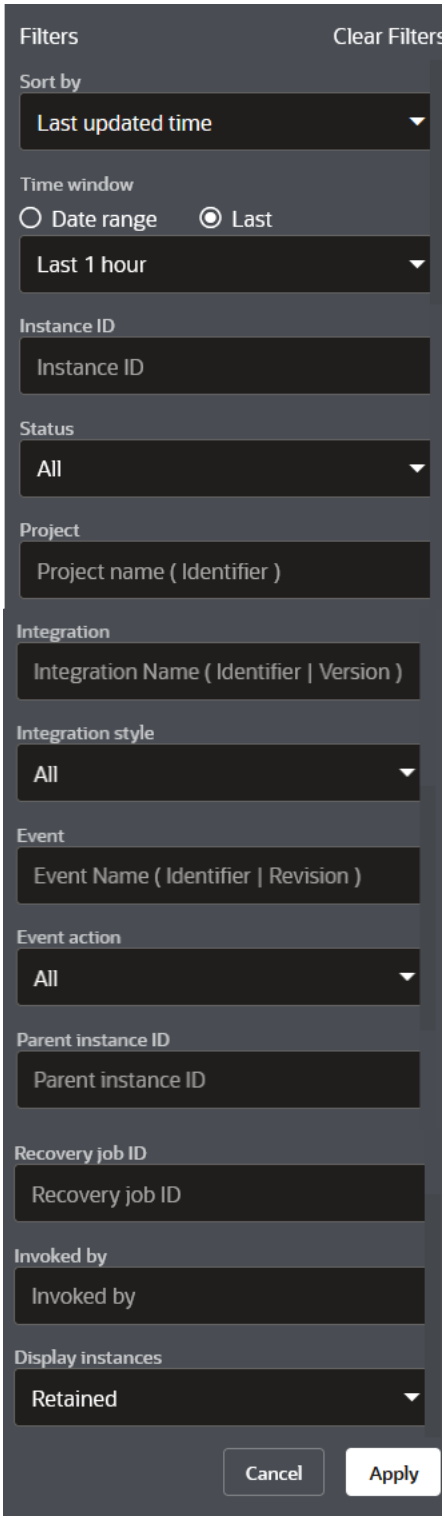
only a maximum duration is specified, then all instances less than that duration are returned. If only a minumum duration is specified, then all instances greater than that duration are returned.

- Instance ID
- Status
- Project name
- Integration name
  Begin typing an integration name. Do not use copy and paste. The filter is refreshed to display fields for the defined secondary business identifiers.

  You can now begin typing the values of the business identifiers to narrow down the search. This input follows the same rules (for example, for exact matches and phrase matches) as described in Search for Primary and Secondary Business Identifier Values.
- Integration style (all, application, schedule, or event)
- Event name
- Instances published and/or subscribed to that event name
- Parent instance ID
- Recovery job ID
- Integration that invoked the instance
- Instance state (all, retained, purged, replayed, or replayable).

You can enter variable values with special characters or variable values of multiple words with white space in double quotes in the **Search** field (for example, `&filename` or `"file tracking"`).

Copyright © 2022, 2026, Oracle and/or its affiliates.



3. Click **Date Range** to specify a custom time range with which to filter the search for instances with the **From** and **To** buttons. 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 Instances page to the Monitor Integrations page or Errors page).

Copyright © 2022, 2026, Oracle and/or its affiliates.

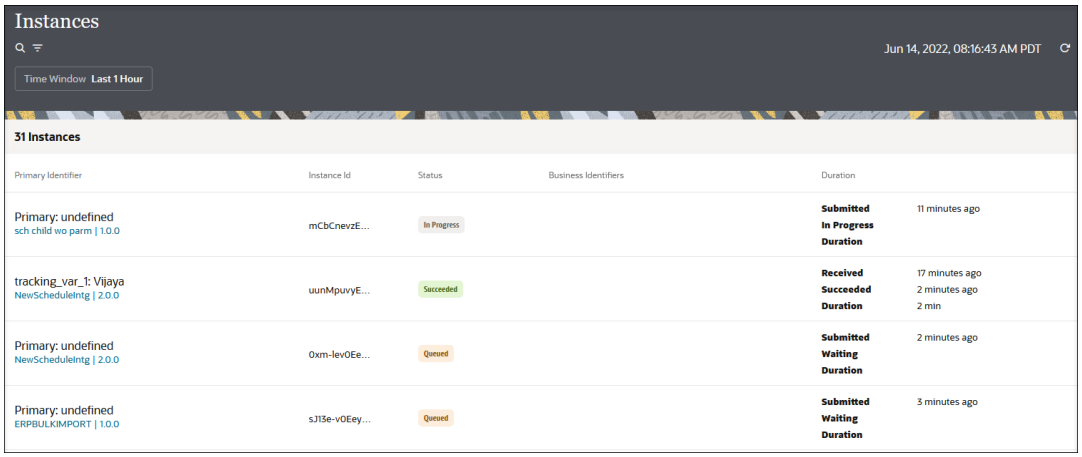
> **Note**
>
> The time range specified must be within the retention period. Data beyond the retention period is not retained. If the selected time range is beyond the retention period, only available data is displayed.

Search for Business Identifier Values

You can search for primary and secondary business identifier values on the Instances page during runtime. You can perform simple searches, phrase match searches, and exact match searches. This section provides examples of performing each search type.

1. In the navigation pane, click **Observability**, then **Instances**.
2. Use the **Search** field to search for values across both primary and secondary business identifiers.
   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 canceled instances, the running time does *not* point to the actual time the canceled (faulted) instance was running. Instead, it consists of the total time span between when the instance began and when the instance was manually canceled.



The **Search** field provides many capabilities:

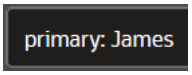
- Search for Primary and Secondary Business Identifier Values
- Perform a Simple Search
- Perform a Phrase Match Search
- Perform an Exact Match Search

Copyright © 2022, 2026, Oracle and/or its affiliates.

> **Note**
>
> - Partial word searches are not supported. For example, assume one business identifier value is `New York` and another is `New Orleans` and both belong to different instance IDs. If you enter `New` as your search criterion, both instances are displayed because `New` and `York` are considered separate words. White space is considered a word separator.
> - Underscores are not considered word separators. For example if the values are `New_York` and `New_Orleans`, and you enter `New` as the search criterion, neither instance is displayed because `New_York` is considered a complete word and partial word matches/searches are not supported. This is the case only with underscores. All other special characters (for example, periods, commas, and others) are considered word separators.

Search for Primary and Secondary Business Identifier Values

1. Enter a value, and press **Enter**. For example, assume the primary value is `James`. In the search results, any instances that have a primary value of **James** are returned. Any instances with a secondary value that includes **James** (for example, **Mark.James@asc.com**) are also returned.
2. Search only for primary business identifier values by entering the value as `primary: value` or `Primary: value` in the **Search** field.



> **Note**
>
> - If you enter a primary business identifier in the **Search** field, but do not press **Enter**, then select a value from the time period drop-down list, note that the instances are filtered considering the string entered in the **Search** field, even though **Enter** was not pressed. This is the expected behavior and is true for other landing pages in Oracle Integration.
> - The search facility on the Instances page is case sensitive.

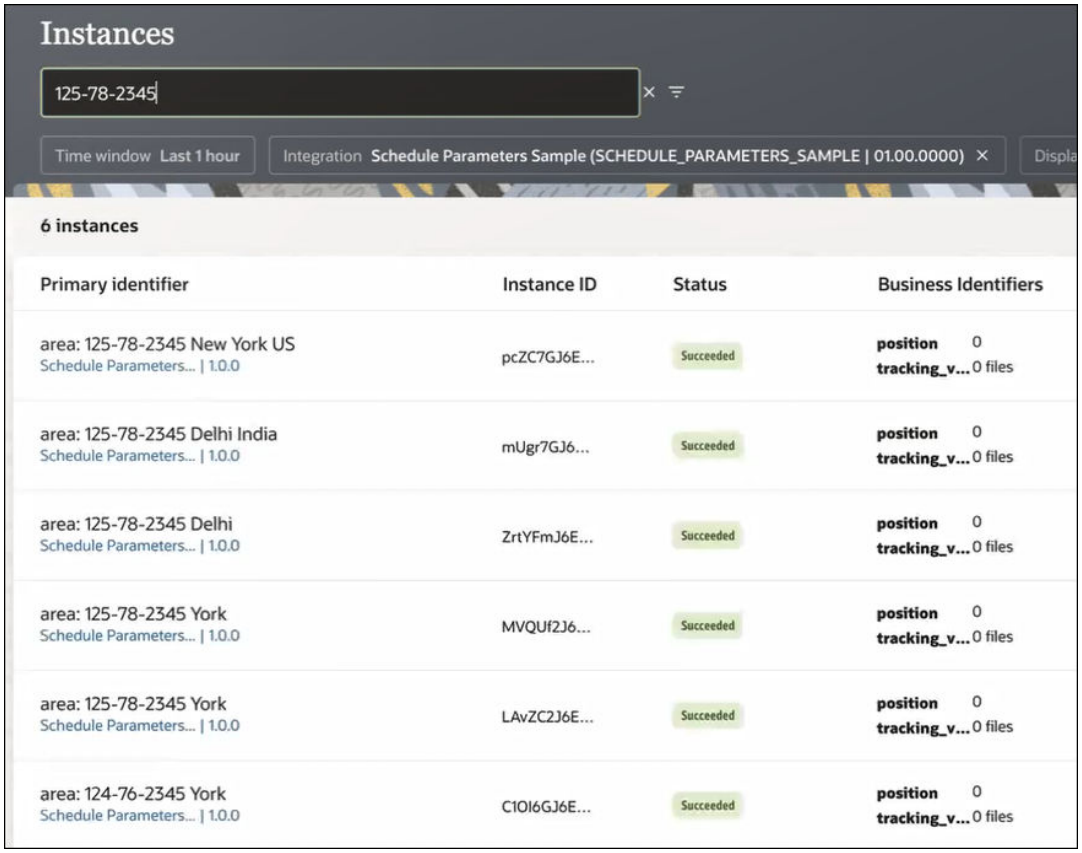
Perform a Simple Search

1. Enter a value in the **Search** field, and press **Enter**. For example:

   ```
   125-78-2345
   ```

   This returns all instances that include part of that business identifier value.

Copyright © 2022, 2026, Oracle and/or its affiliates.

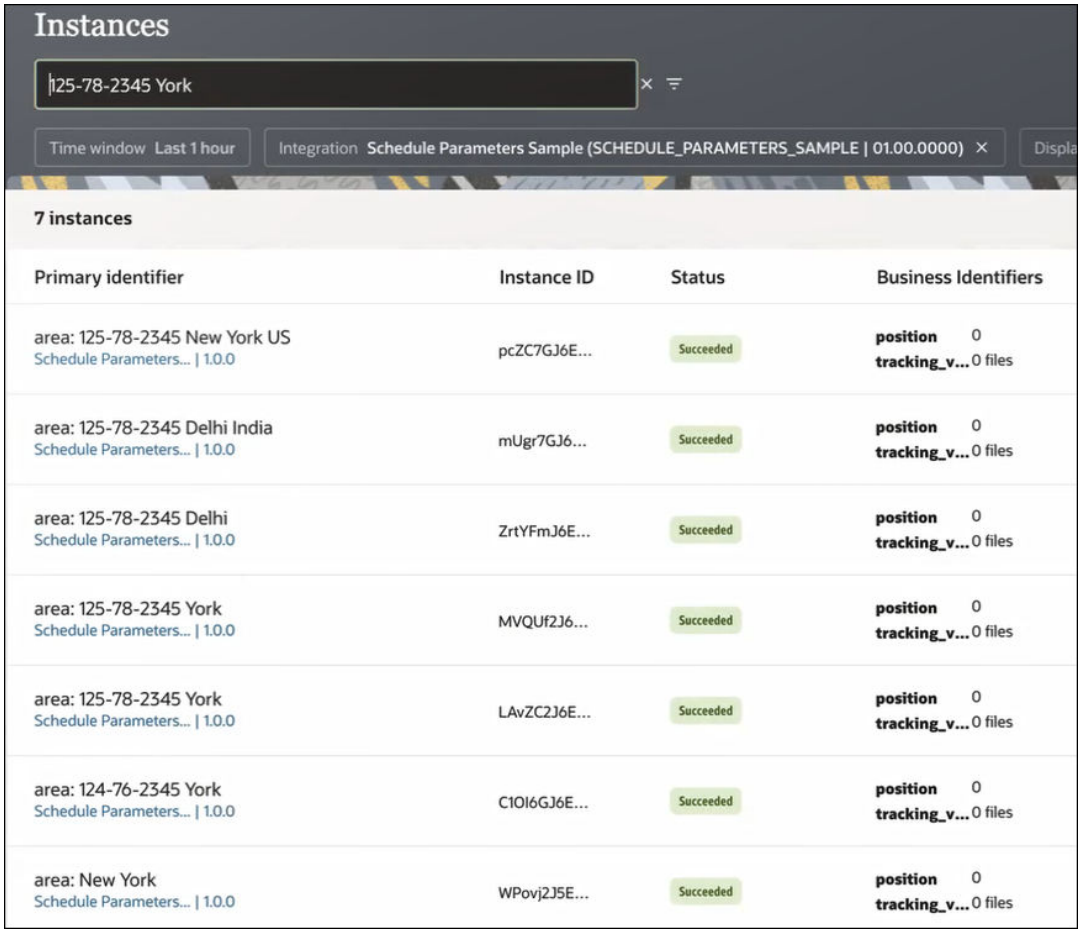


2. Enter a value with white space in the **Search** field, and press **Enter**. For example:

   `125-78-2345 York`

   Note that each individual part of the value is treated as a separate value (`125`, `78`, `2345`, and `York` are all considered separate). Any business identifier value found containing at least one of these values is returned. A simple search is similar to a web browser search.

Copyright © 2022, 2026, Oracle and/or its affiliates.



Perform a Phrase Match Search

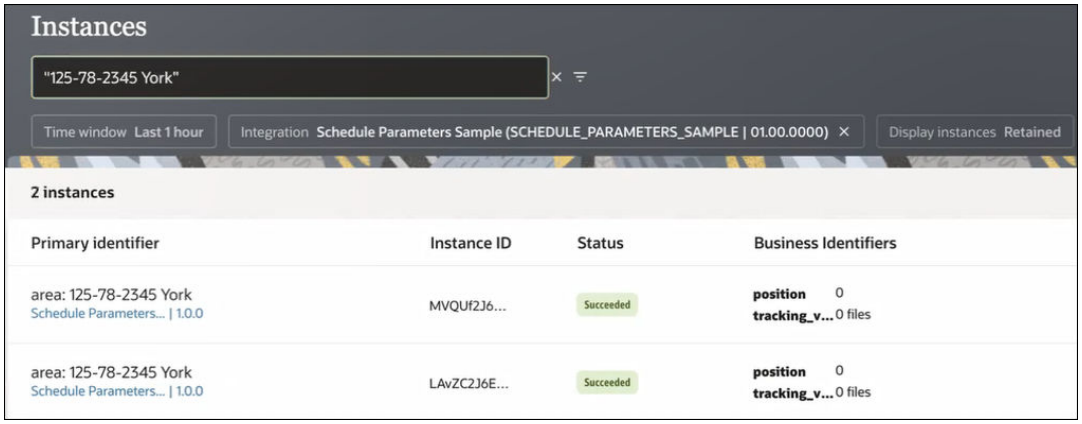
You can perform a phrase match search by enclosing your value in double quotes. Phrase matches are useful when you need to search for primary and secondary business identifier values containing special characters and white spaces.

1. Enter a value in the **Search** field, and press **Enter**. For example:

   ```
   "125-78-2345 York"
   ```

2. Note that exact values are returned. Without the double quotes, each individual part would be considered a separate searchable value: `125`, `78`, `2345`, and `York`.

Copyright © 2022, 2026, Oracle and/or its affiliates.

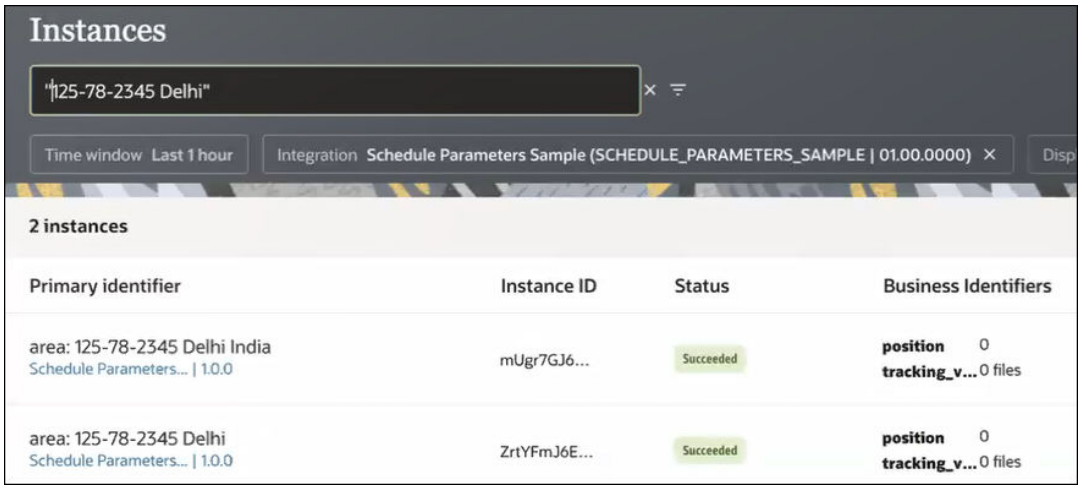


Note that exact phrase match searches can also return the exact match, plus any additional values at the beginning or end of the enclosed value. Consider the following example.

3. Enter a value in the **Search** field, and press **Enter**. For example:

   ```
   "125-78-2345 Delhi"
   ```

   This returns any value that matches, plus any additional values. In this case, both `125-78-2345 Delhi` and `125-78-2345 Delhi India` are returned.



Perform an Exact Match Search

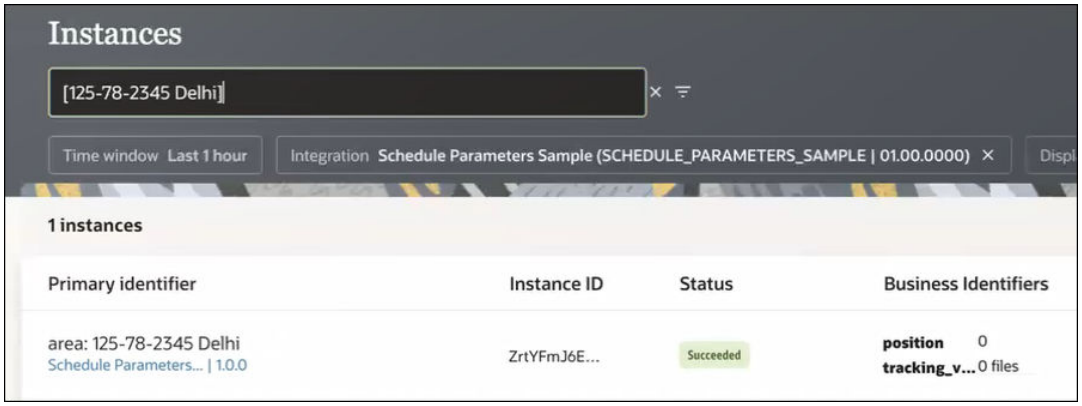
You can return an exact match search by enclosing the business identifier value in brackets ([and]). This search returns only exact match values.

1. Enter a value in brackets in the **Search** field, and press **Enter**. For example:

   ```
   [125-78-2345 Delhi]
   ```

2. View the results. Only exact matches are returned.

Copyright © 2022, 2026, Oracle and/or its affiliates.



Track the Status of Integration Instances

You can track fields in messages on which you have defined business identifiers on the Instances page during runtime. These fields are available for tracking on the Instances page because you defined at least one business identifier in the Business Identifiers panel during design time.

1. In the navigation pane, click **Observability**, then **Instances**.
2. Perform the following tasks:
   - View the Slowest Running Activities in an Integration Instance
   - Discard an Instance on the Instances Page
   - View the Status of Future Schedule Integrations
   - View the Activity Stream on the Instances Page
   - View the Child Instances of a Parent Instance
   - View the Activity Stream of a Child Integration Separately from the Parent Integration
   - View the Display of HTTP Headers Separate from the Payload
   - View File Metadata for Large Payloads in the Activity Stream
   - View the Activity Stream for Adapters Created with the Rapid Adapter Builder

You can also view the activity stream, business identifiers and values, and message payload in failed instances. See Manage Errors.

View the Slowest Running Activities in an Integration Instance

To diagnose performance issues, you can view the activities in the activity stream by their elapsed time. This ability ensures that the most time-consuming activities show first in the list and the best performing activities appear at the bottom. This functionality provides you with an immediate performance view and eliminates the time-consuming task of clicking each individual message or a large number of iterations in a looping action to identify time-consuming activities.

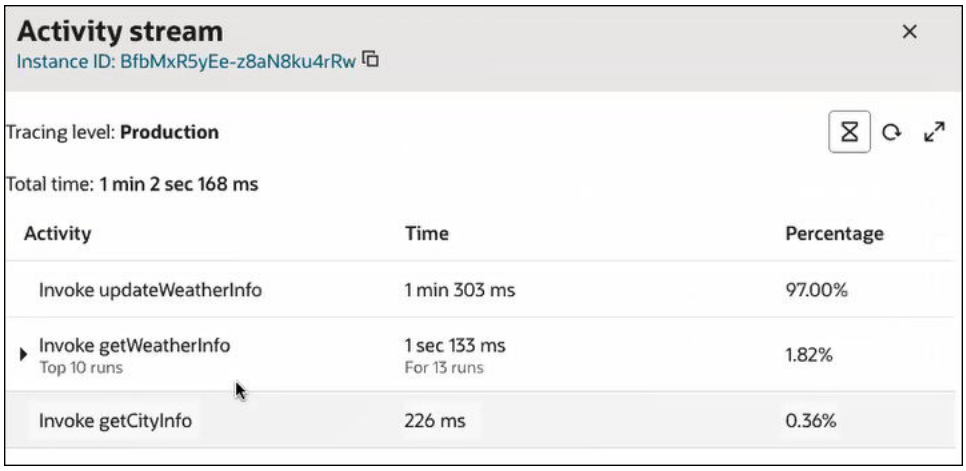
Understand the following behavior:

- The duration is *not* shown for all activities. The duration for an activity is the time from the beginning to the end of activity execution. Not all activities have both times (for example, a trigger or a throw action) because both are not applicable.

Copyright © 2022, 2026, Oracle and/or its affiliates.

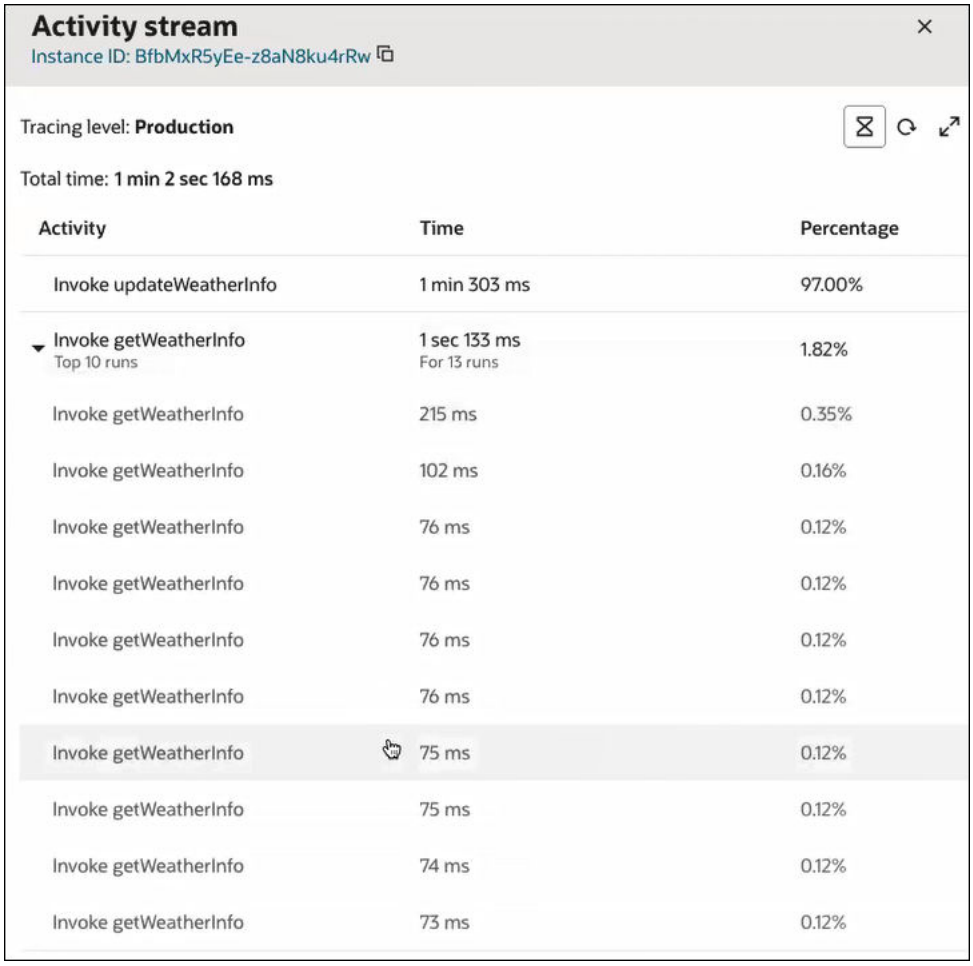
- The duration may not be available even for activities that have both times (such as a map action). This is because availability of these times depends on the integration tracing level. For example, an invoke action always appears, but other actions appear only when the instance is executed with the tracing level set to debug.
- Only the top 10 slowest activities are shown in the table. Your integration may have many more actions.
- You can view the overall time a looping action (for example, a while action or a for-each action) took to execute all its iterations. You can also expand the looping action to view the following:
  - If your looping action had less than 10 iterations, all iterations and their times to execute are shown.
  - If your looping action has more than 10 iterations, only the ten slowest iterations and their times to execute are shown.

1. Go to the row of the integration instance to view. The **Duration** column shows the time that an instance took to execute.
2. Click **View Details** .
   The activity stream is displayed.
3. Click **View slowest activities** .
   The total duration of the instance is displayed at the top. The activities that took the longest amount of time to execute are displayed in a table. The final column shows the percentage of time each activity took to execute based on the total duration of the instance. For this example, **getWeatherInfo** included a looping action that iterated 13 times.



4. Expand the looping action (**getWeatherInfo**) to view the iterations. For this example, the looping action was executed 13 times. The 10 slowest-running iterations are shown.

Copyright © 2022, 2026, Oracle and/or its affiliates.



5. Use this functionality to identify and debug slow running activities.
6. Click **View slowest activities** again to return to the activity stream. This link enables you to toggle back and forth.

Discard an Instance on the Instances Page

1. Hover over the row of the instance to discard, then select **Abort** .



This option is only visible under the following scenarios:

- In-progress instances of asynchronous integrations (both scheduled and nonscheduled).
- Waiting and paused instances of schedule integrations. Attempting to discard other types of instances results in an error message being displayed.

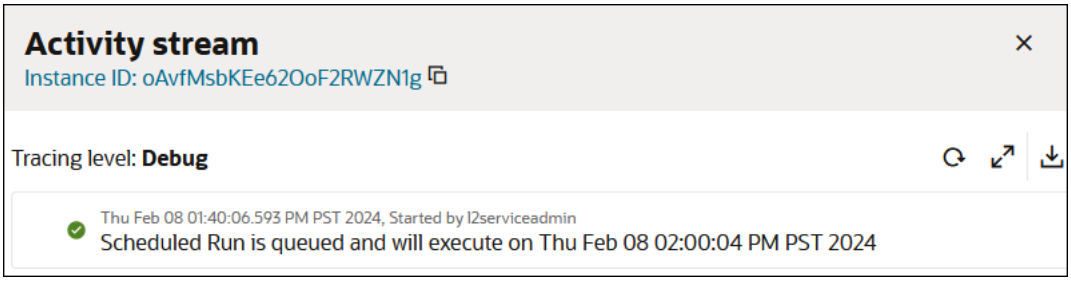
Copyright © 2022, 2026, Oracle and/or its affiliates.

View the Status of Future Schedule Integrations

Schedule integrations that have not yet run are displayed as **Queued** on the Instances page. The **Duration** column shows the label **Scheduled on** and the time at which the queued integration is scheduled to run (for this example, in four minutes).



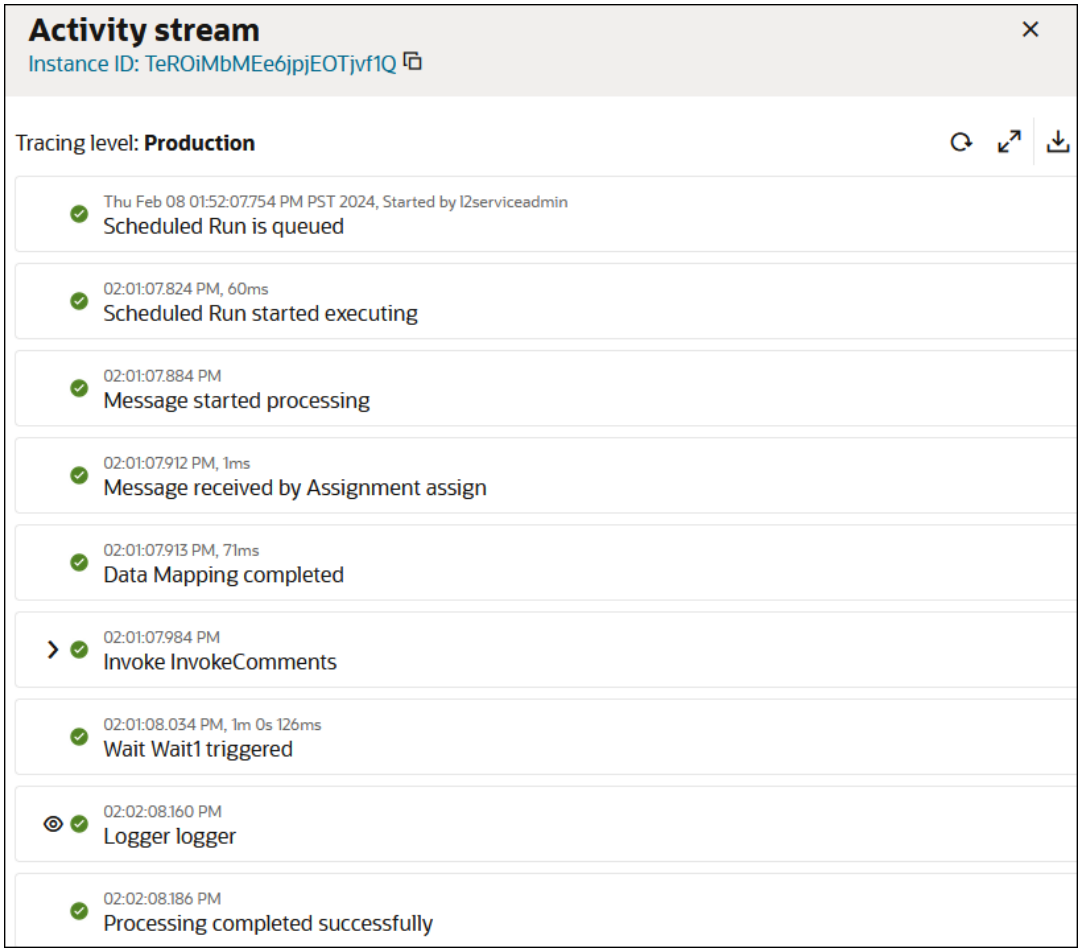
1. Hover over the queued integration in the **Duration** column to display the full date and time at which the queued integration is scheduled to run.
2. Click **View Details** or click the business identifier in the **Primary Identifier** column of a queued integration. The Activity Stream panel opens in either place and shows the time at which the queued integration is scheduled to run.



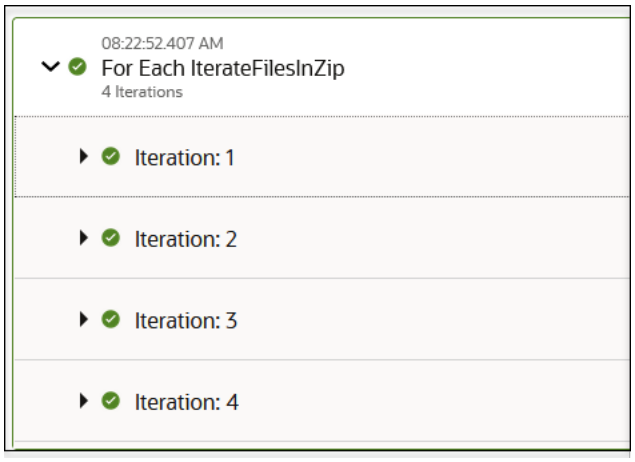
View the Activity Stream on the Instances Page

1. Hover over an instance, then select **View Details** to view the activity stream of the instance.
   The activity stream shows details about the movement of the message through the integration, including where any failures occurred. The message flow through each action is also shown.
2. Expand the invoke connection.
   Key child messages of the parent invoke connection are displayed.

Copyright © 2022, 2026, Oracle and/or its affiliates.



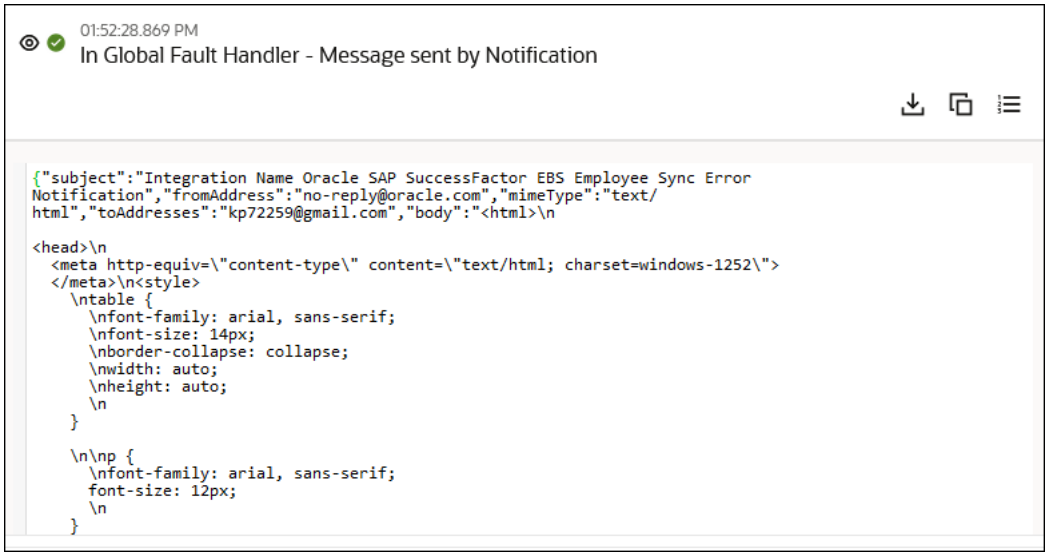
The date and time according to your user preferences are displayed. If your instance contains logic actions (for example, a while action or a for-each action with multiple iterations), each iteration is shown in the activity stream and can be expanded to show specific details about the message flow.



3. Click **Download** at the top of the activity stream to download the activity stream logs for the entire instance.

Copyright © 2022, 2026, Oracle and/or its affiliates.

4. Click **View Payload** to view the payload for that part of the integration flow.



If you view the message payload data in the activity stream of an integration instance in which the tracing level is set to debug or audit, then the time, your user name, and a message indicating that you viewed message payload data are written to the audit log. If the tracing level is set to production, your user name is not logged to the audit log because message payload data is not visible.

5. Perform tasks within the payload.

Element	Description
Summary	Click to view a summary of payload details.
Details	Click to view more specific payload details.
Download Payload	Click to download this part of the payload to your desktop.
Copy Payload	Click to copy this part of the payload for pasting into a document.
Toggle Line Numbers	Click to show or hide line numbers in the payload.

View the Child Instances of a Parent Instance

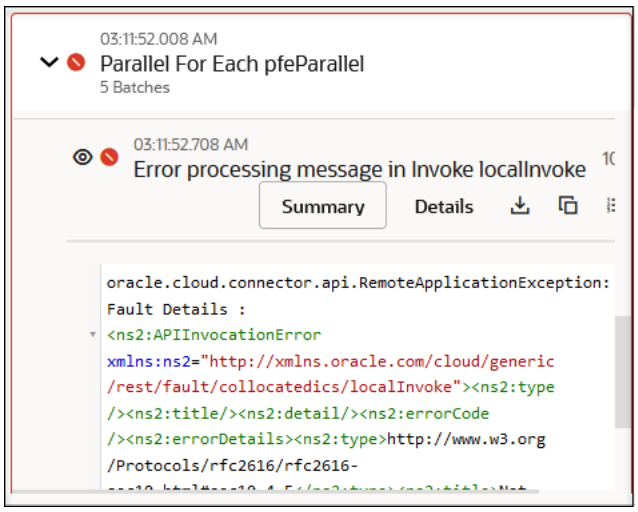
If your instance invokes child instances (for example, an integration action in your flow is invoking child integrations), you can view those instances on the Instances page.

1. Hover over a row and select **View Child Instances**.



You can also view invocations of child instances (local invokes) in the activity stream.

Copyright © 2022, 2026, Oracle and/or its affiliates.



View the Activity Stream of a Child Integration Separately from the Parent Integration

You can view a child integration being invoked by a parent integration in a separate activity stream. You can also view the separate activity streams for one integration that invokes another integration, which then invokes a child integration. Each activity stream is connected by a button that provides easy access.

See Invoke a Child Integration from a Parent Integration.

1. Hover over the instance of a parent integration that invokes a child integration.
2. Click **View Details** .
   The activity stream is displayed.
3. Expand the invoke connection.
   The **View child instance details** link is displayed. This is where the instance of the child integration is invoked.



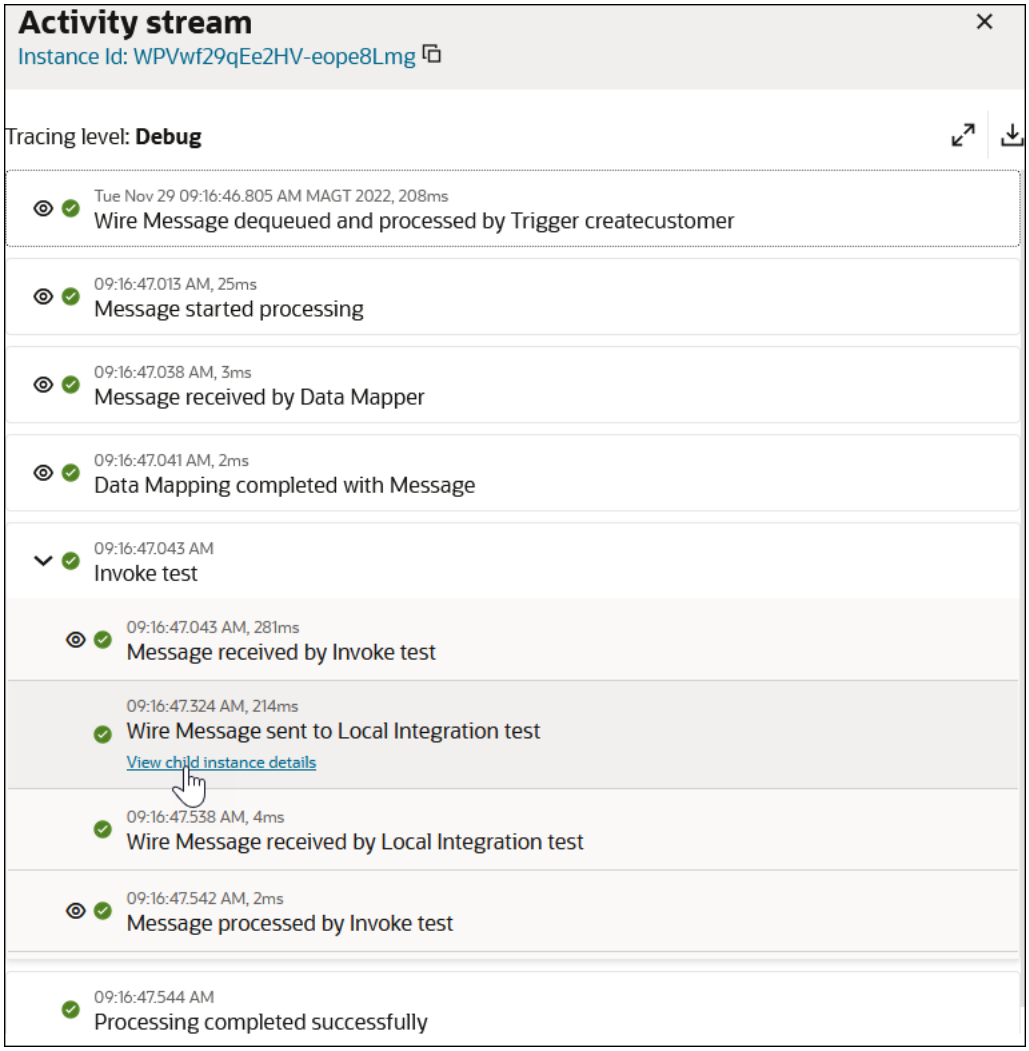
Copyright © 2022, 2026, Oracle and/or its affiliates.

4. Click **View child instance details** to access the instance of the activity stream for the child integration. Note that **Integration: Child** and the integration version number are also displayed after the instance ID at the top.



5. Click < to return to the instance of the activity stream for the parent integration.
6. Return to the Instances page and hover over the instance of an integration that invokes another integration, which then invokes a child integration.
7. Click **View Details** .
8. Expand the invoke connection.
9. Click **View child instance details** to access the instance of the activity stream for the integration being invoked.

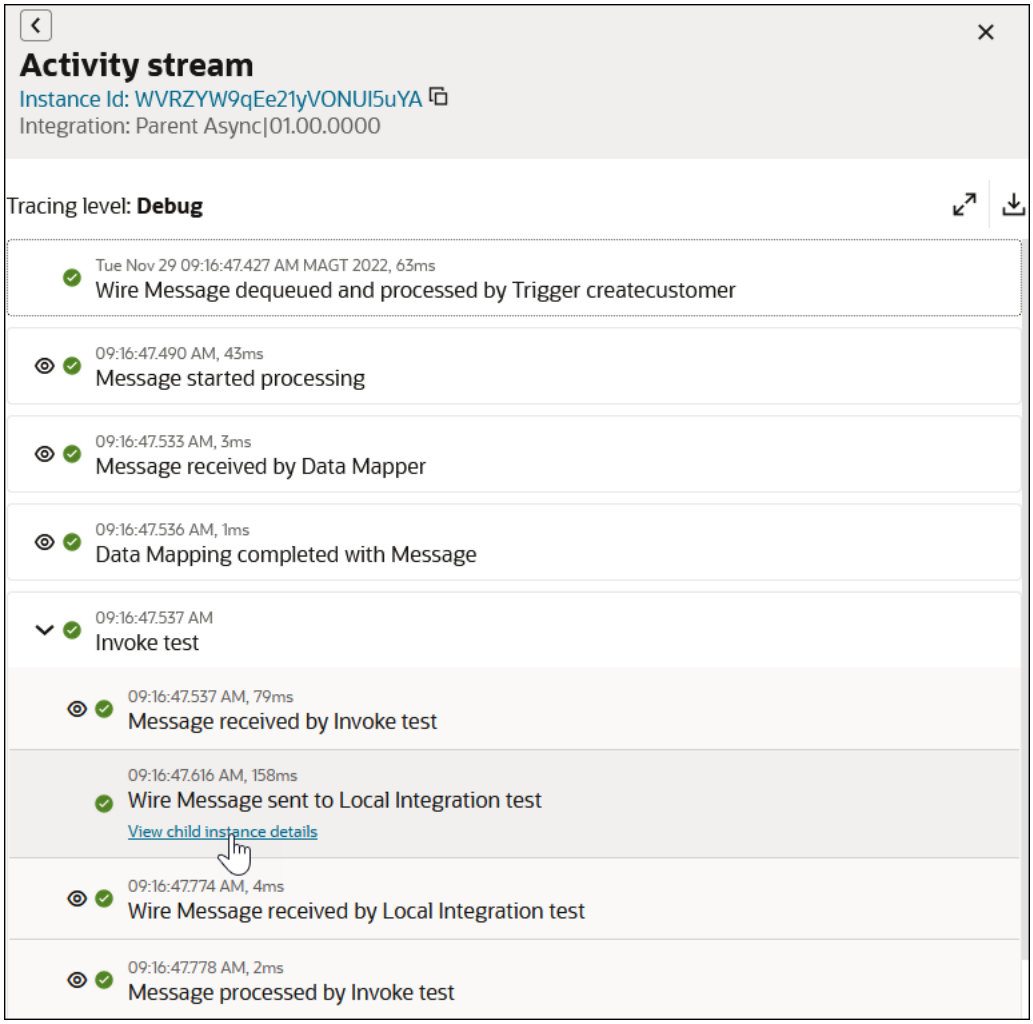
Copyright © 2022, 2026, Oracle and/or its affiliates.



Note that the parent name and the integration version number are displayed at the top.

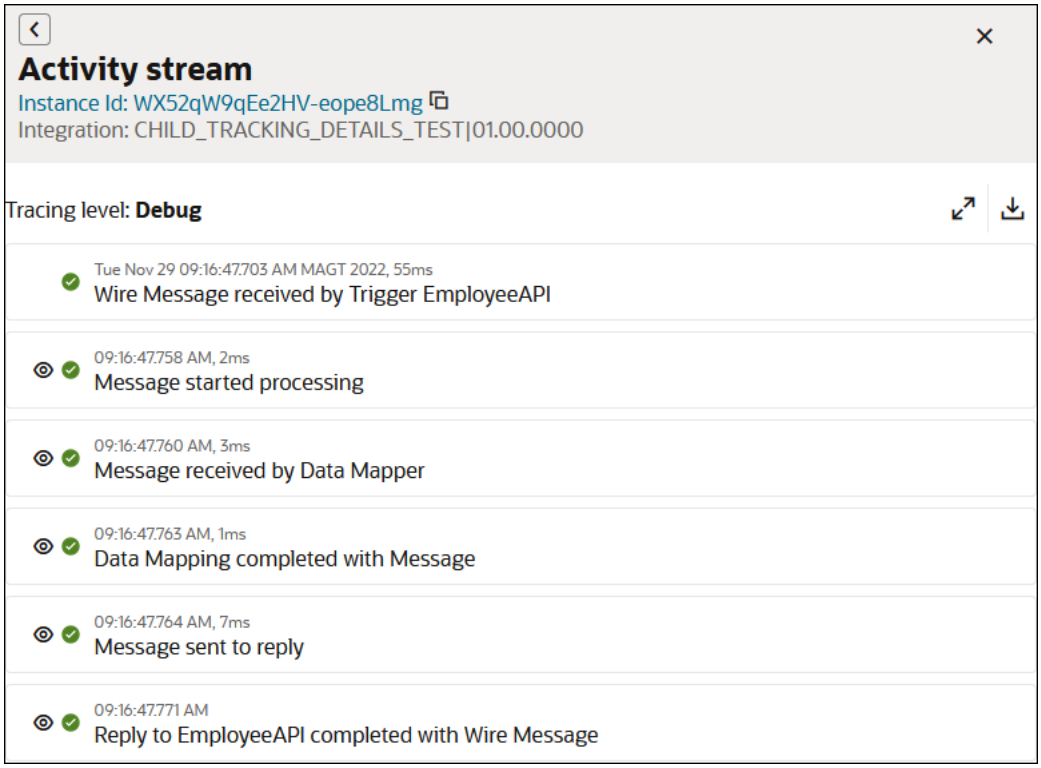
10. Expand the invoke connection.

Copyright © 2022, 2026, Oracle and/or its affiliates.



11. Click **View child instance details** to access the instance of the activity stream for the child integration being invoked.
    Note that the child name and the integration version number are displayed at the top.

Copyright © 2022, 2026, Oracle and/or its affiliates.



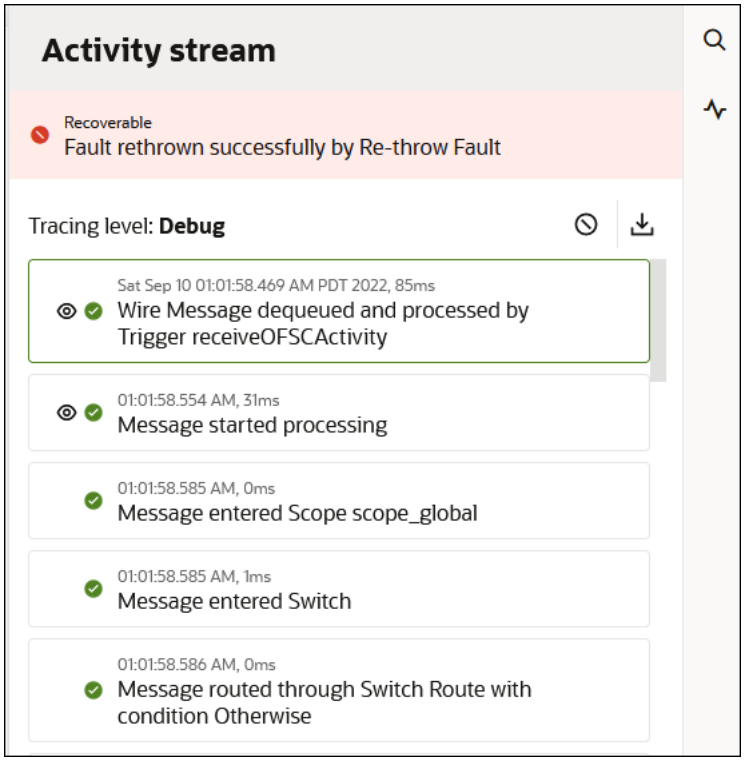
12. Click < twice to return to the instance of the activity stream for the top level integration.

View the Display of HTTP Headers Separate from the Payload

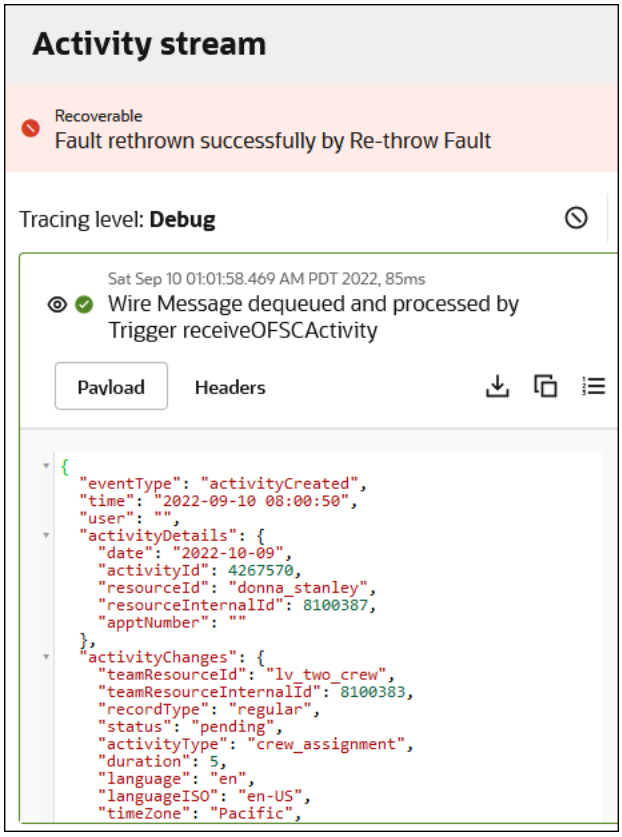
HTTP headers associated with a payload are displayed separately from the payload for ease of viewing in the activity stream.

1. Click **View Details** .

Copyright © 2022, 2026, Oracle and/or its affiliates.

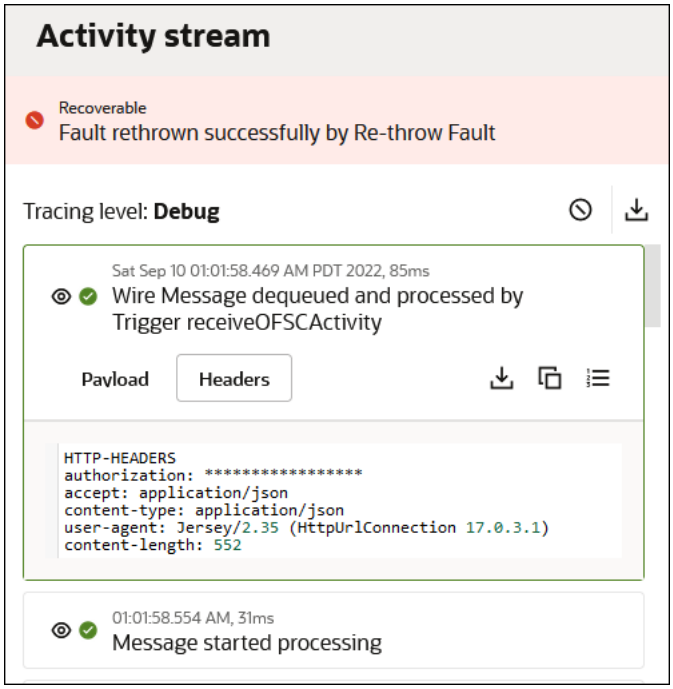


If HTTP headers are associated with the payload, two tabs are displayed: **Payload** and **Headers**.



Copyright © 2022, 2026, Oracle and/or its affiliates.

2. Click **Headers** to view details about the HTTP headers associated with the payload.



View File Metadata for Large Payloads in the Activity Stream

You can view file metadata for large payloads in the activity stream.

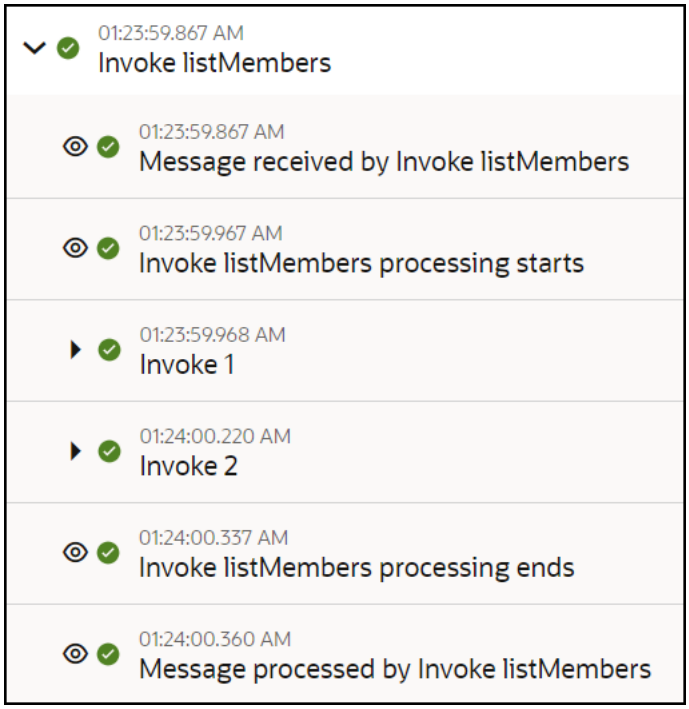
For example, if an integration performing inbound polling sends a CXML payload in GZIP format, the payload is shown in readable format in the activity stream. The payload size and media type are also shown.

View the Activity Stream for Adapters Created with the Rapid Adapter Builder

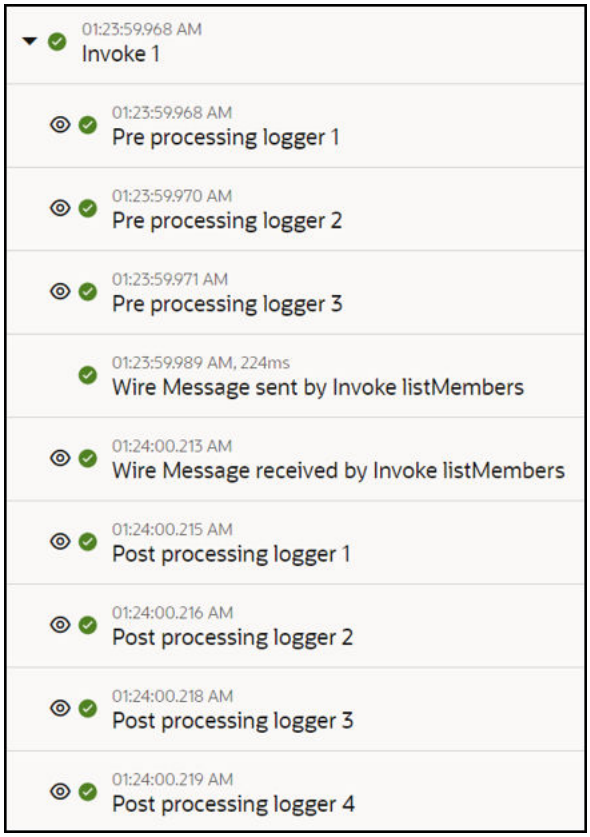
When the tracing level is set to **Debug**, you can view messages for the top level invoke connection and any invoke connections it calls. You can expand each invoke connection being called to view messages for preprocessing logs, wire messages, and postprocessing logs.

1. Click **View Details** for an instance that includes an adapter created with the Rapid Adapter Builder.
2. View the activity stream and note that a single invoke connection (**listMembers**) for an adapter created with the Rapid Adapter Builder is calling multiple invoke connections.

Copyright © 2022, 2026, Oracle and/or its affiliates.



3. Expand an invoke. Each invoke can have preprocessing and postprocessing loggers before and after the invocation. Wire messages sent and received by **listMembers** are also shown.



Copyright © 2022, 2026, Oracle and/or its affiliates.

4. Click **View Details** to view details about the payload and the messages being logged.

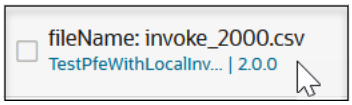
Track the Integration Flow on the Instance Details Page

The Instance Details page provides a graphical view of the instance flow and access to the activity stream.

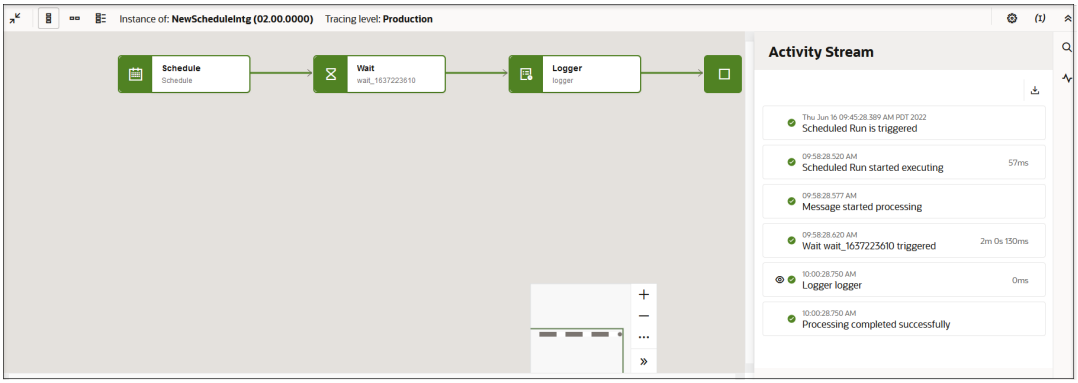
- View the Instance Elements and Activity Stream on the Instance Details Page
- View the Results of Iterations of Logic Actions on the Instance Details Page
- Access Information Through Inline Menus on the Instance Details Page

View the Instance Elements and Activity Stream on the Instance Details Page

1. Click a specific business identifier of an integration instance in the **Primary Identifier** column.



The integration flow of the instance and Activity Stream panel are displayed.



2. Click **Download** at the top of the activity stream to download the activity stream logs for the entire instance.
3. Click **View Payload** to view the payload for that part of the integration flow.
4. Perform tasks within the payload.

Element	Description
Summary	Click to view a summary of payload details.
Details	Click to view more specific payload details.
Download Payload	Click to download the payload to your desktop.
Copy Payload	Click to copy the payload for pasting into a document.

Copyright © 2022, 2026, Oracle and/or its affiliates.

Element	Description
Toggle Line Numbers	Click to show or hide line numbers in the payload.

5. Click **Activity Stream**

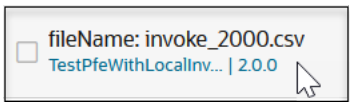


to close the activity stream.

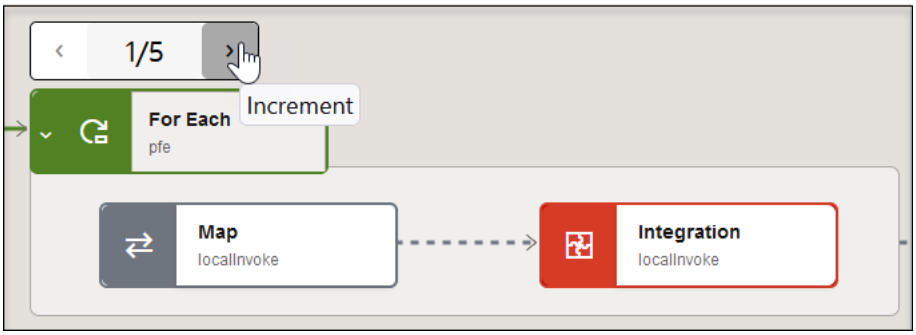
View the Results of Iterations of Logic Actions on the Instance Details Page

You can view the results of iterations of logic actions (such as a for-each, switch, while, and others) in the activity stream on the Instance Details page.

1. Click a specific business identifier of an integration instance in the **Primary Identifier** column.

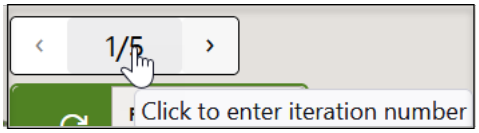


2. Go to the logic action in the integration (for this example, a for-each action is shown).
3. You can move to another iteration of the loop in either of two ways.
   a. Click **>** to access a view of the next iteration in the loop. For this example, there are five iterations in this for-each action.



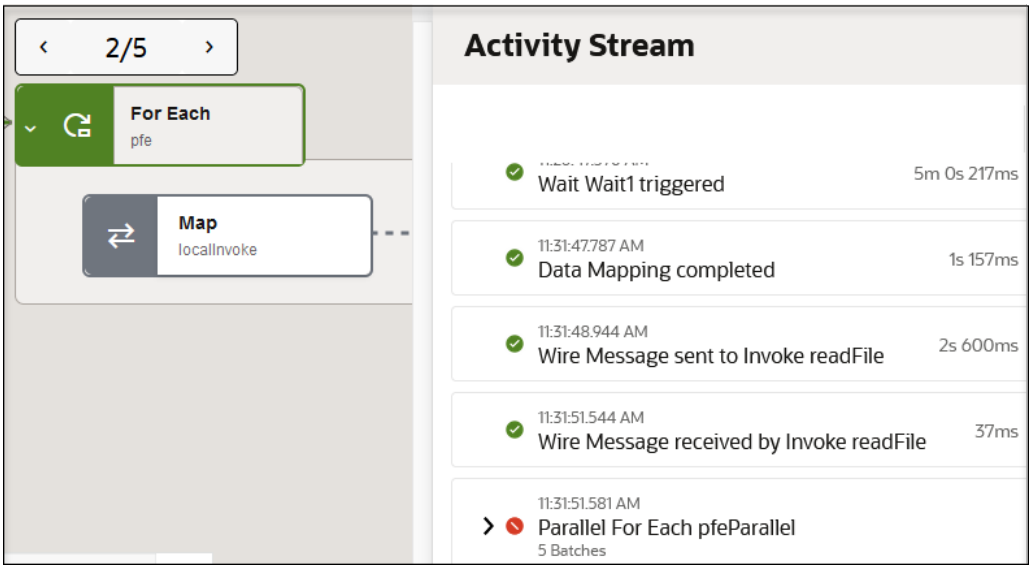
   or

   a. Hover over and click the iteration box of a logic action in the integration.



   b. Enter a specific value and click **OK** (for example, 5). This method is useful when the logic action has iterated many times.
   The activity stream shows the specified iteration of the logic action.

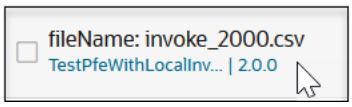
Copyright © 2022, 2026, Oracle and/or its affiliates.



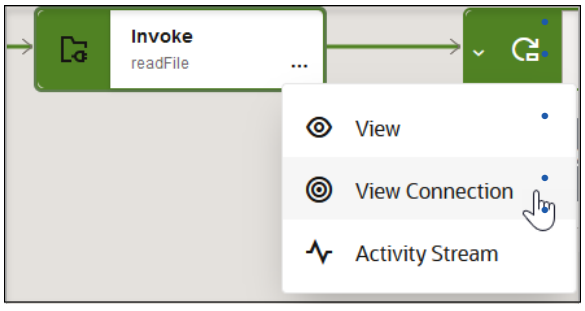
Access Information Through Inline Menus on the Instance Details Page

The inline menus of the triggers, invokes, and actions in the integration flow provide access to additional information.

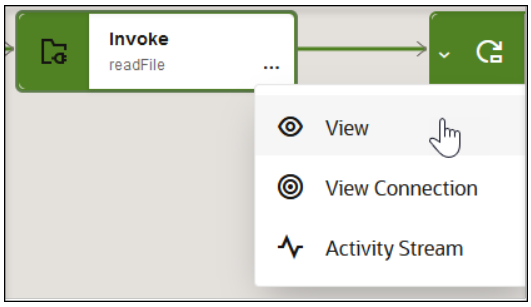
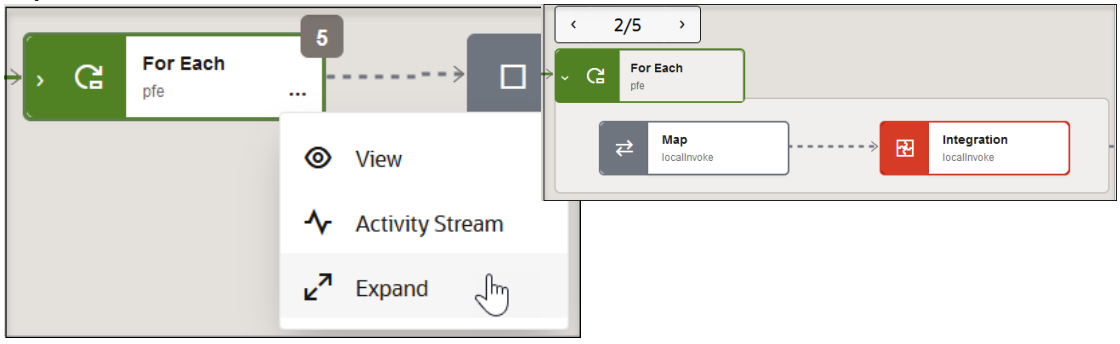
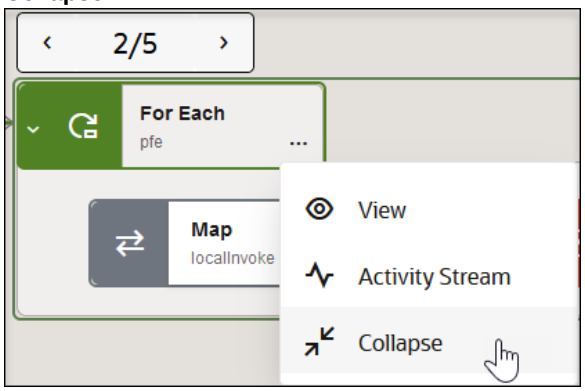
1. Click a specific business identifier of an integration instance in the **Primary Identifier** column.



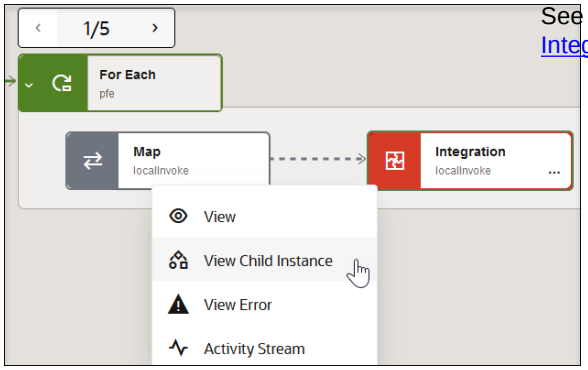
2. Select from a set of options. The options that appear are based on the element you selected.

Option	Description
For trigger or invoke connections, click **Actions** ⋯, then **View Connection**. 	Displays configuration details for the connection: • Selected roles for the connection (trigger, invoke, or trigger and invoke) • Identifier value and date last updated • Integrations using the connection (click to view the list) • Connection and security properties defined on the Connections page • Status of agent group association • Configuration progress (the connection properties and security policy must be completely configured and tested to be listed as 100%) The **Actions** ⋯ menu in the upper right corner enables you to edit or delete the connection (if not currently used in an activated instance).

Copyright © 2022, 2026, Oracle and/or its affiliates.

Option	Description
For trigger, action, and invoke connections, click **Actions** ⋮, then **View**. 	Displays the following details: • For schedules, shows the configured scheduled parameters. • For triggers and invokes, the Adapter Endpoint Configuration Wizard is invoked to show the configuration settings. • For maps, the mapper is invoked to show the mappings. • For other actions, the specific configurations are shown.
For logic actions (such as a scope, for-each, while, wait, and switch), click **Actions** ⋮, then **Expand**. 	Displays each action and invoke within the logic action.
For logic actions (such as a scope, for-each, while, wait, and switch), click **Actions** ⋮, then **Collapse**. 	Displays the collapsed logic action.

Copyright © 2022, 2026, Oracle and/or its affiliates.

Option	Description
For parent integrations that call child integrations, click **Actions** ⋮, then **View Child Instances**. 	Displays the child integration called by the parent. See Invoke a Child Integration from a Parent Integration.

Determine Why an Integration Instance Stopped Running

When a scheduled or asynchronous integration instance stops running, the reason for the termination appears in the activity stream.

The following sections describe the message displayed in the activity stream based on the type of instance termination.

- Someone Stopped an Asynchronous Integration Instance
- Someone Deactivated an Asynchronous Integration While Instances Were Running
- Someone Stopped a Running Schedule of an Integration
- Oracle Integration Terminated an Asynchronous or Schedule Instance That Exceeded the Processing Deadline
- Oracle Integration Terminated a Stuck Scheduled Instance

Someone Stopped an Asynchronous Integration Instance

1. In the navigation pane, click **Observability**, then **Instances**.
2. Hover over the row of the active integration instance to terminate, and select **Abort** .
3. Click **Confirm** when prompted.
4. Click **View Details** .
   The Activity Stream panel opens. A red icon and the reason for the termination are displayed.

   `Instance aborted by` *`username`*

Someone Deactivated an Asynchronous Integration While Instances Were Running

1. In the navigation pane, click **Design**, then **Integrations**.
2. Hover over the row of the integration instance to deactivate, and select .
3. In the navigation pane, click **Observability**, then **Instances**.
4. Click **View Details** .

Copyright © 2022, 2026, Oracle and/or its affiliates.

The Activity Stream panel opens. A red icon and the reason for the termination are displayed.

```
Instance aborted due to deactivation of integration by username
```

Someone Stopped a Running Schedule of an Integration

The scheduled run is either currently running successfully or has exceeded six hours in duration (now considered an out-of-bounds run).

1. In the navigation pane, click **Observability**, then **Instances**.
2. Hover over the row of the active, schedule integration instance to terminate, and select **Abort**  .
3. Click **Confirm** when prompted.
4. Click **View Details**  .
   The Activity Stream panel opens. A red icon and the reason for the termination are displayed.

   ```
   Instance aborted as schedule was stopped by username
   ```

Oracle Integration Terminated an Asynchronous or Schedule Instance That Exceeded the Processing Deadline

1. In the navigation pane, click **Observability**, then **Instances**.
2. In the **Duration** column, find the integration that has exceeded the processing deadline (for example, an integration's duration reached eight hours and two minutes).
3. Click **View Details**  .
   The Activity Stream panel opens and displays the following details for the instance terminated internally by Oracle Integration.

   ```
   Instance aborted as it has reached maximum duration allow for a flow
   ```

Oracle Integration Terminated a Stuck Scheduled Instance

This rare scenario is for scheduler recovery jobs and only applies when a schedule instance is not terminated internally by Oracle Integration.

1. In the navigation pane, click **Observability**, then **Instances**.
2. In the **Duration** column, find the integration that has exceeded the processing deadline.
3. Click **View Details**  .
   The Activity Stream panel opens and displays the following details for schedule instances.

   ```
   Instance aborted as it has reached maximum duration allow for a scheduled
   flow
   ```


Copyright © 2022, 2026, Oracle and/or its affiliates.

Replay Integration Instances

You can replay REST Adapter-triggered and AS2 Adapter-triggered integration instances that you configured as replayable during integration activation. This feature enables you to quickly retest your integration without the need to re-enter the data used to initially trigger the integration.

- Capabilities and Restrictions
- Understand the Differences Between Replaying and Recovering Instances
- Replay an Integration Instance

Capabilities and Restrictions

- You can only replay instances in which you explicitly select **Allow to run again** in the Activate integration panel.
- You can only replay instances in which the trigger adapter connection is a REST Adapter or AS2 Adapter.
- You can only replay instances with structured payload content. You cannot replay instances with unstructured payload content.
- You can replay instances regardless of their state (for example, you can replay successful, faulted, and aborted instances).
- You can replay instances outside and inside of projects.
- You cannot replay instances where the connectivity agent is used.
- You can only replay a single instance at a time.
- There must be an active integration with the same major version as the replayed instance.
- A parent instance can have many child-replayed instances.
- A replayed instance has a single parent instance.
- You can replay both asynchronous and synchronous instances.
- Payload data for replayed instances is captured for 32 days (for the Production and Audit tracing levels).

Understand the Differences Between Replaying and Recovering Instances

Instance replay and instance recovery (performed with the **Resubmit** option on the Errors page) are not the same. Understand the differences to determine which feature is best for your scenario. Instance recovery includes the following restrictions.

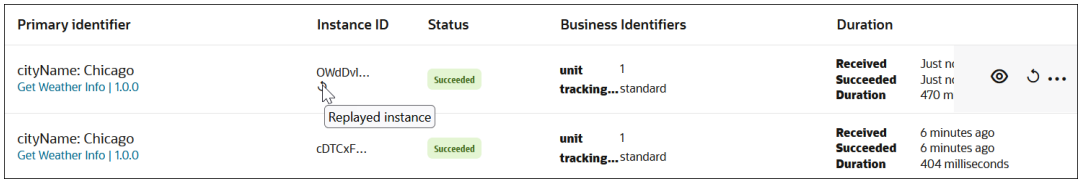
- You can only recover a faulted instance that is identified as recoverable.
- You cannot recover synchronous instances.
- You cannot recover scheduled integration instances with scheduled parameters.
- Recovery only updates the existing instance. No new instance is created.
- Recovery is independent of adapter type or payload content.
- Recovery only restarts from the last dehydration point in an instance (for example, a wait action), whereas replay starts completely from the beginning of the instance.

Copyright © 2022, 2026, Oracle and/or its affiliates.

Replay an Integration Instance

This section describes how to replay an integration instance on the Instances page outside of a project. You can also replay an integration instance inside a project.

1. In the navigation pane, click **Design**, then **Integrations**.
2. Hover over the integration to activate, then click **Activate** .
3. Select the level of tracing appropriate to your integration. This level describes the amount of data to capture in the activity stream. See Activate an Integration.
4. In the **Configure runtime options** section, select **Allow to run again**, then click **Activate**. This selection enables you to replay an instance of an integration during runtime on the Instances page. This selection is only available with specific types of integrations. See Capabilities and Restrictions.
5. Run the integration. For REST Adapter trigger connection-based integrations, you can use the Configure and run page. See Test Integrations from Outside the Integration Canvas. The Activity Stream panel opens.
6. Go to the Instances page in either of two ways:
   a. On the Configure and run page, close the activity stream and click **Track instances**.
   b. In the navigation pane, click **Observability**, then **Instances**.
7. Hover over the row of the instance to replay, and click **Run again** .
   The Run again panel opens.
8. Select the tracing level to use for this instance replay. The level you select is unique to this replay only and does not impact the tracing level you set when activating the integration.
9. Click **Run again**.
   When you replay an instance, a new instance ID is created and an icon is added to identify this as a replayed instance. For this example, the replayed instance and the original instance immediately below are shown. You can continue to replay the original instance or the replayed instance.



> **Note**
>
> The **Display instances** option of the **Filter** enables you to filter on replayed or replayable instances.

10. Click **View details** to view the activity stream.
11. Click **Actions** and note the selections for both the original and replayed instances.
    - The original instance lets you view the replayed child instance.

Copyright © 2022, 2026, Oracle and/or its affiliates.

- The replayed instance lets you view the replayed parent instance and the replayed child instances.

You can replay the original instance again or replay the previously-replayed instance again.

If you no longer want the integration to be replayable, you can change the configuration.

12. In the navigation pane, click **Design**, then **Integrations**.
13. Hover over the row of the integration to configure, and select **Actions** ⋮, then **Configure activation**.
14. In the **Configure runtime options** section, deselect **Allow to run again**, then click **Save**. You do *not* need to deactivate the integration. The change is applied to the active integration.
15. Run your integration again and go to the Instances page to review the results.
16. Hover over the row for that instance and note that there is no **Run again** ↻ icon for replaying this specific instance.

> **Note**
>
> The instances that you created previously when the integration was replayable still show the **Run again** ↻ icon. However, if you attempt to replay these instances, you receive an error. This is because the integration is no longer replayable.

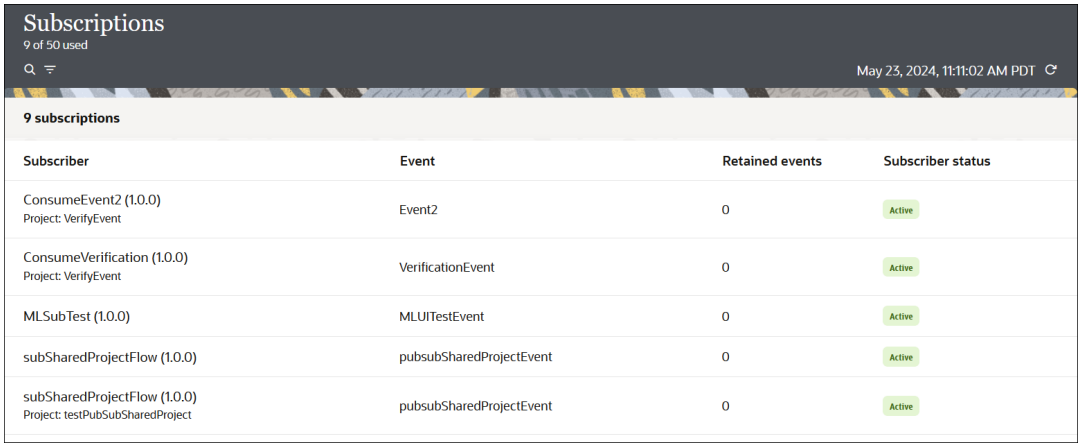
You can also reprocess (replay) B2B for Oracle Integration instances. See B2B Tracking in *Using B2B for Oracle Integration 3*.

Monitor Integrations That Subscribe to Events

You can monitor the number of integrations that subscribe to events, as well as their status, on the Subscriptions page. This page includes subscribed integrations that are outside of projects (globally available) and inside of projects.

1. In the navigation pane, click **Observability**, then **Subscriptions**.

   The Subscriptions page shows the active event subscriptions.
   For this example, 9 of the maximum number of 50 integrations allowed for subscribing to events per service instance are currently available. Subscribed integrations inside of projects are designated with the word **Project** in the **Subscriber** column.



Copyright © 2022, 2026, Oracle and/or its affiliates.

2. Click **Filter** to filter by event subscription integrations or retained events (events published, but not yet received by the event subscription integrations).

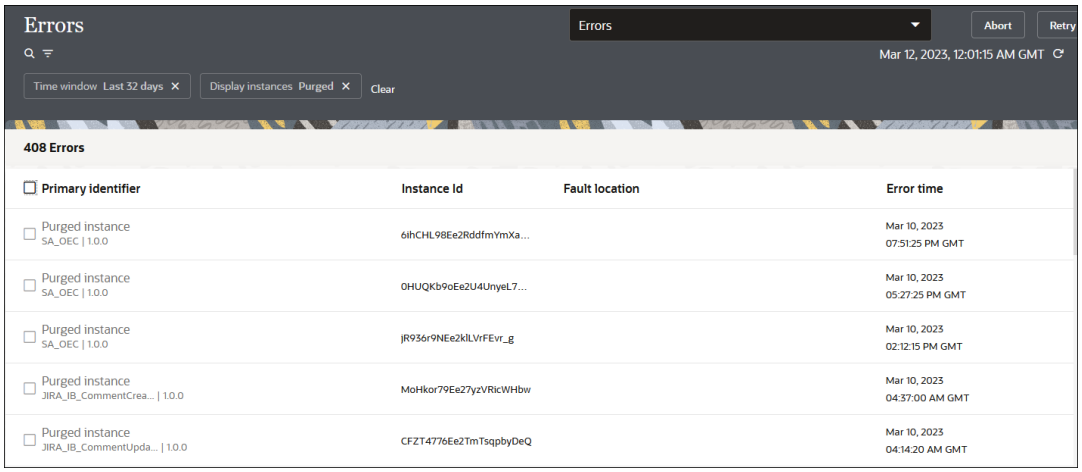
Details about publishing and subscribing to events in integrations are provided. See Create Event Integrations.

View Minimal Details About Debug Tracing Level Instances for 32 Days

When you select the **Debug (not recommended)** option as your tracing level in the Activate integration panel, data is retained for 24 hours before being purged. However, you can view minimal details about the instances for up to 32 days in the Instances page and Errors page. Only the instance ID, instance status, and time duration information are retained. All payload, business identifiers, activity stream, and error message data is purged.

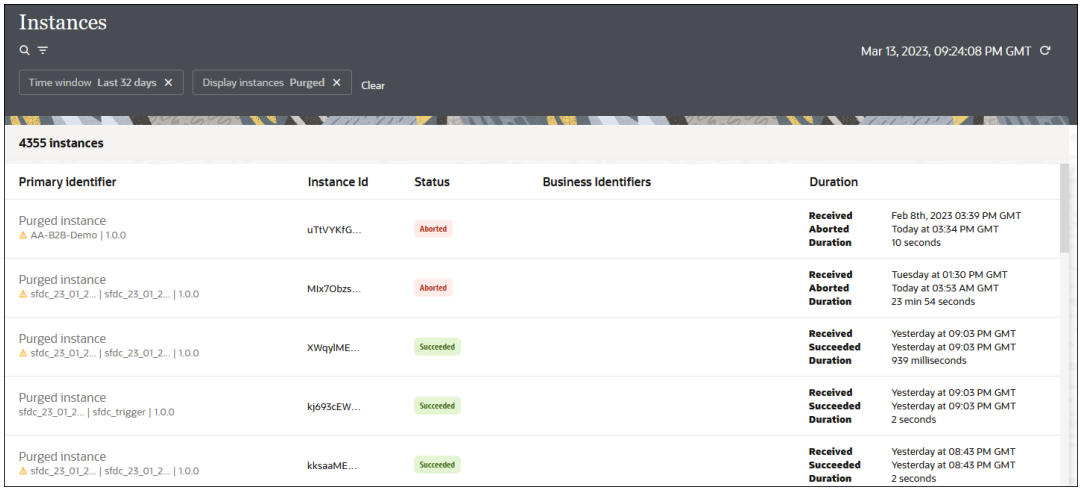
1. Go to the Errors page or the Instances page.
   - In the navigation pane, click **Observability**, then **Errors**.
   - In the navigation pane, click **Observability**, then **Instances**.
2. From the **Filter**, select **Display instances**, then **Purged**.

   On the Errors Page, the instance ID and time of the error are displayed for up to 32 days.



On the Instances page, the instance ID, instance status and instance duration are displayed for up to 32 days.

Copyright © 2022, 2026, Oracle and/or its affiliates.



Note the following:

- In the **Primary Identifier** column, the **Purged instance** label is displayed for each purged instance.
- You cannot click the purged instance identifier to access graphical details about the instance.
- You cannot perform any actions on this instance, such as resubmitting or aborting the instance on the Errors page, selecting options from **Actions** ⋮ on the Instances page, and viewing the activity stream on either page. If you select the activity stream, the following message is displayed:

  ```
  This data is no longer available
  ```

View the Calendar of Schedule Integration Runs

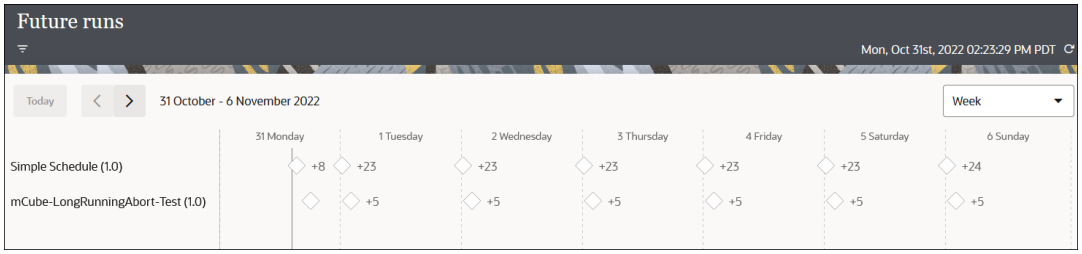
If too many schedule integrations start at the same time, they might experience contention. Ease the contention by staggering the start times. The Future runs page provides an overview of all runs scheduled to trigger in the future, across all schedule integrations whose schedules have started.

> **Note**
>
> The Future runs page works on the browser time zone, and does not support the selected time zone under the **Preferences** option of the **User** menu in the upper right corner of the page.

1. In the navigation pane, click **Observability**, then **Future runs**.

   The Future runs page appears.

Copyright © 2022, 2026, Oracle and/or its affiliates.

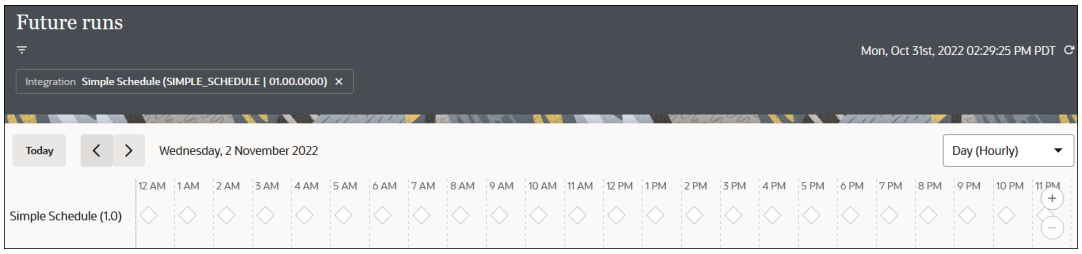


2. At the far right, use the drop-down list to select the time period to view (by minutes in a day, by hours in a day, by week, or by month).

   - For the week and month views, each icon in the chart represents the first run of that day, followed by the count of runs for that day. For example, **+23** means there are 24 runs for that day for a particular schedule. Hovering over the icon shows details, including the time and information about the first run (represented by the icon).
   - For the day view, both by minutes and hours, all future runs are displayed. Every run is represented by the icon. Hovering over them shows the processing time for that run and the sequence (for example, 2 of 20) of that specific run in the day.

   You can also zoom in and zoom out, when needed.

3. Click a specific number (for example, **23** for **Simple Schedule** on Wednesday) to show the scheduled runs for the selected time period (for this example, hours in a day).



4. Hover your cursor over each entry to display the schedule integration name, runtime, and run sequence.
5. Click the schedule integration name in the first column to access the Schedule and future runs page for that schedule.
6. Click to filter by integration name and view future runs of that integration.

Diagnose and Manage Event-Based Oracle Fusion Applications Integrations

You can self-diagnose the delivery of business events between Oracle Fusion Applications and Oracle Integration, such as determining if delivery issues are occurring in Oracle Fusion Applications or Oracle Integration. You can also perform some management tasks, such as retrying the delivery of business events that have failed.

The event handling framework (EHF) capability of Oracle Fusion Applications enables application business event propagation beyond the boundary of Oracle Fusion Applications.

Copyright © 2022, 2026, Oracle and/or its affiliates.

When an Oracle Fusion Applications business event is generated, the EHF ensures that the event information is enriched according to the needs of the subscriber and passed to the subscriber across the boundary. In the case of Oracle Integration, the EHF provides business event support for the following adapters configured using the Basic Authentication and OAuth security policies:

- Oracle ERP Cloud Adapter
- Oracle CX Sales and B2B Service Adapter (previously known as the Oracle Engagement Cloud Adapter)

An integration subscribes to business events with these adapters. When the business event is raised in the application, the EHF passes the event payload to Oracle Integration, which invokes the integration.

With the self-diagnostic capabilities of Oracle Integration, you can:

- Diagnose Oracle Fusion Applications event-triggered integrations.
- Minimize the need to file Oracle Fusion Applications escalations and service requests.
- Review and self-service business event integration-related issues.

For example, assume you have the following scenario in which business events are not received by Oracle Integration:

- User A manages logistics for ABC Corporation and is informed about a large shipment dispatch for an important customer.
- User A observes that the notification has not appeared in the application.
- User A contacts User C, who manages Oracle Fusion Applications at ABC Corporation, and shares their concern about the lack of a notification.
- User C sees that the business event has been correctly created.
- However, neither User A nor User C have any idea about how to diagnose where the event disappeared. They consider logging a service request with Oracle for troubleshooting. With the self-diagnostic capabilities in Oracle Integration, these users can self-diagnose message delivery, thereby reducing the need to create a service request for some basic information.

> **Note**
>
> When business events are not delivered to Oracle Integration, self-diagnosis typically indicates that the error occurred within Oracle Fusion Applications boundaries. Open a service request with Oracle Fusion Applications and include all necessary error messages to aid support in troubleshooting the issue.

1. In the navigation pane, click **Observability**, then **Fusion Applications**.

> **Note**
>
> The **Fusion Applications** option in the navigation pane is only visible to users with the ServiceAdministrator role. For all other users, this option is not visible.

The Fusion Applications page shows the list of deployed adapter connections and their role (trigger, trigger and invoke). To be shown in the Fusion Application page, the connections must satisfy the following conditions:

Copyright © 2022, 2026, Oracle and/or its affiliates.

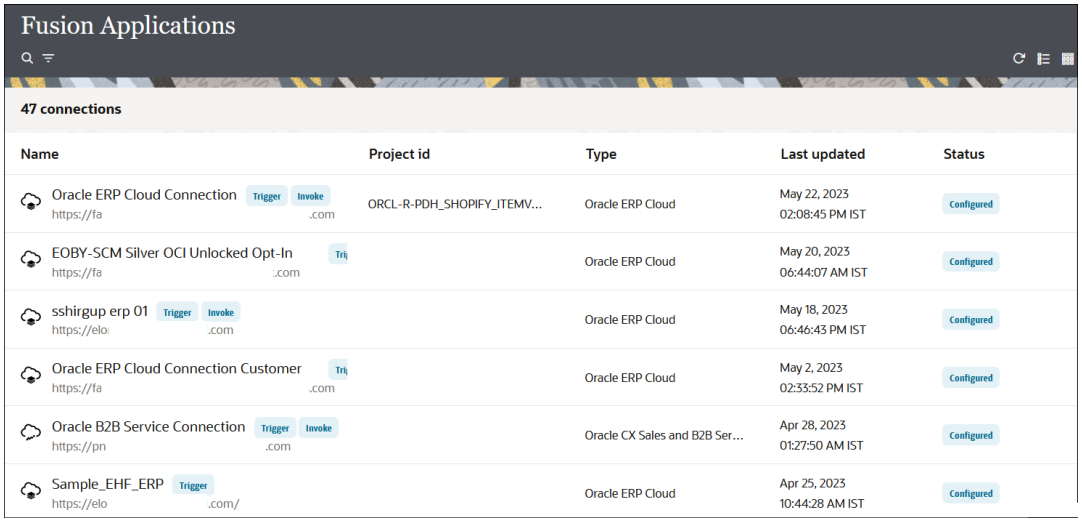
- Must be either the Oracle ERP Cloud Adapter or Oracle CX Sales and B2B Service Adapter.
- Must be a trigger or trigger and invoke role. Invoke alone is not supported.
- Must be in a configured state.

From this page, you can:

- Review basic event information about configured applications.
- Browse the selected application and see subscribed events in detail.
- Review detailed information about failed messages for the selected application.

Key features in the table include the following:

- The **Filter** enables you to search on connection name, user that created the connection, user that updated the connection, adapter type (Oracle ERP Cloud Adapter or Oracle CX Sales and B2B Service Adapter), and connection role (all, trigger, or trigger and invoke)
- The **Name** column shows the connection name and the Oracle Fusion Applications hostname that was specified in the **Properties** section of the Connections page. This enables you to identify the host on which the event was generated.
- The **Project id** column shows the ID of the project to which a connection belongs. If the connection is not part of any project (known as a connection that is globally available), the column cell is empty.
- The **Type** column shows the adapter used by this connection.



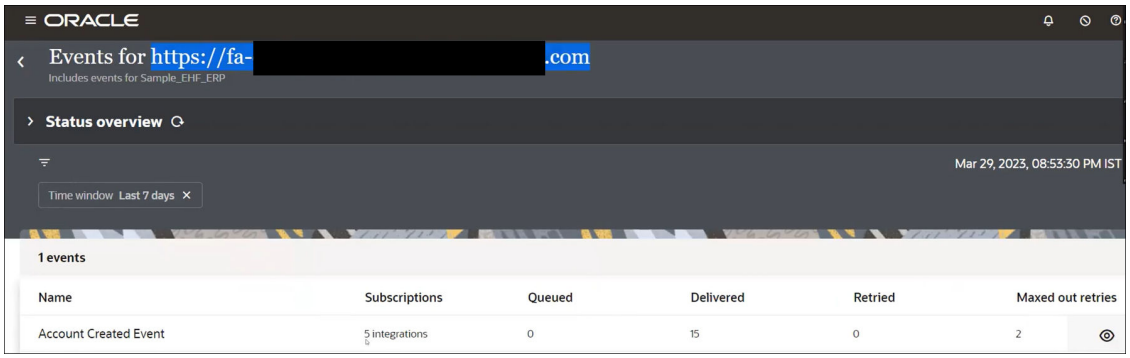
2. Hover over a connection row, and click **View event list for the Fusion Applications instance** to view details about business event subscriptions on the Oracle Fusion Applications host. See Monitor Business Event Delivery Status.

Monitor Business Event Delivery Status

You can view details about business event subscriptions, including their status (queued, delivered, retried, and retries that have reached their limit) and potential reasons for why a

Copyright © 2022, 2026, Oracle and/or its affiliates.

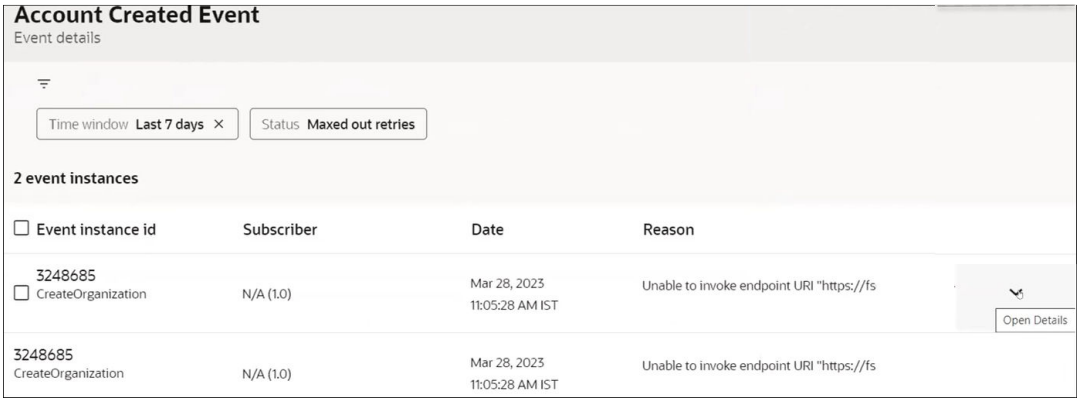
business event has not been delivered to Oracle Integration. You can also retry the delivery of business events.



1. Click **Filter** to adjust the time range for gathering event details.
2. Hover your cursor over a business event row, and click **View** .

   The Event details panel opens. By default, the status is set to **Maxed out retries**.
3. Click **Filter** if you want to change the time range and the status of business events to display:
   - **Maxed out retries** (the default status)
   - **Retried**
   - **Queued**
   - **Delivered**
4. Click **Apply**.

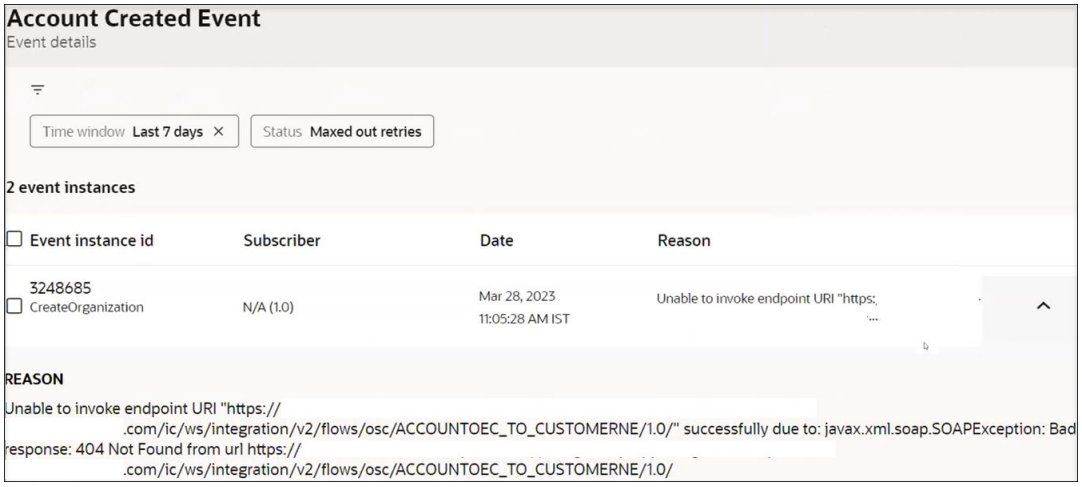
   For the following example, **Maxed out retries** is selected and business events that have reached their retry limit are displayed.



5. Click **Open Details**

   A reason for the error is displayed. For this example, the business event never triggered the integration in Oracle Integration. This may be because the message is stuck in Oracle Fusion Applications or there is a networking issue.

Copyright © 2022, 2026, Oracle and/or its affiliates.



6. If you want to attempt business message delivery again, select the event instances to retry and click **Retry**. You can also click **Event instance ID** to retry all event instances. The maximum number of retries is 10. After that, the event's status is moved to **Maxed out retries**. The event then requires a manual intervention (retry). Once you click **Retry**, the event is queued for retry, but its state has not yet changed. Therefore, that event is still shown. You must refresh the page and check to see if the event's status has changed.

Send System Status Reports with 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, and when a service is down or an agent becomes unavailable). You can also remove email addresses from the suppression list.

> **Note**
>
> This is a system level setting. Therefore, user A cannot create one notification setup while user B creates another notification setup. User B can see the notification setup created initially by user A and can modify it.

Topics:

- Understand DKIM Authentication and SPF Record Details
- Configure Notification Emails
- Remove Email Addresses from the Suppression List
- Change the Delivery Time of Notification Emails

Understand DKIM Authentication and SPF Record Details

When you configure notification emails to send service failure alerts, system status reports, and integration error reports, you can optionally specify Domain Keys Identified Mail (DKIM) authentication and Sender Policy Framework (SPF) record details.

Details and examples are provided to better understand DKIM and SPF. See Configure Email Authentication Settings for SPF and DKIM in *Provisioning and Administering Oracle Integration 3*.

Copyright © 2022, 2026, Oracle and/or its affiliates.

Configure Notification Emails

You can configure notification emails to contact interested parties about system status on a regular basis and alert them about service failures. Detailed system status reports include metrics on instances and runs processed by the system, including failures. You can select to send an hourly or daily report and specify the list of recipients.

- Contents of Email Notification Reports
- Decide Which Email Delivery Service to Use
- Configure Notification Emails

Contents of Email Notification Reports

The report includes the following information:

- POD name (also included in the subject of the email)
- Time period
- Total messages received, processed, succeeded, aborted, and failed
- Percentage of successful messages
- Errors by integration instance and run including messages received, processed, succeeded, and failed

The email 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.

Decide Which Email Delivery Service to Use

The Notifications page enables you to manage email delivery through either of two options. Review the following table to determine which option is best for your business environment.

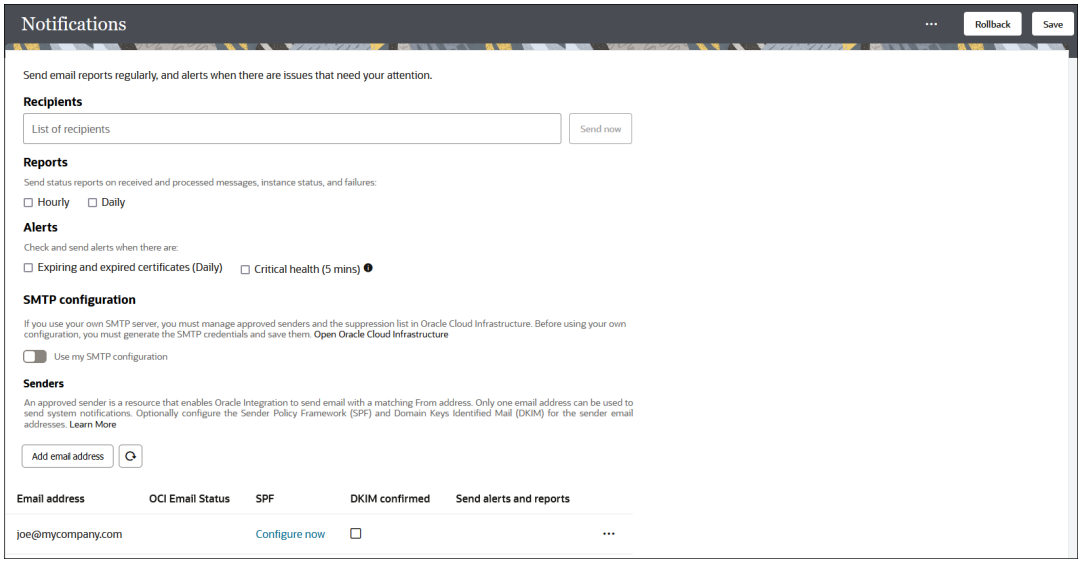
Copyright © 2022, 2026, Oracle and/or its affiliates.

Email Delivery Option	Benefits	Limitations
Oracle Cloud Infrastructure Email Delivery service of your customer tenancy (Recommended)	• Limit is controlled by the Oracle Cloud Infrastructure Email Delivery service of your customer tenancy. The limit can be increased to more than 10,000 messages per rolling 24-hour period by working directly with the Oracle Cloud Infrastructure Email Delivery service. See Email Delivery Service Capabilities and Limits. • Better control over email suppression lists. • Better general management of sent emails. • Better security because email domain and DKIM setup is done on the customer's tenancy. • Ability to register and use senders without an active email inbox.	• Requires explicit setup of SMTP credentials, approved senders, and email domains. To use the Oracle Cloud Infrastructure Email Delivery service of your customer tenancy, you must do the following: • Create SMTP credentials in the Oracle Cloud Console in order to complete the **SMTP configuration** section in Step 5. Only Oracle Cloud Console SMTP credentials are supported. See Creating SMTP Credentials. • Register the default sender email address and the sender email address for system notification with Oracle Cloud Infrastructure in your customer tenancy. • Manage the approved senders and suppression list in your customer tenancy. • Check the service limits. See Overview of Email Delivery. **Note**: You *cannot* bypass the Oracle Cloud Infrastructure Email Delivery service to use email services such as Microsoft 365, GMail, and others.
Default option provided with Oracle Integration	• Out-of-the-box support for sending emails. • Can maintain the approved senders and suppression list from the Notifications page in Oracle Integration.	• Limited to a maximum of 100 messages per rolling 24-hour period for trial instances and 10,000 messages per rolling 24-hour period for paid instances. • Approved senders registered need an active email inbox. • For changes in email domain and Domain Keys Identified Mail (DKIM), you must work with Oracle Support to get the updates reflected.

Configure Notification Emails

1. In the navigation pane, click **Settings**, then **Notifications**.

Copyright © 2022, 2026, Oracle and/or its affiliates.



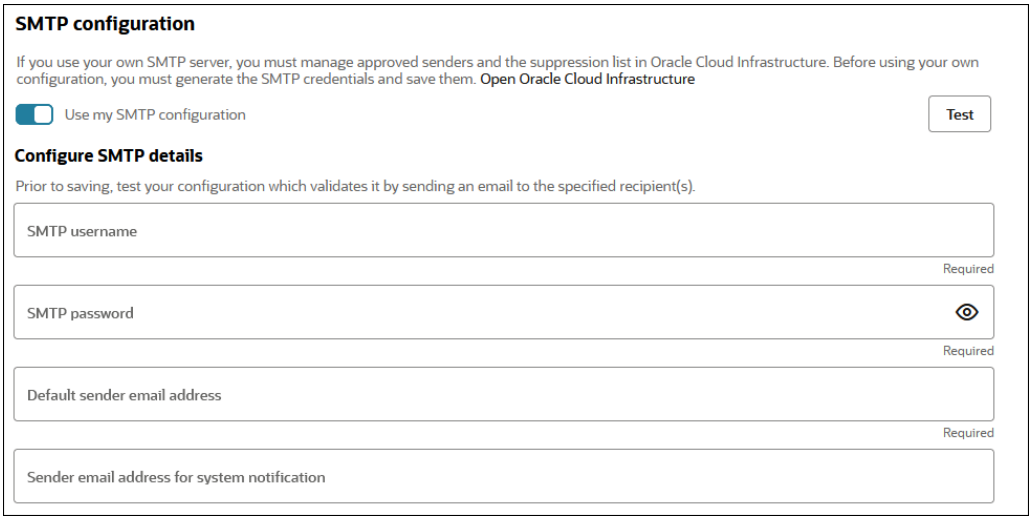
2. In the **Recipients** field, enter the email address to which to send the reports. You can specify multiple email addresses, each separated by commas.
3. In the **Reports** field, select the type of report to send.
   - **Hourly**: Send a detailed report every hour.
   - **Daily**: Alerts are sent once a day.

   The report you receive can include the names and versions of a maximum of five integrations that failed in that time period. Any additional integration failures can be viewed by clicking the **View Errors** button in the email, which opens the Errors page in Oracle Integration.
4. In the **Alerts** section, select to send alerts when the following occurs:

Option	Description
Expiring and expired certificates (Daily)	Select to send alerts when the following occurs: • Certificates are set to expire within seven days. • Certificates have expired. The check is made for certificates you uploaded on the Certificates page. Certificate expiration status is checked daily. Oracle Integration provided-certificates are not checked.
Critical health (5 mins)	Select to send alerts when one or more of the following conditions are detected: • The email quota exceeded the threshold. • One or more connectivity agent instances became unavailable.

5. If you want to use the Oracle Cloud Infrastructure Email Delivery service of your customer tenancy to manage email delivery, click  to enable the customer tenancy mode.
   a. Specify the SMTP user name and password and the default sender email address (that is, the `from` address). These three fields are required. You can also optionally specify a default address for system notifications.

Copyright © 2022, 2026, Oracle and/or its affiliates.



b. Click **Test** to validate the SMTP credentials.

c. Click **Save**.

> **Note**
>
> If you want to return to using the default option, click to disable the SMTP credentials, and click **Save**.

6. If you want to use the default option provided with Oracle Integration to manage email delivery, follow these steps:

   a. In the **Senders** section, click **Add email address** to add approved senders.

   b. Specify the following details:

Element	Description
Email address	An approved sender enables Oracle Integration to send email with a matching **From** address. An email is sent to each of the email addresses listed as approved senders. Upon approval, these email addresses can be used as the **From** address in a notification action of an integration and/or system notification.

Copyright © 2022, 2026, Oracle and/or its affiliates.

Element	Description
OCI Email Status	Perform the following steps: i. Enter an approved sender email address and click **Save**. A verification code is sent to that address and verification status changes to **In Progress**. ii. Click **Enter verification code** and enter the code received by that email address. Upon successful verification, status is changed to **Verified**. Note the following: • Existing sender emails that were previously approved, but never formally verified with this process, can continue to be used for now. It is recommended that you delete these entries, re-enter them, and enter the verification code sent to the email address. • Don't share approved sender email addresses across service instances (for example, test and production instances). Create different approval emails addresses for each service instance, even if they are in the same tenancy.
SPF	Optionally configure the Sender Policy Framework (SPF) for the sender email addresses. You configure SPF support outside of Oracle Integration.
DKIM confirmed	Optionally confirm the DKIM for the sender email addresses. You configure DKIM support outside of Oracle Integration.
Send alerts and reports	Select to use this email address for the system notifications you selected at the top of the page. However, only a single email address can be selected to send system notifications.

Recommendations are provided to ensure that you configure notification email correctly. See Troubleshoot Notification Email Configuration.

c. Click **Save**.

7. Perform other email notification tasks on this page.

To ...	Description
Receive an immediate email about system status	Click **Actions** ⋮, then select **Send Now**.
Clear the distribution list, selected alert/reports options, and specified sender email addresses.	Click **Actions** ⋮, then select **Reset All Notifications**.
Remove email addresses from the suppression list **Note**: This task is only required if you are using the default option.	Click **Actions** ⋮, then select **Email Suppression List**. See Remove Email Addresses from the Suppression List.

Copyright © 2022, 2026, Oracle and/or its affiliates.

Remove Email Addresses from the Suppression List

You can remove email addresses from the suppression list. The suppression list lets you refresh each individual email address to see the current state. You can also see the date on which an email sent to that email address was last suppressed. The date is refreshed each time an email is sent to that address.

Adding users to a suppression list is a common task in email address management. Reasons for adding email addresses to the suppression list can include the following:

- An employee on an email distribution list left the company and their email address was deactivated.
- SPF and DKIM not being configured on the sender's address domain server (DNS) result in soft bounces and subsequently end up on the suppression list.

Oracle Integration provides a self-service feature that enables you to view and remove email addresses from the suppression list, when necessary. This feature eliminates the need to file a service request to remove addresses from the suppression list.

Suppressed email addresses can lead to errors in your integration. The activity stream of the instance can show that a notification action failed if the email address used to send the notification is on the suppression list.

1. Click **Actions** and select **Email suppression list**.

   The Email suppression list panel opens to show all email addresses that have been suppressed. Email addresses on the suppression list cannot send or receive email notifications.

2. Browse or search for the email address to remove.

3. Click **Refresh** to ensure that you are seeing the latest state for this specific email address in this service instance. Performing a refresh may remove that email address (because the suppression list was updated behind the scenes), change the email address state message (for example, from hard bounces to repeated soft bounces), or update the timestamp.

4. Next to the email address to remove, click **Delete** .
   This action removes this address from the suppression list. You *cannot* remove more than one address at a time.

   Need to troubleshoot suppression list issues? See Suppression List.

Change the Delivery Time of Notification Emails

You can change the delivery time of notification emails.

The tracking metrics daily report runs by default at 00:00 UTC time, which can differ from your time zone. For example, the central time zone is 6 hours behind the UTC time zone. When the report runs at 00:00 UTC on November 14th, it is processing at 6:00pm central time on November 13th.

The report time is based on UTC time, which ensures a common reference point (particularly for the log files). This enables anyone in any time zone to identify the time of processing. This is important for global companies with operations in multiple countries/time zones.

Copyright © 2022, 2026, Oracle and/or its affiliates.

As a workaround, set up notifications to run hourly and use a tool such an Outlook rule to automatically delete the 23 unwanted emails, keeping the one email with the tracking metrics you want.

See Troubleshoot Notification Email Configuration.

Copyright © 2022, 2026, Oracle and/or its affiliates.

14

Troubleshoot Oracle Integration

Learn about Oracle Integration troubleshooting tasks, including configuration recommendations to ensure proper email delivery, connectivity agent issues, integration runtime issues, integration design-time issues, schedule integration issues, and integration activation issues.

> **Notes**
>
> - There are overall service limits for Oracle Integration. A service limit is the quota or allowance set on a resource. See Service Limits in *Provisioning and Administering Oracle Integration 3*.
> - Cloud Customer Connect provides a community in which to ask questions, connect with experts, and share your thoughts and ideas about Oracle Integration.
> - Need to troubleshoot an OAuth issue? See Troubleshoot OAuth.

Topics:

- Troubleshoot Notification Email Configuration
- Troubleshoot the Connectivity Agent
- Troubleshoot Private Endpoints
- Troubleshoot Integration Runtime
- Troubleshoot Integration Design Time
- Troubleshoot Schedule Integrations
- Troubleshoot Integration Activations

Troubleshoot Notification Email Configuration

Follow these recommendations to correctly configure and use the default `from` address and suppression list. These recommendations help you to avoid notification email delivery issues.

- Default From Address
- Suppression List
- Invalid Email Addresses in Distribution Lists (DL)

Default From Address

- Don't use `no-reply@oracle.com` as the `from` address.
- Don't use the `oracle` domain.
- Change the default `from` address from `no-reply@oracle.com` to `no-reply@mail.integration.`*`region`*`.ocp.oraclecloud.com`.
  The `region` attribute is provided by Oracle Integration.

Copyright © 2022, 2026, Oracle and/or its affiliates.

- Change the `from` address in your integrations from `no-reply@oracle.com` to `no-reply@mail.integration.region.ocp.oraclecloud.com`.
  The `region` attribute is provided by Oracle Integration.

Suppression List

- If any email address in the `To` list is suppressed, all email addresses in the list are impacted. That is, email is not delivered to any address in the `To` list.
- Add `To` addresses to the suppression list for a number of reasons:
  - As of now, the `recipient` address when a hard bounce occurs (emails go undelivered for permanent reasons), when a soft bounce occurs (emails go undelivered for temporary reasons), and when a large number of emails are received are some of the reasons to add the `To` address to the suppression list.
- If DomainKeys Identified Mail (DKIM) and Sender Policy Framework (SPF) are not configured for the `from` address domain, the likelihood of having a bounce or messages being silently dropped by the receiving infrastructure is higher.
- You can remove email addresses from the suppression list. See Remove Email Addresses from the Suppression List.

Invalid Email Addresses in Distribution Lists (DL)

If an email DL includes an invalid email address, *all* email addresses in the list are impacted. When a notification alert is sent, none of the recipients in the DL receive the alert.

Troubleshoot the Connectivity Agent

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

Topics:

- Don't Change the Default Token Expiry Value for the OAuth Client Application
- Connectivity Agent Fails to Start After Changing the InstallerProfile.cfg File Associated with the Agent Group
- Troubleshoot Agent Unavailable Issues
- Unable to Start Connectivity Agent Because Agent Status Cannot Be Obtained
- Connectivity Agent Startup Fails With a "java.lang.RuntimeException: Agent Startup Failed - Unable to get agent status" Error
- Connectivity Agent Installation Error if /etc/hosts File is Not Configured
- Restrictions on Using the Stage File Action with the Connectivity Agent
- Java Memory Errors
- 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
- 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

Copyright © 2022, 2026, Oracle and/or its affiliates.

- 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
- Integration Activation Error Due to Change in JDK Location
- Troubleshoot Network Connectivity Issues
- HTTP 401 Unauthorized Error Occurs During Connectivity Agent Installation

Don't Change the Default Token Expiry Value for the OAuth Client Application

Don't modify the default token expiry value for the OAuth client application. This value is set to `3600` seconds (one hour) by default. The connectivity agent regenerates the token expiry value right before token expiration to avoid in-flight request failures instead of waiting for the token to expire. If you manually set this value to under one hour, `401` errors can occur, which result in intermittent integration failures.

Connectivity Agent Fails to Start After Changing the InstallerProfile.cfg File Associated with the Agent Group

You cannot use the `InstallerProfile.cfg` file associated with one agent group (for example, AGENT_GROUP_1) with another agent group (for example, AGENT_GROUP_2).

For example, assume you perform the following steps:

1. Download the specific agent group's `InstallerProfile.cfg` file to the `oic_conn_agent_installer` directory on the local host.
2. Install and start the connectivity agent.
3. Stop the connectivity agent and replace the `InstallerProfile.cfg` file in the `oic_conn_agent_installer` directory with a different `InstallerProfile.cfg` associated with a different agent group.
4. Restart the connectivity agent.

   ```
   $ java -jar connectivityagent.jar
   ```

The following error occurs during startup:

```
Existing Agent installation found... Starting Agent for message processing.
Checking for already running instances of this agent. This might take up to
15 seconds ...
Initializing the credential store ...
Error encountered. See logs for more details.
Shutting down ...
```

The agent diagnostics log also shows the following error message.

```
[2025-03-17T18:29:01.286Z] [SEVERE] [ThreadID: 24] [Logger:
oracle.cloud.cpi.agent.util.CpiUtils] [SRC Class:
oracle.cloud.cpi.agent.util.CpiUtils; Method: getKey] Issue when retrieving
agent instance specific key Unable to retrieve key for agent instance CA-
SHADOWFAX-1742233228361. Secret not found in secure credential store Http
Status 500
[2025-03-17T18:29:01.287Z] [SEVERE] [ThreadID: 24] [Logger:
oracle.cloud.cpi.agent.ProxyCpiAgentLoader] [SRC Class:
oracle.cloud.cpi.agent.ProxyCpiAgentLoader; Method: initialize] Agent Startup
```


Copyright © 2022, 2026, Oracle and/or its affiliates.

```
Failed java.lang.Exception: /key endpoint returned a null or empty response.
at oracle.cloud.cpi.agent.util.CpiUtils.getKey(CpiUtils.java:749)
at
oracle.cloud.cpi.agent.util.CpiUtils.initializeCredentialStore(CpiUtils.java:6
93)
at
oracle.cloud.cpi.agent.ProxyCpiAgentLoader.initialize(ProxyCpiAgentLoader.java
:121)
at
oracle.cloud.cpi.agent.ProxyCpiAgentLoader.runAgent(ProxyCpiAgentLoader.java:4
8)
at java.base/jdk.internal.reflect.NativeMethodAccessorImpl.invoke0(Native
Method)
at java.base/
jdk.internal.reflect.NativeMethodAccessorImpl.invoke(NativeMethodAccessorImpl.
java:77)
at java.base/
jdk.internal.reflect.DelegatingMethodAccessorImpl.invoke(DelegatingMethodAcces
sorImpl.java:43)
at java.base/java.lang.reflect.Method.invoke(Method.java:569)...
```

Instead, create a new agent group, download the connectivity agent, download the `InstallerProfile.cfg` file, replace the existing `InstallerProfile.cfg` file with the downloaded file, and install the connectivity agent. See Download and Install the Connectivity Agent.

Troubleshoot Agent Unavailable Issues

Connectivity agent status can be determined by selecting **Observability** > **Agents** to access the Monitor agents page. If the agent status is displayed as unavailable, follow this troubleshooting workflow.

Copyright © 2022, 2026, Oracle and/or its affiliates.

Number	Step	Details

1	**Check the agent process is running**	Check whether the agent is running. Run one of these commands from the host where the agent is installed.
		If you don't see `connectivityagent.jar` as a listed process, the agent is not running. For example: `19054 connectivityagent.jar`
		Linux:
		`ps –ef | grep connectivityagent.jar`
		Windows:
		`jps -l`
		If the agent process is not running, start it:
		Linux:
		Start the connectivity agent as a background process:
		`nohup java -jar connectivityagent.jar &`
		Windows:
		`java -jar connectivityagent.jar`
		Note
		Starting the agent as a Windows service is not supported.
		If you cannot start the agent, continue to check network connectivity.
		If the agent process is running, restart it:
		Follow these steps to restart the agent.

Copyright © 2022, 2026, Oracle and/or its affiliates.

Number	Step	Details
2	**Check network connectivity**	**Check that the agent host can connect to the Oracle Integration instance:** On the host where the agent is installed, run this command, replacing with your own path: `curl https://myoic-tenancy-ia.integration.ocp.oraclecloud.com/ic/home` **If the response is 302 Found**, the agent is able to resolve the Oracle Integration instance host. You have network connectivity. **If the response is not 302 Found**, there may be network issues. Check the `/etc/hosts` file: 1. Check for the correct IP address and host name for your Oracle Integration instance. 2. Troubleshoot with your network administrator to check that connectivity to the instance is not blocked by a firewall or proxy. Here's agent troubleshooting information.
3	**Download the Configuration File**	Download the agent configuration file again and restart the connectivity agent after replacing the previous agent configuration file.
4	**Log a Service Request**	If you've gone through the troubleshooting steps, can't find anything in the diagnostic log, and the agent status is still unavailable, create a service request with Oracle Support. Include the diagnostic log in the service request.

Unable to Start Connectivity Agent Because Agent Status Cannot Be Obtained

Connectivity agent startup can fail with the following error:

```
java -jar connectivityagent.jar &


[2024-12-09T19:30:16.392Z] [INFO] [ThreadID: 15] [Logger:
oracle.wsm.agent.WSMAgent] [SRC Class: oracle.wsm.agent.WSMAgent; Method:
processRequest] pool-1-thread-1Entering processRequest() -  12/09/2024
19:30:16:392
[2024-12-09T19:30:16.396Z] [INFO] [ThreadID: 15] [Logger:
oracle.wsm.agent.WSMAgent] [SRC Class: oracle.wsm.agent.WSMAgent; Method:
processRequest] pool-1-thread-1Exiting processRequest() -  12/09/2024
19:30:16:396
[2024-12-09T19:30:16.473Z] [INFO] [ThreadID: 15] [Logger:
oracle.cloud.cpi.agent.util.CpiUtils] [SRC Class:
oracle.cloud.cpi.agent.util.CpiUtils; Method: createAuthToken] OAuth token
url https://idcs-xxxxxx.identity.oraclecloud.com:443/oauth2/v1/token
[2024-12-09T19:30:46.505Z] [SEVERE] [ThreadID: 15] [Logger:
oracle.cloud.cpi.agent.util.CpiUtils] [SRC Class:
oracle.cloud.cpi.agent.util.CpiUtils; Method: createAuthToken] IDCS Token
Generation: Call to create token failed due to
java.net.SocketTimeoutException: Connect timed out
[2024-12-09T19:30:46.507Z] [INFO] [ThreadID: 15] [Logger:
oracle.wsm.agent.WSMAgent] [SRC Class: oracle.wsm.agent.WSMAgent; Method:
```


Copyright © 2022, 2026, Oracle and/or its affiliates.

```
processRequest] pool-1-thread-1Entering processRequest() -  12/09/2024
19:30:46:507
[2024-12-09T19:30:46.507Z] [INFO] [ThreadID: 15] [Logger:
oracle.wsm.agent.WSMAgent] [SRC Class: oracle.wsm.agent.WSMAgent; Method:
processRequest] pool-1-thread-1Exiting processRequest() -  12/09/2024
19:30:46:507
[2024-12-09T19:30:46.550Z] [SEVERE] [ThreadID: 15] [Logger:
oracle.cloud.cpi.agent.install.AgentInstaller] [SRC Class:
oracle.cloud.cpi.agent.install.AgentInstaller; Method: checkVersion] Agent
Installation Failed !!! Incorrect OIC Username / Password provided.
[2024-12-09T19:30:46.551Z] [SEVERE] [ThreadID: 15] [Logger:
oracle.cloud.cpi.agent.ProxyCpiAgentLoader] [SRC Class:
oracle.cloud.cpi.agent.ProxyCpiAgentLoader; Method: configureAgent] Agent
Installation Failed !!! java.lang.reflect.InvocationTargetException
        at java.base/
jdk.internal.reflect.NativeMethodAccessorImpl.invoke0(Native Method)
        at java.base/jdk.internal.reflect.NativeMethod
```

This error occurs because the connectivity agent cannot query the agent status from the Oracle Integration server.

1. Allowlist your identity domain, design time IP address, and runtime IP address.
2. Check connectivity to Oracle Integration 3.
3. (Optional) Download the configuration file again and restart the agent if no issue is found with Oracle Integration 3 connectivity.

Connectivity Agent Startup Fails With a "java.lang.RuntimeException: Agent Startup Failed - Unable to get agent status" Error

Connectivity agent startup can fail with the following error:

```
[2021-06-19T10:06:06.567Z] [SEVERE] [ThreadID: 1] [Logger:
oracle.cloud.cpi.agent.CpiBootstrapLoader]
[SRC Class: oracle.cloud.cpi.agent.CpiBootstrapLoader; Method: main] Error
encountered !!! java.lang.RuntimeException:
 java.lang.reflect.InvocationTargetException
at
oracle.cloud.cpi.agent.AgentStartCommandTask.run(CpiBootstrapLoader.java:452)
at java.util.concurrent.Executors$RunnableAdapter.call(Executors.java:511)
at java.util.concurrent.FutureTask.run(FutureTask.java:266)
at
java.util.concurrent.ScheduledThreadPoolExecutor$ScheduledFutureTask.access$20
1(ScheduledThreadPoolExecutor.java:180)
at
java.util.concurrent.ScheduledThreadPoolExecutor$ScheduledFutureTask.run(Sched
uledThreadPoolExecutor.java:293)
at
java.util.concurrent.ThreadPoolExecutor.runWorker(ThreadPoolExecutor.java:1149
)
at
java.util.concurrent.ThreadPoolExecutor$Worker.run(ThreadPoolExecutor.java:624
)
at java.lang.Thread.run(Thread.java:748)
Caused by: java.lang.reflect.InvocationTargetException
```


Copyright © 2022, 2026, Oracle and/or its affiliates.

```
at sun.reflect.NativeMethodAccessorImpl.invoke0(Native Method)
at
sun.reflect.NativeMethodAccessorImpl.invoke(NativeMethodAccessorImpl.java:62)
at
sun.reflect.DelegatingMethodAccessorImpl.invoke(DelegatingMethodAccessorImpl.j
ava:43)
at java.lang.reflect.Method.invoke(Method.java:498)
at
oracle.cloud.cpi.agent.CpiBootstrapLoader.startAgent(CpiBootstrapLoader.java:1
92)
at
oracle.cloud.cpi.agent.AgentStartCommandTask.run(CpiBootstrapLoader.java:436)
...
...
Caused by: java.lang.reflect.InvocationTargetException
at sun.reflect.NativeMethodAccessorImpl.invoke0(Native Method)
at
sun.reflect.NativeMethodAccessorImpl.invoke(NativeMethodAccessorImpl.java:62)
at
sun.reflect.DelegatingMethodAccessorImpl.invoke(DelegatingMethodAccessorImpl.j
ava:43)
at java.lang.reflect.Method.invoke(Method.java:498)
at oracle.cloud.cpi.agent.CpiAgentLoader.startAgent(CpiAgentLoader.java:27)
...
...
Caused by: java.lang.RuntimeException: Agent Startup Failed - Unable to get
agent status :
at
oracle.cloud.cpi.agent.ProxyCpiAgentLoader.initialize(ProxyCpiAgentLoader.java
:151)
at
oracle.cloud.cpi.agent.ProxyCpiAgentLoader.runAgent(ProxyCpiAgentLoader.java:5
3)
...
...
```

Ensure that connectivity from your agent host to Oracle Integration 3 is working.

1. Allowlist your identity domain, design time IP address, and runtime IP address.
2. Check connectivity to Oracle Integration 3.
3. (Optional) Download the configuration file again and restart the agent if no issue is found with Oracle Integration 3 connectivity.

Connectivity Agent Installation Error if /etc/hosts File is Not Configured

During connectivity agent installation, if the agent installer cannot find details about the host name/virtual machine (VM) on which the agent is being installed, you may receive the following error:

```
Caused by: java.net.UnknownHostException: : Unknown
name or service
```

Resolve this error by adding an entry for the IP address and host name in the `/etc/hosts` file of the host name/VM.

Copyright © 2022, 2026, Oracle and/or its affiliates.

Restrictions on Using the Stage File Action with the Connectivity Agent

Be aware of several restrictions when using some operations of the stage file action with the connectivity agent. See Restrictions on Using Stage File Action Operations with the File/Attachment Features of the Connectivity Agent.

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 resources on the physical machine/virtual machine/server on which the agent is running. 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.

Add or Change the Non-Proxy Host Configuration After Installation in the CpiAgent.properties File

If you need to add, change, or bypass the non-proxy host configuration *after* agent installation, do *not* edit the `InstallerConfing.cfg` file. Proxy host changes made to that file after agent installation do not take effect. Instead, update the host with the `proxy_nonProxyHosts` parameter in the *`Agent_Installation_Location`*`/agenthome/agent/config/CpiAgent.properties` file for your changes to take effect. After editing this file, restart the agent.

Class Loading Conflict When Sharing the JDK Instance with Another Product

When using the connectivity agent, the following error occurs because your JDK instance is shared with another product for which JAR files have been added in the JDK's endorsed directory. This results in a class loading conflict with the agent:

```
ClassCastException: com.sun.xml.messaging.saaj.soap.ver1_1.Message1_1Impl
cannot be cast to oracle.j2ee.ws.saaj.soap.MessageImpl
```


Copyright © 2022, 2026, Oracle and/or its affiliates.

Ensure that your connectivity agent is running with a JDK installation that is not modified because of use with other products.

Connectivity Agent Log File Location

The connectivity agent `agent-diagnostic0.log` file is available under `agenthome/logs`.

Agent Behavior in a Decommissioned Instance or HTTP 404/401 Error Response Codes

The following code in the logs indicates that agent runtime message processing has halted. This occurs if an HTTP `404/401` error code is continuously received by the agent for 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 Connectivity Agent.

```
[2018-10-13T04:30:13.501Z] [SEVERE] [ThreadID: 18] [Logger:
oracle.cloud.cpi.agent.transport.AQRuntimeConsumer] [SRC Class:
oracle.cloud.cpi.agent.transport.AQRuntimeConsumer; Method: run] Terminate
flag activated. Signalling termination of agent runtime poller thread with Id

CPI_RUNTIME_REQ_RES_CHANNEL_worker_0.
```

Verify Endpoint Accessibility when Agent is Installed with a Proxy Host

When the agent is installed with a proxy host, carefully check that the endpoint to access through the agent is reachable through the proxy host. If it is not reachable through the proxy host, you must configure the on-premises endpoint host in the `proxy_NON_PROXY_HOSTS` parameter of the *`Agent_Installation_Location`*`/agenthome/agent/config / CpiAgent.properties` file.

Unlock the Agent Group

When an agent group is in edit mode and the browser crashes, the agent group becomes locked, which prevents it from being edited. This results in the following error:

```
ICS-10507: The agent group cannot be updated because it is locked.
```

To unlock the agent group:

1. Log in again as the same user who was editing the agent group when the browser crashed, then log out. This action unlocks the agent group.

or

1. Wait 30 minutes for the lock to expire after the timeout starts.

Failure to Send a Response Due to a java.net.SocketException: Connection Reset Error

For connectivity agent installations running on an Oracle Integration Classic VM and connecting to Oracle Integration (running on Oracle Cloud Infrastructure), the design-time and runtime operation involving the connectivity agent sometimes fails with a `java.net.SocketException: Connection reset` error.

This can occur because of a Maximum Transmission Unit (MTU) mismatch.

Copyright © 2022, 2026, Oracle and/or its affiliates.

Here is the complete error:

```
[2019-01-03T16:35:12.670Z] [SEVERE] [ThreadID: 50] [Logger:
oracle.cloud.cpi.agent.transport.CpiAgentAQTransporter] [SRC Class:
oracle.cloud.cpi.agent.transport.CpiAgentAQTransporter; Method:
sendOneWayPacket] Exception while sending response back to
ICSjava.net.SocketException: Connection reset
com.sun.jersey.api.client.ClientHandlerException: java.net.SocketException:
Connection reset
at
com.sun.jersey.client.urlconnection.URLConnectionClientHandler.handle(URLConne

ctionClientHandler.java:155)
at com.sun.jersey.api.client.Client.handle(Client.java:652)
at com.sun.jersey.api.client.WebResource.handle(WebResource.java:682)
at com.sun.jersey.api.client.WebResource.access$200(WebResource.java:74)
```

To make this connection work, set the MTU of the Oracle Integration Classic VM (where the agent is installed) to `1500` from the current value of `8900`.

Perform the following steps:

1. Run `ifconfig -a` as the root user (sudo) and note down the network interface.
2. Run the following command as the root user (sudo) for the network interface (assuming the network interface is `eth0`):

   ```
   ifconfig eth0 mtu 1500
   ```

Anytime the agent VM is restarted (note that it is not an agent restart, but the host where the agent is installed), the changes must be done for the network interface before restarting the agent.

Agent Installation on Linux Fails When Using an Installer Copied Using winscp

During connectivity agent installation in Linux environments, the installation sometimes fails with the following error:

```
On premise agent is throwing the following error: java.lang.RuntimeException:

Agent Startup Failed - java.lang.IllegalArgumentException: URI is not
absolute
```

Installation failure occurs for the following reason:

1. The agent is to be installed in a Linux environment.
2. The agent installer ZIP file is downloaded in a Windows environment and transferred to a Linux environment for installation using a Windows tool called `winscp`.
3. Even when the binary option is enabled in the `winscp` tool, the installer ZIP used to run and install the agent fails with the above error.

As a workaround, perform the following steps:

1. Download the agent installer to a Linux environment directly and do not transfer it from a Windows environment.

Copyright © 2022, 2026, Oracle and/or its affiliates.

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
   ```

Integration Activation Error Due to Change in JDK Location

If integration activation fails with the following error, this is likely the result of the agent installation using a JDK whose location has been changed (for example, removed). This can occur if the agent was installed and running with a version of the JDK whose location was removed and a newer version was installed in a different location. If the JDK installed with the agent is removed, ensure that you restart the agent with the newer version (and location) of the JDK.

```
Caused by: java.lang.Error: Circular loading of installed providers
detected at
java.nio.file.spi.FileSystemProvider.installedProviders(FileSystemProvider.jav

a:161)
at java.nio.file.FileSystems.newFileSystem(FileSystems.java:324)
at java.nio.file.FileSystems.newFileSystem(FileSystems.java:276)
at
oracle.cloud.cpi.agent.store.StoreUtils.extractZipBundle(StoreUtils.java:49)
at
oracle.cloud.cpi.agent.store.FileAgentStore.addActivation(FileAgentStore.java:
40)
at
oracle.cloud.cpi.agent.ActivationCpiCommand.doRequest(ActivationCpiCommand.jav

a:80)
```

Troubleshoot Network Connectivity Issues

Ensure that network connectivity is working correctly when persistent connectivity failures are encountered with an Oracle Integration instance.

> **Note**
>
> This does not apply to any intermittent failures because the connectivity agent is resilient to temporary conditions and recovers when the situation is resolved.

- Run the following command:

  ```
  nslookup hostname
  ```


Copyright © 2022, 2026, Oracle and/or its affiliates.

- Run the following command for a period of five minutes to also capture any transient failures:

```
while true;
do
curl https://Oracle_Integration_hostname/ic/home >> file.txt 2>&1;
sleep 1;
echo "trying again";
done;
```

HTTP 401 Unauthorized Error Occurs During Connectivity Agent Installation

If connectivity agent installation or restart fails with an `HTTP 401 Unauthorized` error, download the agent configuration file from Oracle Integration 3 and restart the connectivity agent.

Troubleshoot Private Endpoints

Get help troubleshooting issues with private endpoints.

Troubleshoot Integration Runtime

Learn about troubleshooting integration runtime in Oracle Integration.

Topics:

- An Integration Runs Slowly
- Fewer Integration Instances Ran Than Usual
- Using Basic Authentication to Call Services When the Identity Domain is Configured With Multi-Factor Authentication Causes a 401-Unauthorized Error
- Perform Remedial Actions When the Payload Exceeds the Size Limit
- Remove Unprintable Control Characters from the Incoming Payload
- Notification Action Returns a <from> value is invalid Error
- Synchronous Integration Timeout Failure
- Activity Stream Details
- Message Dequeuing Takes Three to Four Seconds in Asynchronous Integrations
- 403 Error When Accessing an API Protected Using OAuth
- Troubleshoot Certificate Import Failures

An Integration Runs Slowly

Several common issues can result in performance issues for an integration. For example, when the target application takes longer than expected to respond, the integration also runs slower than expected.

How to Determine If Your Integration Is Running Slowly

See Determine an Integration's Typical Processing Times, and then take the appropriate action:

Copyright © 2022, 2026, Oracle and/or its affiliates.

- If the integration has always run slower than expected, determine whether it contains a common pitfall.

  See Design Best Practices.

- If the integration has detectable performance outliers, follow the troubleshooting steps on this page.

Step 1. Investigate the Target Application

The target application for an invoke connection might be taking longer than usual to respond. **A slow target application is the most common cause of slow-running integrations.**

To determine whether the target application caused the slowness for your integration, check the logs of the application. The logs help you determine whether the application was experiencing an issue when Oracle Integration connected to it.

Step 2. Check the Number of Incoming Messages

When an integration receives a large number of messages to process, Oracle Integration adds the messages to a queue and processes them until it clears the queue. A long queue can impact the overall performance of your system. For example, a retailer might experience higher message volume during a promotion.

To see how many messages Oracle Integration processed in the last hour:

1. In the navigation pane, click **Observability**, then **Integrations**.
2. Click **Filter**, and update the following filtering requirements:
   - **Time window:** Last 1 hour.
   - **Status:** All.
3. Click **Apply**.
4. Review the numbers in the **Processed** column.

Keep in mind that the number of messages is just one factor in performance. Smaller messages that contain very large incoming payloads can also impact performance.

Step 3. Determine Whether the Message Counts Are Too High

1. Work with an administrator to check whether your organization purchased enough message packs for the messages that you're using. See Edit the Edition, License Type, and Message Packs of an Instance in *Provisioning and Administering Oracle Integration 3*.
2. If the integration is synchronous, determine whether the integration is trying to process too many requests at a time. See Calculate Requests Per Second in *Provisioning and Administering Oracle Integration 3*.

Step 4. Check for a Clustered Schedule (Schedule Integrations Only)

Schedule integrations can run slower than expected when too many integrations start at the same time.

To determine whether your schedule integrations are clustered together, see View the Calendar of Schedule Integration Runs.

Step 5. Look For Inefficiencies

If the integration processes records, you might be able to fine-tune the efficiency of the processing.

Copyright © 2022, 2026, Oracle and/or its affiliates.

Action for processing records	Recommendations
For-each action	You typically can process 10,000 or fewer records using the for-each action without experiencing performance issues. To process 10,000 or more records, use a for-each action within a parallel action. The parallel action allows you to process the records in parallel threads, thereby improving the performance of the integration.
Stage file action	You typically can read and write 10,000 or fewer records within a file using the stage file action without experiencing timeouts. To process a file with 10,000 or more records, use a stage file action within a parallel action. The parallel action allows you to read and write the records in chunks, thereby improving the performance of the integration.

Haven't Solved the Problem?

Get help from Oracle by entering a service request.

Fewer Integration Instances Ran Than Usual

Consider a scenario in which 100 integration instances typically run by a specific time of day, but only 80 integration instances have run by that time today. You need to quickly determine what happened with your missing integration instances.

> **Note**
>
> The Observability pages show information about the integrations that are running and have already run, not any missing instances. To identify whether any integration instances are missing, you must know the integrations that usually run.

Why Integrations Don't Run

An integration might not run for a number of reasons. For example:

- Someone deactivated an integration.
- The external application that triggers an integration is down.
- Someone changed the schedule for a schedule integration, such as changing it from running hourly to running daily.

To Troubleshoot Missing Integration Instances:

1. Determine how many integration instances have run so far.
   a. In the navigation pane, click **Observability**, then **Instances**.
   b. Click **Filter**, and update the filtering requirements for the page.
      For instance, update the time window to the last 1 hour, 6 hours, or 1 day.
   c. At the top of the page, review the count of the number of integration instances that ran.
2. If fewer integration instances have run than expected, troubleshoot the missing instances.
   For instance:
   - Check whether the integrations are active.

Copyright © 2022, 2026, Oracle and/or its affiliates.

- Check whether an external application that triggers an integration is down.
- Check whether someone changed the schedule for a schedule integration.
- If needed, consult the audit trail to determine who last updated an integration.

  See Check the Design Time Audit History for an Integration or Other Component.

Using Basic Authentication to Call Services When the Identity Domain is Configured With Multi-Factor Authentication Causes a 401-Unauthorized Error

If the identity domain is configured with multi-factor authentication (for example, user name / password + multi-factor authentication) and you use basic authentication to invoke Oracle Integration services, you receive a `401-Unauthorized` error. None of the adapters or the connectivity agent in Oracle Integration support multi-factor authentication.

As an alternative:

- Disable multi-factor authentication from the account with which you are using basic authentication.
- Use OAuth instead.

See Managing Multi-Factor Authentication.

Perform Remedial Actions When the Payload Exceeds the Size Limit

When the message payload in an integration exceeds the size limit, the message is rejected and an error message is displayed. The error message describes remedial actions to perform.

Copyright © 2022, 2026, Oracle and/or its affiliates.

Issue	Error Message	Remedial Action to Perform
Payload size limit exceeded	• When REST, File, SOAP, FTP, and SAP Adapter connections that support payload sizes of up to 50 MB exceed the limit: `CASDK-0059: Unable to process the response received from the client application. The response received by the OIC integration flow is rejected. The response received has been rejected since the content received of length 5723727000 bytes exceeds the maximum allowed threshold of 52428800 bytes (50MB).`  • When all other adapter connections that support payload sizes of up to 10 MB exceed the limit:  `CASDK-0059: Unable to process the response received from the client application. The response received by the OIC integration flow is rejected. The response received has been rejected since the content received of length 1723727000 bytes exceeds the maximum allowed threshold of 10485760 bytes (10MB).`	The preferred way to send large data sets is by using attachments. Check whether the target application exposes a SOAP API to return data sent as an attachment. If yes, the integration can be designed to configure an invoke connection using the SOAP Adapter to accept an MTOM attachment. The attachment can be an archive containing an XML/comma-separated value (CSV) document.
Attachment size limit exceeded	`CASDK-0059: Unable to process the response received from the client application. The response received by the OIC integration flow is rejected. The response received has been rejected since the content received of length bytes exceeds the maximum allowed threshold of 1073741824 bytes (1GB).`	Check whether the attachment can be broken and sent through multiple APIs. For example, if the attachment is a ZIP archive containing a large CSV file and unstructured documents, have one API that returns the CSV file and another API that retrieves the unstructured documents.

For additional details about 50 MB payload support, limits, and best practices, see Service Limits in *Provisioning and Administering Oracle Integration 3*.

Remove Unprintable Control Characters from the Incoming Payload

Oracle Integration does not support certain non-printable control characters in the following range.

```
['\u0000', '\u0001', '\u0002', '\u0003', '\u0004', '\u0005', '\u0006',
'\u0007', '\u0008', '\u000b', '\u000c', '\u000e', '\u000f', '\u0010', '
```


Copyright © 2022, 2026, Oracle and/or its affiliates.

```
\u0011', '\u0012', '\u0013', '\u0014', '\u0015', '\u0016', '\u0017', '\
u0018', '\u0019']
```

The presence of these control characters in the incoming payload results in the following error during processing:

```
{
  "type" : "10.4.1",
  "title" : "exception occurred while translating the request from native to
xml.",
  "detail" : "oracle.cloud.connector.api.CloudInvocationException:
ORABPEL-15235\n\nTranslation Failure.\nFailed to translate JSON to XML.
org.codehaus.jackson.JsonParseException: Illegal unquoted character
((CTRL-CHAR, code 1)): has to be escaped using backslash to be included in
string value\n at [Source: java.io.BufferedReader@.#; line: 2, column:
22]\nThe incoming data does not conform to the NXSD schema. Please correct
the problem.\n",
  "o:errorCode" : "TRANSLATION-ERROR-00",
  "o:errorDetails" : [ {
    "type" : "NA",
    "instance" : "NA",
    "title" : "NA",
    "o:errorPath" : "NA",
    "o:errorCode" : "NA"
  } ]
}
```

If you receive this error and are using the REST Adapter, you can use the **Skip Control Characters** property in the mapper. See Set REST Adapter Connectivity Properties in the Mapper. Otherwise, contact Oracle Support for information on how to preprocess incoming JSON payloads and filter unprintable control characters.

Notification Action Returns a <from> value is invalid Error

If the notification action in an integration validates successfully, but the following error occurs during integration runtime, the value specified in the **Parameter Value** field of the notification action may be incorrect.

```
:
summary=<summary>The <from> value is invalid.
Xpath expression associated with <from> in copy assign activity is invalid.
There is an user error that results in missing element value(s) in the xpath
query.
Please review the payload and modeling to ensure that all elements defined in
the <from> xpath query have valid non-null values .
</summary>
:
,query=<query>fn:current-dateTime()</query>
```

The error can be confirmed by viewing errors for the integration instance on the Instances page or the response of a REST testing tool such as `Postman`.

Copyright © 2022, 2026, Oracle and/or its affiliates.

Verify that the value specified in the **Parameter** section of the notification action is correct. For example, this value may require conversion to a string:

```
string(fn:current-dateTime())
```

Synchronous Integration Timeout Failure

If a synchronous integration fails to complete within 300 seconds, the following error appears in the activity stream that is viewable from the Instances page or Errors page.

```
CompositeException in:
io.reactivex.internal.operators.single.SingleOnErrorReturn$OnErrorReturn.onErr
or2 exceptions occurred
```

No details are returned when the call is processed (for example, no `502` status error is returned).

Optimize your integration and/or the endpoint you invoke to complete instance processing in under 300 seconds.

See Service Limits in *Provisioning and Administering Oracle Integration 3*.

Activity Stream Details

You can enable tracing with the payload, view the activity stream logs on the Instances page, and download the activity stream logs with the REST APIs.

- Enable Tracing with the Payload and View the Activity Stream on the Instances Page
- Download the Activity Stream Logs with the REST APIs

Enable Tracing with the Payload and View the Activity Stream on the Instances Page

Documentation is available that describes how to enable tracing with the payload and view the activity stream for integrations on the Instances page:

- See Manage Tracing Levels on Integrations
- See Activate an Integration

Download the Activity Stream Logs with the REST APIs

Documentation is available that describes how to download activity stream logs in Oracle Integration or with the REST APIs:

- See Download a Log File (by specifying the `icsflowlog` parameter)

> **Note**
>
> If you decide to use absolute values (10 MB, instead of a certain size), downloading the file downloads the latest 10 MB file only.

Copyright © 2022, 2026, Oracle and/or its affiliates.

Message Dequeuing Takes Three to Four Seconds in Asynchronous Integrations

You may observe that Oracle Integration sometimes takes three to four seconds to go from `Message Received:Instance created and enqueued for processing` to `message dequeued and processed by Trigger` *`trigger_name`*. This increases the end-to-end processing time of a particular instance.

This is the expected behavior with asynchronous requests. Asynchronous requests are processed using a queue. The first two entries in the flow logs show the enqueue and dequeue times. Therefore, a few seconds lag is expected.

403 Error When Accessing an API Protected Using OAuth

OAuth enables Oracle Integration to access a user's resources on their behalf. REST and other REST-based adapters (such as Google Calendar Adapter, Microsoft Office 365 Outlook Adapter, and others) often access APIs that are protected using OAuth. If these calls result in a `403` error (forbidden), it usually indicates a lack of permissions to access the API.

You need to carefully examine your OAuth configuration and work with the target API owner/administrator to resolve this issue. There may be several conditions leading to this, but the exact cause is best described by the authorization server. Some probable causes of `403` errors are as follows:

- The access token was procured for a scope that does not cover the API being accessed.
- During the phase to provide consent, the user that provided the consent is not the resource owner or owner of the API being accessed.
- The access token expired or is not valid to access the resource any longer.
- The authorization server revoked access privileges to a particular resource.

Troubleshoot Certificate Import Failures

The certificate that you receive from a Certificate Authority (CA) such as Verisign, Entrust, or others can fail during import into Oracle Integration with the following error in the log file:

```
java.security.cert.CertificateException: Unable to initialize,
java.io.IOException: extra data given to DerValue constructor
```

To resolve this issue, either:

- Obtain a fixed certificate from the CA.
- Convert the certificate to a Distinguished Encoding Rules (DER) binary certificate using tools such as `openssl`. For example:

  ```
  openssl x509 -outform der -in sfdc-client.crt -out sfdc-client.der
  ```


Copyright © 2022, 2026, Oracle and/or its affiliates.

Troubleshoot Integration Design Time

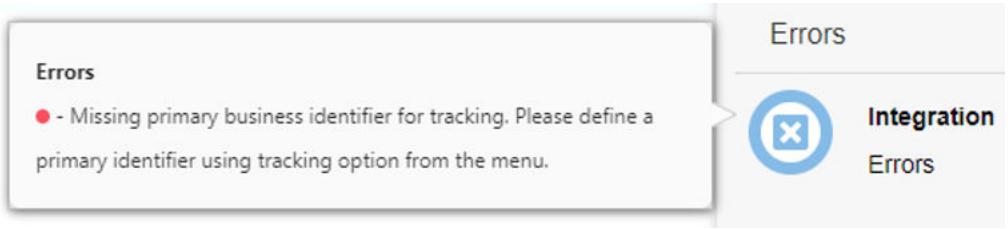
Learn about troubleshooting integration design time in Oracle Integration.

Topics:

- Missing a Primary Business Identifier in an Integration
- Unable to Call a replaceAll JavaScript Function
- 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
- Invoke Subprocess Integrations Using a Single REST API Connection
- Trigger Connection Cannot Receive an Empty Array in JSON if the Corresponding JSON Received Has an Empty JSON
- Object Storage Bucket Error Creation
- Manage Your Registered AI Engines

Missing a Primary Business Identifier in an Integration

You must configure at least one business identifier tracking field to make your integration ready for activation. This is known as the primary business identifier. Business identifiers enable you to track payload fields in messages during runtime. If you have not yet configured at least one business identifier, the following error message is displayed when you hover your cursor over the **Integration Errors** icon in the design canvas of your integration.



Configure at least one business identifier to remove this error message. See Assign Business Identifiers.

Unable to Call a replaceAll JavaScript Function

If you attempt to call a `replaceAll` JavaScript function in Oracle Integration, you receive the following error:

```
ode=<code>JavaScript execution error</code>\n,detail=<detail>Review
JavaScript error(s) and fix them:\nerror: null@3 -> TypeError: Cannot find
function replaceAll in object
```

The `replaceAll` function is not supported by the JavaScript engine in Oracle Integration.

Use one of the following alternatives to achieve `replaceAll` functionality:

Copyright © 2022, 2026, Oracle and/or its affiliates.

- Use the `replace` function with the global option, as follows:

```
function myFunction(Input) {
  var res = Input.replace(/<\/>/g,'');
 return res;
}
```

- Use the `split` and `join` functions instead of `replaceAll`.

Conflict with Using Variables of String Data Type and Applying Mathematical Comparator Expressions

All variables created in Oracle Integration are currently only string data types. Using these variables directly and applying mathematical comparator expressions results in unusual behavior.

For example, in an assign action, two variables, `var1` and `var2`, are created with values of `1` and `10`, respectively. In a while action, if you then set the condition as `$var1 < $var2`, the loop terminates after two iterations. The correct usage is to set the condition by wrapping the variables with a `number` XPath function: `number($var1) < number($var2)`. This loop iterates 10 times before terminating.

Reduce Network Calls that Slow Down Processes Caused By Writing Record By Record

Writing record by record using the **Append to Existing File** option on the Operations page of the Adapter Endpoint Configuration Wizard of the FTP Adapter creates too many network calls and eventually slows down the process. Instead, use a stage file action and write the records to a stage directory. Once all records are written, use the **List File** option in the stage file action and the FTP Adapter **Write File** option to transfer the file to an FTP location. This approach reduces processing time and prevents too many calls to the FTP server.

Invoke Subprocess Integrations Using a Single REST API Connection

Assume you follow a modular design approach in which you create separate integrations (think of them as subprocesses) to perform specific tasks. For example:

- File - Staging
- Translation
- Stage - Core
- Trigger Job Processes

Each of these subprocesses is created as a WSDL connector. However, having to maintain multiple connectors complicates the situation. Creating a single REST API and triggering subprocesses using API calls is ideally 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 this is a schedule integration. In addition, the schedule now API runs the specified integration immediately and this type of integration cannot be configured to take input parameters.

Copyright © 2022, 2026, Oracle and/or its affiliates.

As an alternative, you can create an application integration with a REST connection used as the trigger that can be configured to take input parameters. That integration can be invoked as a REST API itself, without the need for the schedule now API to invoke it. The URL for the schedule now API in the v2 version of the REST API is:

```
http://host:port/icsapis/v2/integrations/SAYHELLO%7c01.00.0000/schedule/jobs
```

That is, the URL contains `CODE|VERSION` to specify which integration to run.

Note that a SOAP connector is the only option available that can call subprocesses.

Trigger Connection Cannot Receive an Empty Array in JSON if the Corresponding JSON Received Has an Empty JSON

An adapter does not receive any element from the mapper if the element is an empty array. To get the element from the mapper in case of an empty array, change the XSL file manually.

1. Export the integration.
2. Manually change the mappings in the XSL file to ensure that the default elements are generated in the target even when the source elements do not exist or are empty. For example:

   Before editing the XSL file:

   ```
   <xsl:for-each
   select="$Cheers_API/nsmpr0:executeResponse/nsmpr1:response-wrapper/nsmpr1:
   employeeEarnedDetails/nsmpr1:earnedBadges" xml:id="id_24">
   <nstrgdfl:earnedBadges xml:id="id_23">
   <xsl:value-of select="." xml:id="id_25" />
   </nstrgdfl:earnedBadges>
   </xsl:for-each>
   ```

   After editing the XSL file:

   ```
   <nstrgdfl:earnedBadges xml:id="id_23">
   <xsl:for-each
   select="$Cheers_API/nsmpr0:executeResponse/nsmpr1:response-wrapper/nsmpr1:
   employeeEarnedDetails/nsmpr1:earnedBadges" xml:id="id_24">
   <xsl:value-of select="." xml:id="id_25" />
   </xsl:for-each>
   </nstrgdfl:earnedBadges>
   ```

3. Import the integration.

Object Storage Bucket Error Creation

If your object storage bucket is not in the default domain, you can receive the following error when creating the bucket on the Instance storage page.

```
An error occurred while saving storage changes
{ "servlet":"jersey", "message":"insufficient content written",
"url":"/ic/api/common/v1/storages", "status":"500" }
```


Copyright © 2022, 2026, Oracle and/or its affiliates.

Add the domain name in front of your user name in the **Username** field on the Instance storage page:

domain_name/username

Manage Your Registered AI Engines

If you previously registered AI engines in Oracle Integration (before June 2025), you can remove them.

AI features are supported in select regions, as mentioned in AI Feature Availability in *Provisioning and Administering Oracle Integration 3*.

To remove your registered AI engines:

1. In the navigation pane, click **Settings**, then **AI engines**.
2. Next to an entry, click ⋯, then click **Delete**.

After you delete all your AI engine entries, the **AI engines** settings page disappears.

Troubleshoot Schedule Integrations

Learn about troubleshooting schedule integrations in Oracle Integration.

Topics:

- Schedule Integrations Are Not Running on Time
- When a Schedule Integration Instance Gets Terminated

Schedule Integrations Are Not Running on Time

Schedule integrations may be queued and not run on time as scheduled for the following reasons.

- There is a running instance of the same schedule integration. No new instance of the same integration is run until the running instance completes.
- The throttling limit for schedule integrations has been reached. No new schedule integrations are run until the number of schedule integrations being run falls under the throttling limit.

> **Note**
>
> The throttling limit is automatically adjusted based on available system resources.

When a Schedule Integration Instance Gets Terminated

An instance of a schedule integration instance can be terminated by a user or Oracle Integration for the following reasons.

- A user cancels a schedule integration instance.
- A user call reactivates a schedule integration, which causes the currently-queued scheduled instance to terminate.

Copyright © 2022, 2026, Oracle and/or its affiliates.

- Scheduled flow processing exceeds six hours, which causes Oracle Integration to terminate the schedule integration instance.

Troubleshoot Integration Activations

Learn about troubleshooting integration activations in Oracle Integration.

Topics:

- Troubleshoot Integration Activations
- Delay in Message Processing of First Instance after Activation

Troubleshoot Integration Activations

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

Activation Failure Causes	Corrective Actions
WSDL issues: • Duplicate schema issues in the WSDL • WSDL/schema validation failures	1. Check the WSDL for duplicate schemas. 2. Check that the WSDL is valid. 3. If necessary, correct the WSDL in the application. 4. Reconfigure the integration by refreshing the endpoints. 5. Retry the activation.
Deployment timeout issues.	Retry the activation because the server may be slow to react.

Database issues:
• Reached the size limit:
`java.sql.SQLException: ORA-01654: unable to extend index`
• Database is down.
• Connection pool issues:
`Could not create pool connection for datasource 'SOALocalTxDataSource'. The DBMS driver exception was: Listener refused the connection with the following error: ORA-12516, TNS:listener could not find available handler with matching protocol stack`
	1. Check that the database is running and has enough space.
	2. If not, the database administrator must increase the database size limit or connection pool size.
	3. Retry the activation.

Copyright © 2022, 2026, Oracle and/or its affiliates.

Activation Failure Causes	Corrective Actions
A connection is down or was modified.	1. Check if the service/application is accessible. If it is down, retry the activation later. 2. If the connection is invalid, update the connection configuration. 3. Reconfigure the integration by refreshing the endpoints. 4. Retry the activation.
Managed server unavailable.	1. The administrator must restart the server. 2. Retry the activation.
If an integration contains too many actions (for example, it has 25 or more stage write actions), activation failure can occur: `RegularFileObject[/u01/data/domains/iscs /servers/integration_server1/dc/ 338c94 f1-d3ee-45b9-bbf8-43c9203df851/SCA-INF/ bpel/ StageCurrentGTHROFileForEachBpel/ src/orabpel/ stagecurrentgthrofileforeach bpel/BIN0.java]:51 code too large`	Optimize the integration. For example, split the integration into multiple integrations.

Delay in Message Processing of First Instance after Activation

After activation of an integration, there is a several second delay between when the instance is created and when the message is processed. This is the expected behavior.

On the first request for all integrations, the artifacts and back-end processes are lazy loaded as a first step. This can make for a longer time period spent between the first two steps shown in the activity stream when compared to subsequent invocations. This is the expected behavior for the initial request, and by design.

For example, you may see the following in the activity stream:

```
Sep 3rd 12:43:16.442 PM UTC 2020 Message received: Instance create
Sep 3rd 12:43:23 Message processed by Trigger
```

Note the seven second delay between when the message was received and processed.

Copyright © 2022, 2026, Oracle and/or its affiliates.

A

Oracle Integration Use Cases

Use Oracle Integration in scenarios such as the following.

Topics:

- Extract Content from an Invoice PDF Document with a Document Understanding Action
- Extract Information from a Custom Document with a Document Understanding Action
- Manage Oracle Cloud Infrastructure Buckets and Objects from an Integration
- Read Large XML Files Containing Multiple Namespaces
- Use a Stage File Write Operation with an Opaque Schema
- Test REST Adapter Trigger-Based Integrations with Multipart Attachments on the Configure and run page
- Calculate the Sum of a Column or the Count of Rows While Processing Large Files

Extract Content from an Invoice PDF Document with a Document Understanding Action

You can use AI to extract information from invoices, receipts, passports, and drivers licenses with a document understanding action. This section provides a high-level use case on how to exact information from an invoice in PDF format and convert it to JSON format in Oracle Integration.

Ensure that you have completed all prerequisites for using the document understanding action. See Extract Document Information with a Document Understanding Action.

1. Create an application integration.
2. Drag a REST Adapter into the integration canvas.
3. Configure the REST Adapter.
    a. On the Basic Info page, enter a name.
    b. On the Resource Configuration page, specify the necessary information. For example:
        - Enter a relative resource URL: **/process**.
        - Select the **POST** action.
        - Select **Configure a request payload for this endpoint**.
    c. On the Configure Request page, select **Binary** as the request payload format.
    d. Complete the wizard.
4. Click **Actions**  and drag the **OCI Document Understanding** action into the integration canvas.
5. Enter the following information in the wizard.
    a. Enter a name and optional description.

Copyright © 2022, 2026, Oracle and/or its affiliates.

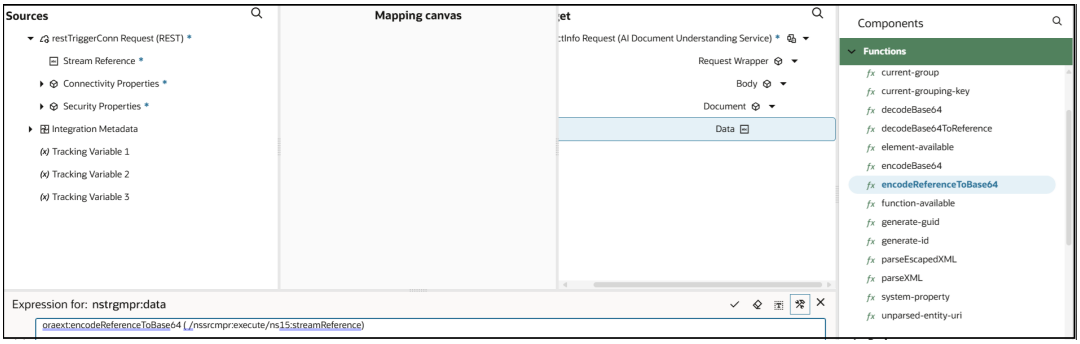
b. From the **Select Categories** list, select **Pre-trained Models**.

c. From the **Action** list, select **Analyze document**, then click **Continue**.

d. From the **Compartment Name** list, select the compartment in which your instance is running.

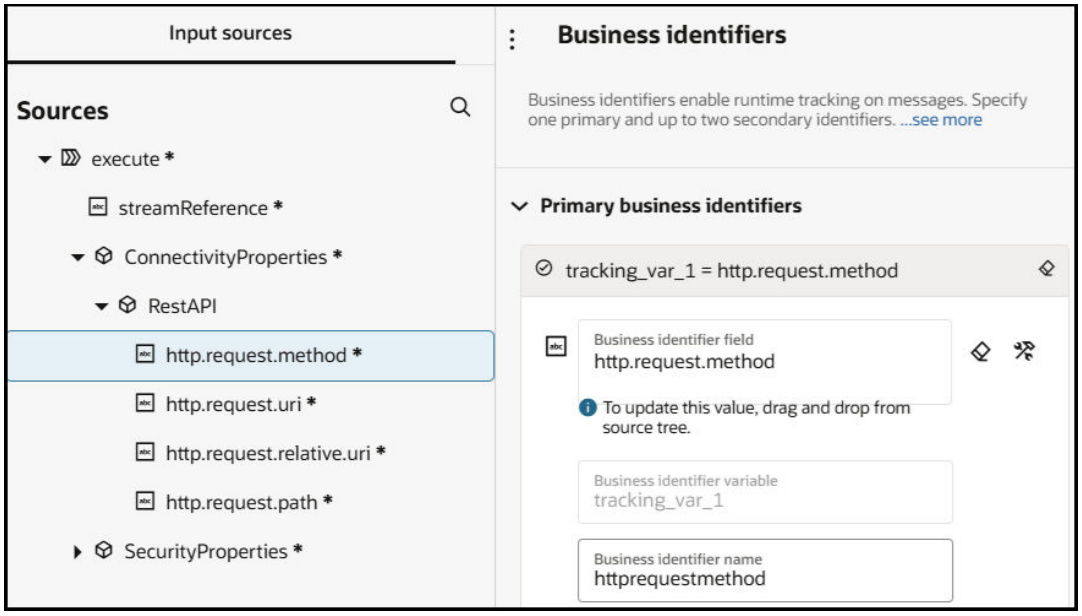
e. From the **Document Type** list, select **Invoice**.

6. Open the mapper. The invoice PDF document must be sent in base-64 format.
7. Expand the target **Document** element.
8. Right-click **Data**, and select **Create target node**.
9. In the target section, click **Functions**  .
10. In the Expression Builder, click **Design View**  .
11. In the **Functions** section, expand **Advanced**, and drag **encodeReferenceToBase64** into the Expression Builder. This step is required for extracting data from all supported document types (invoice, receipt, drivers license, and passport).
12. Map **Stream Reference** from the **Sources** section to the **encodeReferenceToBase64 ()** function in the Expression Builder.



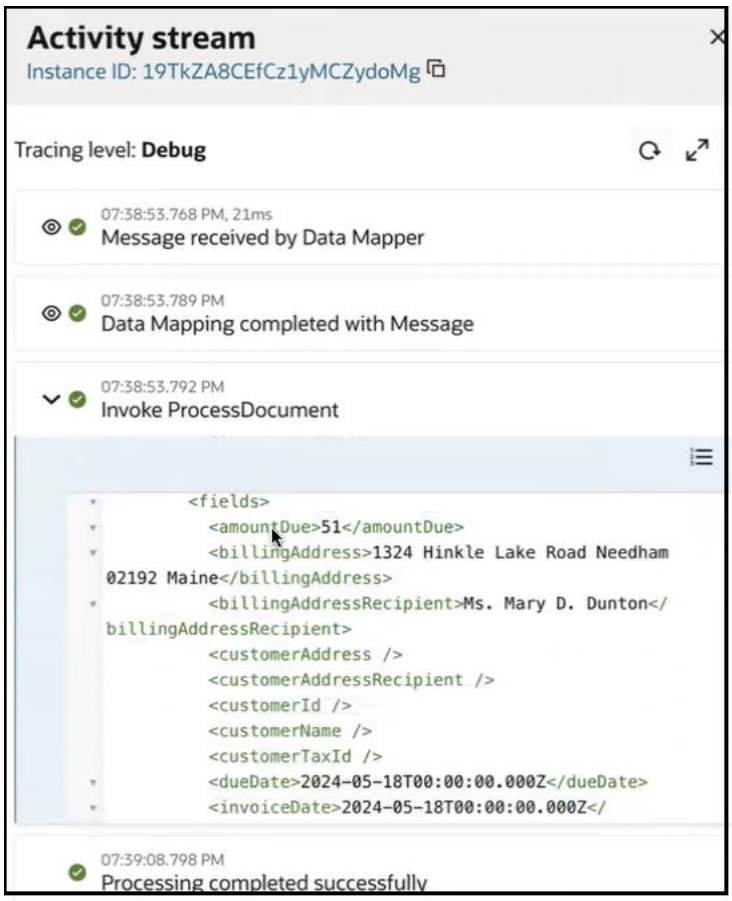
13. Return to the integration canvas. This action validates your expression.
14. Assign a business identifier.

Copyright © 2022, 2026, Oracle and/or its affiliates.



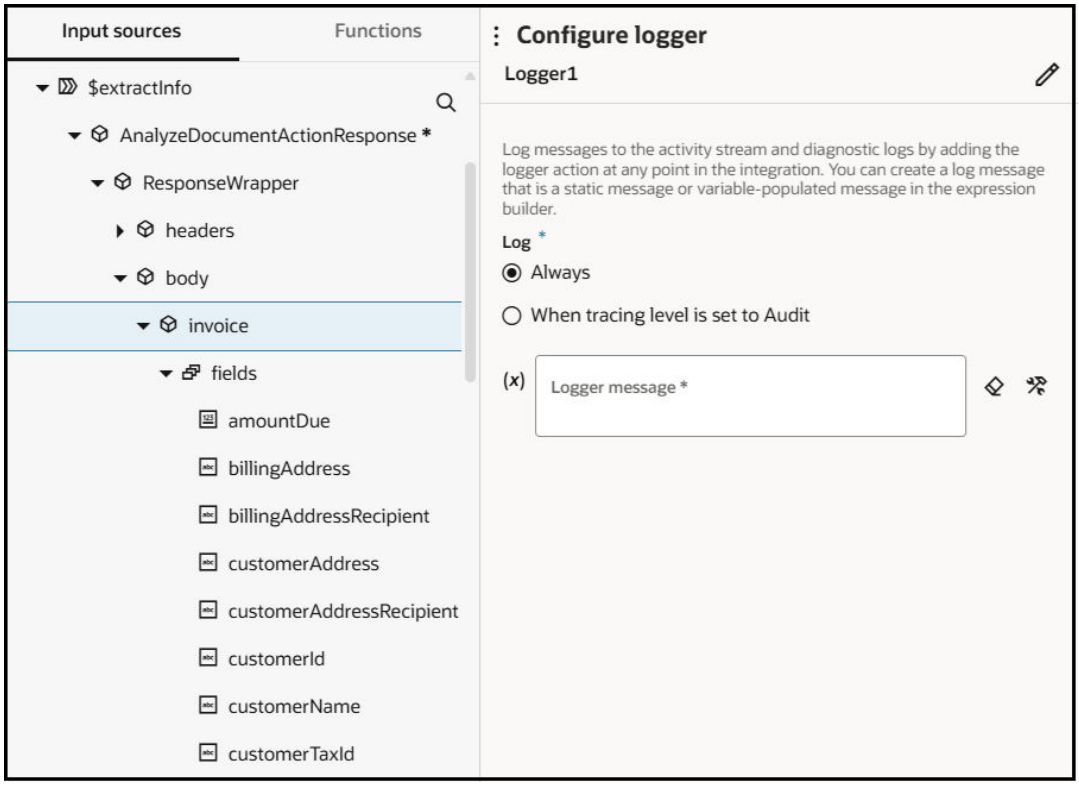
15. Save and activate the integration.
16. From the **Actions** ⋯ menu, select **Run**.
17. In the **Body** tab of the **Request** section, click **File**, then **Browse**.
18. Upload the PDF file of your invoice.
19. Click **Run**.
20. When processing completes, expand the invoke connection in the activity stream.
21. Scroll through the payload and note that the fields from the invoice PDF document have been converted to JSON format.

Copyright © 2022, 2026, Oracle and/or its affiliates.



A log action was not added to this integration. However, adding one to the end of the integration also enables you to see the fields that are returned as output by the Document Understanding API. An invoice returns the following specific fields. The fields returned are different and visible in the log action for each document type that you select in the wizard (receipt, drivers license, or passport). See Analyzing with the API.

Copyright © 2022, 2026, Oracle and/or its affiliates.



Extract Information from a Custom Document with a Document Understanding Action

You can use AI to extract information from a custom document with a document understanding action. This section provides a high-level use case on how to exact information from a custom document (for this example, extract an insurance quote from a JPG file) and convert it to key-pair values in JSON format.

Ensure that you have completed all prerequisites for using the document understanding action. See Extract Document Information with a Document Understanding Action.

1. Create an application integration.
2. Drag a REST Adapter into the integration canvas.
3. Configure the REST Adapter.
   a. On the Basic Info page, enter a name.
   b. On the Resource Configuration page, specify the necessary information. For example:
      - Enter a relative resource URL: **/test** is used for this example.
      - Select the **POST** action.
      - Select the following request/response options:
        – **Configure a request payload for this endpoint**
        – **Configure this endpoint to receive the response**
   c. On the Request page, select **Binary** as the request payload format.

Copyright © 2022, 2026, Oracle and/or its affiliates.

d. On the Response page, select **JSON Sample** as the response payload format, and specify the content:

```
{
"companyName":""
}
```

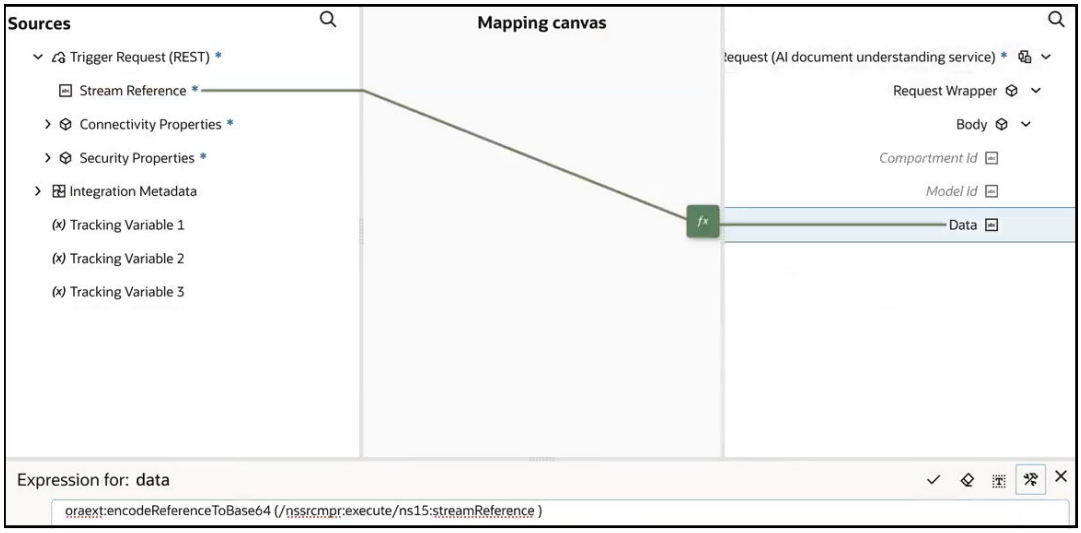
e. Complete the wizard.

4. Click **Actions** and drag the **OCI Document Understanding** action into the integration canvas.
5. Enter the following information in the wizard.
   a. Enter a name and optional description.
   b. From the **Select Categories** list, select **Custom models**, then click **Continue**.
   c. From the **Action** list, note that **Analyze document** is the only option available and cannot be deselected, then click **Continue**.
   d. Select the following information, then click **Continue**.

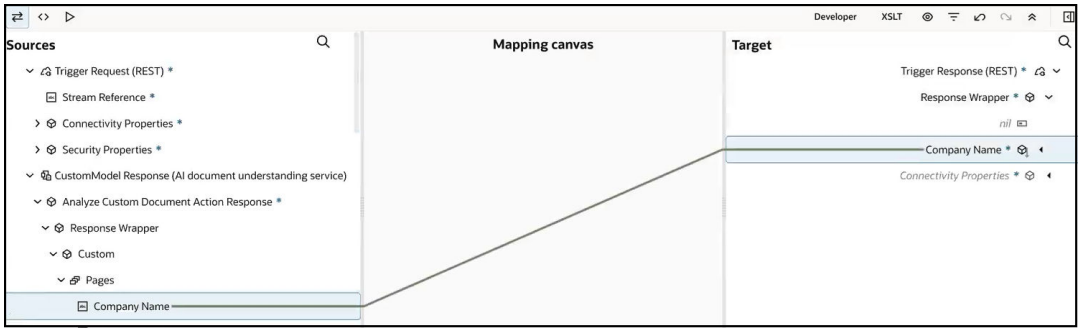
Element	Description
Compartment	Select the Oracle Cloud Infrastructure compartment in which Oracle Integration is installed.
Model	Select the model ID to use that you created in Oracle Cloud Infrastructure. See Custom Models.
Input source	Select **Inline** as the location from which to get the input document.

6. In the request mapper, expand the source and target sections. The insurance quote custom document must be sent in base-64 format.
7. Expand the target **Document** element.
8. Right-click **Data**, and select **Create target node**.
9. In the target section, click **Functions** .
10. In the Expression Builder, click **Design View** .
11. In the **Functions** section, expand **Advanced**, and drag **encodeReferenceToBase64** into the Expression Builder. This step is required for extracting data from the custom document.
12. Map **Stream Reference** from the **Sources** section to the **encodeReferenceToBase64 ()** function in the Expression Builder.

Copyright © 2022, 2026, Oracle and/or its affiliates.

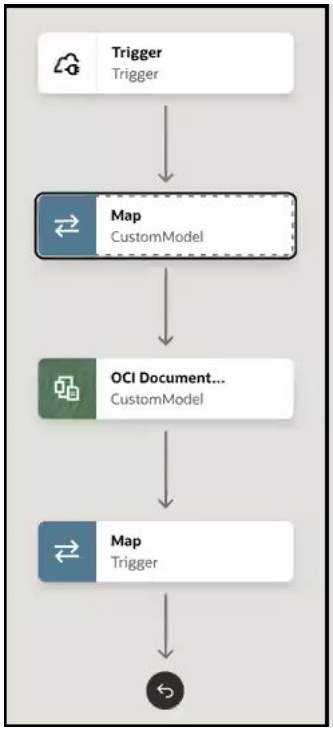


13. In the response mapper, expand the source and target sections.
14. Map the target **Company Name** element to the source **Company Name** element.



15. Return to the integration canvas. This action validates your expression.
16. Assign a business identifier.
17. Save and activate the integration.

Copyright © 2022, 2026, Oracle and/or its affiliates.



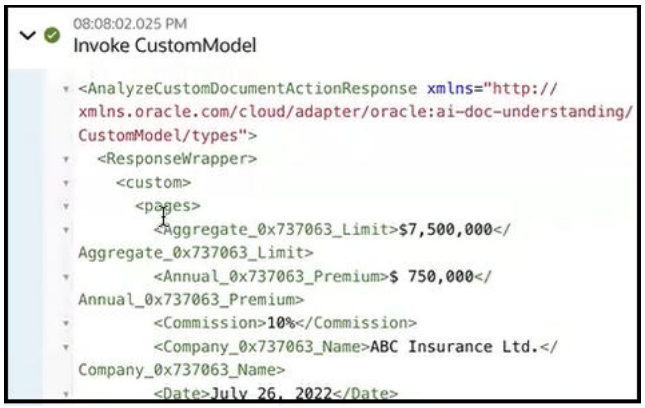
18. From the **Actions** ⋯ menu, select **Run**.
19. In the **Body** tab of the **Request** section, click **File**, then **Browse**.
20. Upload the JPG file that includes the insurance quote.
21. Click **Run**.
22. View the `companyName` returned in the response.



23. When processing completes, expand the invoke connection in the activity stream.

Copyright © 2022, 2026, Oracle and/or its affiliates.

24. Scroll through the payload and note that the fields from the insurance quote in the JPG file have been converted to key-pair values. Here is a partial view of the insurance quote details.



Manage Oracle Cloud Infrastructure Buckets and Objects from an Integration

You can manage Oracle Cloud Infrastructure Object Storage buckets and objects directly from the integration canvas through use of the **OCI Object storage** action. This section provides high-level use cases on how to create a bucket and upload objects directly from an integration.

- Create an Object Storage Bucket
- Upload an Object

Understand the concepts and satisfy all prerequisites before using this feature. See Invoke Oracle Cloud Infrastructure Object Storage from an Integration with an OCI Object Storage Action.

Create an Object Storage Bucket

1. Add a REST Adapter trigger connection to the integration.
   a. On the Basic Info page, enter a name, and click **Next**.
   b. On the Resource Configuration page, enter the following information.

Field	Description
What is the endpoint's relative resource URI	Enter the following: `/trigger`
What action do you want to perform on the endpoint	Select **POST**.
Select any options that you want to configure	Scroll to the bottom of the page and select the following options: • **Add and review parameters for this endpoint** • **Configure a request payload for this endpoint**

Copyright © 2022, 2026, Oracle and/or its affiliates.

c. On the Request Parameters page, create the following query parameters.

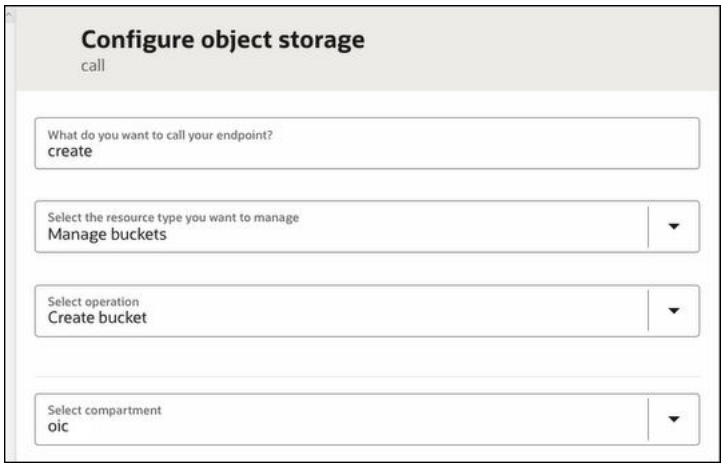
Name	Data Type
`bucketName`	string
`objectName`	string

d. On the Request page, create the following query parameters.

Field	Description
Select the request payload format	**Binary**
What is the media-type of Request Body?	**application/octet-stream**

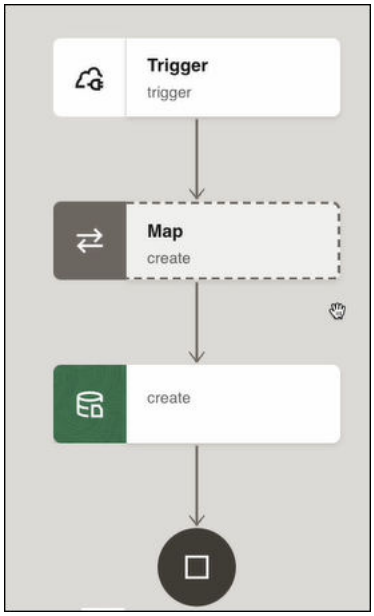
e. On the Summary page, click **Finish**.

2. Add an **OCI Object storage** action to the integration.
3. Enter a name.
4. Select **Manage buckets**.
5. From the **Select operation** list, select **Create bucket**.
6. Select the compartment in which to create the bucket.
For this example, an object storage bucket is selected for creation in the **oic** compartment in Oracle Cloud Infrastructure.

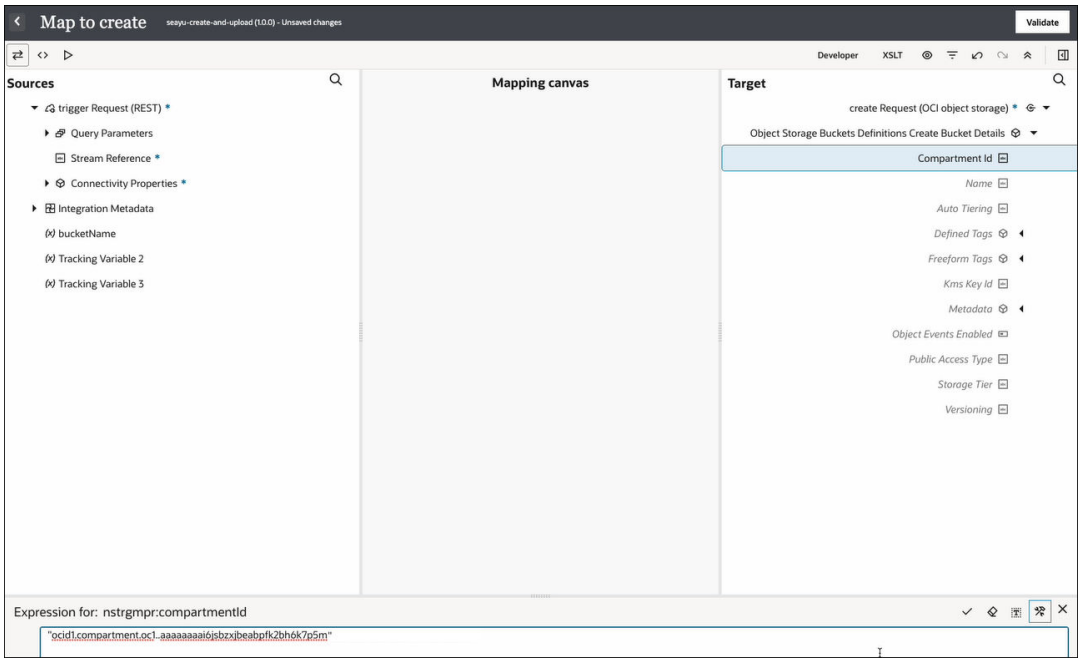


7. Click **Continue**.
8. Review your selections on the Summary page. The provided name, selected compartment, and the OCID for that compartment are displayed.
The integration looks as follows.

Copyright © 2022, 2026, Oracle and/or its affiliates.

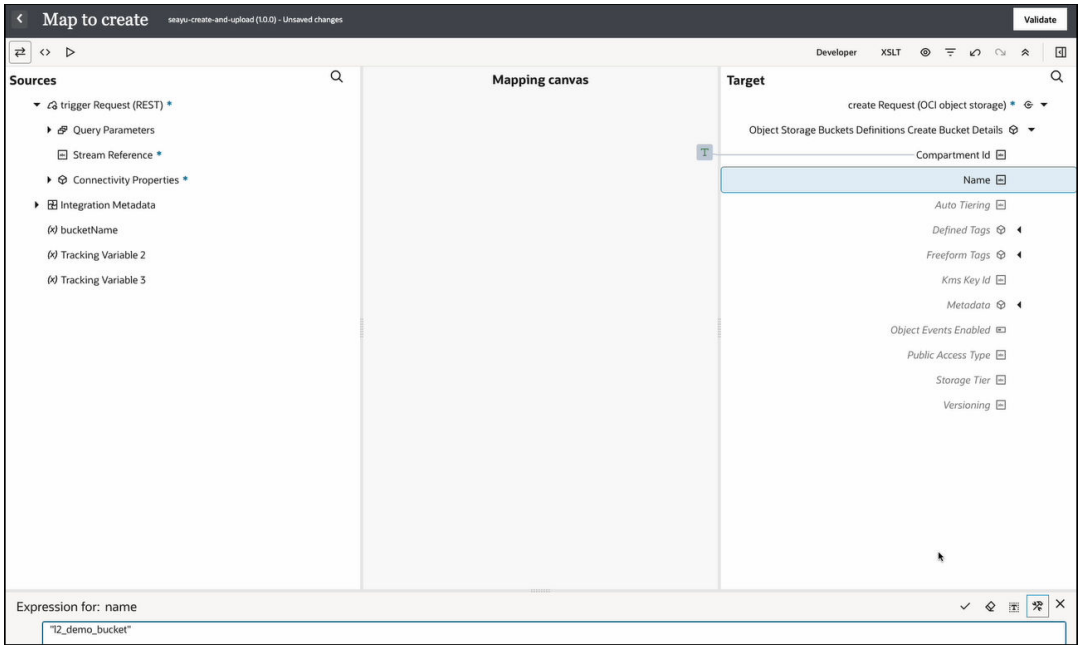


9. Double-click the mapper.
10. In the **Target** section, right-click the **Compartment ID** element and select **Create target node**.
11. In the Expression Builder, specify an OCID value. The value you map to the **Compartment ID** takes precedence over the design-time selection, even if it's a different OCID or a blank value (empty string).

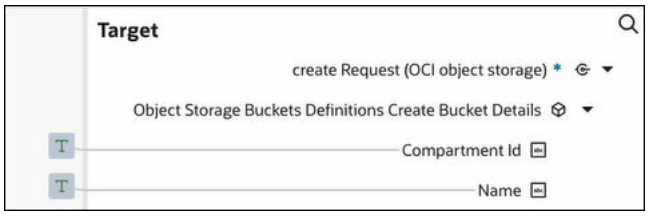


12. In the **Target** section, right-click the **Name** element and select **Create target node**.
13. In the Expression Builder, enter the name to use for the new bucket (for this example, `l2_demo_bucket` is entered).

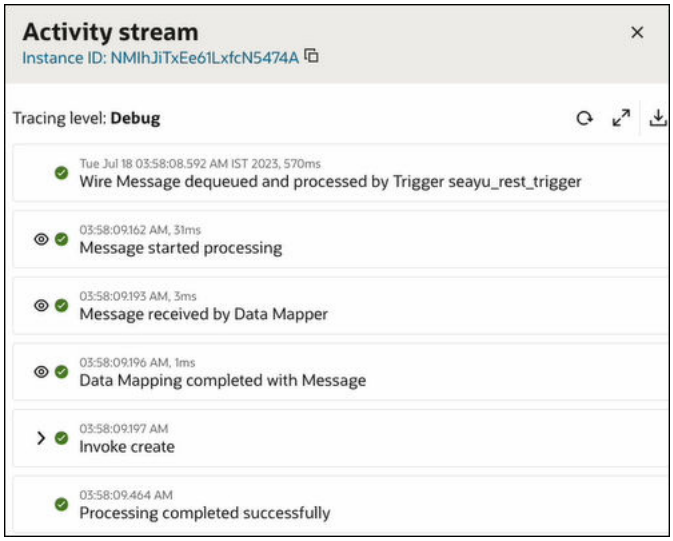
Copyright © 2022, 2026, Oracle and/or its affiliates.



Configuration of the mapper is now complete.

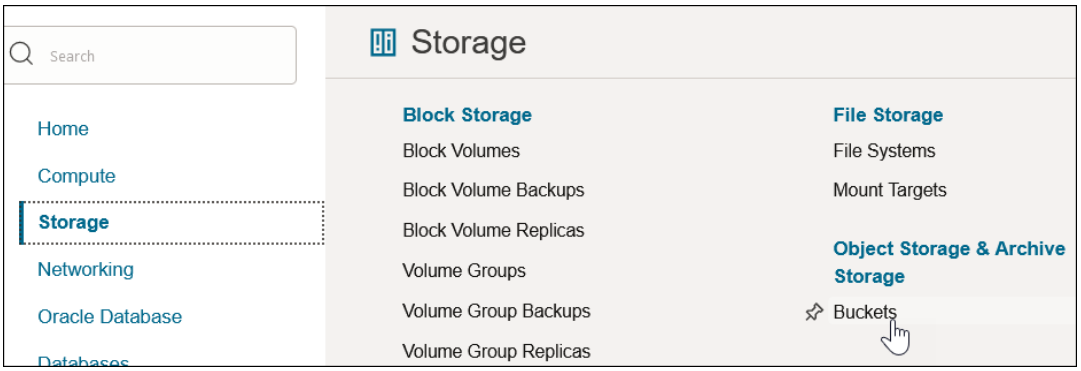


14. Save your changes, and exit the integration.
15. Hover your cursor over the integration, and select **Actions** ⋮, then **Run**.
16. On the Configure and run page, click **Run**.
17. Review the activity stream.



Copyright © 2022, 2026, Oracle and/or its affiliates.

18. Go to the Oracle Cloud Console.
19. In the navigation pane, select **Storage**, then **Buckets**.



The new bucket (for this example, **l2_demo_bucket**) is visible in the table.



Upload an Object

1. Add a REST Adapter trigger connection to an integration.
   a. On the Basic Info page, enter a name, and click **Next**.
   b. On the Resource Configuration page, enter the following information.

Field	Description
What is the endpoint's relative resource URI	Enter the following: `/trigger`
What action do you want to perform on the endpoint	Select **POST**.
Select any options that you want to configure	Scroll to the bottom of the page and select the following options: • **Add and review parameters for this endpoint** • **Configure a request payload for this endpoint**

   c. On the Request Parameters page, create the following query parameters.

Copyright © 2022, 2026, Oracle and/or its affiliates.

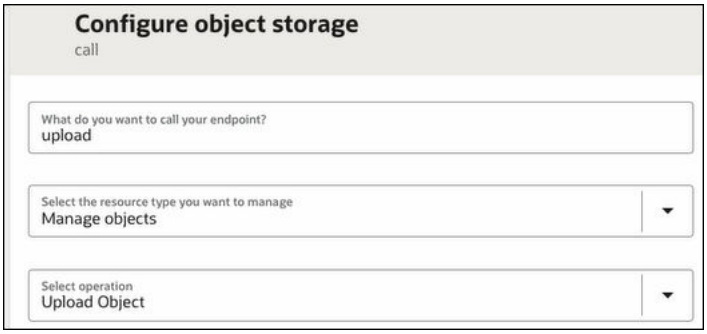
Name	Data Type
bucketName	string
objectName	string

d. On the Request page, create the following query parameters.

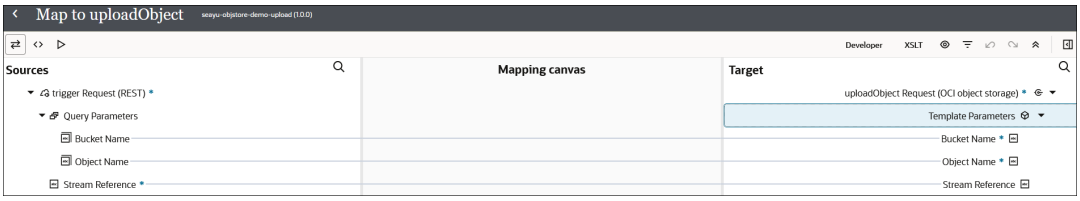
Field	Description
Select the request payload format	**Binary**
What is the media-type of Request Body?	**application/octet-stream**

e. On the Summary page, click **Finish**.

2. Add an **OCI Object storage** action to an integration.
   a. Enter a name.
   b. Select **Manage objects**.
   c. From the **Select operation** list, select **Upload Object**, then click **Continue**.

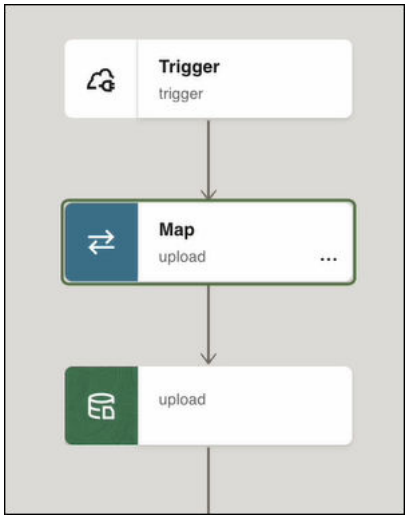


   d. Review your selections on the Summary page, then click **Finish**.
3. Click the mapper.
4. Perform the following source-to-target mappings.
   - Map the source **Bucket Name** to the target **Bucket Name**.
   - Map the source **Object Name** to the target **Object Name**.
   - Map the source **Stream Reference** to the target **Stream Reference**.

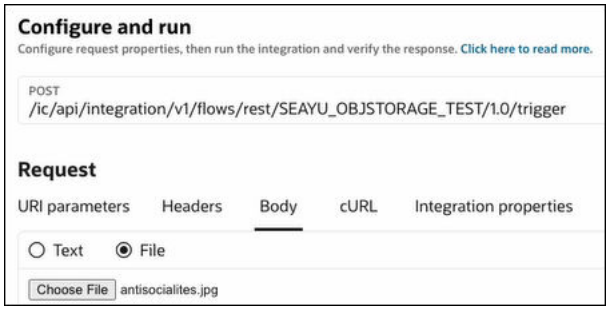


5. Save the integration.
   The integration looks as follows.

Copyright © 2022, 2026, Oracle and/or its affiliates.

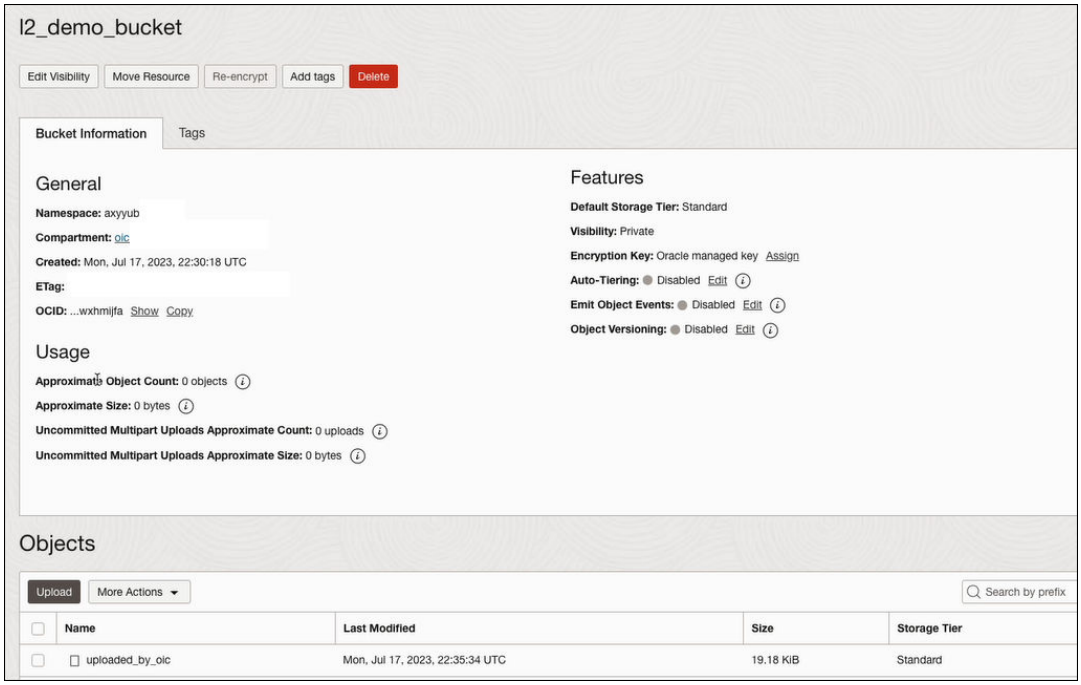


6. Exit the integration canvas.
7. Hover your cursor over the integration, and select **Actions** ⋮, then **Run**.
8. In the **Request** section, click **File** and upload the object to add to Oracle Cloud Infrastructure (for this example, a jpg file), then click **Run**.



9. Go to the Oracle Cloud Console.
10. In the navigation pane, select **Storage**, then **Buckets**.
11. Click the bucket in which you uploaded the object (for this example, **l2_demo_bucket**). The object is displayed in the **Objects** section. Note that the object is renamed from what you uploaded on the Configure and run page.

Copyright © 2022, 2026, Oracle and/or its affiliates.



Read Large XML Files Containing Multiple Namespaces

You can use a stage file action to read a large XML file that contains multiple namespaces. You upload a ZIP archive file containing the contents. While this use case describes how to use the FTP Adapter with the stage file action, other adapters can receive an XML document as an attachment in the response (for example, the SOAP Adapter, REST Adapter, and Oracle Taleo Enterprise Edition Adapter).

The following XML example shows an export of a candidates list obtained from Oracle Taleo Enterprise Edition. The XML file includes a repeating structure and contains more than one namespace in the document.

Copyright © 2022, 2026, Oracle and/or its affiliates.

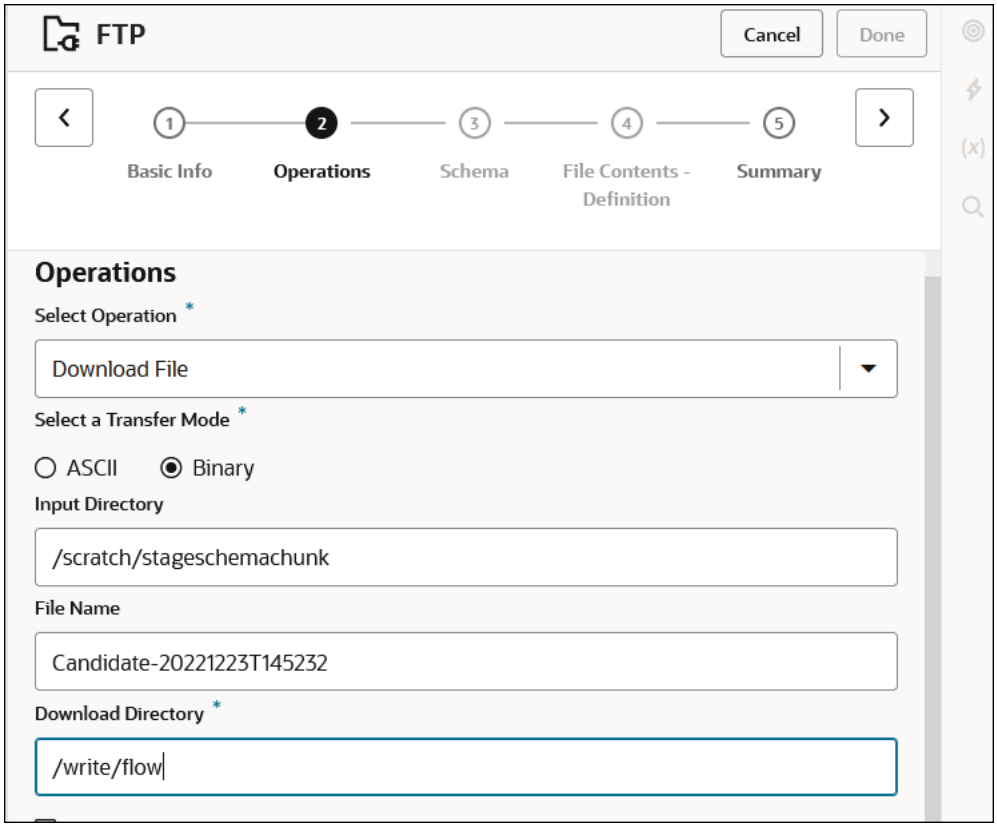


```
<Document xmlns="http://www.taleo.com/ws/integration/toolkit/2011/05">
    <Attributes>
        <Attribute name="count">3327</Attribute>
        <Attribute name="duration">0:00:01.443</Attribute>
        <Attribute name="entity">Candidate</Attribute>
        <Attribute name="mode">XML</Attribute>
        <Attribute name="version">http://www.taleo.com/ws/tee800/2009/01</Attribute>
    </Attributes>
    <Content>
        <ExportXML xmlns="http://www.taleo.com/ws/integration/toolkit/2005/07/action/export">
            <record>
                <field name="Number">10160</field>
                <field name="FirstName">Performance</field>
                <field name="LastName">HR Admin</field>
                <field name="EmailAddress">hradmin@invalidmail.com</field>
                <field name="CreationDate">2011-11-16T14:36:14-05:00</field>
            </record>
            <record>
                <field name="Number">10163</field>
                <field name="FirstName">Performance</field>
                <field name="LastName">Manager 1</field>
                <field name="EmailAddress">manager1@invalidmail.com</field>
                <field name="CreationDate">2011-11-16T14:42:10-05:00</field>
            </record>
            <record>
                <field name="Number">10261</field>
                <field name="FirstName">Performance</field>
                <field name="LastName">Matrix Manager</field>
                <field name="EmailAddress">tpmatrix@invalidmail.com</field>
                <field name="CreationDate">2012-03-12T08:28:44-04:00</field>
            </record>
            <record>
```

This use case provides a high-level overview of the steps to read a large XML file with multiple namespaces in segments in an integration. Two FTP Adapters are used to download a large ZIP file from a remote FTP server and write the staging file to a remote FTP server.

1. Add an FTP Adapter to a schedule integration.
2. Configure the FTP Adapter with the **Download File** operation and specify the input directory, file name, and ZIP file download directory.

Copyright © 2022, 2026, Oracle and/or its affiliates.



3. Add a stage file action to the integration and configure it as follows:
   a. On the Configure Operation page, configure the following options:
      - Select **Read File in Segments** from the **Choose Stage File Operation** list.
      - Select **Yes** for **Configure File Reference**.
      - Specify the file reference for the ZIP file downloaded from the FTP server in the **Specify the File Reference** section.
   b. On the Schema Options page, select **XML schema (XSD) document**.
   c. On the Format Definition page, configure the following options:
      - Click **Browse** to upload the archive of the schemas (XSDs) that represents the message structure.
      - Select the root element that contains the repeated element.
      - Click the **Edit** icon to select the repeating batch element in the Expression Builder.

      The Summary page shows your selections.
4. Add a second stage file action to write the records in comma-separated value (CSV) format.
   a. On the Configure Operation page, configure the following options:
      - Select **Write File** from the **Choose Stage File Operation** list.
      - Specify the file name of the CSV file in the **Specify the File Name** section.
      - Specify the output directory in the **Specify the Output Directory** section.
      - Select the **Append to Existing File** check box.

Copyright © 2022, 2026, Oracle and/or its affiliates.

b. On the Schema Options page, select **Sample delimited document (e.g. CSV)**.

c. On the Format Definition page, configure the following options:

- Click **Browse** to select the sample CSV file.
- Specify the record and record set name.

5. Perform the mapping between the two stage file actions.
6. Add a second FTP Adapter to write the staging file to an output directory on a remote FTP site.
7. Design the remaining parts of the integration.

Use a Stage File Write Operation with an Opaque Schema

You can use an opaque schema in a stage file action Read File or Write File operation without concern for a schema for the file. The only condition is that whatever is sent to the opaque element in the opaque schema must be base64-encoded data.

For the following use case:

- The integration downloads a zip file using the FTP Adapter Download File operation and extracts and processes each file.
- The files are staged temporarily in Oracle Integration using a stage file action List Files operation.
- The files are iterated over using a for-each action.
- The files are written to a staged directory using a stage file action Write File operation with an opaque schema.
- The files are written to an output directory using the FTP Adapter Write File operation.

1. Add a stage file action to an integration. See Configure a Stage File Action.
2. On the Configure Operation page, select the **Write File** operation and specify the file name and output directory.
3. On the Schema Options page, select **XML schema (XSD) document**.
4. On the Format Definition page, make the following selections.

   a. Select a new file (for this example, **opaque_schema.xsd** is selected).

      The contents of the XSD file are as follows for this example:

```
<?xml version = '1.0' encoding = 'UTF-8'?>
<schema targetNamespace="http://xmlns.oracle.com/pcbpel/adapter/opaque/"
            xmlns="http://www.w3.org/2001/XMLSchema" >
      <element name="opaqueElement" type="base64Binary" />
    </schema>
```

   b. Select the schema element (for this example, **opaqueElement** is selected).

5. Map the **FileReference** of the for-each repeating element (**ICSFile**) to **opaqueElement** of **StageFileWrite** using an **encodeReferenceToBase64** function.

Copyright © 2022, 2026, Oracle and/or its affiliates.

Test REST Adapter Trigger-Based Integrations with Multipart Attachments on the Configure and run page

You can test REST Adapter trigger-based integrations with multipart attachments on the Configure and run page. This section provides high-level use cases for testing the following multipart attachment types.

- multipart/form-data
- multipart payload with structured data (for this example, JSON)

Test Multipart/Form-Data

1. Configure a REST Adapter as a trigger connection and select the following options on the Request page:
   - **Request is multipart with payload**
   - **Multipart request is of type multipart/form-data with HTML form payload**
2. Design the remaining parts of your integration. For this example, the integration includes a REST Adapter trigger connection, notification action, mapper, and return action.
3. Activate the integration.
4. Hover over the REST Adapter trigger connection-based integration to run.
5. Click **Actions** ⋮, then select **Run**.
6. In the **URI Parameters** section of the Configure and run page, specify values for the parameters.
7. In the **Headers** section, view the header details.
8. In the **Body** section, select the attachments file to upload and specify a text description for the email.
9. Click **Run** to invoke this REST Adapter-triggered integration.
   The activity stream results are displayed on the right side of the page.
10. Check your response email for the file attachment you selected and the text description you specified.

    ```
    This document describes oracle rest standards
    ```

Test a Multipart Payload with JSON Structured Data

1. Configure a REST Adapter as a trigger connection and select the following options on the Request page:
   - **Request is multipart with payload** as the multipart attachment processing option.
   - **JSON Sample** as the request payload format.
   - **multipart/mixed** or **multipart/form-data** as the media type of the request body.
2. Design the remaining parts of your integration. For this example, the integration includes a REST Adapter trigger connection, notification action, mapper, and return action.
3. Activate the integration.
4. Hover over the REST Adapter trigger connection-based integration to run.

Copyright © 2022, 2026, Oracle and/or its affiliates.

5. Click **Actions** ⋮, then select **Run**.
6. In the **URI Parameters** section of the Configure and run page, specify values for the parameters.
7. In the **Body** section, select the attachments file to upload and specify a text description for the email.
8. Click **Run** to invoke this REST Adapter-triggered integration.
   The activity stream results are displayed on the right side of the page.
9. Check your response email for the file attachment you selected and the text description you specified.

   ```
   Oracle rest standards document
   ```

The Configure and run page provides a variety of testing features. See Test Integrations from Outside the Integration Canvas.

Calculate the Sum of a Column or the Count of Rows While Processing Large Files

You can process large comma-separated value (CSV) files (up to 1 GB in size) using the Read File In Segments operation of a stage file action. You may also have a requirement to calculate the sum of a column or the count of rows.

Consider the following payload. Assume you want to calculate the sum of all the values in the column **Amount1**. You may typically declare a variable upstream of the stage file action and keep updating this variable with the computation done in each chunk of the stage file action.

Sno	Amount1	Amount2	Amount3	Amount4
Inte	Inte	Inte	Inte	Inte
Man	Man	Man	Man	Man
1	100	100	10.0	600
2	200	100	10.1	600
3	300	100	10.2	100
4	400	100	10.3	200

However, updating the upstream variables inside the stage file action Read File in Segments operation impacts performance and does not allow stage file action processing in parallel. You observe the following warning message in the integration canvas:

```
Stage File Read File in Segments includes action that will result in segments
being processed sequentially
```

As a solution for this use case, perform the following steps to use the aggregate functions sum and count while processing larger files:

Copyright © 2022, 2026, Oracle and/or its affiliates.

1. Add a stage file action in the integration canvas and select the **Read File in Segments** operation to read the large file.
2. Inside the stage file action that uses the **Read File In Segments** operation, create a second stage file action that uses the **Write File** operation and select the **Append to Existing File** check box.
3. Get the count and sum of each segment and write the results into the second stage file action.

   - `count ($StageSegments/.../Record)`

   - `sum ($StageSegments/..Record/ns22:Amount1)`

4. After completion of the stage file action using the **Read File in Segments** operation, create a stage file action that uses the **List File** operation to list the file that was created in Step 2. The response of the stage file action **List File** operation contains the file size. The file size determines the next action to perform.
5. If the file size is less than 10 MB, read the file with a stage file action that uses a **Read Entire File** operation and create an assign action with variables to store the grand totals.
6. Otherwise, repeat Step 1 through Step 4 with the new file.

Copyright © 2022, 2026, Oracle and/or its affiliates.