

Oracle® Cloud

Using Oracle Cloud Infrastructure Process Automation



F34768-25
March 2025



Oracle Cloud Using Oracle Cloud Infrastructure Process Automation,

F34768-25

Copyright © 2022, 2025, 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

1 Introduction

Overview of Oracle Cloud Infrastructure Process Automation	1-1
Process Automation Product Types	1-2
Set Up Process Automation as an Individual Service	1-3
Use Process Automation with Fusion-Based Oracle Cloud Applications	1-3
Prerequisites	1-3
Synchronize Users From Fusion Applications Identity Provider to IDCS Stripe	1-4
Assign IDCS Application Roles	1-5
Get the Process Automation Instance URL	1-6
Example of Required User Created in IDCS	1-7
Use Process Automation with Oracle Integration 3	1-8

2 Work with Applications

Applications at a Glance	2-1
Add and Edit Applications	2-3
Create an Application	2-3
Clone an Application	2-4
Rename an Application	2-4
Export an Application	2-5
Import an Application	2-5
Work with Snapshots	2-6
Work with Versions	2-7
Delete an Application Version	2-8
Activate Applications	2-8
Activate an Application Version	2-9
Deactivate an Application Version	2-10
View Application API Endpoints	2-11
View Activated Application Versions	2-12
Work with Process Applications	2-13
Create a Process Application	2-13
Add Design Components	2-13
Activate and Test Process Applications	2-14
Localize Applications	2-15

Configure Searchable Fields Based on Business Data	2-17
Configure Roles for Process Applications	2-18
About Process Application Roles	2-19
Users, Groups, External Applications and Permissions	2-20
Work with Roles in Designer	2-22
Create an Application or Global Role	2-22
Link a Global Role to Your Application	2-23
Assign Roles to User Tasks in Structured Processes	2-24
Assign Roles to User Tasks in Dynamic Processes	2-25
Enable Users to Start Your Application	2-25
Remove Roles in Designer	2-26
Work with Decision Applications	2-26
Create a Decision Application	2-27
Activate and Use Decision Applications	2-27
Create an Application from a Recipe	2-28

3 Design Structured Processes

Typical Workflow for Designing a Structured Process	3-1
Use the Structured Process Editor	3-2
Add and Edit Swimlanes	3-4
Create a Structured Process	3-5
Clone a Structured Process	3-5
Delete a Structured Process	3-6
Work with Activities	3-6
Add Activities	3-8
Edit Activity Properties	3-8
Find Activities in the Palette	3-9
Add Human Interaction	3-9
Work with Events	3-10
Timer Start Event	3-10
None Start Event	3-12
Form Start Event	3-12
Message Start Event	3-13
Timer Catch Event	3-13
End Event	3-14
Error End Event	3-14
Terminate End Event	3-15
Error Boundary Event	3-15
Control Process Flow with Gateways	3-15
Exclusive Gateway: Take Only One Path (Data Condition)	3-16
Inclusive Gateway: Take One or More Paths	3-17

Parallel Gateway: Take All Paths Simultaneously	3-18
Create a Gateway	3-19
Connect to Other Processes and Services	3-21
About Reusable Processes	3-23
About Subprocesses	3-24
Use a Send Activity	3-29
Send Notifications	3-30
Use Send and Receive	3-31
Communicate Between Processes Using Correlation	3-32
Example of Using Correlation with Send and Receive	3-33
Handle Errors with Event Subprocesses	3-39
Configure a Multi-instance Activity	3-40
Configure an Upload Form Activity	3-42
Use Abstract Activity	3-44
About Draft Property	3-45
Use Data Mapper Activity	3-47
Work with Sequence Flows	3-48
Unconditional Sequence Flows	3-48
Conditional Sequence Flows	3-49
Default Sequence Flows	3-49

4 Design Dynamic Processes

How do dynamic processes work in process applications?	4-1
Model a Dynamic Process in Design Time	4-2
Create a Dynamic Process	4-2
Clone a Dynamic Process	4-3
Create Activities	4-3
Create Stages	4-4
Define Stage Properties	4-5
Enable Markers for a Stage	4-5
Define Activity Properties	4-7
Enable Markers for an Activity	4-7
Implement a Human Task Activity	4-8
Implement a Process Activity	4-9
Implement a Service Activity	4-10
Define Conditions for Stage or Activity	4-10
Define Process Input and Output	4-11
Define Process Data Objects	4-13
Define Process Completion and Termination	4-14
Work with Inline Validations	4-15

5 Design Forms and User Interfaces

Forms and Linked UIs at a Glance	5-1
Work in the Forms Editor	5-3
Position Controls on Forms	5-3
Configure Basic Controls	5-5
Configure Text Input and Area Fields	5-5
Configure Simple Text Fields	5-6
Configure Buttons	5-6
Configure Drop-down Select Fields	5-7
	5-8
Configure Check Lists and Check Boxes	5-8
Configure Radio Buttons	5-9
Configure Number Fields	5-10
Configure Date and Time Fields	5-11
Configure Email Fields	5-12
Configure Web Address URL Fields	5-13
Configure Message Fields	5-13
Configure Links	5-14
Configure Advanced Controls	5-15
Configure Currency (Money) Fields	5-16
Configure Phone Number Fields	5-17
Include Images	5-18
Upload and Preview a Base64 Image	5-18
Include Videos	5-19
Configure Identity Browser Fields	5-20
Place Controls in Panels, Sections, or Tabs	5-22
Format the Title and Description of a Panel	5-23
Indent Sections	5-24
Configure Static and Dynamic List of Values Fields	5-24
Add an Option to a List of Values Field	5-25
Clear All Options From a List of Values Field	5-25
Configure Tables	5-26
	5-28
Configure Repeatable Sections	5-29
	5-30
Configure Train Controls	5-30
Configure Divider Controls	5-32
Configure File Manager Controls	5-32
Configure File Upload Control	5-33
Configure File List Control	5-35
Implement Intelligent Document Processing in Forms	5-36

Configure Document Understanding Controls	5-37
Reuse Forms	5-40
Work with Presentations	5-40
Change Form Presentations Dynamically	5-42
Change Form Presentations Dynamically	5-43
Work with Styling	5-47
Styling Properties	5-47
Work with CSS	5-49
Apply a Stylesheet to a Form	5-49
Style Form Controls	5-50
Bind Form Data with Controls	5-51
Create a Form Based on a Business Type	5-53
Create Computed Controls	5-53
Specify Filters for Controls	5-55
Add Dynamic Behavior to Forms	5-58
Configure Events	5-59
Specify Actions	5-61
Specify Conditions	5-63
Specify Functions	5-66
Example of Current User Function	5-74
Specify Filters in Events	5-76
Reuse Event Snippets	5-80
Extract a Snippet	5-80
Use a Snippet	5-81
Execute Loops in Events	5-81
Example of a Simple Loop	5-81
Example of a Loop with an Array	5-82
Execute REST Connector Calls in Events	5-83
Populate Controls Using REST Calls	5-86
Link and Refresh List of Value Fields	5-88
Preview Forms and Their Payload	5-89
Export Forms as PDF	5-90
Configure a Linked UI	5-91
Linked UIs at a Glance	5-91
Add a Linked UI Connection	5-92
Use a Linked UI in a Human Task	5-93
Integrate a Linked UI with the Task List	5-94

6 Work with Human Task Activities

Understand End User Tasks	6-1
Create Human Task Activities	6-1

Typical Workflow for Creating a Human Task Activity	6-2
Create Submit and Approve Tasks	6-2
Implement Human Task Activities	6-2
Assign Human Task Activities	6-3
Configure the Title and Task Summary	6-4
Configure Action Rules	6-5
Use Forms or Linked UIs to Display Task Information	6-5
Define an Approval Pattern	6-6
Bypass the Approval Chain	6-6
Exclude Participants	6-7
Configure the Due Date and Priority	6-7
Configure Task Escalation or Expiration	6-8
Configure Task Reminders	6-9
Customize Notification Emails	6-10
About Notification Email and Templates	6-10
Configure Email Notifications	6-11
Manage Email Templates	6-12
Configure Email Templates	6-12

7 Model Decisions

Decisions at a Glance	7-1
Understand Local and External Decisions	7-3
Create a Local or External Decision	7-4
Understand the Graph View	7-5
Add a Decision Logic Type	7-6
Define Expressions with the Friendly Enough Expression Language (FEEL)	7-8
Data Types	7-8
Grammar Rules	7-9
Built-In Functions	7-10
Conversion Functions	7-10
Boolean Functions	7-11
String Functions	7-11
List Functions	7-12
Numeric Functions	7-12
List Iteration Expressions	7-13
Date, Time, and Duration Functions	7-13
Conversion Examples	7-13
Arithmetic Operation Examples	7-14
Comparison Operation Examples	7-14
Define Decision Input and Type	7-15
Create Input Data	7-15

Define Custom Data Types	7-16
Model Decision Logic	7-19
Create Empty Logic Decisions	7-19
Create Decision Tables	7-19
Define Decision Table Input	7-21
Define Decision Table Output	7-23
Configure Rules	7-24
Configure a Hit Policy	7-24
Create Expressions	7-29
Create If-Then-Else Statements	7-30
Create Functions	7-32
Create Contexts	7-33
Create Lists	7-37
Create Relations	7-38
Create Loops	7-40
Test Decision Models	7-42
Expose Decisions as Services	7-42
Add Decision Services to Processes	7-43

8 Manage Data

About Managing Data	8-1
Work with Data Objects	8-1
Create a Data Object	8-2
Edit or Delete a Data Object	8-2
Work with Business Types	8-3
Create a Business Type	8-3
Create a List of Values	8-5
Use Business Types	8-6
Create a Business Exception	8-8
Configure Data Association	8-9
Data Association Tips	8-10
Human Task Exec Variables	8-11
Process Variables	8-12
Define Conditions for Data Associations	8-12
Work with Transformations	8-14
Append to Array	8-16
Filter Arrays	8-17
Work with Expressions	8-18
Use Business Keys	8-26

9 Integrate with Applications and Services

Work with Connectors	9-1
Connectors at a Glance	9-1
Configure REST Connectors in Designer	9-3
Add a REST Connector	9-3
Add REST Connector Resources and Operations	9-4
Define REST Connector Input and Output Types	9-6
Secure the REST Connector	9-7
Specify Connector Visibility in Process Editors	9-11
Use Connectors in Process Applications	9-12
Delete a Connector	9-13
Work with Integrations	9-13
Use Integrations	9-14
Create a Connector with an Integration	9-14
Implement an Integration in Processes and Forms	9-15
View Integration Details	9-15
About Integration Statuses	9-15
Discover and Use Integrations Within Oracle Integration 3 Projects	9-16
Embed a Start Form in an External Application	9-16
Embed a Task List in an External Application	9-18
Embed a Task Detail in an External Application	9-20

10 Explore Workspace

Start an Application	10-1
Work on Tasks	10-2
Find a Task	10-2
Search for a Task	10-2
Use the Task Tab Filters	10-5
Use the Task Lifecycle and Due Date Filters	10-6
Apply Advanced Search Filters	10-6
Perform Task Actions	10-7
View and Complete Tasks	10-9
Track Processes	10-10
Find a Process	10-10
Filter Processes	10-11
Sort Processes	10-12
About Process States	10-12
Work on Processes	10-13
View Process Details	10-13
Work with Activities	10-14

View Process Audit	10-15
Post and View Comments	10-16
Upload and View Documents	10-16
View Process Data	10-16
View Process Information	10-17
End Dynamic Processes and Activities	10-17
Alter the Flow of a Process	10-18
Use Analytics	10-20
Analytics: Summary View	10-22
View Analytics for an Application	10-28
View Analytics for a Process	10-32
View Analytics for Activities	10-36
View Analytics for a Human Task Activity	10-39
View Analytics for a Service Task Activity	10-41
About Instance Status	10-43
Analytics: Instance Tracking	10-45
Set to Out of Office	10-46
Create Out of Office Record	10-46

11 Workspace Administration

Manage Roles in Workspace	11-1
View or Edit Roles After Activation	11-1
Add a Global Role in Workspace	11-2
Assign the Process Application Administrator Role	11-3
Delete Roles in Workspace	11-4
Manage Notifications	11-5
Manage Credentials in Workspace	11-6
About Credentials	11-6
Edit Application Credentials	11-7
Manage Global Credentials	11-8
Add Global Basic Credential	11-8
Add Global OAuth Credential (Client Credential)	11-9
Add Global OAuth Credential (JWT Assertion)	11-10
Add Global OCI Signature Policy Credential	11-11
Edit or Delete Global Credentials	11-12
Manage Out of Office Records	11-12
Configure Out of Office Records for Users	11-13
Enable Break Glass	11-13
Temporary Access Approval	11-14
Provide Your Own Key	11-14
Create and Import Your TDE Master Key	11-14

Update the Key	11-16
Specify Data Management Settings	11-16
Register Services	11-17
Prerequisites for Registering a Service	11-17
Register a New Service	11-19

A Quick Reference

Preface

This document describes how to use Oracle Cloud Infrastructure Process Automation to design process applications and its components in Designer (design-time), and work on user tasks in Workspace (runtime).

Topics:

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

Audience

Using Oracle Cloud Infrastructure Process Automation is intended for users who want to design process applications and work on end user tasks.

Documentation Accessibility

For information about Oracle's commitment to accessibility, visit the Oracle Accessibility Program website at <https://www.oracle.com/corporate/accessibility/>.

Access to Oracle Support

Oracle customers that have purchased support have access to electronic support through My Oracle Support. For information, visit <https://support.oracle.com/portal/> or visit [Oracle Accessibility Learning and Support](#) if you are hearing impaired.

Diversity and Inclusion

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

Related Resources

For more information, see these Oracle resources:

- Oracle Cloud Infrastructure Process Automation documentation in the Oracle Cloud Library on the Oracle Help Center.
- Oracle Cloud at <http://cloud.oracle.com>.

Conventions

The following text conventions are used in this document.

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

1

Introduction

Oracle Cloud Infrastructure Process Automation is a multi-tenant Oracle Cloud Infrastructure (OCI) service. Use the low-code designer environment to quickly create structured and dynamic process workflows. Leverage capabilities such as reusable business rules, configurable web forms, connectivity to prebuilt application integrations, and much more to simplify repetitive tasks and automate your business processes.

Topics:

- [Overview of Oracle Cloud Infrastructure Process Automation](#)
- [Process Automation Product Types](#)

Overview of Oracle Cloud Infrastructure Process Automation

Use Oracle Cloud Infrastructure Process Automation to rapidly design, automate, and manage business processes.

Key capabilities include:

- Structured and dynamic process modeling
- Decision modeling
- Connectivity to outside apps and integrations
- Web forms
- User task management and tracking

Explore the Designer and Workspace environments

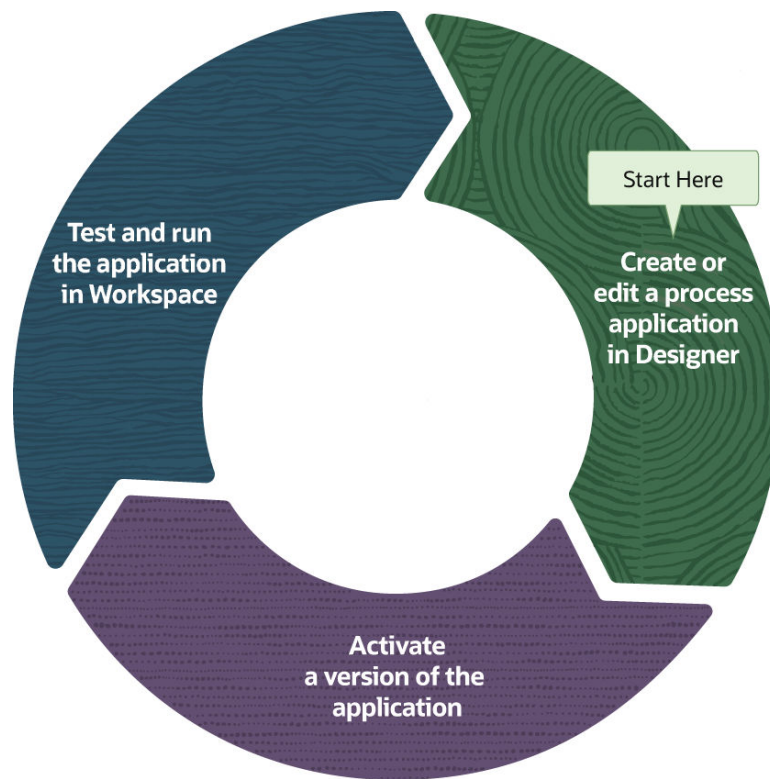
The Oracle Cloud Infrastructure Process Automation development and runtime management environments enable you to perform iterative process automation development:

- Use **Designer**, the design-time development environment, to create and edit applications and their components. This environment is for Process Automation developers.

The Designer URL ends in: `/process/designer/`

- Use **Workspace**, the runtime environment, to test, run, monitor, and administer applications. This environment is for Process Automation Administrators and end users.

The Workspace URL ends in: `/process/workspace/`



Navigate easily between the design time and runtime environments:

- Click **Workspace** in the left navigation pane of the Process Automation Designer home page to navigate to Workspace from Designer.
- Click the user icon on top right of Process Automation Workspace user interface, and choose **Designer** to navigate to Designer from Workspace.

Explore Process Automation REST APIs

Work with applications, processes, and user tasks using REST APIs.

See available endpoints in *REST API for Oracle Cloud Infrastructure Process Automation*.

Process Automation Product Types

Oracle Cloud Infrastructure Process Automation is available in three models - as an individual Oracle Cloud Infrastructure (OCI) service, as a service paired with a Fusion-based Oracle Cloud Applications service (such as HCM Cloud or CX Cloud), and as a service enabled with Oracle Integration 3.

- **Individual Oracle Cloud Infrastructure (OCI) service:** When you order Process Automation as an individual service, you provision and set up the service instance from the OCI console.
See [Set Up Process Automation as an Individual Service](#).
- **Paired with Fusion-based Cloud applications:** Process Automation instances paired with Fusion-based Oracle Cloud Applications such as Oracle Human Capital Management (HCM) Cloud or Oracle Customer Experience (CX) Cloud, are designed for you to use and extend existing out of the box features that are provided by Fusion-based Oracle Cloud Applications.

When you get Process Automation in this way, it is automatically provisioned for you. You give team members access to the instance in the Oracle Identity Cloud Service (IDCS) application of the instance.

See [Use Process Automation with Fusion-Based Oracle Cloud Applications](#).

- **Enabled with Oracle Integration 3** : You can enable a Process Automation instance with an Oracle Integration 3 Enterprise Edition instance. When you enable Process Automation with Oracle Integration, a Process Automation service instance is automatically provisioned and associated with the Oracle Integration service instance.

See [Use Process Automation with Oracle Integration 3](#).

Set Up Process Automation as an Individual Service

If you want to set up Oracle Cloud Infrastructure Process Automation as an individual service, you need to perform certain provisioning and administering tasks from the Oracle Cloud Infrastructure (OCI) console.

For information on provisioning and setting up Oracle Cloud Infrastructure Process Automation as a standalone OCI service, see *Administering Oracle Cloud Infrastructure Process Automation*.

Use Process Automation with Fusion-Based Oracle Cloud Applications

If you have an Oracle Cloud Infrastructure Process Automation instance that is paired with a subscription to a Fusion-based Oracle Cloud Applications service, such as Oracle Human Capital Management (HCM) Cloud or Oracle Customer Experience (CX) Cloud, Oracle Cloud Infrastructure Process Automation is automatically provisioned for you.

For Oracle Cloud Infrastructure Process Automation instances that are paired with Fusion-based Oracle Cloud Applications, you use Oracle Identity Cloud Service (IDCS) to manage access to Oracle Cloud Infrastructure Process Automation. See *Administering Oracle Identity Cloud Service*.

- [Prerequisites](#)
- [Synchronize Users From Fusion Applications Identity Provider to IDCS Stripe](#)
- [Assign IDCS Application Roles](#)
- [Get the Process Automation Instance URL](#)
- [Example of Required User Created in IDCS](#)

Prerequisites

To use Oracle Cloud Infrastructure Process Automation with Fusion-based Oracle Cloud Applications, ensure that you have met the prerequisites listed here.

1. Your organization must have purchased Oracle Cloud Infrastructure Process Automation for Fusion Applications.
2. Complete any prerequisite requirements detailed in the Fusion Applications documentation.

Synchronize Users From Fusion Applications Identity Provider to IDCS Stripe

Enable and verify user synchronization in Oracle Identity Cloud Service (IDCS). IDCS is an intermediary to synchronize user accounts between your Fusion-based Oracle Cloud Applications and Oracle Cloud Infrastructure Process Automation.

1. Sign in to IDCS as a Cloud Account Administrator.

The URL and credentials for your IDCS instance are provided in the welcome email that you receive when you create a new cloud account.

2. In the IDCS navigation pane, click **Oracle Cloud Services**.
3. Select the IDCS application configured for your Fusion-based cloud application, for example, **Oracle Applications Cloud (Fusion)**. Click the application name to open the application.

4. Enable user synchronization.

- a. Click the **Provisioning** tab.

- b. Turn on **Enable Provisioning**.

For more information, see *Enable and Configure Connectivity for Provisioning for Your Application* in *Administering Oracle Identity Cloud Service*.

- c. In the Select Provisioning Operations section, select the **Authoritative Sync** check box.

For more information, see *Select Provisioning Operations* in *Administering Oracle Identity Cloud Service*.

- d. Turn on **Enable Synchronization**.

For more information, see *Enable and Configure Synchronization for Your Application* in *Administering Oracle Identity Cloud Service*.

- e. Click **Save**.
5. Verify that the users are available and synchronized in IDCS.
 - a. In the IDCS navigation pane, click **Users**.
 - b. If the users that you want to access Oracle Cloud Infrastructure Process Automation are not listed, click **Import** to import those users.



Note:

In order for some Fusion-based Oracle Cloud Applications services to use Oracle Cloud Infrastructure Process Automation, you may need to create specific users in IDCS. See [Example of Required User Created in IDCS](#).

Assign IDCS Application Roles

You must assign predefined Oracle Identity Cloud Service (IDCS) application roles to users or groups in Oracle Identity Cloud Service to allow them to work with Oracle Cloud Infrastructure Process Automation.

In Process Automation, the ServiceAdministrator and ServiceDeveloper IDCS application roles are functional roles that determine whether or not a user has access to the administration and Designer user interface and APIs.

The Cloud Account Administrator on your team (typically, the person who signed up for the service) can grant team members these roles.

 **Note:**

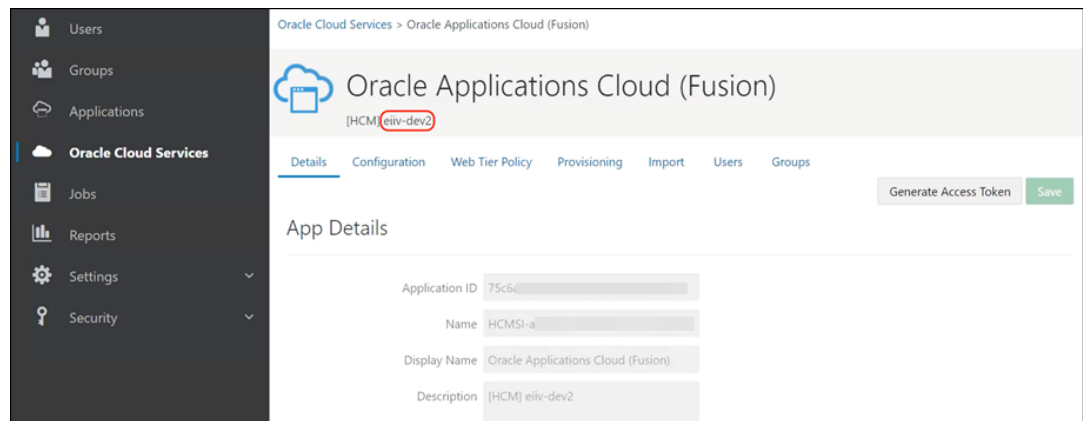
These roles are not required for accessing the Process Automation Workspace. Any authenticated user can work in Workspace or access runtime APIs.


Note that to further define data access and task permissions for users and groups in Process Automation, you can configure roles specific to process applications. See [Configure Roles for Process Applications](#).

1. In the IDCS navigation pane, click **Oracle Cloud Services**.
2. As the display name of the Process Automation IDCS application is prefixed with the display name of the IDCS application for your Fusion based cloud application, you can search for the Process Automation IDCS application by entering the display name of the IDCS application for your Fusion based cloud application in the **Search** field.

For example, if the display name of the IDCS application for your Fusion based cloud application is **exqx-test** then enter `exqx-test` in the **Search** field. The Process Automation IDCS application appears in the search result. Verify it is the right application by confirming that its display name starts with **exqx-test** (for example, `exqx-testdemoxyz`).

The following image shows the *display name* for the IDCS application of a Fusion based cloud application (HCM).



3. Select and click the Process Automation IDCS application to open it.
4. Navigate to the **Application Roles** tab.
5. In the tile for the **ServiceAdministrator** or **ServiceDeveloper** role:
 - a. Click  and select **Assign Users** or **Assign Group**.
 - b. Select the users or groups to whom you want to assign the role and click **Assign**.

Get the Process Automation Instance URL

To access Oracle Cloud Infrastructure Process Automation, you have to get the Process Automation URL. Bookmark the URL and share it with your developers.

1. In the IDCS application for your Process Automation instance, click the **Configuration** tab.
2. Expand the **Resources** section of the page.
3. Copy the value of the **Primary Audience** field and paste it to a text file.

4. Append `/designer` to the end of the URL. This is the URL for the Process Automation instance's Designer user interface.
5. Copy the value of the **Primary Audience** field again and append `/workspace` to the end of the URL. This is the URL for the Process Automation instance's Workspace user interface.
6. Distribute these URLs to team members who require access to the Process Automation instance.

Example of Required User Created in IDCS

To use Oracle Cloud Infrastructure Process Automation with some Fusion-based cloud applications, you may need to create specific users in Oracle Identity Cloud Service (IDCS).

For example, if you want to use Oracle Cloud Infrastructure Process Automation with Oracle HCM Cloud, you must create the `FA HCM Journey Admin` user in IDCS.



Note:

The `FA HCM Journey Admin` user must be in IDCS only, and should not exist in FA.

To create the required user `FA HCM Journey Admin` in IDCS for Oracle HCM Cloud:

1. As the Cloud Account Administrator, sign in to IDCS.

The URL and credentials for your IDCS instance are provided in the welcome email that you receive when you create a new cloud account.
2. In the IDCS navigation pane, click **Users**.
3. On the Users page, click **Add**.
4. In the Add Users dialog, add the `FA HCM Journey Admin` user, as follows:
 - **First Name:** `FA HCM`
 - **Last Name:** `Journey Admin`
 - **User Name:** `fa_hcm_journey_admin`
 - **Email:** An email address where this user can receive email notifications

Add User ×

Step 1: Add User Details

The user name will be used to log in, and the email address will be flagged as the primary email address for the user.

First Name	<input type="text" value="FA HCM"/>
* Last Name	<input type="text" value="Journey Admin"/>
* User Name	<input type="text" value="fa_hcm_journey_admin"/>
Email	<input type="text" value="facilities.freddy@gmail.com"/>

Use the email address as the user name

5. Click **Finish**.

Use Process Automation with Oracle Integration 3

To use Process Automation with Oracle Integration 3, you must enable it with an Oracle Integration service instance from the Oracle Cloud Infrastructure (OCI) console.



Note:

You can use Process Automation with only Oracle Integration 3 *Enterprise Edition*.

When a Process Automation instance is enabled with an Oracle Integration instance, the Process Automation instance gets automatically provisioned with the Oracle Integration instance. Users assigned the IDCS application role ServiceDeveloper can access both the Process Automation and Oracle Integration design-time.

See *Enable Process Automation with Oracle Integration 3* in *Administering Oracle Cloud Infrastructure Process Automation*.

Another way to use Process Automation with Oracle Integration 3 is by registering an Oracle Integration service instance with your Process Automation instance from Workspace.

To learn how to register a service such as Oracle Integration from Workspace, see [Register Services](#)

Once Process Automation is enabled with Oracle Integration or Oracle Integration is registered with Process Automation from Workspace, you can use activated integrations from Oracle Integration that are designed with REST triggers into your process applications. See [Work with Integrations](#).

2

Work with Applications

Applications are the containers for all design components such as processes, UIs, decisions, connectors and so on that you create in Designer.

Topics:

- [Applications at a Glance](#)
- [Add and Edit Applications](#)
- [Activate Applications](#)
- [Work with Process Applications](#)
- [Configure Roles for Process Applications](#)
- [Work with Decision Applications](#)
- [Create an Application from a Recipe](#)

Applications at a Glance

Start by familiarizing yourself with these application concepts.



How can I create applications?

Create applications in one of these ways.

Method	Description
Create	Create a new process application or decision application.
Import	Import a previously exported process application from an .exp file or a previously exported decision application from a .dmnx file.
Recipe	Use a prebuilt, ready to use application.
Clone	Save a copy of an existing application under a new application name.
Version	Save a copy of an existing application under the same application name. (for example, 2.0).

What are the different types of applications that you can design?

You can design either a process application or a decision application. All process applications are listed in the Process Applications page, while all decision applications are listed in the Decision Applications page. The All Applications page lists all created process and decision applications. You can distinguish between a process and a decision application by the different icons.

- Process application 
- Decision application 

See [Work with Process Applications](#) and [Work with Decision Applications](#).

What are snapshots?

A snapshot refers to the application at a point in time, and is used when activating. While designing, name and save as many snapshots as you want, so you can return to one to view or activate if needed. Snapshots are read only and can't be edited. See [Work with Snapshots](#).

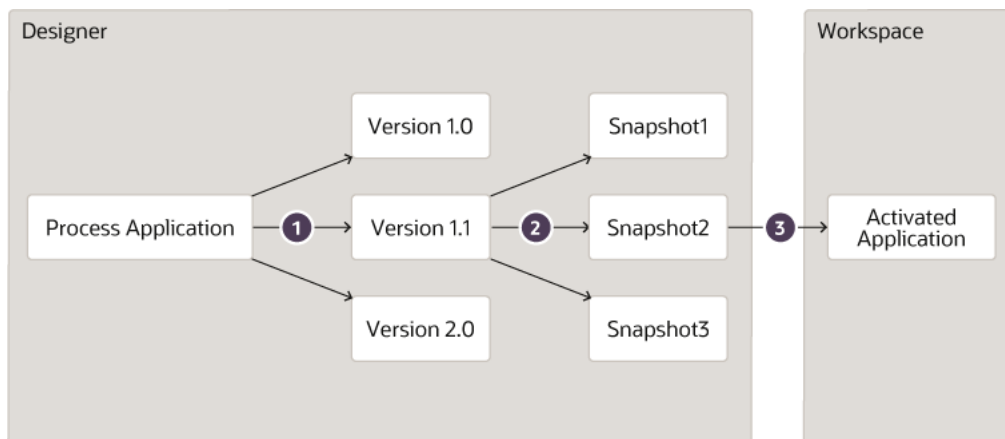
How do versions work?

Want to branch your application, starting with its current state but grouping it under the same application? Create a new version, such as 1.1 based on 1.0. You can create and activate many versions in parallel, and switch between them in Designer.

An application's version always appears next to its name. Click the version icon to view other versions, open a different version, or create a new version.

While you can create and activate many application versions, just one version is set as the default version in Workspace.

See [Work with Versions](#).




1. Create multiple independent versions of an application
2. Save a snapshot of a version at any time
3. Activate the application based on a snapshot

How are applications validated?


Validation ensures that your application can be activated. It reports errors (which prevent activation) and warnings (which don't prevent activation) found so you can resolve them.

Validation happens automatically as part of activation, but you can also validate any time by

clicking **menu**  and choosing **Validate** from the menu.

Can multiple users work on the same application at once?

Yes. Only one user can have access to a resource, such as a process or form, at a time. Within

an application, view other users and all activity by clicking **Main menu**  and choosing **View Members**. If you try to open a resource being edited by another user, a message notifies you.

How do I move applications from one system to another?

You can export and import process applications as .exp files and decision applications as .dmnx files. See [Export an Application](#) and [Import an Application](#).

Add and Edit Applications

Learn all about designing and working with applications in Designer.

Topics:

- [Create an Application](#)
- [Clone an Application](#)
- [Rename an Application](#)
- [Export an Application](#)
- [Import an Application](#)
- [Work with Snapshots](#)
- [Work with Versions](#)
- [Delete an Application Version](#)

Create an Application

When you create an application, you provide a name, version, and optional description with which you can identify the application later on.

To create an application:

1. From the Designer home page (All Applications), click **Create**.
2. In the Creation application pane, you can create an application using one of the options available.

Option	Description
Create process application	Create a new process application. See Create a Process Application .
Create decision application	Create a new decision application. See Create a Decision Application .
Import application	Import a process application as an .exp file or a decision application as a .dmnx file. See Import an Application .
Create application from recipe	Create an application from a recipe which is a pre-designed, run-ready application. See Create an Application from a Recipe .

The newly created application is listed under the All Applications page, and also on the Process Application page if it is a process application or the Decision Application page if it is a decision application.

Besides the above options to create a new application, you can also:


- Save a copy of an existing application under a new application name by cloning the application. See [Clone an Application](#).

- Save a copy of an existing application under the same application name. (for example, 2.0). See [Work with Versions](#).


Clone an Application

When you clone an application, a copy of the selected application is created with a new name and version that is specified by you.

Clone from the All Applications, Process Applications or Decision Applications page

1. From the list of applications, hover at the far side of the application you want to clone and choose **Clone** from the  menu.
2. In the Clone application pane, change the name, description, version tag and identifier name for the application as needed.
3. Click **Clone**. The newly cloned application appears in the list of applications.


Clone from within a selected application

1. At the top of the page, click the down arrow after the application name and version.
2. In the Version pop up that appears, click **View more options**  and choose **Clone**.
3. In the Clone application pane, change the name, description, version tag and identifier name for the application as needed.
4. Click **Clone**. The newly cloned application appears in the list of applications.



Rename an Application

You can change an application's title and description only. You can't change its identifier or version.

Rename from the All Applications, Process Applications or Decision Applications page

1. From the list of applications, hover at the far side of the application you want to rename and choose **Info** from the  menu.
2. In the Application Info pane, change the application name and description. Note that these changes apply only to the selected application version.
3. Click **Save**. The newly saved application changes appear in the list of applications.

Rename from within a selected application

1. At the top of the page, click **Main menu**  and choose **Info**.
Alternatively, click the down arrow after the application name and version on top of the page. In the Version pop up that appears, click **View more options**  and choose **Info**.
2. In the Application Info pane, change the application name and description. Note that these changes apply only to the selected application version.
3. Click **Save**. The newly saved application appear in the list of applications.

Export an Application

Exporting saves an application's metadata as an .expx file (in case of a process application) or a .dmnx file (in case of a decision application) that can be imported as a new application on another system.

Note:

You should not modify the contents of an exported file. If the contents of an exported file is modified, then the file becomes corrupt and it cannot be imported.


What's exported?

All metadata (design time settings) from the application is exported, except for:


- Credentials
- Role assignments

Credentials and roles need to be reconfigured after import.

Export from the All Applications, Process Applications or Decision Applications page

1. From the list of applications, hover at the far side of the application you want to rename and choose **Export** from the  menu.
2. In the dialog that appears, choose **Save File** and click **OK**.
3. Specify a name and location for the .expx file or the .dmnx file, and click **Save**. Note that you can specify any name for the .expx or .dmnx file while saving it.

Export from within a selected process application

1. Click the down arrow next to the application's name and version.
2. In the Version pop-up dialog that appears, click **View more options**  and choose **Export**.

You get a confirmation dialog that the file has been exported successfully and the file gets automatically downloaded to your local system.

Import an Application

Import a previously exported process application or decision application to create a new application.

What's imported?


You can import the following files:

- Previously exported .expx or .dmnx files from Oracle Cloud Infrastructure Process Automation.
- Previously exported .exp and .dmn files that are created in Oracle Integration Generation 2.

Once you import the .exp or .dmn file, Oracle Cloud Infrastructure Process Automation displays a migration report that contains the following sections:

- **Migrated**
Review this section to understand what has been migrated.
- **Review and finalize**
Review this section and take the required action. For example, you have to manually reassign the role members.
- **Couldn't migrate**
Review this section to understand functionality that could not be imported. For example, references to Insight activity may not be available in Process Automation.

 **Note:**

Once you fix the problems, validate the application. You can always access the original migration report by clicking **Main menu**  at the top of the application's page and choosing **View Migration report** from the menu.

All metadata (design time settings) from the application is imported, except for:

- Credentials are not imported.
- Users or groups assigned to roles may be unavailable.

You can also import .exp and .dmn files that are created in Oracle Integration Generation 2. Once you have imported the file, Process Automation displays a migration report that contains the following sections:

To import a saved application:

1. On the Designer home page (All Applications), click **Create**.
2. In the Create application pane, choose **Import application**.
3. In the Import application pane, drag and drop the .exp file or the .dmn file into the Drag and Drop area.

You can also import .exp and .dmn files into the Drag and Drop area.

4. Click **Import**.

The newly imported application appears in the list of applications.

If you have imported a .exp or a .dmn file, Process Automation displays a migration report that you need to read and take the necessary action.

Work with Snapshots

A snapshot is a copy of the current application that you're working on at a given point of time.

Create a snapshot of your current application

You can create a snapshot of the application that you're working on currently to save the latest changes.

1. At the top of the page, click the saved timestamp or click  and choose **View Activity**.

2. In the Activity pane that appears, click **Take Snapshot**.
The snapshot is named with its timestamp and listed topmost in the activity list.
3. To edit the default name of the snapshot, click the snapshot and edit the name in the **Name** field that appears.
You can also enter a meaningful description in the **Description** field.
4. Click the three horizontal dots next to the snapshot name field to display a menu with the following options:
 - **View**: Choose **View** to view the contents of the selected snapshot.
 - **Activate**: Choose **Activate** to activate the selected snapshot.
 - **Delete**: Choose **Delete** to delete the selected snapshot.

Create a snapshot during activation

Snapshots also get created when you activate an application.



1. In Designer, click **Activate**.
The Activate pane is displayed. Process Automation assumes you want to create a new snapshot that contains your most current changes.
2. In the **Select a snapshot** field, select another snapshot rather than creating a new one that contains the latest changes.
3. In the **Snapshot name** field, optionally change the default timestamp name that will be assigned to a new snapshot.
4. When you select **Activate** or **Override**, a message lets you know that the snapshot is being created.

Work with Versions

You can create a new version of an application at any time. The new version retains the same application name (although you can change it later) but has a different version tag, such as 1.1 versus 1.0. Once activated, application versions become different applications.

You can activate multiple application versions, but an application has only one default version, and that version is displayed by default in the Start Requests page in Workspace. When you activate a version, you specify if it's the default version.

Create a new version

1. For an application, open the Version pop-up by clicking the down arrow  next to the application name and version breadcrumb.
The selected version is listed at the top. Note that a green dot is displayed next to the default version of the application.
In this dialog, you can clone, export, delete, rename, or create a new version from any version.
2. Create a new version. If creating from the selected version, click **New Version**. If creating from a version below the selected version, click its **New version**  icon.
A New Version side pane opens.
3. Enter a new version tag (for example, 1.1) or leave the default suggested tag. This tag will differentiate the application versions.
4. Click **Create**.
The new version is created but not selected.

5. Click **Open Now** to open the new version. Alternatively, open the Version pop-up by clicking the down arrow ▼ next to the application name and version breadcrumb. In the Versions pop-up, select the new version to edit it.

Switch between versions

If you want to edit a different application version, use the Versions dialog to select and open the one you want.

1. For an application, display the Versions pop-up by clicking the down arrow ▼ next to the application name and version. In the Versions pop-up, select the new version to edit it.
2. Click **New Version**.
3. Select another version to open and edit it.

Change the default version of a process application

The default version is set at activation of a process application. To change the default version, you must deactivate the current default version, then activate the new default and select the **Make it default** field.

Delete an Application Version

Deleting an application version removes it and all of its components and snapshots permanently. Activated application versions must be deactivated before they can be deleted.

You can delete a version from the Applications page or from within an application version.

1. For an application, open the Versions pop-up by clicking the down arrow ▼ after the version indicator.

The Version dialog appears.

2. Delete an application version.
 - To delete the selected version, click **View more options** ⋮ and choose **Delete**.
 - To delete another version, select the version and click **Delete** 🗑️.
3. In the side pane that appears, click **Delete** to confirm the deletion.

If the application version is activated, a message alerts you that it must be deactivated first. Click **Go to deactivate**.

Activate Applications

After designing an application, you can activate your application and make its resources available to be used by end users.

When you activate an application:

- You specify a snapshot (application metadata) to use.
- You're notified of any validation issues. You can still activate with warnings but not with errors.
- You can deactivate a previous activation, and activate your current one.

After activating an application:

- A process application's resources moves from design time (Designer) to runtime (Workspace), where users can run it in either a test or production capacity.

- You can test the process application in Workspace, provided you have permission to start requests for it.
- You can view and use REST API endpoints for the processes in a process application.
- You can view and use REST API endpoints for the decision services in a decision application.

Topics:

- [Activate an Application Version](#)
- [Deactivate an Application Version](#)
- [View Application API Endpoints](#)
- [View Activated Application Versions](#)

Activate an Application Version

You can activate an application to make its resources available to end users.

If you've previously activated, and want to deactivate and purge the previous activation, see [Deactivate an Application Version](#).

1. From a process application, click **Activate**.

The Activate pane opens, listing the version to be activated.

2. Specify a snapshot to activate.

- a. In the **Select a snapshot** field, leave **New snapshot** selected to create a new one, or select a previously created snapshot.
- b. In the **Snapshot name** field, optionally change the default name based on date and time.

3. Resolve validation issues, if needed.

Validation results for the selected snapshot are listed under **Validation Summary** with a **View validation issues in detail** link to drill down to warnings or errors. You can still activate with warnings, but errors must be fixed before you can activate.

In many cases, you'll be taken to the place to fix the validation issue. For example, if a warning appears that the application's description summary is empty, clicking the link takes you to the application info pane to complete the missing description. After resolving validation issues, click **Activate** again.

4. For process applications, note that the **Make it default** checkbox is selected. Selecting the checkbox will make this application version the default version to start new application instances under Start Requests in Workspace, after activation.

Note that you must deactivate an application version to change this setting.

5. Click **Activate**.

If you're creating a new snapshot, a **Taking Snapshot** status appears. Once complete, an **Activating** status appears. As soon as the application version completes activating:

- the **Activate** button changes to **Deactivate and purge** for process application and **Deactivate** for decision application. This allows you to iteratively design and activate.
- details about the activation appear under **Last Activation**.


6. Optionally, for process applications you can test the activated application in Workspace. See [Activate and Test Process Applications](#).

Deactivate an Application Version

After making changes to an activated application, in order to include the latest changes and also to test the changes in Workspace (in case of a process application), you must first deactivate the activated application version.



Note:

Not sure which application version is active? Click  and choose **View Activated Versions**. The active version is identified by its version tag and shows a **Running** status.

To deactivate an activated application version:

1. From the application's main page, click **Activate**.

The activation pane for the active version of your application opens.

Note that instead of **Activate**:

- the **Deactivate and purge** button is displayed for *process applications*.
- the **Deactivate** button is displayed for *decision applications*.

Under **Last Activation**, clicking **View more options**  displays other available options such as **View Activated Versions** and **View API**.

2. For process applications, click **Deactivate and purge**.

In the resulting pane, you get a message that if you choose to deactivate and purge it will delete the currently activated version of the application, including the running and completed process instances, and all associated analytics data. You're asked to confirm the action.

Caution: If working in a production environment, use the **Deactivate and purge** option with care. Deactivating and purging a process application permanently deletes all existing (running or completed) process instances and associated analytics data for your application version.

3. For decision applications, click **Deactivate**.

In the resulting pane, you get a message that if you choose to deactivate the decision application, decision services of this version will be unavailable. You're asked to confirm the action.

Caution: If working in a production environment, use the **Deactivate** option with care. When you deactivate a decision application, the decision services exposed by it will no longer be reachable. Only decision service endpoints of activated decision applications can be reused by other process applications as well as external clients.

4. Click **Confirm**.

An intermediate message lets you know that your application is being deactivated. You get a confirmation after deactivation, and the Activate pane opens.

5. Specify a snapshot to activate. See [Work with Snapshots](#).

- a. In the **Select a snapshot** field, leave **New snapshot** selected to capture your most recent changes.
- b. In the **Snapshot name** field, optionally change the default date/time name.

6. Resolve validation issues, if needed.

Validation results for the selected snapshot are listed under **Validation Summary** with a **View validation issues in detail** link to drill down to warnings or errors. You can still activate with warnings, but errors must be fixed before you can activate.

7. For process applications, the **Make it default** checkbox displays. Keep the checkbox selected if you want to make this version (that you're going to activate) as the default version. The default version of a process application will be visible in the Start Requests page of Workspace.

8. Click **Activate**.


If you're creating a new snapshot, a **Taking Snapshot** status appears, followed by a message that the application is being activated. Once complete, you get a confirmation that the application is activated.

9. For a process application, you can test the new activation in Workspace.
 - a. Click **Test in Workspace**. A new browser tab opens with Workspace. Notice that the URL is the same except for workspace at the end rather than designer.
 - b. On the Start Requests page, select your activated application, listed by its application name and process.
 - c. Run the application as an end user would. For example, if the application starts with a form, complete and submit the form.

View Application API Endpoints

See the API endpoints used in an application at the point of activation.

1. From a process application, click **Activate**.
2. Activate the application, if it's not already activated.

The application must be activated to view its endpoints. See [Activate an Application Version](#).
3. In the Activate pane, click **View more options**  and select **View API**.
4. Expand API endpoints.

Depending on your application's components, endpoints are displayed. For example, in a process application you can view endpoints for connectors and processes; and in a decision application you can view endpoints of the exposed decision services.

Version 1.0 is active ✕

Choose "Deactivate and purge" to delete existing process instances (running or completed) and associated analytics data for this application. Once the current version is deactivated, you will be able to activate it again to push your changes.

i If you want to retain your existing data, then create a new version of this application and deploy it.

Deactivate and purge

Snapshot

Snapshot Tue Dec 05 2023 - 11:03:24 AM

Last Activation

◆ Tue Dec 05 2023 by opauser1 Test in Workspace ⋮

Rest API ✕

Operations ↗

OpenAPI [https://opatest\[redacted\].oci.oc-test.com/process/api/v1/applications/Correlation/versions/1.0/processes/Order/openapi](https://opatest[redacted].oci.oc-test.com/process/api/v1/applications/Correlation/versions/1.0/processes/Order/openapi)

Operations ↗

[Explore complete API](#)

5. If needed, click **Explore complete API** to view all available endpoints for Process Automation. See available endpoints in [REST API for Process Automation](#).

View Activated Application Versions

Suppose you have multiple activated versions of your application, if required you can view all the activated application versions and work with them.

To view activated application versions:

1. From an application's main page, click **Main menu** ⋮ and choose **View Activated Versions**.

The Activated Versions page appears. It displays all activated versions of the application.

2. You can identify an active version by its version tag, its status, the date of creation, and the user who created it.
3. Click **View more options** (three horizontal dots) on an activated application version's row. If required, select any of the following options:
 - **Test in Workspace:** Click this option if you want to test the activated application version in workspace. This option only appears for a process application. See [Activate and Test Process Applications](#).

- **Deactivate:** Click this option if you want to deactivate the activated application version. See [Deactivate an Application Version](#).
- **View API:** Click this option if you want to view REST API endpoints used in the activated application version. See [View Application API Endpoints](#).

Work with Process Applications

A process application is the primary type of application in Process Automation that can contain various design components such as processes, UIs, roles, and connectors. Once activated, a process application becomes available in Workspace for users to start requests and complete tasks.

Topics:

- [Create a Process Application](#)
- [Add Design Components](#)
- [Activate and Test Process Applications](#)
- [Localize Applications](#)
- [Configure Searchable Fields Based on Business Data](#)

Create a Process Application

To create a new process application:

1. On the Designer home page (All Applications), click the **Create** button on top right of the page.
2. In the Create application pane, click **Create process application**.
3. In the Create process application pane, enter details for the following fields:
 - a. **Title:** Enter a name for the process application. Give a meaningful name so that users get an idea of what the application is and why they may want to use it.
 - b. **Identifier Name:** This is the unique identifier of the process application. By default, this value is same as the title. But you can edit it if required.
 - c. **Version Tag:** The alphanumeric tag that is used to identify this version after activation. The default tag is 1.0.
 - d. **Description:** Enter a suitable description for the process application.
4. Click **Create**.

A confirmation dialog lets you know the process application has been created. You can click the **Open now** link to open the newly created process application.

The newly created process application is listed under the All Applications and the Process Applications pages.

Now, build your process application by adding design components to it. See [Add Design Components](#).

Add Design Components

After you've created a process application, you can add design components to the process application and model it according to your business needs.

A process application can contain some or all of the following design components:

Components	Description
Processes	<p>Model your business process using structured or dynamic processes.</p> <ul style="list-style-type: none"> • A structured process follows a defined flow of human or automated tasks. See Design Structured Processes. • A dynamic process depends on human actions, events, and data values. See Design Dynamic Processes.
UIs	<p>Create user interactions that display or request data from users.</p> <ul style="list-style-type: none"> • Quick Form: Quickly lay out a form. Use more advanced controls later, if needed. See Work in the Forms Editor. • Web Form: Design using the full range of options. See Work in the Forms Editor. • Linked UI: Connect and exchange data using a UI designed externally. See Add a Linked UI Connection.
Decisions	<p>Create decision models to automate the decision logic inherent in your business processes. You can either add an inline decision created within the context of your process application or add an external decision service. See Model Decisions.</p>
Connectors	<p>Connect to an external system using REST APIs. See Work with Connectors.</p>
Types	<p>Create business types and list of values to group related data types and define the structure of data used.</p> <ul style="list-style-type: none"> • Business Types are created by adding attributes. See Create a Business Type. • List of Values are created by adding set of predefined values. See Create a List of Values.
Roles	<p>Specify who can do what in your process application. See Configure Roles for Process Applications.</p>

Activate and Test Process Applications

Activating a process application moves its resources from design time (Designer) to runtime (Workspace), where users can run it in either a test or production capacity.

To activate and test a process application:

1. Activate the process application. See [Activate an Application Version](#).

To know more about working with snapshots and versions, see the following topics:

- [Work with Snapshots](#)
- [Work with Versions](#)

2. Test the process application in Workspace.

- a. In the Activate pane, click **Test in Workspace**.

A new browser tab opens in Workspace. Notice that the URL is the same except for `workspace` at the end rather than `designer`.

- b. On the Start Requests page, select your activated process application. If you deselected the **Make it default** field, click the **All** tab to display all activated versions rather than just default versions.
- c. Run the process application as an end user.
 - [Start an Application](#)

- [Work on Tasks](#)
- [Track Processes](#)

Troubleshoot: If you can't see your activated process application in the Start Requests page in Workspace, ensure that you've configured a role that can start the process in the application. See [Enable Users to Start Your Application](#).

Localize Applications

Process Automation supports localization for a number of components such as process applications, processes (structured and dynamic), UIs (forms and external UIs), and decisions.

In addition, localization is also supported for dynamic labels of the following:


- Forms
- Structured process - activity name in process flow, human task title, and start form title
- Dynamic process - stage name, activity name, human task title
- Process application title and description

By default, all process applications are created in English. To localize an application, first choose and add a locale from the available languages, and then provide localized strings for the new locale in the Localization editor. The editor's interface provides a consolidated view of all default and translated resources and allows you to manage them from one place.

When the process application is activated in Workspace, users can view labels in the localized language based on their browser language setting.

Note that if the browser language chosen by the user was not added as a locale in Designer, then labels are displayed in the default language (English).

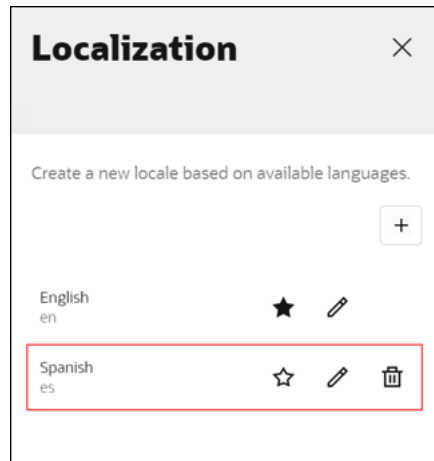
To localize an application:

1. Open an application.
2. Click **Main menu**  on the top right of the application page, and select **View Localization**.
3. In the Localization pane, click **Add**.
4. Choose a locale from the **Select Locale** drop-down field.

You can also search for a language in the resulting **Search** field while you choose a language.

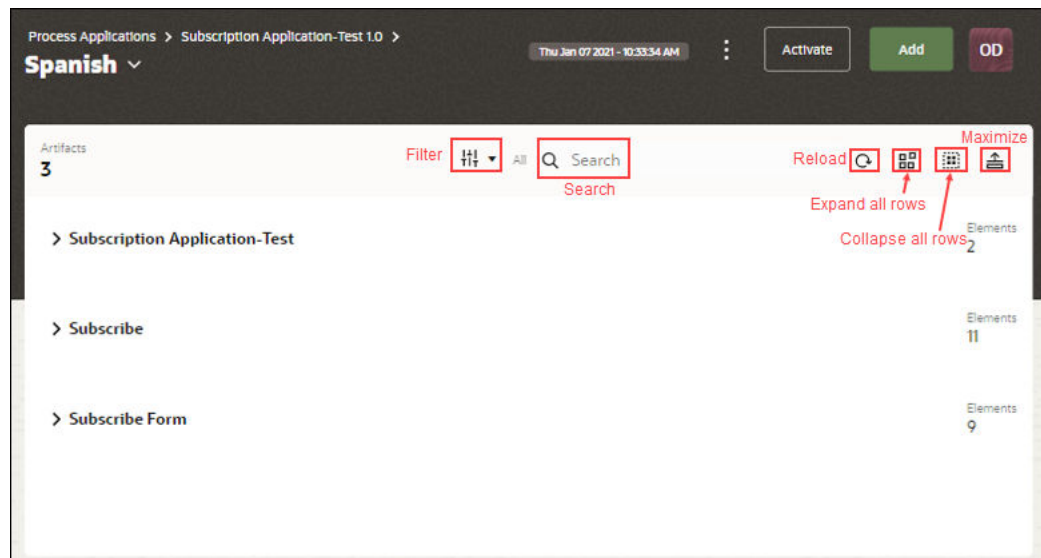
5. Optionally, select the **Make it default** check box to make it the application's default language.
6. Click **Create**.

The new locale is displayed in the Localization pane.



Notice the three icons.

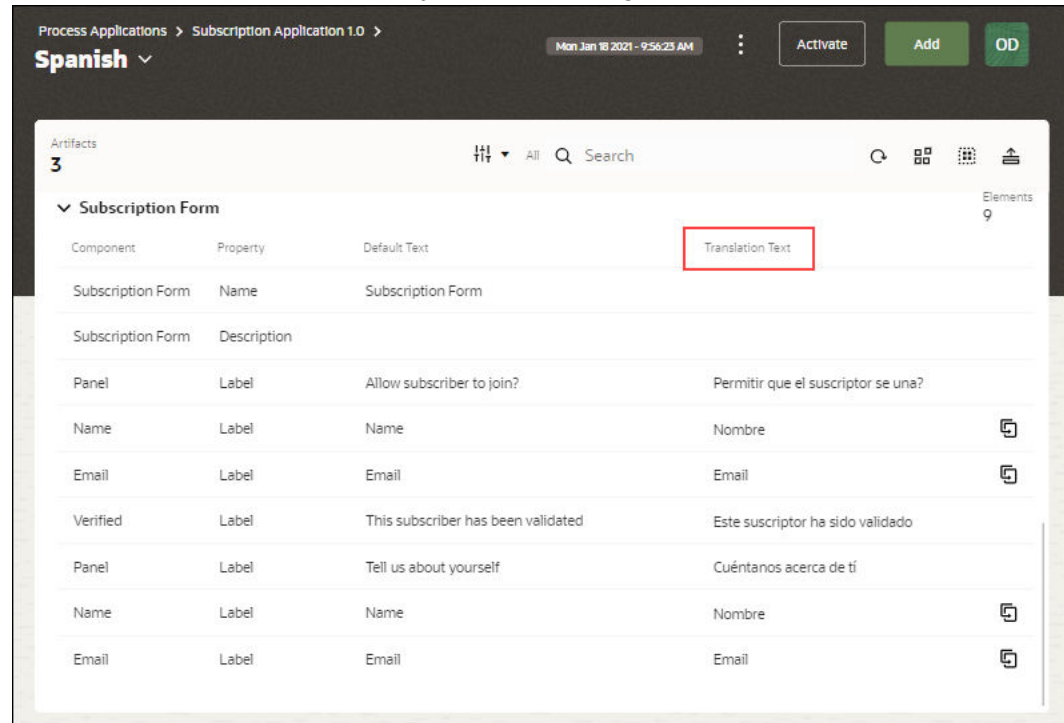
- **Make it default** icon: Click to make the selected locale the default language of the application.
 - **Edit** icon: Click to open the Localization editor.
 - **Delete** icon: Click to delete the locale.
7. Click **Edit** to open the application's artifacts in the Localization editor.
 8. Notice the following in the Localization editor:




- a. The main canvas of the editor displays the process application's artifacts. You can view the total number of artifacts displayed on the page in the top left corner.
- b. At the right of each artifact, you can see the number of components or elements present under each artifact.
- c. **Filter** with a drop-down arrow next to it lets you select the artifacts (Project, Process, and Form) that are to be displayed on the page.
- d. A search field lets you search components by property. For example, if you search by Title, all components in the application with property Title get displayed.
- e. **Reload** icon lets you reload the page so that the latest changes get reflected.

- f. **Expand all rows** icon expands all artifacts so that you can view or edit any of the elements.
 - g. **Collapse all rows** icon collapses all artifacts, and gives you a concise view of the artifacts.
 - h. **Maximize** icon expands the central pane with the four columns – Component, Property, Default Text, and Translation Text.
9. Click the **Translation Text** field of a component, and enter the translated text.

The localization editor automatically saves the changes.




Note the **Duplicate**  icon besides some of the rows. When you click the **Duplicate** icon besides a row, changes made to the translated text of that row gets applied to all its duplicate counterparts.

Configure Searchable Fields Based on Business Data

Based on your business needs, you can map certain fields to business data in your process application and configure these business critical fields as searchable. In runtime, these fields can be used as search keywords or filters in the Tasks page to let users search for instances and tasks.

To configure searchable fields:

1. Open a process application for which you want to define search fields based on business data.

2. Click **Main menu**  on top right of the process application page. Select **Define Business Search**.

The Business Search window opens.

3. Click **Add field**.

Notice that all process and form data objects present in the process application are displayed under the Attributes pane, on the left of the Business Search window. You can add search fields using these process or form data attributes which are the business data of your process application.

4. Select a process or form data object under the Attributes pane, and click **Add Attribute**. The attribute gets added to the list on the right of the Business Search window. You can also drag and drop an attribute to the list.

Notice the following in the attribute list table:

- By default, the attribute is added as searchable data into the Tasks page in runtime.
 - If you don't want to add the attribute to the Tasks page, then deselect the corresponding check box in the Task column.
 - To delete the attribute from the list, click the Delete icon under the Action column.
5. If required, you can change the name and type of the search field.
 - The name should always start with a lower case alphabet.
 - You can choose a type from the **Type** drop-down field.
 - Select the **Match whole words** check box to enable whole word search. Leave this check box deselected, if you want to enable partial search of words.
 6. Click **Apply**.

The field is added to the Business Search window.

In runtime, you can use the configured searchable fields to search for process instances or tasks. See [Search for a Task](#) and [Apply Advanced Search Filters](#).

Configure Roles for Process Applications

You can configure roles specific to process applications to specify data access and task permissions for users and groups.

To configure process application roles for users or groups in Designer you have to be a Process Automation Designer (assigned the ServiceDeveloper IDCS application role). These roles take effect upon activation of the process application.

Topics:

- [About Process Application Roles](#)
- [Users, Groups, External Applications and Permissions](#)
- [Work with Roles in Designer](#)

As a Process Automation Administrator (assigned the ServiceAdministrator IDCS application role), you can manage process application roles in Workspace. See [Manage Roles in Workspace](#).

Note:

The users and groups assigned process application roles do not need the ServiceAdministrator or ServiceDeveloper IDCS application roles.

To know about the different types of roles in Oracle Cloud Infrastructure Process Automation, see Process Automation Roles.

About Process Application Roles

Process application roles determine what data a user or group can access, and what actions they can perform in Process Automation instances, interfaces, and user tasks.

A process application role consists of users/groups and permissions. Learn more about how users, groups and permissions are defined in roles. See [Users, Groups, External Applications and Permissions](#).

Types of Process Application Roles

There are two types of process application roles: Application roles and Global roles.

- **Application roles** are specific to a particular process application and can be used only for that process application. They can't be shared with other process applications.
- **Global roles** are defined once and then can be linked and used wherever they're needed in process applications.

Some key points about *Global* roles:

- A global role's users/groups are global, but their permissions are application-specific, meaning that their permissions can vary in each process application in which the role is used. For example, a HR Recruiter global role contains all HR recruiters, and these users may have Use permission in one process application but only Read permission in another process application.
- Keep in mind that if you change a global role's users/groups, this change affects all process applications that link to that role.
- Process Automation Administrators can see all users/groups assigned to a global role in Workspace.
- A global role must be present in an activated application or created in Workspace before it is available to be linked to process applications in Designer.

Recommended best practice: Using global roles can result in fewer roles. If your organization has many process applications running in the same domain, using global roles linked to process applications may make sense. On the other hand, if your organization has very different types/domains of process applications, it may be best to keep roles specific to their process applications.

See [Create an Application or Global Role](#).

Default Process Application Roles

- **Process Application Administrator:** This global role gets created by default with a Process Automation instance. The Process Application Administrator role has full permissions (that is, the **Manage** permission) on all applications, versions, and resources. To know more about permissions, see [Users, Groups, External Applications and Permissions](#).

As a Process Automation Administrator (assigned the ServiceAdministrator IDCS application role), you can assign users or groups to the Process Application Administrator role in Workspace. See [Assign the Process Application Administrator Role](#).

 **Note:**

Process Application Administrator is a system generated role and can't be deleted.

- **ProcessUser:** This application role gets automatically created when you create a structured process, and gets displayed in the swimlane. You can either use it for task assignments or delete it.

The following table lists how and where Process Automation Designers and Process Automation Administrators work on process application roles.

Who	What can they do and where?
Process Automation Designers	In Designer , Process Automation Designers add roles, and then assign these to user interactions such as human tasks in structured and dynamic processes. Process Automation Designers also need to allow specified users to start process requests in Workspace.
Process Automation Designers	In Designer , Process Automation Designers activate process applications, and the role assignments are made available in Workspace.
Process Automation Administrators	In Workspace , Process Automation Administrators can view all application and global roles, including their assigned users/groups and the process applications in which they're used. Process Automation Administrators can edit and delete application or global roles, and also add global roles. In addition, they can also update role permissions.

Users, Groups, External Applications and Permissions

A process application role consists of users and/or groups, and permissions.

Users and Groups

Users and groups for your Process Automation instance are created in Oracle Identity Cloud Service (IDCS) or Oracle Cloud Infrastructure Identity and Access Management (IAM). Note that to create users and groups in IDCS or IAM, you have to be assigned the identity domain administrator or user administrator role.

See Manage Oracle Identity Cloud Service Users and Manage Oracle Identity Cloud Service Groups in *Administering Oracle Identity Cloud Service*.

See [Managing Users](#) and [Managing Groups](#) in Oracle Cloud Infrastructure documentation.

Once created, users and groups will be available to be assigned to application and global roles by Process Automation Designers during design and testing. If required, Process Automation Administrators can later update them for production in Workspace.

External applications

You can also authenticate and authorize an external application in a flow. In a business scenario that uses machine to machine flows, the process instance and decision services can handle requests that contain bearer tokens generated through the OAuth client credential flow. To use an external application in a flow, as a prerequisite, you must have an Oracle Cloud Infrastructure Identity and Access Management (IAM) application configured to authenticate a client using the client credentials grant. This results in a scoped bearer token that can access the Process Automation instance.

You can either use the default Oracle Cloud Infrastructure Identity and Access Management (IAM) application created for your Process Automation instance or create a new integrated application and configure it appropriately.

To initiate a Process Automation instance using the client credentials flow, add the external application to a role with Use permission.



Note:

You can perform GET and POST operations at all instances level, while you can perform only the GET operation on a specific instance.

Permissions

Users, groups, and/or external applications added to a process application role have to be assigned one of four permissions, listed in the table below from least amount of access to the most. For example, **Inspect** grants the least control and **Manage** grants the most. Behind the scenes, permissions use a data access control (DAC) model.

Permission	Description	Target Users
Inspect	<p>Allows users to list tasks and processes, but not view their details.</p> <p>For example, users assigned the Inspect permission can review a list of pending tasks without going into individual tasks and seeing their details.</p>	<p>Viewers</p> <p>Users who want to get an overview of pending tasks in a process application.</p>
Read	<p>Allows users to Inspect plus the ability to perform the following:</p> <ul style="list-style-type: none"> View details of processes and their activities, including attachments and comments. View process's task audits. 	<p>Reviewers</p> <p>Users who want to view details and review tasks and processes in a process application.</p>
Use	<p>Allows users to Read plus the ability to perform the following:</p> <ul style="list-style-type: none"> Start an application. Work on processes. Work on tasks, such as add comments and attachments. 	<p>Users/Process Users</p> <p>Users who want to review as well as work on tasks and processes in a process application.</p>

Permission	Description	Target Users
Manage	<p>Allows users to Use plus the ability to perform the following:</p> <ul style="list-style-type: none"> Withdraw and reassign tasks. Terminate processes and their activities. Alter the flow of a process. View and work on analytics in Workspace. 	<p>Power Users and Process Application Administrators</p> <p>Users who want to have maximum permission for a process application.</p> <ul style="list-style-type: none"> Any user who is assigned the Manage permission automatically becomes a Power User as such a user have the maximum level of permission for a process application. The Process Application Administrator role by default has Manage permission on all process applications of a Process Automation instance. To assign users to the Process Application Administrator role, see Assign the Process Application Administrator Role.

Work with Roles in Designer

Learn about working with roles in Process Automation Designer.

- [Create an Application or Global Role](#)
- [Link a Global Role to Your Application](#)
- [Assign Roles to User Tasks in Structured Processes](#)
- [Assign Roles to User Tasks in Dynamic Processes](#)
- [Enable Users to Start Your Application](#)
- [Remove Roles in Designer](#)

If you're a Process Automation Administrator (assigned the IDCS application role ServiceAdministrator), learn about managing roles in Workspace. See [Manage Roles in Workspace](#).

Create an Application or Global Role

In Designer, you can create and define a role such as set its scope as either application or global, assign users/groups, and add permissions.

- In Designer, open a process application.
- Click **Add** in the upper corner. The Add component pane appears.
- Expand **Roles**, and click **New**.
- Under **Create role** options, complete these fields. Note that these entries can't be changed later.
 - Title:** Enter a name that describes who is assigned the role or what they can do (for example, *Subscriber* or *Approver*).
 - Identifier:** You can leave this unique identifier that's created for you, or change it. You can't change this identifier later.

- **Scope:** Choose **Application** to use the role in this application only. Choose **Global** to allow this role to be used in other applications.
5. Click **Create**. The role is created.
 6. Open the role from the **Roles** tab.
 7. Assign users, groups and/or external applications to the role.
 - a. In the **Category** field, select **Users, Groups, or External applications** from the drop-down list.
The search field displays the appropriate text depending on the category that you selected.
 - b. Click the search field, a list of available users, groups, or external applications are displayed.
 - c. Select the user, group, or external application to add it to the role. The selected user, group, or external application gets listed on the roles page indicating that it has been assigned to the role.

You can assign a combination of users, groups, and external applications to a role. Process Automation Administrators can also select users, groups, external applications later in Workspace to add to the role.

8. Under **Choose permission**, choose a level. The default is **Use**.

Permission	Allows Users to...	Example Use
Inspect	List resources, but not read them	Allow a user to review of a list of pending tasks without going into individual tasks and seeing their details.
Read	View resource details, but not edit them	<ul style="list-style-type: none"> • View details for tasks and structured and dynamic processes, including attachments and comments • View structured and dynamic process definitions and task audits.
Use	Perform actions on resources	<ul style="list-style-type: none"> • Start a structured and dynamic process instance. • Complete an activity.
Manage	All the above plus delete processes, withdraw, release or reassign tasks, alter the flow of a process, and view as well as work on analytics data of instances.	<ul style="list-style-type: none"> • Withdraw, reassign, or release tasks • Terminate structured or dynamic processes

You can now use the role in structured and dynamic processes. See:

- [Assign Roles to User Tasks in Structured Processes](#)
- [Assign Roles to User Tasks in Dynamic Processes](#)

Link a Global Role to Your Application

You can link to a global role instead of creating a new one for an application. Linking means finding an existing global role and bringing it into your application to use in a structured or dynamic process, or to define permissions for this application.

Note that a global role must be activated (exist in an application that's been activated) or created in Workspace to be available for linking.

1. In Designer, open a process application.
2. Click **Add** in the upper corner. The Add component pane appears.
3. Expand **Roles**, and click **Global**.
4. Under **Link role** options, click the **Search** field. Search for and select a global role to use in your application.
5. Click **Link**.

The role is added to your application. You can now use it in structured and dynamic processes, or to define permissions for this application. Process Automation Administrators can change the global role's users and groups, and permissions in Workspace.

Assign Roles to User Tasks in Structured Processes

When selecting assignees for user tasks, you have multiple options. When the task executes, users' assigned tasks display under **My Tasks** in Workspace.

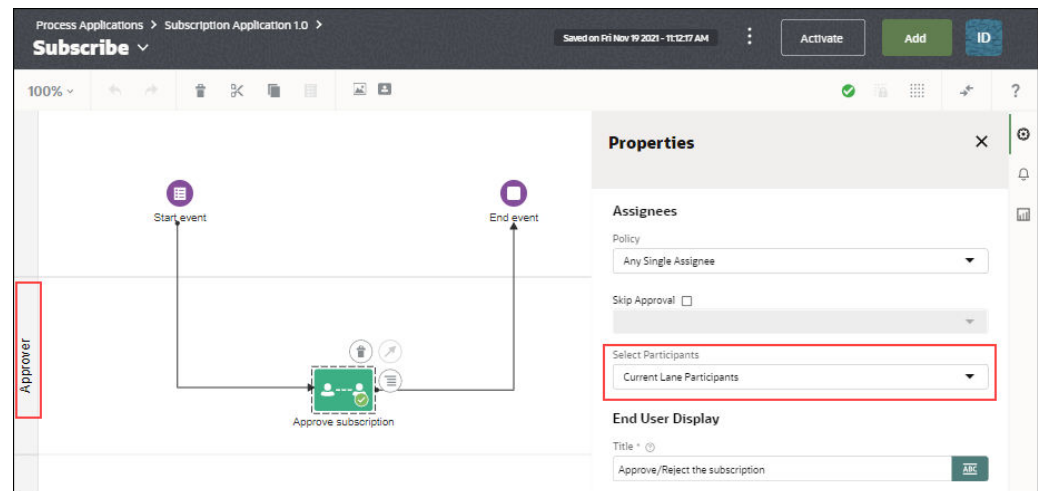
You can select assignees based on:

- The role assigned to the swimlane
- The selected assignee (user, role, or expression)

To select an assignee:

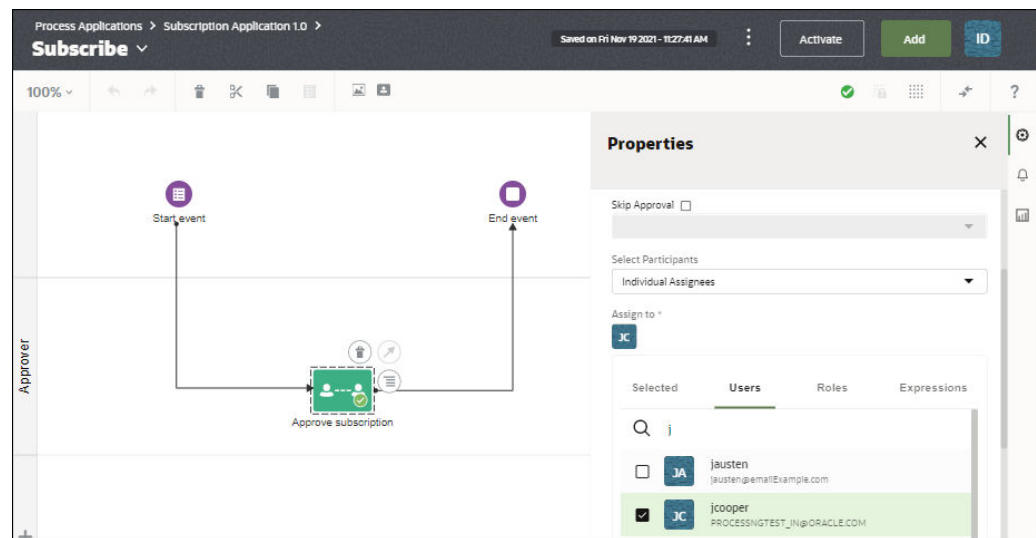
1. In Designer, open a process application.
2. In a structured process, open the Properties pane of an activity and make a selection in the **Select Participants** field.
 - a. Choose **Current Lane Participants** to assign based on the swimlane's role.

In the example below, users or groups assigned to the Approver role will be assigned the Approve subscription task.



- b. Choose **Individual Assignee** under **Assignees** to select a specific assignee. Click **Assign to**, then choose a role, roles, or expression. Use the search field to search for specific user or role.

In the example below, the user jcooper is assigned to the Approve subscription task.



Assign Roles to User Tasks in Dynamic Processes

When selecting assignees for user tasks, you can select a local or global role. When tasks execute in runtime, users' assigned tasks display under **My Tasks** in Workspace.

Note that you can also assign a task directly to a specific user on the **Users** tab, or use an expression to specify the user.

1. In Designer, open a process application.
2. In a dynamic process, add or open a human task activity and edit its properties.
3. In the Properties pane, click **Assign to** under **Assignees**.
4. Click the **Roles** tab and select a local or global role for the task.

Note that tasks assigned to a role display under Team Tasks in Workspace. Task directly assigned (single-user assignment) display under My Tasks.

Enable Users to Start Your Application

A process application starts when its structured or dynamic process starts. Typically, a process starts by a user who selects the process under **Start Requests** in Workspace, and submits an associated form. In the process start, enable users to start the application.

When specifying who can start the process, you can specify:

- All users and external applications
This setting enables any authenticated user who can access Workspace or a Process Automation embeddable component (snippet) can start the process.
- Role members with at least Use permission
To assign a role's permissions, see [Create an Application or Global Role](#).

To specify who can start the process:

1. In Designer, open a process application.
2. Create or open a process.
An application can start with a structured or dynamic process.
See [Create a Structured Process](#) or [Create a Dynamic Process](#).

3. Open the process start properties.
 - a. In a **structured process**, drag a Form Start or a Message Start activity onto the canvas as the process start. Select the start event and choose **Open Properties**.
 - b. In a **dynamic process**, click **Start Process** at the top of the canvas.
4. Select who can start the process.
 - a. In a **structured process**, select from the **Who can start the process?** field of the Properties pane.
 - b. In a **dynamic process**, select from the **Who can start the process?** field in the **Start Process** dialog.

In either type of process, select one of these options:

- **Role members with at least Use permission** Allows any user assigned a role with Use or Manage permission to see and start the application. Note that in a structured process, a user does not need to be assigned to the swimlane to start the process.
- **All users and external applications** Allows any user or external application with access to Process Automation to see and start the application.


Remove Roles in Designer

You can remove a role if it is no longer needed in your application. Removing a role removes it from any dynamic or structured processes in which it is used.

Removing a role is different for application versus global roles:

- You **delete** an application role from your application.
- You **unlink** a global role from your application. The global role can still be used in other applications. Global roles must be deleted by Process Automation Administrators in Workspace.

To remove a role:

1. In Designer, open a process application.
2. Select the **Roles** tab. If you have application and global roles, notice that their icons are different.
3. Click  for the role you want to remove, and choose **Delete** (for an application role) or **Unlink** (for a global role).
4. Confirm that you want to remove the role.

The role is removed.

Work with Decision Applications

Decision applications are applications that only contain decisions. They do not contain any other design components. Once activated, the decision services exposed by the decisions in the decision application can be used in other process applications and also by external applications.

Topics:

- [Create a Decision Application](#)
- [Activate and Use Decision Applications](#)

Create a Decision Application

To create a decision application:

1. On the Designer home page (All Applications), click the **Create** button on top right of the page.
2. In the Create application pane, click **Create decision application**.
3. In the Create decision application pane, enter details for the following fields:
 - a. **Title:** Enter a name for the decision application. Give a meaningful name so that users get an idea of what the application is about and why they may want to use it.
 - b. **Identifier Name:** This is the unique identifier of the decision application. By default, this value is same as the title. But you can edit it if required.
 - c. **Version Tag:** The alphanumeric tag that is used to identify this version after activation. The default tag is 1.0.
 - d. **Description:** Enter a suitable description for the decision application.
4. Click **Create**.

A confirmation dialog lets you know the decision application has been created.

The newly created decision application is listed under the All Applications and the Decision Applications pages.

Open the decision application by clicking the **Open now** link in the confirmation dialog. Alternatively, select and open the decision application from the All Applications or Decision Applications page. By default, it opens in the Decision editor canvas where you can model the decision:

- [Add a Decision Logic Type](#)
- [Define Decision Input and Type](#)
- [Model Decision Logic](#)
- [Expose Decisions as Services](#)

Activate and Use Decision Applications

After modeling a decision and exposing it as a service, you must activate the decision application so that the decision service exposed can be used by other process applications and external applications.

1. Activate the decision application. See [Activate an Application Version](#).

To know more about working with snapshots and versions, see the following topics:

- [Work with Snapshots](#)
- [Work with Versions](#)

2. Use the activated decision application in process applications as external decision. See [Understand Local and External Decisions](#).

You can call the decision services of an activated decision application from other process applications. First, create an external decision with the activated decision application and then use the exposed decision services in processes of your process application.

- [Create a Local or External Decision](#)

- [Add Decision Services to Processes](#)
3. Use the activated decision application in external applications.
You can call the decision services of an activated decision application from external applications. You use the decision service REST API endpoint URL, the request payload and the response payload in a `POST` operation to call the decision service from an external application.
See [Expose Decisions as Services](#).

Create an Application from a Recipe

Process Automation offers some pre-built, run-ready applications called **recipes**. Recipes help you to quickly create an application and jump-start your application design as all the required resources and components for the application are provided in the recipe.

To create an application from a recipe:

1. On the Designer home page (All Applications) page, click the **Create** button on top right of the page.
2. In the Create application pane, click **Create application from recipe**.
3. In the Choose a recipe page, find the recipe.
Click the recipe card to view details about the recipe such as the date it was published, keywords, description, documentation link and so on.
4. Click **Select**.
The Create application from recipe pane opens.
5. Review the name, version, and description for the application. If required, you can update these fields.
6. Click **Create**.
You get a confirmation that the application is being created. Once created, the application's main page opens, displaying all the resources of the recipe.

3

Design Structured Processes

A key part of your application is the business process. When developing a structured process, your first step is to determine the people and roles required to complete each task that requires user interaction. You then use the various activities, such as human tasks, system activities, and other events, to design the flow of your process.



Note:


In Oracle Cloud Infrastructure Process Automation, you can create two types of process: **structured** (described below), and **dynamic**, described in [Design Dynamic Processes](#).

Topics:

- [Typical Workflow for Designing a Structured Process](#)
- [Use the Structured Process Editor](#)
- [Create a Structured Process](#)
- [Work with Activities](#)
- [Work with Sequence Flows](#)

Typical Workflow for Designing a Structured Process

Use this typical workflow to design a structured process in Designer.

Task	Description
Create a process	Create a structured process from the Add component pane. See Create a Structured Process .
Assign roles to the process	Assign roles and determine which members of your business organization are responsible for completing the tasks and activities within the structured process. See Work with Roles in Designer .
Design the flow	Drag and drop activities such as human tasks, system, events onto the canvas. See Work with Activities
Configure activity properties	Select an activity in your process diagram, click  , and select Open Properties to open the Properties pane. Configure properties specific to the activity. For example, configure a form, service call, or send task and receive task.
Define the data	Define the data the activity uses. See Work with Data Objects

Task	Description
Associate the data	Link data to the activity inputs and outputs. See Configure Data Association





Use the Structured Process Editor









The structured process editor provides tools to easily create and edit a structured process.

The editor is divided into four areas - Toolbar, Canvas, Activities palette, and Swimlane.

Process Editor Toolbar

The process editor toolbar provides quick access to tools related to process design.

Toolbar Icon	Name	Keyboard Shortcut	Description
	Set View	Not available	Set the view size.
	Undo	Ctrl+Z	Revert to the last change made to your process.
	Redo	Ctrl+Y	Reverse the last Undo action you performed.
	Delete	Delete key	<p>Delete the selected items from your process.</p> <p>When you delete an activity that contains an incoming and outgoing sequence flow, the incoming sequence flow is automatically connected to the outgoing sequence flow. However, you may need to manually re-configure the surrounding activities.</p> <p>When you delete a sequence flow from a business process, any implementation details you may have configured for that sequence flow are lost.</p>



Toolbar Icon	Name	Keyboard Shortcut	Description
	Cut	Ctrl+X	<p>Cut the selected items and copy them to the clipboard.</p> <p>When you cut an activity that contains an incoming and outgoing sequence flow, the incoming sequence flow is automatically connected to the outgoing sequence flow. However, you may need to manually re-configure the surrounding activities.</p>
	Copy	Ctrl+C	<p>Copy the selected items to the clipboard.</p> <p>Note that when you copy a human task, the associated form is not copied.</p>
	Paste	Ctrl+V	Paste the items currently in the clipboard.
	Change Icon	Not available	<p>Change the icon of the selected item. You can also click it to return to the default.</p> <p>Change Icon is available only when a valid activity is selected.</p>
	Change Type	Not available	<p>Change the type of the selected item.</p> <p>Change Type is available only when a valid activity is selected.</p>
	Snap to Grid	S	<p>Center the activities in your process on the closest grid axis.</p> <p>Existing activities are automatically centered and new activities are automatically centered when they're added.</p> <p>Snap to Grid is active only when the grid is shown.</p>
	Show/Hide Grid	G	Toggle the display of a grid.
	Data Association	A	Opens the data association window where you can link data to the activity inputs and outputs.

Process Editor Canvas

The process editor canvas is the central area of the process editor window. Use the canvas to draw a diagram that represents your business process using the activities available in the Activities palette.

Activities Palette

Use the Activities palette to add activities and sequence flows to your business process by dragging and dropping activities to the process editor canvas.

Click **Expand**  to expand a specific type of palette item and see all the activities available for that type. Click the icon again to collapse. Click **Show/Hide Palette**  to close and open the palette. See [Work with Activities](#).

Swimlanes

Swimlanes are the horizontal lines that run across the process editor canvas. All activities must be placed within a swimlane. Swimlanes that contain user tasks must have roles assigned to them. Swimlanes visually display the role responsible for performing each activity within your business process.

You can have multiple swimlanes that are assigned to the same role. Swimlanes can make your business process more readable when you must use the same role in different parts of the same business process.

When you create a new business process, a default swimlane is created. You can add additional swimlanes to your business process as necessary. For example, for a travel request approval business flow, you may have two swimlanes. The Employee role is assigned to the first swimlane and the Approver role is assigned to the second swimlane.

See [Add and Edit Swimlanes](#).

See [Configure Roles for Process Applications](#).

Add and Edit Swimlanes

To add a new swimlane and edit swimlane properties:

1. Open your application and open the structured process.
Note that a default swimlane is already created for your process.
2. Click **Create New Lane** (the plus sign) below the default swimlane.
3. Click the new swimlane, and then click the edit icon that appears.
The Properties pane displays.
4. Choose a role that you had defined from the **Roles** field.
5. Optionally, change the color of the swimlane by choosing a color of your choice in the **Background Color** field.
6. Close the Properties pane.

The changes you made to the swimlane gets reflected in the editor.

Create a Structured Process

A structured process contains a start event, an end event, and possibly other activities. You can start your process from scratch (that is, with an empty start event), with a form, or when a message is received.

To create a structured process:

1. On the application main page, click **Add**.
2. In the Add component pane, expand **Processes**, and then choose **Structured**.
3. Enter a title in the **Title** field.

The field is populated with a default name, such as *MyStructuredProcess*, *MyStructuredProcess1*, and so on.

Note that the **Identifier Name** field gets auto-populated with the name you entered as the title, but without any spaces.

4. Optionally enter a suitable description (such as, what it's about, and why you would use it) in the **Description** field.
5. Click **Create**.

A confirmation dialog lets you know that the process is created. You can open the process by clicking the **Open now** link in the dialog.

The newly created process gets listed in the Processes page. If you missed the **Open now** link, click the process from the Processes page and open it.


Clone a Structured Process

You can clone a structured process to create an exact copy of the process. This is helpful when you need to quickly create and use a replica of an existing process.

When you clone a structured process, the cloned process will have all activities and properties that were configured in the original process. All underlying artifacts (such as forms, business objects, integrations) used by the original process will also be used by the cloned process. Note that the artifacts itself are not cloned; but are just used by both the original and the cloned process.

The cloned process will be a standalone process that can be executed and activated independently. You can also modify and customize the cloned process to meet your business needs.

To clone a structured process:

1. Select the structured process from the process application's main page or from the Processes page.
2. Hover over **Menu**  on the far side of the selected process, and choose **Clone**.
The Clone component pane displays.
3. In the Clone component pane, enter a name in the **Title** field and a suitable description in the **Description** field.
4. Click **Clone**.

A confirmation dialog lets you know that the component is being cloned. You can click the **Open now** link to open the cloned process.


The cloned process gets displayed as a new process in the Processes page as well as the process application's main page.

Delete a Structured Process

You can delete structured processes from your application anytime.

Before you delete, make sure that the process isn't referenced anywhere in your application. You get validation errors if any remaining references to the deleted process exist.

To delete a process:

1. Open the application.
2. In the application main page, select the structured process you want to delete.
3. Click the options menu icon  and then click **Delete**.

Work with Activities

Activities are the building blocks of your process. They usually represent the work that is performed within the process. Use the Activities palette to add activities to a process by dragging and dropping them onto the process editor canvas. After you add an activity, you define its properties, data association, forms, and sequence flows where they apply.

Activities are divided into the following types:

- Human
- System
- Events
- Gateway
- Other

Human

Human activities represent tasks where a process participant is required to perform the work. The task can be a simple interaction, such as filling out a form, or part of a more complicated workflow that requires input from multiple process participants.

- A **submit task** provides a form or submit action that the user acts on to create a request or provide information about a certain subject.
- An **approve task** provides a form for review or an approve/reject action that the user acts on to approve or reject the request.

See [Add Human Interaction](#).

System

System activities allow you to define interactions across business processes and services. For example, you can use a Service activity to invoke an external service or process, or a Call activity to call a reusable process from within the current process.

Activity	Description
Abstract	Use to designate a placeholder for another activity.
Call	Use to call a reusable process from within the current process.

Activity	Description
Data Mapper	Use to assign values to data objects and variables within the process.
Decision	Use to incorporate decision services exposed through a decision created within your application, into your business process.
Notify	Use to send an email notification to a user.
Send	Use to asynchronously start or send a message to another process.
Receive	Use to receive a message from another process. Send and receive activities are typically used as pairs to communicate between asynchronous processes. See Use Send and Receive .
Service	Use to communicate with other processes and services.
Subprocess	Use to group related activities in your business process into inline subprocesses.
Event Subprocess	Use to handle exceptions that occur in the runtime life cycle of a process.
Upload Form	Use the upload form activity to upload snapshot of a form in a process instance to an external database or service.

Events

Event activities can be divided into two types:

- Start and End activities that define the starting and ending points of a process
- Intermediate activities that can either occur within the typical flow of your process or trigger an interruption with your process

Activity	Description
Start	Use when no instance trigger is defined, such as when a process instance is created by another activity, or as a placeholder when the start event isn't known.
Form Start	Use to trigger a process instance when a user submits a form.
Message Start	Use to trigger a process instance when an email message is received.
Timer Catch	Use to control the flow of your business process using a time condition.
End Event	Use to mark the end of a process path, or as a placeholder.

Gateway

Gateway activities determine the path a token takes through a process. They define control points within your process by splitting and merging paths.

Activity	Description
Exclusive	Splits a process into multiple paths, where process flow continues down only one of the paths. The decision about which path the process should proceed along is based on data-specific conditions.
Inclusive	Splits a process into multiple paths, where process flow can continue down multiple paths depending on conditional sequence flow.
Parallel	Splits a process into multiple paths, where process flow continues down all paths simultaneously.

Integrations

Use the integrations activity to quickly add integrations to your structured processes. You can do so by either of the two ways:

- Drag and drop an already configured integration connector from under the **Integrations** category in the Activities palette.
- Drag and drop an integration activity from under the **System** category in the Activities palette, and configure the connector's properties from the activity's properties pane.

See [Work with Integrations](#).

Other

Notes are equivalent to sticky notes. They're temporary and you should use them more as a reminder and delete them as soon as the information is used.

Add Activities

You can add an activity to your business process by dragging it from the Activities palette onto the process editor canvas.

To add an activity to a process:

1. In the Activities palette, click **Expand** ▼ next to an activity type.
2. Click and drag an activity onto the process editor canvas where you want to add it.
The cursor displays the icon associated with the activity type.
3. Position the cursor at the point in your process where you want to add the activity.

Note:

If you position the cursor over a sequence flow, incoming and outgoing sequence flows for the new activity are automatically created.

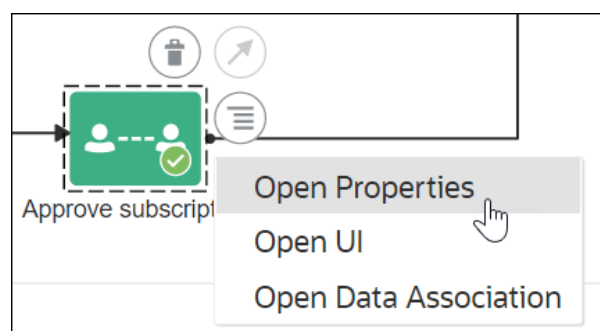
Adding a gateway activity, such as an exclusive gateway, requires more steps. See [Create a Gateway](#).

Edit Activity Properties

You can edit the basic properties of an activity from its properties pane in the process editor.

To open the properties pane of the activity, double-click the activity, or select the activity, click

, and select **Open Properties**.



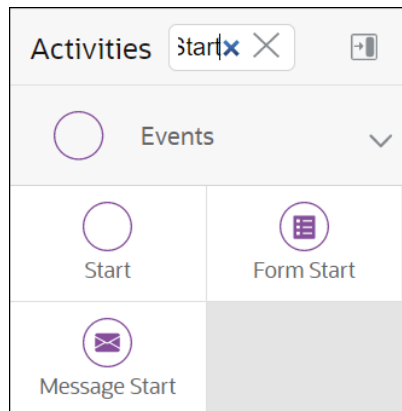
Looking for a keyboard shortcut? After you select an activity, press **p** to open the properties of the activity.

Find Activities in the Palette

You can search for an activity by entering its name in the **Search** field of the Activities palette.

To search for an activity in the palette, enter the name of the activity you want to include in your process.

You can also search for an activity by using a text string in its name. For example, the following search results are displayed when you enter the text string `start` in the **Search** field.





Add Human Interaction

Use human task activities to model the user interaction with the application. In Process Automation, process participants interact with your business application during runtime.

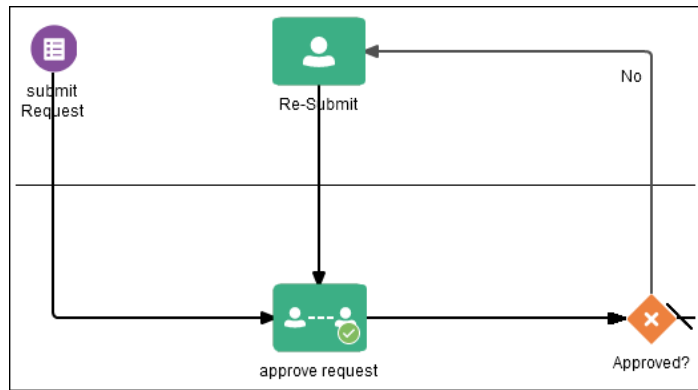
The human task activity represents a part of your process where a process participant is required to perform an action. The task can be a simple interaction, such as entering a form, or part of a more complicated workflow that requires input from multiple process participants.

After you create the human task activity, you can configure implementation details such as its assignment, priority, due date, reminders, and deadlines. Human task activities may also contain incoming and default outgoing sequence flows.

Different types of human task activities let you model different types of interactions.

Human Task Activity	Description
 Submit	Lets you display a form that the user must submit to create a request or to provide information about a certain subject.
 Approve	Lets you display a form that the user must review or complete, and then perform a certain action. The user might approve or reject the request. You can also define custom actions for the user to perform. Approval tasks lets you define an approval pattern. Generally, you use the outcome of the approval task to drive the rest of the process flow.

When a token reaches a human task activity, the corresponding task is performed. The token waits until the human task activity is completed before continuing to the next activity. For example, look at the following process where the loan request is submitted using the form start event:



After the user enters information into the form, the process flow passes the outgoing sequence flow to the loan approval human task activity. If the loan is approved, then the flow passes to the fulfill human task activity. If the loan application is rejected, then the flow passes to the re-submit human task activity for more information from the applicant.

See [Create Human Task Activities](#).

Work with Events

Event activities of your process determines the start, end, and sometimes intermediate flow and behavior of your business process.

- [Timer Start Event](#)
- [None Start Event](#)
- [Form Start Event](#)
- [Message Start Event](#)
- [Timer Catch Event](#)
- [End Event](#)
- [Error End Event](#)
- [Error Boundary Event](#)
- [Terminate End Event](#)

Timer Start Event

The timer start event is used when you want to start or initiate structured process flows on a predefined schedule. You can monitor the scheduled processes in the instance tracking and analytics pages.



This event is useful in business usecases that require periodic repetitive actions. For example, you can use the timer start event to initiate a process to send weekly email reminders to all the employees to submit their timesheets.



Note:

To stop a schedule, you have to deactivate the application.

You can configure hourly, daily, weekly, or monthly schedules as shown in the following screenshot:

Timer ✕
Activity: Start event

Recurrence
Daily

Repeat every (in days)
1

Repeat time
 At specific time
 Every certain time

Add time
12:00:00 AM UTC +
12:00:00 PM UTC ✕

The timer start event also allows you to schedule triggering of process flows at repetitive time intervals. Process Automation calculates all the schedules according to UTC time zone.

Additionally, to configure advanced schedules using cron expressions, select **Advanced** from the **Recurrence** dropdown list, and then specify the cron expression.

Here is how you can use the cron expressions:

Example 1: Trigger a process flow every five minutes

```
0 0/5 * * * ?
```

Example 2: Trigger a process flow every 5 minutes, and every 10 seconds after the minute (i.e. 10:00:10 am, 10:05:10 am, etc.).

```
10 0/5 * * * ?
```

The subexpressions are separated with a white space.

 **Note:**

- The timer start event rounds off to the nearest hour if the nearest hour is earlier than the next scheduled time.

For example, if an event is scheduled for 8:11 a.m, the process flow is triggered at 8:00 a.m which is 11 minutes earlier and at the beginning of the hour according to UTC.

- Due to unknown circumstances, if there is any delay in starting a schedule, the delayed start impacts subsequent scheduled triggering of the processes. For example, consider a schedule that starts at 8:00 a.m and repeats every 15 minutes. However, due to unknown circumstances, the process starts at 8:05 a.m. The next scheduled process starts at 8:20 a.m, 8:35 a.m, and 8:50 a.m. However, the next scheduled process starts at 9:00 a.m because there is an overlap between the next scheduled start at 9:05 a.m, and 09:00 a.m.

None Start Event

The none start event is used when no instance trigger is specifically defined. The none start event can be used as a placeholder when the required start event of a process is unknown, not yet defined, or implemented later by process developers.

- None start events specify the beginning of a process where the process instance is created by another event or activity. The none start event doesn't trigger a new process instance.


Similar to other start events, the none start event can't have incoming sequence flows. It can only have default out-going sequence flows.

 **Note:**

None start events are always used to define the beginning of reusable processes and subprocesses.

Form Start Event

The form start event triggers a process instance when a user submits a form. The form is specified in the implementation for the form start event.

-  The form is designed to get input from the user and present information relevant to the workflow. As you build the form, a business object is created to store the form data.

Message Start Event

The message start event triggers a process instance when a message is received. This message can be sent from another business process or from a service.



Messages are types of data used to exchange information between processes. Just as data objects are used to define the data used within an application, messages are used to define the data used between processes or between a process and a service.

Similar to other start events, the message start event can't have incoming sequence flows. Message start events require a default outgoing sequence flow.

You can expose a business process as a service that allows other processes and applications to invoke the process. To expose a process as a service, your process must begin with a message start event. Additionally, you must define the input arguments to the process, which are the data objects passed to the message start event.

The message start event allows you to specify input arguments to a process. These arguments define the message that other processes or services must send to the process during invocation.



Note:

Any process configured with a message start event will not appear in Workspace.

You can start a process configured with message start event using REST APIs.

- Run `/process-interfaces` API to get the `processDefinitionId`.
- Run `/instances` API with appropriate arguments to invoke a process.

See [Create a Process Instance](#) under *REST API for Oracle Cloud Infrastructure Process Automation*.

Timer Catch Event

Timer catch events lets you control the flow of your business process using a time condition.



Possible uses of the time catch event include:

- Creating a delay before running an activity
- Configuring a deadline for an activity
- Configuring a deadline for a process
- Triggering additional activities after an elapsed time

You can use timer events as boundary events on an activity. Timer events can be defined as either interrupting or non-interrupting boundary events.

When an interrupting timer event fires, the token leaves the main process flow to follow the process flow the timer event defines. The process flow that an interrupting timer event defines can return directly to the main process flow.

When a non interrupting event fires, a copy of the token is created and passes through the process flow the timer event defines. The process flow that a non-interrupting event defines can't return to the main process flow.

End Event

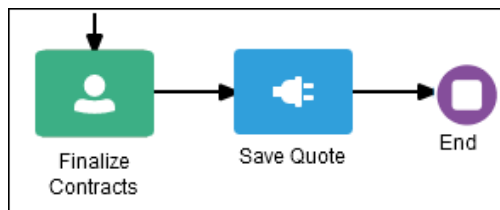
The end event marks the end of a process path. When a token reaches a none end event, it's consumed. If there are no other tokens within the process instance, then the instance is complete.



Use the end event:

- When your process isn't required to perform any action after it completes
- As a placeholder to be changed later during implementation by a process developer

For example, in a Sales Quote business process, the Sales Quote service task saves information about the sales quote to a database. Because no other work can be performed when the token reaches the end of a process, a none end event is used. After all process tokens reach the none end event, the process instance completes.



A process should have at least one end event. However there can be multiple end events based on how the process is designed.

Error End Event

The error end event lets you throw an error that can be caught by an error event subprocess activity or an error boundary event.



Use the error end event:

- When the end of a process is the result of an error condition
- To trigger an exception subprocess or a set of activities to handle a specific error.

You typically use the error end event with the error boundary event. The error boundary event is used to change the process flow based on a specific error. This flow usually ends with an error end event. See [Error Boundary Event](#).

Terminate End Event

The terminate end event stops all parallel executions of a Process Automation instance.

A Process Automation instance can generate multiple parallel tokens by using parallel gateways and non-interrupting timer events. If you use the terminate end event at a process level, then it results in the termination of all parallel token, and the Process Automation instance changes to a completed state. If you use the terminate end event at a subprocess level, then it results in the end of all parallel executions in the subprocess, and the control returns to the next activity in the parent flow.



Use the terminate end event:

- To handle a condition inside a parallel gateway that results in the termination of all tokens.
- To handle a condition after a non-interrupting timer event that results in the termination of all tokens.
- To terminate a concurrent flow that is not required.

The terminate end event immediately stops all work on a process and does not trigger error handling or other cleanup operations of the running process.

Error Boundary Event

Error boundary events are intermediate events that you can use to handle an error that occurs within a process flow.



Use the error boundary event in the following activities that can throw service invocation failure exceptions:

- A service task activity.
- A subprocess activity.

The error boundary event catches an error that can occur during the execution of an activity and interrupts the execution of the process flow, and reroutes the flow out of an activity.

Control Process Flow with Gateways

Gateways are activities that define the flow of your business process. Gateways determine the path a token takes through a business process. They define control points within your business process by splitting and merging paths. When possible, gateways are used for paths that are exceptions to, or deviate from, the default path of the business process.

- [Exclusive Gateway: Take Only One Path \(Data Condition\)](#)
- [Inclusive Gateway: Take One or More Paths](#)
- [Parallel Gateway: Take All Paths Simultaneously](#)

- [Create a Gateway](#)

Exclusive Gateway: Take Only One Path (Data Condition)

The exclusive gateway lets you split your business process into two or more paths. However, the business process only continues down one of the paths even if multiple outgoing sequence flows are present. Exclusive gateways can have conditional outgoing sequence flows and must have at least one default outgoing sequence flow.

You can define expressions that are used to determine if your process continues down a conditional sequence flow.

Evaluating the Flows of an Exclusive Gateway

When a process token reaches an exclusive gateway, the outgoing sequence flows are evaluated until one of them evaluates to *True*.



You can define the order in which the flows are evaluated by configuring the properties for the gateway.

- If only one outgoing sequence flow evaluates to *True*, then the process token continues down that outgoing sequence flow to the next activity.
- If more than one outgoing sequence flow evaluates to *True*, then the process token continues down the first sequence flow according to the order you defined in the gateway properties.
- If none of the outgoing sequence flows evaluates to *True*, then the process token moves down the default outgoing sequence flow. Therefore, you must define a default outgoing sequence flow for the exclusive gateway.

Unlike other gateways, the exclusive gateway doesn't require a corresponding merge to be explicitly defined in your process after splitting.

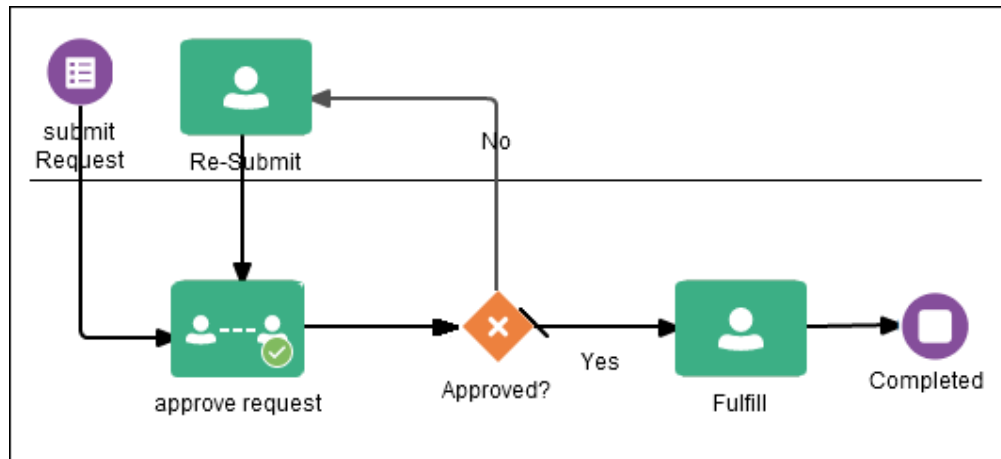
Note:

The exclusive gateway can also merge incoming sequence flows. However, there's no synchronization with other tokens that may be coming from other paths within the process flow. If other tokens arrive at an exclusive gateway merge, then they're passed through as is. If you're synchronizing tokens or performing evaluations on incoming sequence flows, then you should use a different type of gateway.

Example of an Exclusive Gateway

The following diagram shows an example of the exclusive gateway used within a Loan Approval business process. Here, the exclusive gateway is used to evaluate if the loan request is approved or if more information is required.

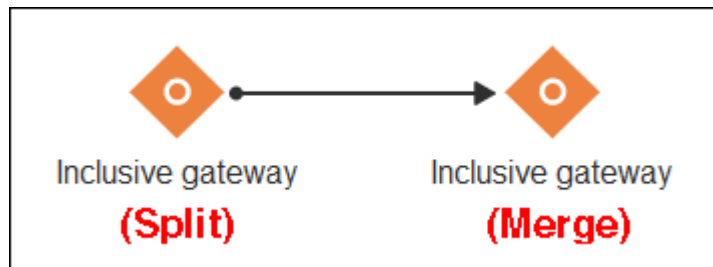
This evaluation is determined by the expression defined for the outgoing conditional sequence flow. If the expression evaluates to true, then the process flow proceeds down the No path. If it evaluates to false, then the process flow proceeds down the path of the default outgoing sequence flow.



Inclusive Gateway: Take One or More Paths

The inclusive gateway lets you split your business process into two or more paths. Unlike the exclusive gateway, however, a token may flow down one or more of these paths depending on how the outgoing conditional sequence flows are evaluated.

The inclusive gateway requires a split-merge pair. When you add an inclusive gateway to your business process, the split and merge flow objects are automatically created:



The merge portion of the gateway is required. However, you don't have to ensure that all paths out of the split return to the merge.

Although it's possible to have process paths that split at a gateway without merging through the gateway, it's not a good practice and not recommended.

Note:

If you delete the merge gateway from a business process, the corresponding split gateway is also deleted.

Splitting and Merging Inclusive Gateways

The inclusive gateway splits a business process similar to the exclusive gateway, but allows tokens to proceed down one OR more outgoing sequence flows. You can define any number of outgoing conditional sequence flows for an inclusive gateway split. You must define at least one default sequence flow.

When a token arrives at an inclusive gateway, the expressions of its conditional sequence flows are evaluated.

Next, a token is generated for each of the conditional sequence flows that evaluates to *True*. A token is generated for the default sequence flow only if none of the conditional sequence flows evaluate to *True*.

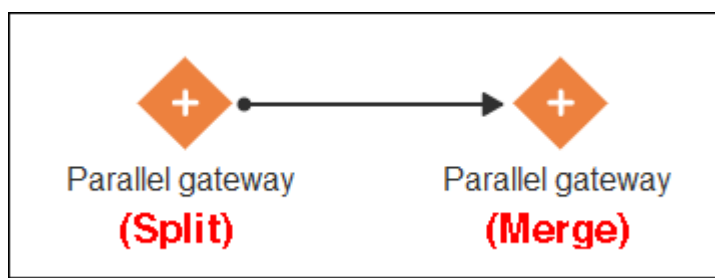
These tokens are joined at the merge of the inclusive gateway. When a token reaches the merge gateway, it waits until all the tokens generated by the split have reached the merge. After all the tokens have reached the merge of the inclusive gateway, the merge is complete, and the token continues to the next sequence flow after the gateway.

The token proceeds along all the sequence flows that evaluate to *True*. If none of the sequence flows evaluate to *True*, then the token passes along the default sequence flow.

Parallel Gateway: Take All Paths Simultaneously

The parallel gateway lets you split your business process into two or more paths when you want your process flow to follow all paths simultaneously. The parallel gateway is useful when your business process must perform multiple tasks in parallel.

The parallel gateway requires a split-merge pair. When you add a parallel gateway to your business process, the split and merge flow objects are automatically created:



The merge portion of the gateway is required. However, you don't have to ensure that all paths out of the split return to the merge.

Although it's possible to have process paths that split at a gateway without merging through the gateway, it's not a good practice and not recommended.

Note:

If you delete the merge gateway from a business process, the corresponding split gateway is also deleted.

Splitting and Merging Parallel Gateways

When a token arrives at a parallel gateway, the parallel gateway creates a token for each outgoing sequence flow. The split of the parallel gateway doesn't evaluate any of the outgoing sequence flows.

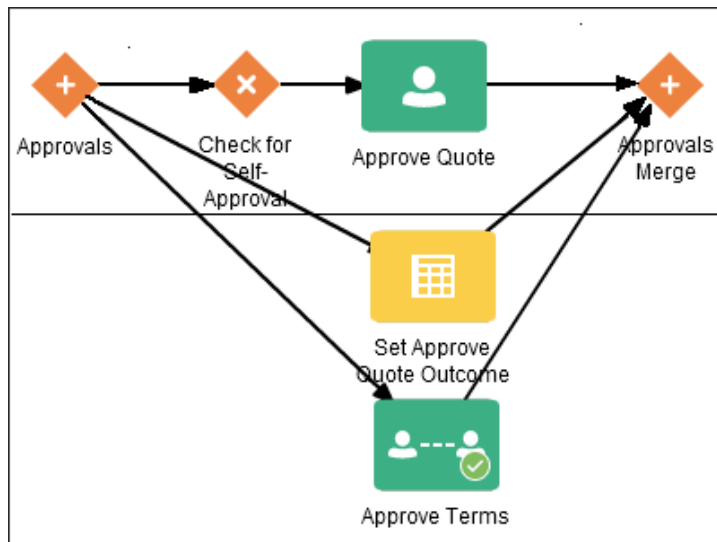
You can also use the parallel gateway to merge process paths split by the parallel gateway. The merge of the parallel gateway waits for a token to arrive from each of the incoming sequence flows. After all tokens arrive, only one token is passed to the outgoing sequence flow.

 **Note:**

You must design your business process so that a token arrives for each incoming sequence flow for the merging parallel gateway. If you don't, your business process can freeze if the merge is expecting tokens that don't arrive.

Example of a Parallel Gateway

In this example, a Sales Quote process uses a parallel gateway at the approval stage. The diagram shows how the parallel gateway is used to execute all process paths at the same time.





Create a Gateway


Use gateways to control how the process flows.



To create a gateway:


1. Open your process.
2. In the Activities palette, click **Gateways** to expand the list.

Here are the gateways you can add to a process:

Gateway Icon	Gateway Type	Description
	Exclusive	Only one of the paths out of the gateway is taken. The decision about which path the process should proceed along is based on data-specific conditions. For example, an exclusive gateway can specify different paths for the APPROVE and REJECT outcomes of an Approval human task.
	Inclusive	One or more paths out of the gateway can be taken, and the paths must converge later in the process. Use this type to perform several optional or conditional tasks at the same time.


Gateway Icon	Gateway Type	Description
	Parallel	All paths out of the gateway are taken, and the paths must converge later in the process. Use this type to perform several required tasks at the same time.

3. Drag and drop the gateway into the process flow.
4. Select the gateway. Click **Sequence Flow**  and drag the icon to create each path that exits from the gateway.
 - For exclusive or inclusive gateways, make sure the default path connects to the correct activity. Default sequence flows represent the path your business process takes out of these gateways when none of the data conditions evaluate to true. Default sequence flows are represented by an arrow with a slash mark on one end.
 - For inclusive or parallel gateways, you can drag paths to the side to separate them.
 - For parallel gateways, you must drop at least one activity onto each path before you can create another path.
5. For exclusive or inclusive gateways, create a data object. Data objects store and organize data the process uses.
 - a. Select the gateway and click the **Data**  icon. The Data pane opens.
 - b. Click **Create Data Objects**.
 - c. In the resulting pane, type a name for this data object. All names must begin with a lowercase letter.
 - d. Select the data type that matches the data flowing into the gateway from the previous task.

In most cases, the type is **boolean** (true or false) or **string** (text). For example, select **string** for the APPROVE and REJECT outcomes of an Approval human task.
 - e. Click **Create**.
 - f. Close the Data pane.
6. For exclusive or inclusive gateways, implement each non-default path. You don't need to implement the default path, but you should name it.
 - a. Select a non-default path, click **Edit**  on the path. The properties pane opens.
 - b. Select the **Conditional Flow** check box.
 - c. In the **Condition** field, type an expression that tests whether the Gateway data object has one of the choices.

For example, type `approvalOutcome == "APPROVE"`. Note the double equal sign.
 - d. Close the properties pane.
7. For exclusive or inclusive gateways, associate the output data with the gateway data object.

A parallel gateway doesn't need data association, but some of the activities on the paths might need it.

- a. Select the task that precedes and sends data to the gateway, and click **Data Association**  in the toolbar.

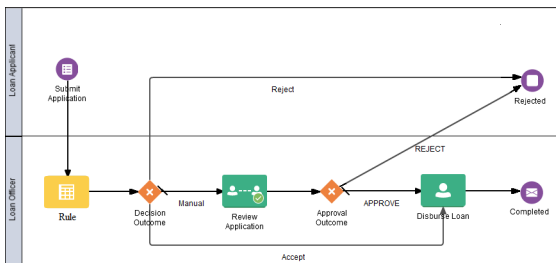
- b. Expand the Process and Data Objects nodes in the right pane.

- c. Drag and drop the gateway data object into the Outputs field.

You can associate inputs as well, which often come from a form. If the form is referenced in the start event or another process component, then the form data object appears in the right pane. Drag and drop each form field from the right pane to the corresponding Inputs field.

- d. Click **Apply**.


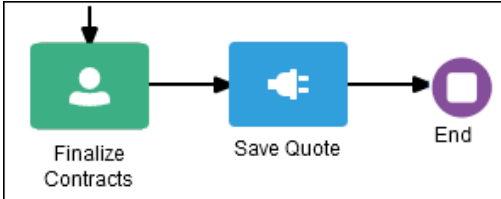




The Data Association editor closes.







Connect to Other Processes and Services

More likely than not, you'll need to define interactions across business processes within a process-oriented application. For example, your business process may need to invoke an external service, call another process from within the current process, or pass a message to a process outside of the current process. Use the following **System** activities to model communication between processes and services.

Processes can communicate with each other either synchronously or asynchronously. Processes interact *synchronously* when one process starts another and waits for a response. Processes interact *asynchronously* when one process starts another and both processes run independently.

Activity	Description
 Service	<p>Use the service activity to communicate with other processes and services. Add the service activity when you know that your business process must invoke an external service or process.</p> <p>For example, the following process uses the service activity to save the finalized sales quote to a database.</p>  <pre> graph LR A[Finalize Contracts] --> B[Save Quote] B --> C[End] </pre> <p>When the service activity invokes a service, the token waits at the service activity until a response is returned. After the response is received, the token continues to the next sequence flow in the process. Note that multiple service activities can also execute at any given point of time depending on the way the user has modeled the application.</p> <p>In case a service activity fails to execute (for example, due to REST endpoint timeout), you get an error message with details about the failure in the process Tracking page in runtime. You can manually retry the service activity from the process Tracking page by clicking the Retry button.</p>
 Notify	<p>Use the notify activity to generate and send notifications. The notify activity, which is similar to the service activity, uses a predefined service to perform notifications. You use expressions to determine which users receive the notifications generated by the notify activity.</p> <p>Currently, email is the only type of notification supported in Oracle Cloud Infrastructure Process Automation. This type of notification sends an email to the users you specify.</p> <p>See Send Notifications.</p>
 Call	<p>Use the call activity to call a reusable process from within the current process. The process being called becomes a child process of the calling process.</p> <p>When calling a reusable process, the call activity of the parent process waits until the child process completes before continuing.</p> <p>Data objects of the parent process aren't automatically available to the reusable process. Data objects must be passed to and from the child process using argument mapping of the call activity.</p> <p>See About Reusable Processes.</p>
 Decision	<p>Use the decision activity to add decision services to your process applications.</p> <p>You can use different types of decision logic, such as decision tables, simple expressions, and so on, to create an executable decision model, then expose it as services. These decision services can be incorporated into your business process to facilitate automated decision-making.</p> <p>Want to know more about working with decision models? See Model Decisions.</p>
 Upload Form	<p>Use the upload form activity to capture snapshot of a form (with its payload) during execution of a process instance and store the form in a database or service for archiving or auditing purposes.</p> <p>For example, you can capture snapshots of a form after a user submits the form and then again after a manager approves the form during a process execution and upload the form snapshots to an external database for auditing purpose.</p> <p>See Configure an Upload Form Activity.</p>

Activity	Description
 Subprocess	<p>Use the subprocess activity to model a subprocess within a business process. The subprocess becomes a child process of the parent process and is always a part of the parent process. A subprocess activity lets you hide the complex details of a part of a process to make the overall process more readable.</p> <p>See About Subprocesses.</p>
 Event Subprocess	<p>Use the event subprocess activity to catch system exceptions that occur in the runtime life cycle of a process and cause it to fail.</p> <p>See Handle Errors with Event Subprocesses.</p>
 Send	<p>Use the send activity to start another process from within the current process. The send activity doesn't wait for the process that it starts to complete. Once the called process is started, the main process continues to the next activity in the process flow. The called process and the main process runs asynchronously. See Use a Send Activity.</p> <p>A send activity can also be used to send a message to a receive activity in another process. See Use Send and Receive.</p>
 Receive	<p>Use the receive activity to receive messages from another process or external system. It waits for the message to arrive from a process or external system. Once the message is received, the receive activity completes and the process continues to the next activity in the flow.</p> <p>A receive activity is usually paired with a send activity to interact between asynchronous processes. See Use Send and Receive.</p> <p>Receive activities paired with send activities can also be used to set up <i>correlation</i> to communicate between asynchronous processes. See Example of Using Correlation with Send and Receive.</p>

About Reusable Processes


Reusable processes are processes that can be called from other business processes. They become child processes within the main parent process.

Reusable processes have the following characteristics:

- Must always begin with a none start event and end with a none end event.
- Can only be called by other business processes by using the *Call activity*.
- Since reusable processes are used as child processes within a main process, they can't contain elements that only make sense in an independent process, such as message start events, form start events or service activities.

To use a reusable process in a business process:

1. Create a reusable process.
Keep in mind the characteristics of a reusable process while creating the process.
2. Call the reusable process in a business process.
 - a. Open the main (parent) process where you want to use the reusable process as a child process.
 - b. Expand the **System** category in the Activities palette, and drag and drop a Call activity onto the process flow.
 - c. Double-click the call activity to open its properties pane.

Select the call activity, then click  and select **Open Properties** to open its properties pane.

- d. Under the **Implementation** section of the properties pane, select the reusable process from the **Process** drop-down field.

The reusable process that you created in step 1 should appear in the **Process** drop-down field.

- e. Close the properties pane to save the changes.
3. Define data associations for the call activity to pass the input data to the reusable (child) process and receive the output data from it.

See [Configure Data Association](#).

About Subprocesses

Subprocesses are embedded or in-line processes within the main business process. They are mainly used to manage complex business processes by grouping related activities together. Subprocesses not only improves the readability of complex processes in design-time but also improves the runtime user experience by enabling users to take collective actions on similar tasks during process execution.

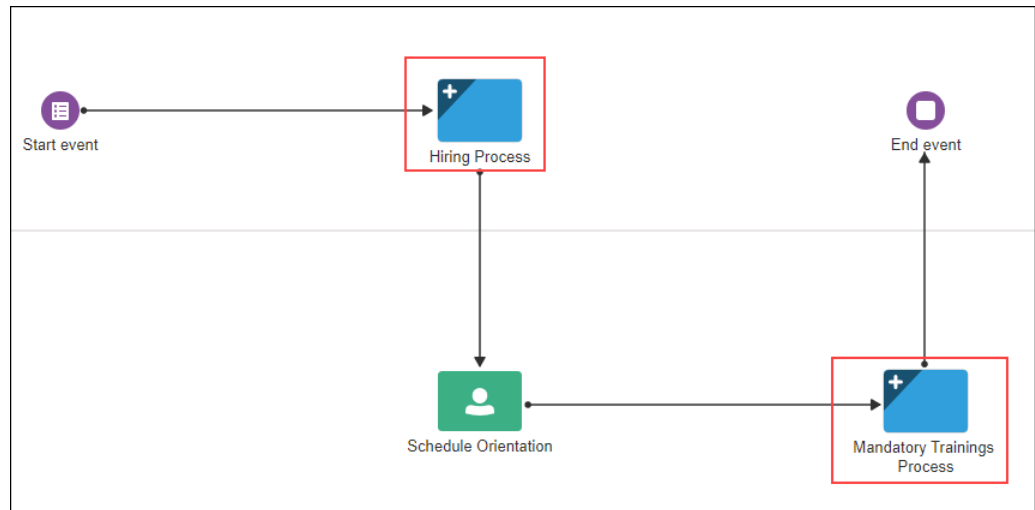
Note the following:

- Subprocesses are child processes within the main (parent) process. But unlike [reusable](#) processes, they can't be called from outside the parent process.
- Subprocesses must always begin with a none start event and end with a none end event.
- Subprocesses are useful in modeling a group of repeatable tasks within the main process as subprocess activities can be configured as multi-instance activities.
- Subprocesses can have sequence flows only within the subprocess. Sequence flows that connect to activities outside the subprocess are not supported.

Let us explore how we can use subprocesses to make a complex process such as an employee onboarding process into a more manageable and readable process in design time, and then explore the runtime experience for the same. The employee onboarding process can have several activities within it, such as activities related to the main hiring process, and activities for training newly hired employees. We'll use the subprocess activity to group related hiring and training activities into inline subprocesses thereby making the main process more readable.

1. Create a structured process.
2. Add subprocess activities onto the process flow and group related activities into the inline subprocesses.
 - a. In the Activities palette, expand **Systems** and drag and drop a subprocess activity onto the process flow.

For example, we have added two subprocess activities `Hiring Process` and `Mandatory Trainings Process` onto the process flow of the `Employee Onboarding Process`. We'll group activities related to hiring and training under these two subprocesses.

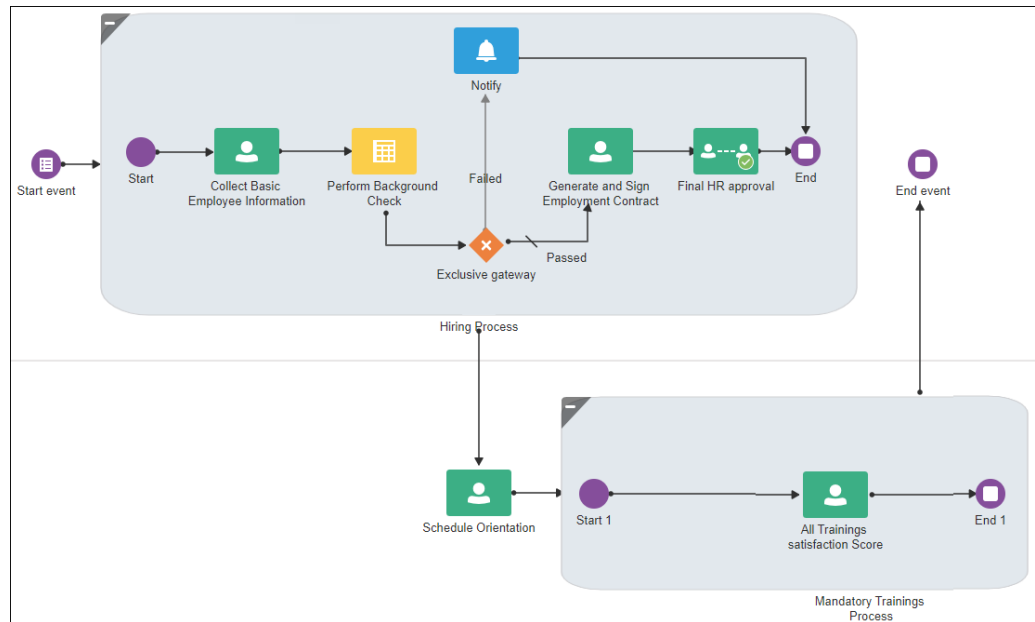


- b. Now expand the subprocess activity, and then drag and drop required activities from the Activities palette into the subprocess.

For example, in the subprocess Hiring Process, we have added the following activities:

- A human task activity `Collect Basic Employee Information` for the recruiter to gather information about the employee.
- A decision activity `Perform Background Check` to check if the information provided by the employee matches the job criteria.
- An exclusive gateway which when evaluates to `True` allows the process to move to the next human activity `Generate and Sign Employment Contract`, otherwise sends a notification with the `Notify` activity.
- A human task activity `Generate and Sign Employment Contract` for the employer to generate and submit the employment contract.
- A human task activity `Final HR Approval` of type `approve` for the HR manager to approve the employment contract for the employee.

You would need to configure the implementation details for the activities. For example, in case of the human task activities you would need to associate specific forms that the end user needs to complete and take actions (Submit or Approve).



The two inline subprocesses - Hiring Process and Mandatory Trainings Process become child processes of the main process (Employee Onboarding Process).

Depending on your modeling requirement, you can also use nested subprocess, that is, add a subprocess inside another subprocess.

3. Configure the subprocess activity as a multi-instance activity. See [Configure a Multi-instance Activity](#).

For example, we have configured the subprocess activity Hiring Process as a multi-instance activity.

Multi-Instance ✕

Activity: Hiring Process

Repetition Cycle

Do not repeat

Generate multiple instances

How do you want to manage your instances?

Sequential

Parallel

How do you want to create them?

Condition

Array

How many instances do you want to create? * Integer

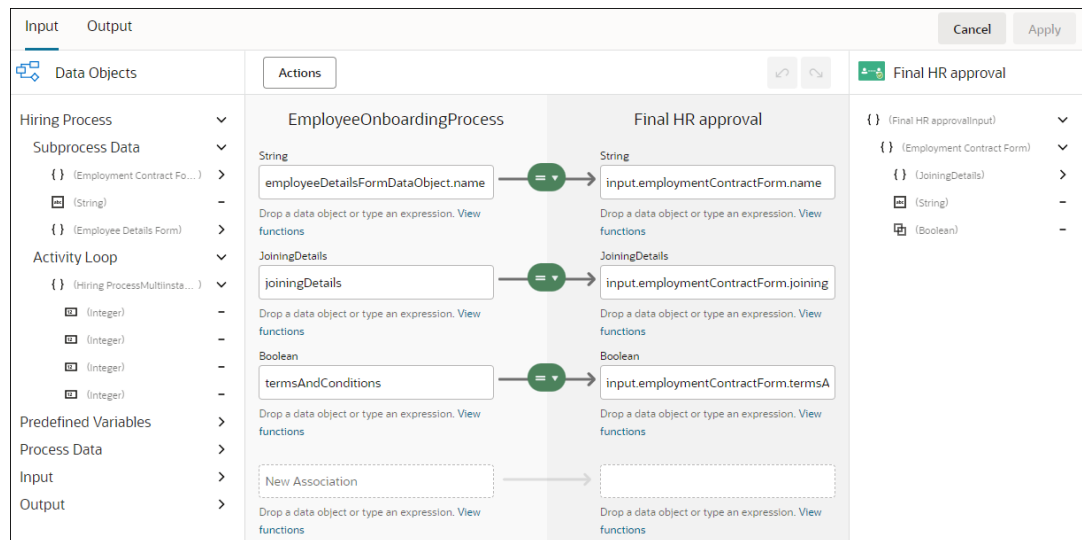
Enter an expression. [View functions](#)

Define a condition that terminates the execution of the activity instances

A marker with two parallel lines appear on the subprocess activity indicating that it has been configured as a multi-instance activity whose instances will execute in parallel in runtime.

4. Define data associations for the activities in a subprocess. See [Configure Data Association](#).

For example, the following figure shows the data associations for the Final HR approval activity.



Notice the following under the **Data Objects** section in the data association editor:

- The data objects for the subprocess are present under **Subprocess Data**.
 - The multi-instance predefined variables for the subprocess (as the subprocess is configured as a multi-instance activity) are present under **Activity Loop**.
5. Deploy and test the process application. Take actions and complete the various tasks in the parent process and the child subprocesses in runtime.

For the Employee Onboarding process, the sequence of the tasks will be as follows:

- The process instance is started by submitting the start form.
- The process moves to the next task, in this case the first activity (Collect Basic Employee Information) in the subprocess Hiring Process.
- After completing each task in the subprocess Hiring Process that is configured as a multi-instance activity with parallel execution, the process moves to the next activity Schedule Orientation in the process flow.
- Once the Schedule Orientation task is completed, the process moves to the subprocess Mandatory Trainings Process.
- After completing each task in the subprocess Mandatory Trainings Process, the process ends and the process instance state changes to Completed.

▼ Audit

Start
Aug 9, 2023 3:58 PM

Aug 9, 2023 3:58 PM

Aug 9, 2023 4:06 PM

Aug 9, 2023 4:07 PM

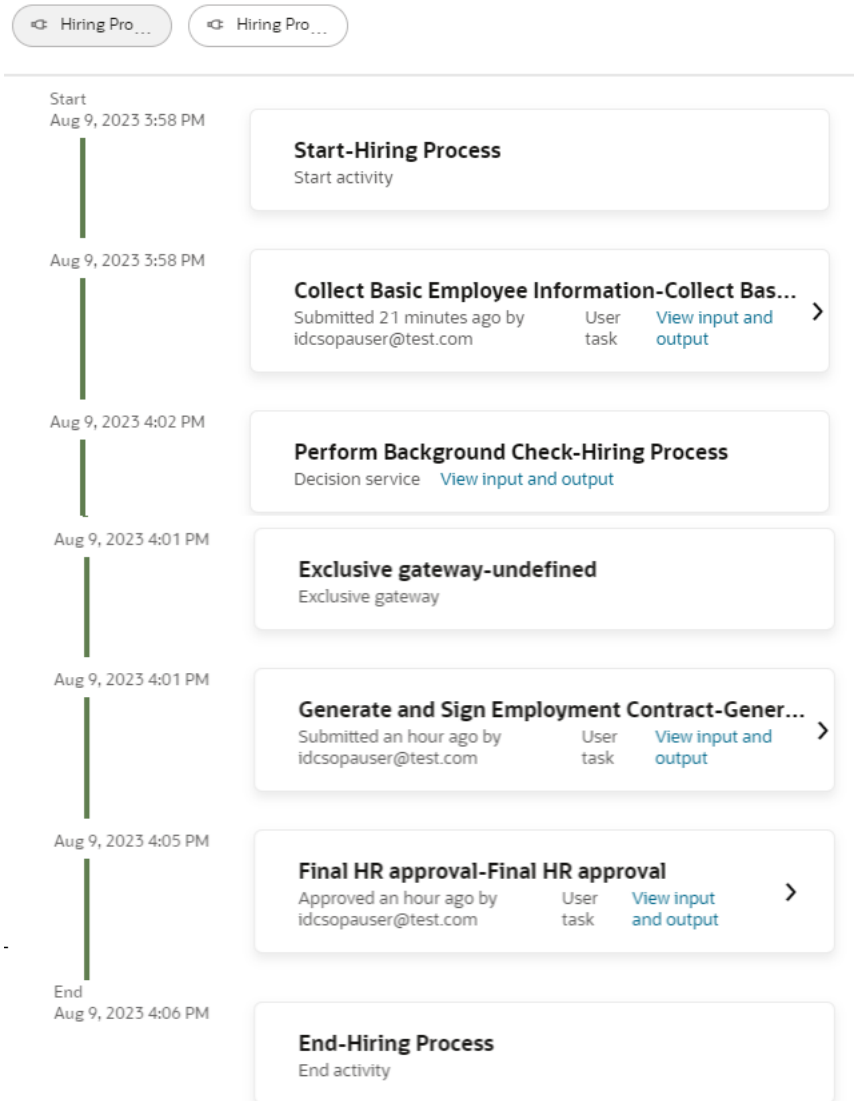
End
Aug 9, 2023 4:12 PM

- Start event-EmployeeOnboardingProcess**
Start activity [View input and output](#)
- Hiring Process-EmployeeOnboardingProcess**
Parallel 2 out of 2 doneMulti Instance Activity [View input and output](#)
- Schedule Orientation-Employee Onboarding Process**
Submitted 5 minutes ago by idcsopauser@test.comUser task [View input and output](#)
- Mandatory Trainings Process-EmployeeOnboardingProcess**
Sub process [View input and output](#)
- End event-EmployeeOnboardingProcess**
End activity

If required, in the Audit window you can drill down and check how each of the parallel instances of the multi-instance subprocess (Hiring Process) executed.

Hiring Process-EmployeeOnboardingProcess

Parallel 2 out of 2 doneMulti Instance Activity [View input and output](#)




Use a Send Activity

Use a send activity to asynchronously start another process. It helps in parallel initiation of multiple processes and can be used to communicate between independent processes.

Note the following:

- A send activity can trigger an independent process that starts with either a form start or message start event.
- A send activity is different from a call activity. Unlike a call activity, it doesn't call a [resuable](#) process but calls and triggers another independent process. Also, unlike a call activity which waits for the called process to complete, a send activity triggers the called process but doesn't wait for the called process to complete. That is, the main process with the send activity and the process triggered by the send activity can run asynchronously.

To use a send task activity in a business process:

1. On the Activities palette, expand **System** and drag and drop a send activity onto the process flow.
2. Double-click the send activity. Alternatively, select the activity, then click  and select **Open Properties** to open the activity's properties pane.
3. In the properties pane of the send task activity, enter a name for the activity in the **Name** field and an optional description in the **Description** field.
4. Under the **Implementation** section of the properties pane, enter the following details:
 - a. In the **Type** drop-down field, choose **Process call**.
 - b. In the **Process** drop-down field, select the process that you want to trigger with the send activity.
 - c. In the **Target node** drop-down field, choose the target node for the selected process.
Note that the target node can be one of the following in the called process:
 - message start event
 - form start event
5. Close the properties pane to save the implementation details in the send activity.
6. In the called process:
 - For a message start event, define input arguments.
 - For a form start event, define data associations to pass the input arguments to the next activity in the process flow. See [Configure Data Association](#).

After activation, in runtime Tracking page you will see that the main process and the process triggered by the send activity (called process) are running and completing asynchronously.

Send Notifications

Use the notify activity to generate and send email notifications to users that you specify. You can create custom email notification templates.

To send email notifications:

1. On the Activities palette, expand **System** and drag and drop a Notify activity onto the process flow.
2. Configure the properties for the notify activity.
 - a. Double-click the notify activity in the process flow, to open its properties pane.
 - b. In the properties pane of the activity, enter the name and description for the notify activity.
 - c. In the Send an e-mail section, enter the following information:

Field	Information to enter
To	Enter the email address of the recipients. To add multiple recipients, enter email addresses separated by commas.
Cc	Enter the email address of the recipients. To add multiple recipients, enter email addresses separated by commas.

Bcc	Enter the email address of the recipients. To add multiple recipients, enter email addresses separated by commas.
Reply	Enter the email address of the recipients. To add multiple recipients, enter email addresses separated by commas.
Subject	Enter the subject of the email notification.

 **Note:**

The combined character limit of **To**, **Cc**, **Bcc**, and **Reply** fields is 200. To add more recipient email addresses, you can add multiple notification activities.

- d. To test the process flow without waiting for all activities in the process to be implemented, select **Draft** checkbox.
3. To send a plain email notification message, select **Body** and enter the message in the **Body** field.
4. To send a customized email notification, select **Template**.
 - a. To use the default template, select Default Template from the drop down.
 - b. To create a custom template, in the Manage Email Templates section, click the plus icon.

Enter the name of the template and the html template for the notification message. You can also create a copy of the default template and modify it to your requirements.

Use Send and Receive

Send and receive activities are usually designed as pairs to communicate between two asynchronous processes.

The following steps outline a possible scenario for using send and receive activities to communicate between two asynchronous processes.

1. *Process A* starts.
2. The flow of *Process A* reaches the send activity.
3. The send activity starts *Process B*.
The send activity implementation specifies which process is started.
4. The flow of *Process A* continues to the next activity.
5. *Process B* runs.
6. Depending on the specific behavior of both processes, the following scenarios may occur:
 - a. If the flow of *Process A* reaches a receive activity paired with a send activity from *Process B*, the flow of *Process A* waits until a response is received.
After the response is received, the flow of *Process A* continues to the next activity.
 - b. If the flow of *Process B* reaches a send activity paired with a receive activity in *Process A*, *Process B* sends a response to *Process A*.
The flow of *Process B* continues to the next activity.

7. Both business processes continue running.

You can use subsequent send and receive pairs to define subsequent communication between the two processes.

Send and receive activities can be also used to set up correlation and communicate between asynchronous processes. See [Example of Using Correlation with Send and Receive](#).

Communicate Between Processes Using Correlation

Correlations enable business processes to communicate with each other based on the state of an instance.

When you define a correlation for a business process, you can identify an instance in another process based on the instance state and then send a message to that specific instance.

For example, you can use correlations to communicate between a sales process with the corresponding shipping and mailing processes. When the customer confirms an order, the sales process sends a message to the shipping and mailing processes using a correlation that defines that it uses the order ID to locate the instances in both processes.

How does correlation work?

Let's explore how correlation works with a simple use case.

Suppose we have a main process (Order) for ordering items. This process calls another process (Vendor) to check and confirm if the items requested in the order are available. Both these processes run asynchronously. While the Vendor process runs and checks for the availability of the item requested, the Order process continues to progress to next activities in the process flow – for example, it moves to an activity to get first level approval from the manager, then it moves to the next activity to get approval from finance. The Order process will continue to run until it reaches the activity in which it has to wait for the response message from the Vendor process. Once the response message is received from the Vendor process, the Order process continues to the subsequent activities and completes.


However, in real life scenario many orders may get initiated simultaneously and wait for response from the vendor for the availability of particular items. So there may be a lot of instances of the Order and Vendor processes running simultaneously. How does the engine determine and correctly match the response of a Vendor process instance to the corresponding request of the main Order process instance? For example: 50 instances of the Order and Vendor processes may have been started simultaneously. How does the engine match the response of the Vendor process instance 25 with the request of the Order process instance 25? This is done by using **correlation**. Using correlation we can match the incoming response message with the appropriate instance. The main component of correlation is the **correlation key**.

What are correlation keys?

Correlation keys are set of correlation properties used to associate an incoming message to a particular instance of a process. For example, in the use case of the Order and Vendor processes, the correlation key property may be the `Order Id`. When you define a correlation key, you provide a unique name to the key so as to identify the properties used in the correlation. The scope of the correlation key is the process, which means that after you define a correlation key you can use it for the correlation definition of an activity in that process.

Where do I define correlation keys?

You can define correlation keys either in the activity level (Receive activity) or process level.

- Define and create correlation key from the properties pane of a receive activity.
- Define and create correlation key in the process level by clicking the **Correlation keys**  icon from the sidebar menu of the process editor. The Correlation keys pane opens where you can define correlation keys for your process. Once defined you can use it in a receive activity in that process.

What are the main steps of implementing correlation?

1. **Define:** Define a correlation key to use. When you define a correlation key you provide a name to identify it. You create the key with one or more properties. For example, order id + item number. Properties are abstractions for representative attributes in the process, like the order ID, the customer name, or the social security number.
2. **Initialize:** Assign a value to the correlation key using literal values or expressions and initialize it from an activity. You can use the arguments and predefined variables of the activity. After you initialize a correlation, you can't change its value because the service engine uses this value to locate the instance.
3. **Correlate:** Specify the input arguments that should match the correlation key. The input arguments are those that you receive in the response message sent by the called process

Example of Using Correlation with Send and Receive

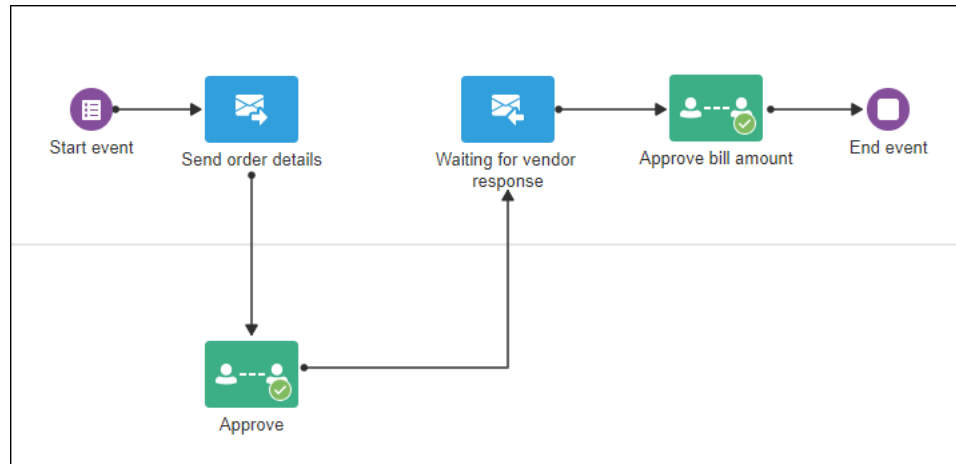
Correlations are typically used with send and receive activities to communicate between processes. Send and receive activities use correlations to make sure that incoming messages are properly correlated with the right process instances at run time.

Let's explore with a simple use case how we can use correlation with send and receive activities to design processes that communicate with each other. Suppose we configure two processes: Order process and Vendor process. The Order process initiates order for an item. It calls the Vendor process to check if the ordered item is available and gets a quote for the item. If the ordered item is available in the vendor's inventory, then the Vendor process sends a message to the Order process with a quote for the item. To ensure that the response sent by the called process is received by the correct calling process instance, correlation is set up.

Overview of the flow:

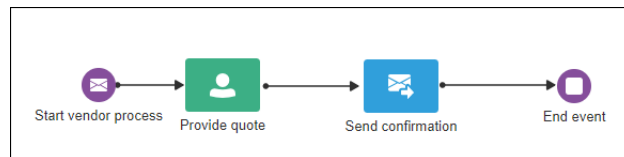
- When the Order process is started, the **correlation key** with the `order id` is initialized. The `order id` is a data object of the Order process. The engine will store the key for that instance of the Order process.
 - The Order process uses a *Send* activity to send the `order id` to the Vendor process.
 - The Vendor process uses a *Send* activity to callback with the `order id`.
 - A *Receive* activity in the Order process checks if the `order id` that was sent by the Vendor process matches the **correlation key** that was initialized when the Order process instance was started. If the `order id` matches the **correlation key** then that confirms that the response received by the Order process instance is from the corresponding Vendor process instance that was called.
1. Create the Order process with the following activities:
 - a form start event (Start event)
 - a send activity (Send order details)
 - an approve human task activity (Approve)
 - a receive activity (Waiting for vendor response)

- a second approve human task activity (Approve bill amount)
- an end event (End event)



2. Create the Vendor process with the following activities:

- a message start event (Start vendor process)
- a submit human task activity (Provide quote)
- a send activity (Send confirmation)
- an end event (End event)



3. Implement a start form for the form start event in the Order process.

The form should have at least two fields: **Order Id** and **Description**. When you add these two fields in the form, notice that two data objects `orderId` and `description` get listed under the **Data** section in the properties pane of the form.

You can implement the same form but with different presentations in the Approve and Approve bill amount activities.

4. Define input arguments for the message start event in the Vendor process.

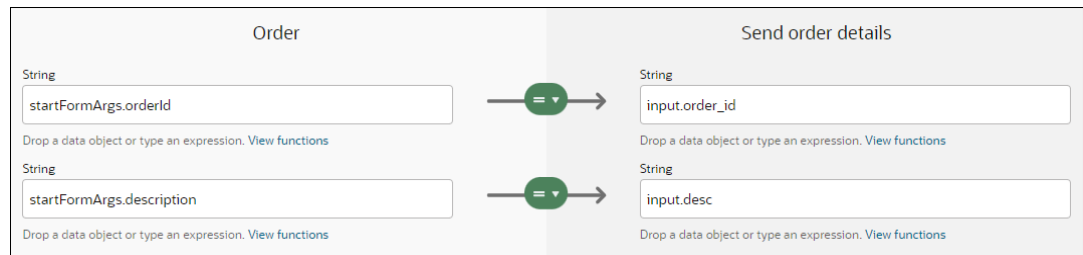
Open the properties pane of the message start event (Start vendor process) to specify the input arguments. Define two input arguments: `order_id` and `desc`.

Arguments Definition		+
<input type="text" value="order_id"/>	String	🗑️
<input type="text" value="desc"/>	String	🗑️

5. Use the send activity in the Order process to call the Vendor process.

- Open the properties pane of the send activity (Send order details).
- In the **Type** drop-down field, choose **Process call**.
- In the **Process** drop-down field, select the **Vendor** process.
- In the **Target node** drop-down field, choose the message start event (**Start vendor process**) of the Vendor process.

6. Define data associations in the Order process' send activity (Send order details) to map `orderId` and `description` to `order_id` and `desc`.
 - `orderId` and `description` are the start form's data objects.
 - `order_id` and `desc` are the message start events input arguments.



7. Define input arguments that you want to receive in the callback from the Vendor process.
 - a. Open the properties pane of the receive activity (Waiting for vendor response) in the Order process.
 - b. In the **Type** drop-down field, select **Define interface** as the implementation type.
 - c. In the **Operation name** field, define the operation: `waitingForVendorResponse`.
 - d. Under **Arguments Definition**, define the input arguments: `orderid` and `quote`.

Arguments Definition		+
<input type="text" value="orderid"/>	String	✕
<input type="text" value="quote"/>	Number	✕

8. Implement correlation in the receive activity of the Order process so that the response message from the Vendor process is mapped to the correct Order process instance.
 - a. Open the properties pane of the receive activity (Waiting for vendor response) in the Order process.
 - b. Define the correlation key.
 - i. Under the Correlation section, click the arrow **>** next to **Key**.
 - ii. In the resulting Create key pane, click the **New** tab.
 - iii. In the **Name** field, enter a name for the key (`orderKey`).
 - iv. Add a property `id` to the key. Enter `id` in the **Attributes** field.

The image shows two overlapping dialog boxes. The left dialog, titled 'Create key', has two tabs: 'Existing' and 'New'. Below the tabs, there is a 'Name' field containing 'orderKey' and an 'Attributes' section with a plus sign and a list containing 'id'. At the bottom are 'Apply' and 'Cancel' buttons. The right dialog, titled 'Properties', is for 'Waiting for vendor response'. It has a 'Name' field with 'Waiting for vendor response', an 'Identifier' field with a long alphanumeric string, and a 'Description' field with the placeholder 'Enter a description'. Below that is a 'Draft' checkbox. The 'Implementation' section has a 'Type' dropdown set to 'Define interface' and an 'Operation name' field with 'waitingForVendorResponse'. The 'Arguments Definition' section has two rows: 'orderid' with a 'String' dropdown and 'quote' with a 'Number' dropdown. At the bottom is a 'Correlation' section with a radio button for 'Key' and a right-pointing arrow.

If required, you can add more than one property to the key. Use + to add more properties.

v. Click **Apply**.

A green check mark next to **Key** indicates that the correlation key has been defined.

- c. Correlate or match the correlation key to the input argument that will be received from the Vendor process in its response message. In our example, we'll match `id` (the attribute of the correlation key) to the input argument attribute `orderid`.
- Click the arrow > next to **Correlate**.
 - In the resulting Correlate key attributes pane, switch to literal mode in the drop-down field, and choose **orderid** from the list.

Correlate key attributes ×

The correlation key identifies which process instance should get the message.

Key
orderKey

Match attributes with message arguments

id* String
orderid

Properties ×
Activity: Waiting for vendor response

Description
Enter a description

Draft

Implementation

Type
Define interface

Operation name
waitingForVendorResponse

Arguments Definition +

orderid	String	✕
quote	Number	✕

Correlation

- Key orderKey >
- Correlate >
- Initialize >

Apply Cancel

iii. Click **Apply**.

A green check mark next to **Correlate** and `id: orderid` indicates that the key has been correlated.

d. Initialize the correlation key.

i. Click the arrow **>** next to **Initialize**.

ii. In the resulting Initialize pane, choose **Start event** in the **Initialize in** drop-down field.

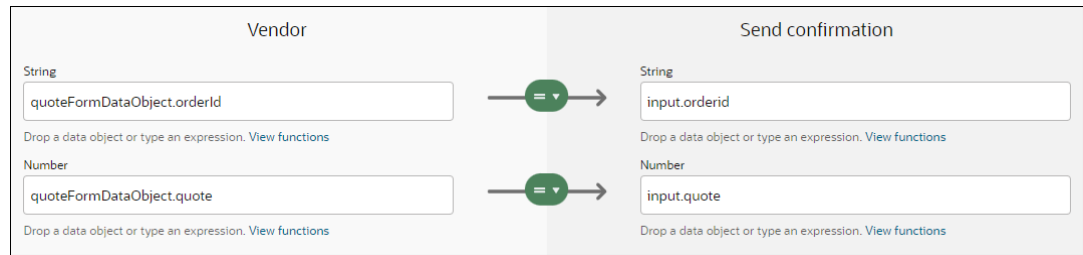
iii. Under **Map attributes to arguments**, map the correlation key property `id` to the start form's data object `orderId`. Use the expression mode to form the expression `startFormArgs.orderId` in the given field.

The screenshot shows two side-by-side panes in a software interface. The left pane is titled 'Initialize' and contains a dropdown menu labeled 'Initialise in' with 'Start event' selected. Below it is a section 'Map attributes to arguments' with a field 'id *' of type 'String' containing the expression 'startFormArgs.orderId'. The right pane is titled 'Properties' and shows 'Activity: Waiting for vendor response'. It has a 'Description' field, a 'Draft' checkbox, an 'Implementation' section with a 'Type' dropdown set to 'Define interface' and an 'Operation name' field containing 'waitingForVendorResponse'. Below that is an 'Arguments Definition' table with two rows: 'orderid' (String) and 'quote' (Number). The 'Correlation' section has three items: 'Key' (checked, orderKey), 'Correlate' (checked, id: orderid), and 'Initialize' (unchecked).

iv. Click **Apply**.

When we initialize the key in the start activity, we assign a value to the correlation key. The value comes from the form's data object `orderId`.

9. Implement a form for the submit human task activity (Provide quote) in the Vendor process. This form should have at least the following fields: **Order Id** and **Quote amount**.
10. Define data association for the message start activity (Start vendor process) in the Vendor process, so that `order_id` is mapped to the submit form's data object `orderId`. This ensures that the order id value is passed into the submit activity (Provide quote) of the Vendor process.
11. Use the send activity in the Vendor process to callback the Order process.
 - a. Open the properties pane of the send activity (Send confirmation).
 - b. In the **Type** drop-down field, choose **Process call**.
 - c. In the **Process** drop-down field, select the **Order** process.
 - d. In the **Target node** drop-down field, choose the receive activity (**Waiting for vendor response**).
12. Define data associations in the send activity (Send confirmation) of the Vendor process to map the `orderId` and `quoteamount` from the Vendor process to the input arguments `orderid` and `quote`. Optionally, you can also map the process input `order_id` to `orderid` instead of the form data object `orderId`.



This ensures that the correlation key value matches the value in the `orderid` input argument and the quote amount for the requested item is passed onto the correct Order process instance.

Now that the processes are set up, you can activate and test the process application in Workspace. Track the running of the Order and Vendor process instances from the Tracking page, and check if the response message from the Vendor process instance is received by the corresponding Order process instance. See [Track Processes](#).

Handle Errors with Event Subprocesses

Use event subprocesses to handle exceptions that occur in the runtime life cycle of a process and cause it to fail.

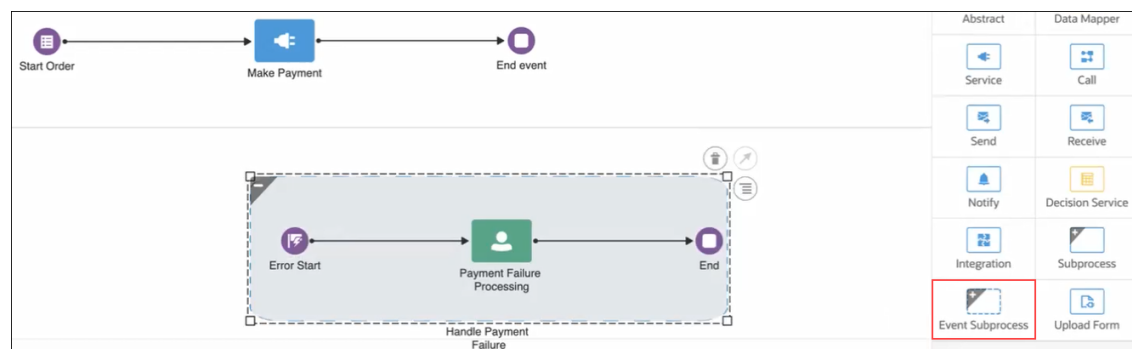
Using an event subprocess activity, you can catch all or specific system exceptions and recover the original process flow or supply an alternative, exception-handling flow.

You can define more than one event subprocess to handle different exceptions within a single process.

Note the following:

- It is not possible to define multiple event subprocesses catching the same error in the same scope.
- It is not possible to have more than one event subprocesses catching all errors in the same scope.
- Subprocesses can have their own event subprocesses to handle errors. An error that occurs in the subprocess can be caught by the subprocess level event subprocess instead of the process level subprocess.

In the example below, an event subprocess `Handle Payment Failure` is used to handle any payment failure errors in the main process. When an error occurs in the `Make Payment` service task, it gets canceled and the `Handle Payment Failure` event subprocess is triggered. The process then continues to run in the event subprocess flow and completes.



- [Add an Event Subprocess](#)

- [Configure a Start Error Event](#)

Add an Event Subprocess

1. In the Activities palette, expand **Systems**.
2. Drag the **Event Subprocess** activity onto the process editor canvas, and drop it at the required position. Note that you should add the activity outside your process flow as event subprocesses don't support incoming or outgoing flows.
3. Expand the event subprocess activity. You'll see a default error-handling flow with a Start Error event and an End event. You can add other activities, such as human tasks, to the flow according to your requirements.

You cannot configure properties or define data association for the event subprocess activity as a whole; however, you can configure the Start Error event to catch and handle exceptions.

Configure a Start Error Event

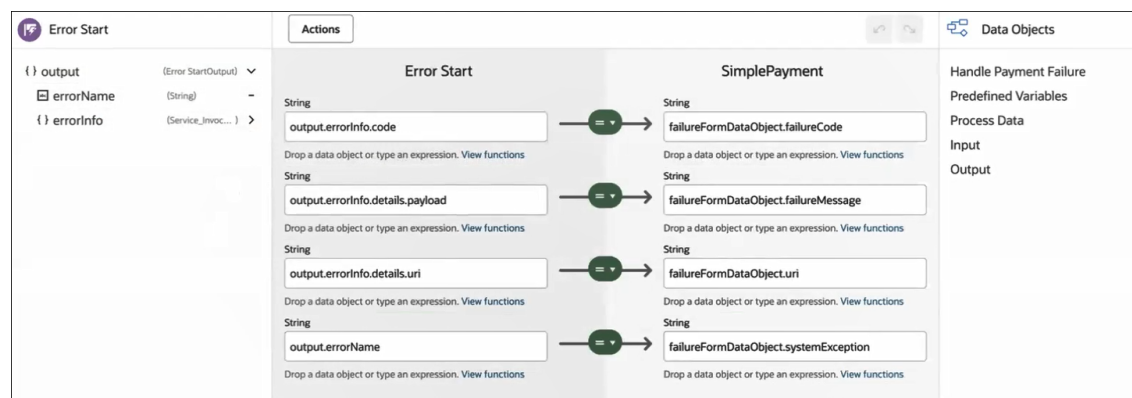
1. Open the properties pane of the Start Error event.
2. In the **Exception** drop-down field, select **Service Invocation Failure**.

Setting this property will catch service invocation failures in the following activities:

- service task
- integration
- form upload

3. Optionally, you can choose to catch all system exceptions in the process. Select **Catch all system exceptions**.
4. Close the properties pane to save the changes.

In order to give users data to analyze the error, some data objects such as error code, error payload, uri and error name are provided out of the box in the data association editor of the Start Error event activity. In runtime, users are able to analyze the cause of the error with this information.





Configure a Multi-instance Activity

You can configure an activity as a multi-instance activity, so that you can run multiple unique instances of the same activity sequentially or in parallel.

The following system activities can be configured as multi-instance activities:

- Call
 - Service
 - Subprocess
 - Integration
1. Open a structured process where you want to add a repeatable or multi-instance activity.
 2. In the process editor, expand **System** in the Activity palette, and drag and drop an activity (Call, Service, Subprocess, Integration) that can be configured as a multi instance activity onto the process flow.

Notice how a multi-instance icon  appears on the right side of the process editor. The icon is displayed only for a call, service, subprocess or integration activity, indicating that you can configure such an activity as multi-instance.

3. If required, complete the implementation details for the activity.
 - a. Double click the activity, or select the activity and click , and then select **Open Properties** to open the activity's properties pane.
 - b. In the activity's properties pane, under **Implementation**:
 - For a call activity, select the process you want to call from the **Process** field.
 - For a service or integration activity, enter the connector details by selecting the service in the **Service** field, the resource in the **Resource** field, and an operation in the **Operation** field.

4. Click the  **Multi-instance** icon.

The Multi-instance properties pane appears. By default, **Do not repeat** is selected under **Repetition Cycle**.

5. Select the **Generate multiple instance** button. A number of fields to specify and manage your multi-instance activity appears.
 - a. Define how you want to manage your instance. If you want the instances to occur sequentially, select **Sequential**. If you want to see all the instances at the same time, select **Parallel**.

Notice that the icon for the multi-instance activity, in the process flow diagram on canvas, changes according to your selection - sequential or parallel.

- b. Specify how you want to create the instances – with a condition or with an array. Depending on the selection, related fields appear.

For example, if you select **Array**, a field appears to enter the array. You can use the in-built expression builder to specify the array. If you select **Condition**, a field appears to specify the number of instances to be created.

See [Work with Expressions](#).

- c. If you want to terminate the activity execution with a condition, select the checkbox for it, and specify the condition in the **Condition** field using the in-built expression editor. Otherwise, de-select the checkbox.
 - d. Close the multi-instance properties pane to save the configurations.
6. Configure [data association](#) with the variables created for the multi-instance activity.

In the Data Association window, notice a new container **Activity Loop** under Data Objects. Expand the **Activity Loop** container to see all the multi-instance related variables and use them for data association.

Predefined Variables

The following predefined variables are available for multi-instance activities.

Name	Data Type	Description	Available for MI Array?	Available for MI Condition?
loopCounter	integer	The sequence number of the iteration.	Yes	Yes
nrOfInstances	integer	The total number of instances.	Yes	Yes
nrOfActiveInstances	integer	The total number of active instances.	Yes	Yes
nrOfCompletedInstances	integer	The total number of completed instances.	Yes	Yes
inputDataItem	Same as input array element type.	Holds a reference to the input array content on the current iteration. For example, if the <code>loopCounter</code> for the iteration is 2, <code>inputDataItem</code> will contain the second element of the input array.	Yes	No
outputDataItem	Same as output array element type.	Holds a reference to the output array content on the current iteration. For example, if the <code>loopCounter</code> for the iteration is 2, <code>outputDataItem</code> will contain the second element of the output array.	Yes	No

Configure an Upload Form Activity

You can configure an upload form activity in your process to generate a form as either a PDF or image (PNG) and upload the base64 string representation of the form to an external service for archiving or auditing purposes. The base64 string of the form content is uploaded via a POST or PUT REST API call.

Note the following limitations:

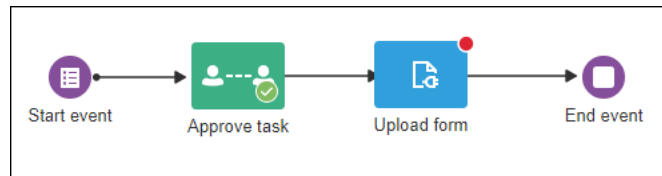
- Localization is not supported for forms uploaded to external services with the upload form activity.
- The maximum size supported for forms uploaded to external services with the upload form activity is 2 MB.
- If a form contains controls that trigger events invoking connectors and actions that load data in other controls based on the connector response, these dynamic controls will not be displayed in the uploaded form.
- If a form contains a file upload (in a file manager) control then the control will not be displayed in the uploaded form.

To configure an upload form activity:


1. Create a process that includes a form and a upload form system activity.

To add a upload form activity to the process, expand **System** on the Activities palette and drag and drop a Upload Form activity onto the process flow.

In the example process, we have included an upload form activity after an approve task to get a snapshot of the approval form.



2. Add and configure a REST connector that will upload the form to an external service. See [Work with Connectors](#).
3. Configure the properties for the upload form activity.
 - a. Double-click the upload form activity in the process flow, to open its properties pane.

Alternatively, select the upload form activity, then click  and select **Open Properties** to open its properties pane.

- b. In the properties pane of the activity, enter the following information :

Field	Information to Enter
UI	Select the form to upload.
Presentation	Select the presentation for the form.
Form data	Select the form's data object that contains its payload.
File format	Select the file format of the form. The available options are: PDF and PNG.
Service	Choose the connector that you configured in step 2.
Resource	Choose the connector resource.
Operation	Choose the operation based on the resource that you selected for the connector. Note that only POST and PUT operations are supported.
File content expression	Enter the path of the form's payload that will be overwritten by the base64 string value in the request body JSON. Note that only value of string type is supported in the expression.

 **Note:**

If the same expression exists in a upload file activity's data association as well as in the file content expression field (in the properties pane), then the execution of the file content expression will be given precedence over data association in runtime.

Properties X
Activity: Upload form

UI *
MyForm

Presentation
MyForm

Form data *
myFormArgs

File format *
PDF

Connector details

Service *
FaCon

Resource *
Resource

Operation *
postResource

File content expression * ⓘ String
input.body.FileContents
Enter an expression.

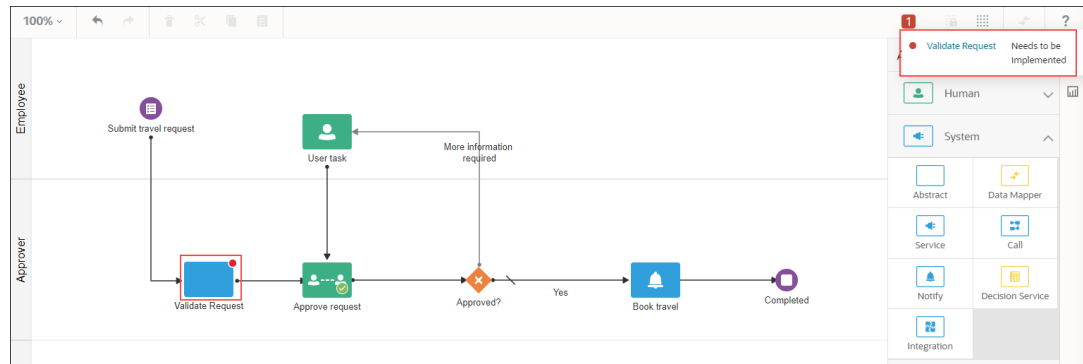
4. Deploy and test the process application, submitting and approving the form as needed. Check if the form data has been uploaded to the external service.

Use Abstract Activity

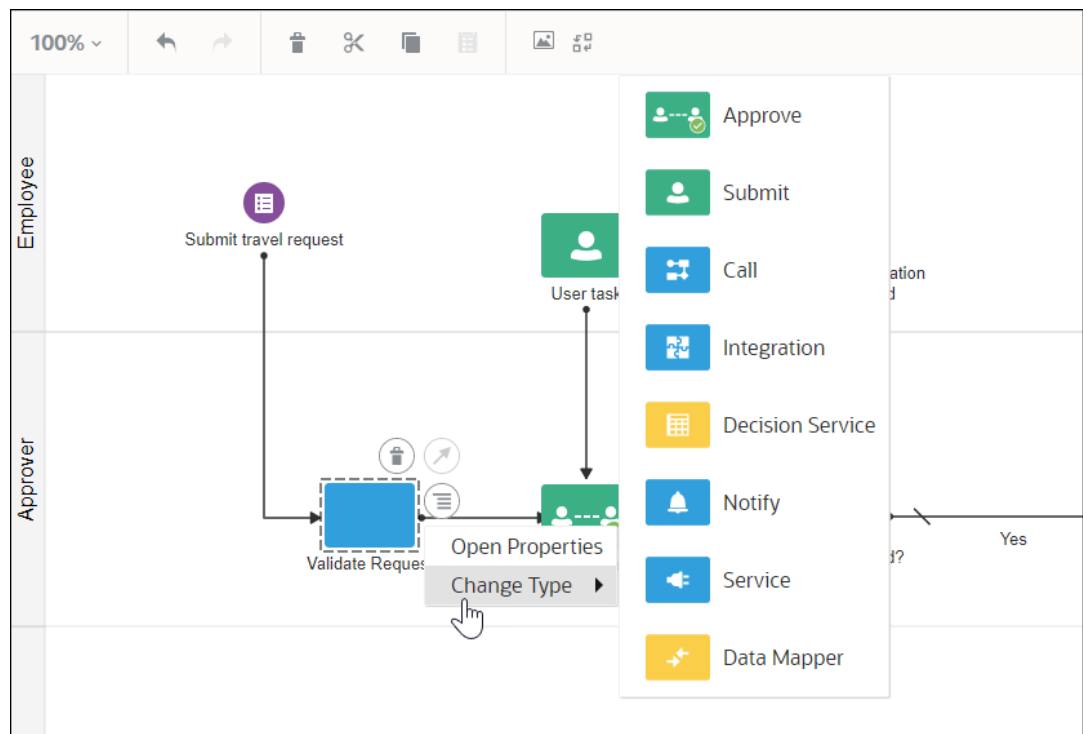
An abstract activity is a modeling placeholder for another activity in a structured process. It is useful when you want to design a structured process but aren't sure about the implementation details for some activity in the process flow.

Let us explore how you can use an abstract activity in a structured process. Suppose you have designed a travel request process. You want that the travel request should be validated before it is sent for approval, but aren't sure at this point if you would want to use a human task activity, service activity, or decision activity for validating. In this case, you can use an abstract activity as a placeholder activity to complete designing the process flow without worrying about the implementation details for the activity.

1. On the Activities palette, expand **System** and drag and drop an Abstract activity onto the process flow.
2. Rename the abstract activity as `Validate Request`. Notice the following:
 - There are no implementation details defined for the activity hence the activity will display an error.
 - You'll get a process level validation error also for the activity which will not allow activation unless implemented.



- Once you're sure about the kind of activity you want to use, you can change the abstract activity type. For example, you decide to use a service activity to validate the travel request. Right-click the abstract activity and select **Change Type**, then choose **Service**.



The abstract activity changes to a service activity.

You decided to use a service activity but still don't have connector details for implementing the service activity. As there are no implementation details for the service task activity, you'll get validation error for the activity and won't be able to activate and test the process in runtime.


If you want to test a process that has an abstract activity or an unimplemented activity, you need to use the draft property for the activity. See [About Draft Property](#).

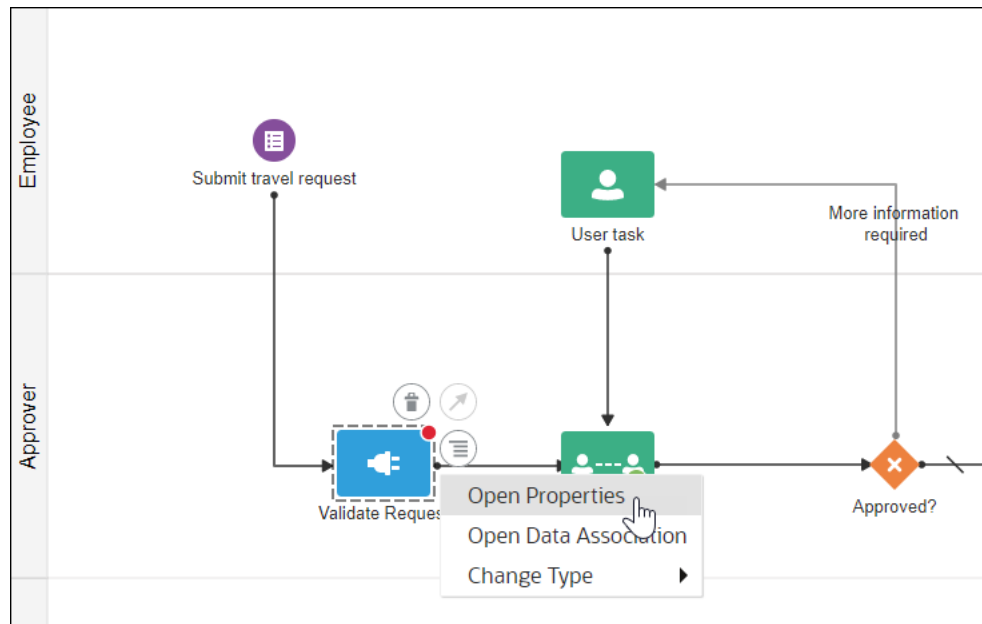
About Draft Property

Draft property enables you to test structured processes that are designed with abstract activities or with activities without implementation details. Draft properties are useful when you want to test the process flow without waiting for all activities in the process to be implemented.

Let us explore how we can use the draft property to activate and test a structured processes with unimplemented activities.

Suppose you want to activate and test a travel request process that has an unimplemented service activity.

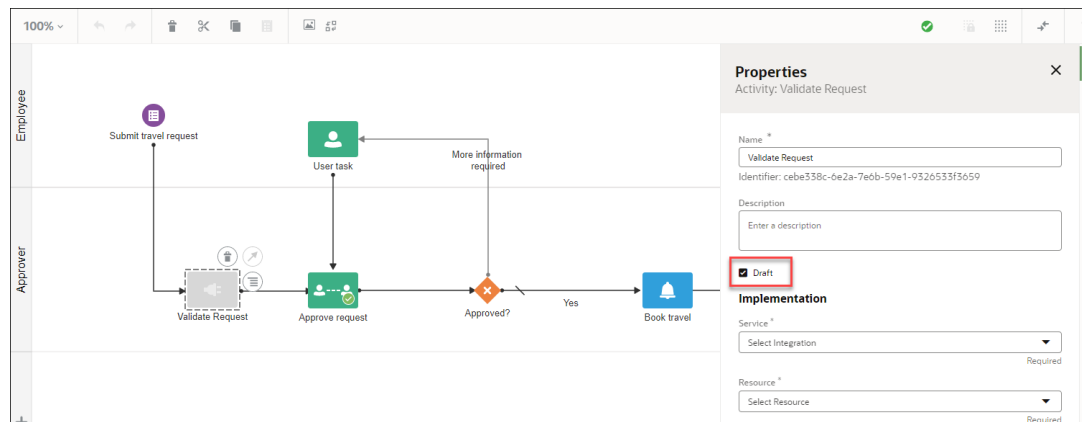
1. Double-click the service activity, click  and select **Open Properties**.



2. In the properties pane for the activity, select the **Draft** check box.

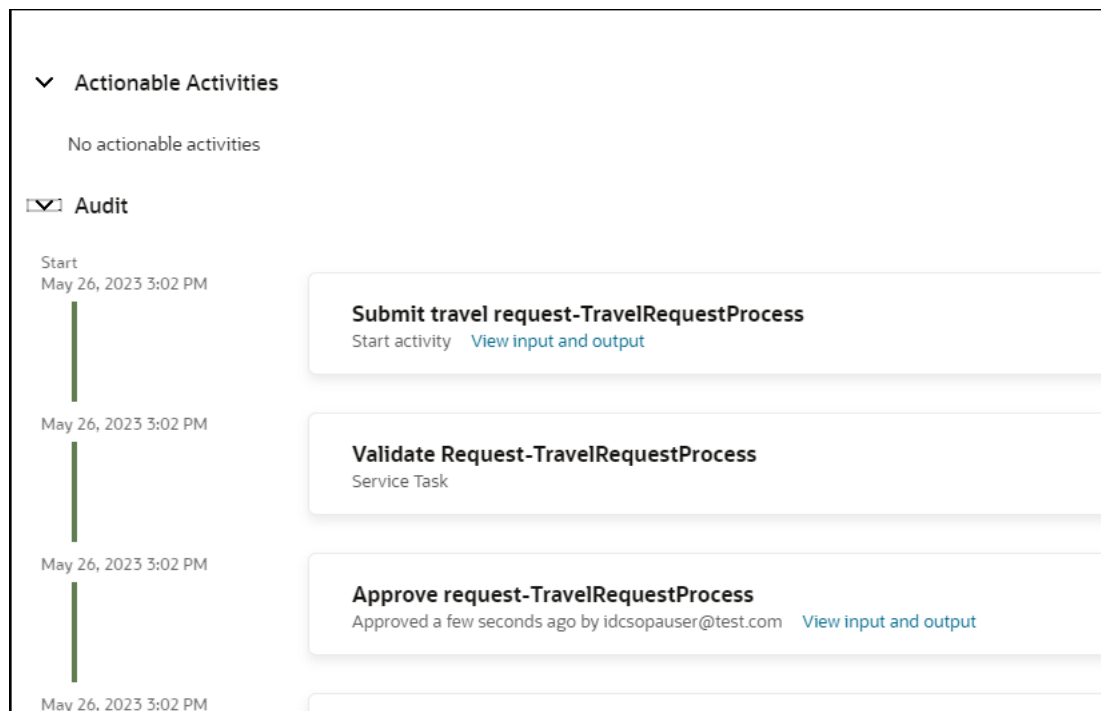
Notice the following:

- The service activity is greyed out.
- The validation error disappears and a green tick mark indicates that the process is valid.



3. Activate and test the process application.
With the draft property selected, validation for the unimplemented service activity will be skipped. You can activate the process application and test it in runtime.


The activated process application will be available in Workspace Start Request page. [Start the application](#) and [complete the assigned tasks](#). If you [view the audit](#) for the completed process, you will find that the unimplemented service task (that had a draft property) was automatically marked as completed and the rest of the process completed successfully.



Use Data Mapper Activity

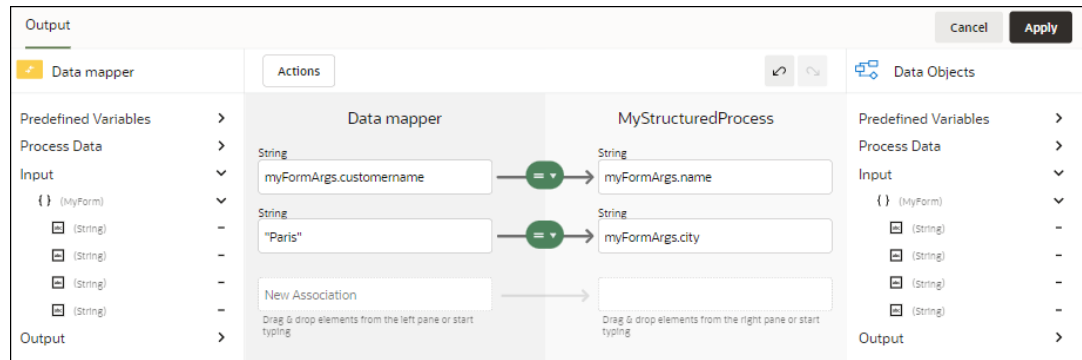
The data mapper activity is used to assign the value of a process data object, predefined variable, or literal to another data object or variable within the process. It can be used in scenarios where it is not possible to assign values to certain data attributes through other activities in your process or to override the values assigned to attributes through the defined process flow.

To define data mappings through the Data Mapper:

1. On the Activities palette, expand **System** and drag and drop a Data Mapper activity onto the process flow.
2. Select the Data Mapper activity, click , and select **Open Data Association**.

The Data Association editor opens. Notice that the editor contains only the **Output** tab because the Data Mapper is just a mapper that serves to associate one process data object or variable with another. Unlike other activities (for example, human tasks), it does not have data objects of its own that require mapping with the process data on the input side.

3. In the editor, drag and drop elements from the left and right panes onto the center pane to map associations. Both the panes display all the available data objects or variables within the process. The object you add on the left serves as the source and the object on the right denotes the target of the mapping. The following figure shows a sample data mapping:



Here, the value of a form data object, *customername* is assigned to another form data object *name*, and a literal value of “Paris” is assigned to a form data object, *city*.

Note:

If you chose a predefined variable as the target, ensure that it isn't read only.

4. Optionally, use the Expression Editor to build an input expression using standard functions and operators.
5. Click **Apply** to save the data mappings.

To define data associations through other activities, such as human task or service activities, see [Configure Data Association](#).


Work with Sequence Flows

Sequence flows define the order or sequence that work is performed within a business process. Sequence flows connect the activities within your business process and determine the path a process token follows through your business process.

Incoming sequence flows are the sequence flows that lead into an activity. **Outgoing sequence flows** are the sequence flows that determine the process path out of an activity.

Most activities contain both incoming and outgoing sequence flows. Exceptions to this are start and end events. Start events can only contain outgoing sequence flows. End events can only contain incoming sequence flows.

To add sequence flows to your business process:

1. In the process editor, click the activity from where you want to create the outgoing sequence flow.
2. Click  and keep the mouse depressed.
This option only appears for activities that don't already have outgoing sequence flows.
3. Move the cursor to the activity you want to connect to, and then release the click.

Unconditional Sequence Flows

Unconditional sequence flows represent the typical path between two activities. Default sequence flows are displayed as arrows:



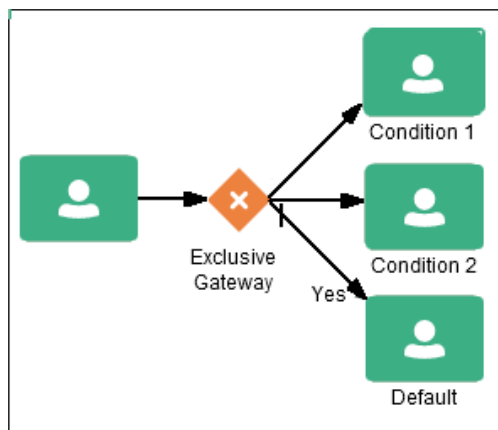
Most activities can contain only one default outgoing sequence flow. Only parallel gateways can contain multiple unconditional sequence flows, which represent the parallel paths of your business process.

Exclusive and inclusive gateways can't have unconditional outgoing sequence flows. These gateways use conditional and default sequence flows to determine the flow of your business process.

Conditional Sequence Flows

Use conditional sequence flows to control the flow of a business process based on certain conditions. Like unconditional sequence flows, conditional sequence flows are represented by an arrow.

Here's an example of two outgoing conditional sequence flows and a default sequence flow (which includes a slash mark at the beginning of its arrow):



Not all activities can use outgoing conditional sequence flows. Only the following types of gateways can have outgoing conditional sequence flows:

- Exclusive gateways
- Inclusive gateways (split)

You use expressions to define the conditions used within a conditional sequence flow.

Default Sequence Flows

Like conditional sequence flows, default sequence flows are used as outgoing sequence flows to exclusive, event-based, inclusive, and conditional gateways.

Default sequence flows represent the path your business process takes out of these gateways when none of the conditions evaluate to true. Default sequence flows are represented by an arrow with a slash mark on one end.

4

Design Dynamic Processes

A key part of your application is its business process or processes. When developing a dynamic process, first determine the activities and stages that may occur. You can then build out related elements, such as web forms for human interaction, milestones, and the data flow between elements.



Note:

In Oracle Cloud Infrastructure Process Automation, you can create two types of processes: **dynamic** (described below) and **structured**, described in [Design Structured Processes](#).

Topics:

- [How do dynamic processes work in process applications?](#)
- [Model a Dynamic Process in Design Time](#)
- [Create a Dynamic Process](#)
- [Create Activities](#)
- [Create Stages](#)
- [Define Stage Properties](#)
- [Define Activity Properties](#)
- [Define Conditions for Stage or Activity](#)
- [Define Process Input and Output](#)
- [Define Process Data Objects](#)
- [Define Process Completion and Termination](#)
- [Work with Inline Validations](#)

How do dynamic processes work in process applications?

Use the dynamic process editor as a canvas to define, configure, and activate dynamic processes for process applications.

Automate unpredictable processes that require expert knowledge or changing circumstances

Many business processes don't follow a structured or sequential path. They might have many possible activities that experts could act on as needed, given the situation. Which tests should a doctor administer? Based on test results, does the patient need surgery? Which documents must be approved before a treatment stage can begin?

Want to learn how to design dynamic processes? See [Model a Dynamic Process in Design Time](#).

Wondering how knowledge workers use activated dynamic processes? See [Work on Processes](#).

Combine dynamic and structured processes in process applications

Combine dynamic and structured processes in a process application to take advantage of their strengths. For example, orchestrate an overall unpredictable process with a dynamic process that accommodates a wide variety of activities, but use structured processes for activities that follow a structured path.

Model a Dynamic Process in Design Time

Within an application, you can create dynamic processes to model flexible and data-intensive business scenarios. For a dynamic process, knowledge workers or process participants can define the process flow in runtime by making real-time decisions and carrying out relevant tasks.

Use the following approach as a general guideline to model a dynamic process; however, you may choose to complete some of these steps in any order. Iteratively refine the dynamic process and its elements as you model them to suit your requirements.

1. To begin with, create a dynamic process with the required details. See [Create a Dynamic Process](#).
2. Create process roles to define responsibilities. See [Work with Roles in Designer](#).
3. Within the process, create activities that make up the process. See [Create Activities](#).
4. You can choose to divide the process into several segments by creating stages and group activities into these stages. See [Create Stages](#).
5. Define input and output arguments for the process. See [Define Process Input and Output](#).
6. Define data objects to be used within the process. See [Define Process Data Objects](#).
7. Define properties for each stage and activity. See [Define Stage Properties](#) and [Define Activity Properties](#).
8. Finally, define conditions for process completion or termination. See [Define Process Completion and Termination](#).

Create a Dynamic Process

A dynamic process is basically a collection of activities or tasks without a predetermined sequence of execution. It provides flexibility to knowledge workers to define the process flow at runtime based on the information available to them.

To create a dynamic process:

1. In the application main page, click **Add**.
2. In the Add component pane, expand **Processes**, and then click **Dynamic**.
3. Enter a title in the **Title** field.
Note that the **Identifier Name** field gets auto-populated with the title you entered (but without any spaces).
4. Optionally, enter a suitable description.
5. Click **Create**.

The dynamic process is created and listed in the Processes page. You can either open it immediately by clicking the **Open now** link in the Confirmation dialog, or select and open it from the Processes page.

Clone a Dynamic Process

You can clone a dynamic process to create an exact copy of the process. This suits business needs where you have to quickly create a copy of an existing dynamic process.

When you clone a dynamic process, the cloned process will have all the activities and properties that were configured in the original process. All underlying artifacts (such as forms, business objects, integrations) used by the original process will also be used by the cloned process. Note that the artifacts itself are not cloned; but are just used by both the original and the cloned dynamic process.

The cloned process will be a standalone process that can be executed and activated independently. You can also modify and customize the cloned process to meet your business needs.

To clone a dynamic process:

1. Select the dynamic process from the process application's main page or from the Processes page.

2. Hover over **Menu**  and choose **Clone**.

The Clone component pane displays.

3. In the Clone component pane, enter a name in the **Title** field and a suitable description in the **Description** field.
4. Click **Clone**.




A confirmation dialog lets you know that the component is being cloned. You can click the **Open now** link to open the cloned process.



The cloned process gets displayed as a new process in the Processes page as well as the process application's main page.

Create Activities

Activities represent the actions for a process to execute. The **Activities** drop-down menu on the dynamic process canvas contains all the activities that can be used within a dynamic process.

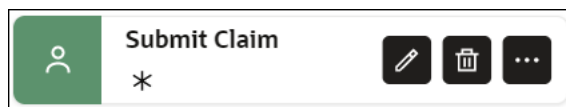
Activities are divided into the following types:

- **Human Task Activity**  – Represents an activity where a process participant is required to perform the work. The task can be a simple interaction, such as filling out a form, or part of a more complicated workflow that requires input from multiple process participants.
- **Process Activity**  – Represents invoking of a structured process within a dynamic process.
- **Service Activity**  – Represents invoking of an external service, such as a REST connector.




- **Milestone**  – Represents a sub-goal within a process. Milestones are typically defined to track progress of a process.
- **Connectors**  – Represents invoking of a REST connector.

To create an activity, select the type of activity that you want to create from the **Activities** drop-down field, provide a suitable name, and click **Add Activity**.

An example human task activity is shown in the following figure:



Select the activity to perform various actions on it. The actions available for each activity are listed as follows:

- **Edit**  – Click this icon to edit the properties of the activity. See [Define Activity Properties](#).
- **Delete**  – Click this icon to delete the activity.
- **More**  – Click this icon to reveal the following additional actions:
 - **Change Type:** Use this action to change the activity type. Select a different activity from the available options.
 - **Data Association:** Use this action to define input and output for activities that require them. See [Configure Data Association](#).
 - **Open Form:** If a human task activity has a form associated with it, use this action to edit or view the form. If no form is associated, this option is grayed out.

In addition, you can select an activity and move it within the stage or drag and drop it outside the stage or into another stage.


Create Stages

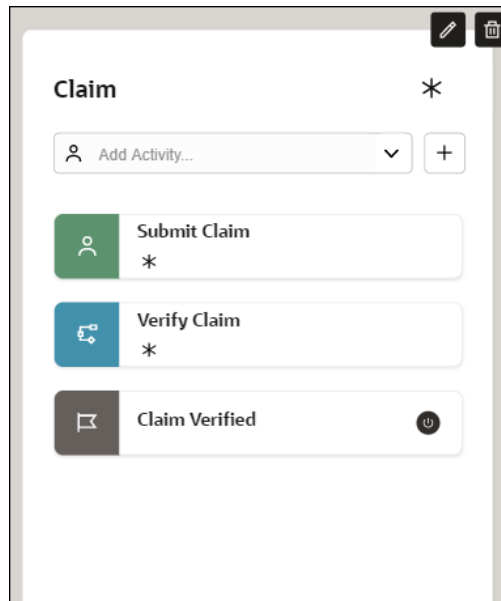
Stages are segments that a dynamic process can be divided into. A stage acts as a container using which you can group and organize activities in a logical way. Typically, activities that must be completed in order for a milestone to be achieved are grouped into a stage.

The execution of stages is not necessarily sequential. Stages can be activated in parallel or upon occurrence of an event.



Note:

Within a process, activities can exist without being grouped into a stage. Such activities are called *Global* activities.

To create a stage on the process canvas, enter a suitable name in the **Add Stage** field, and click **Add Stage**  next to it. An example stage with some activities grouped into it is shown in the following figure:





Select the stage to perform actions on it. The actions available for each stage are listed as follows:

- **Edit**  – Click this icon to edit properties of the stage. See [Define Stage Properties](#).
- **Delete**  – Click this icon to delete the stage.

In addition, you can select a stage and move it to reorganize your process.

Define Stage Properties

You can customize a stage to suit your requirements by defining markers and conditions specific to the stage.

You set properties for a stage in the stage's Properties pane. Select a stage and click **Edit Properties**  that appears to open the Properties pane. Optionally, select a stage and click the **Properties** icon  on the right of the process editor canvas to open the Properties pane.

In the Properties pane, you can do the following:



- Edit the name of the stage. Click the field with the name to modify it.
- Enter a suitable description in the **Description** field. Click the field to enter a description.
- Enable markers for the stage, and if required set conditions for markers.

Enable Markers for a Stage

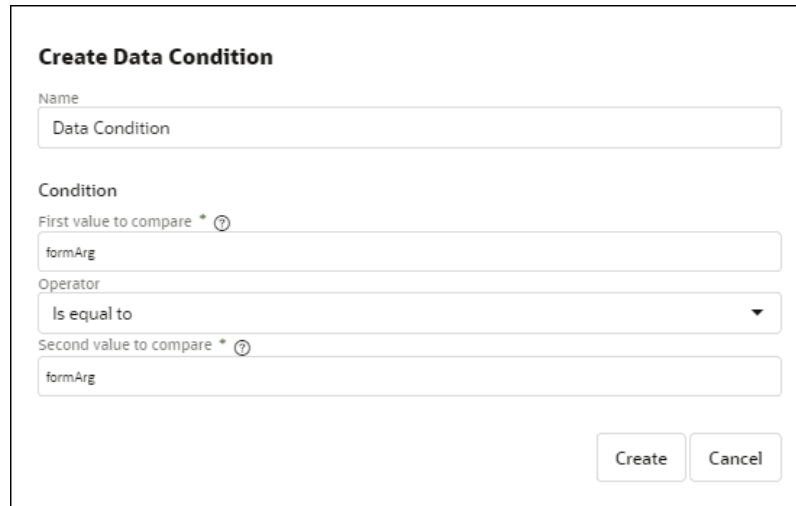
Markers control transitions of a stage between states. In addition, markers can specify if the execution of a stage is mandatory for the process to complete or if the execution must be repeated under certain conditions.

Note that if you apply a marker to a stage without condition then the marker applies to the stage by default.

To enable a marker based on data condition:




1. Select the check box next to the marker.
2. Click the Edit Property icon  that appears next to the selected marker.
3. In the resulting pane, click **Create Data Condition**  to create a new condition.
4. Define the condition.

The following figure shows a data condition that has been defined.



5. Click **Create**.


The following table lists the markers available for a stage.

Marker	Description
Repeatable 	<p>Enable the Repeatable marker to repeatedly execute an entire stage.</p> <ul style="list-style-type: none"> • If this marker is enabled with no data condition, a new instance of the stage is created each time the execution of the stage is complete. • If this marker is enabled with a data condition, a new instance of the stage is created only when the data condition is fulfilled and the stage instance transitions away from the <i>Available</i> state into the next state.
Required 	<p>Enable this marker to make execution of a stage obligatory for the process to complete.</p>
Manually Activated 	<p>Enable the Manually Activated marker if you require a stage to be started by a process participant.</p> <ul style="list-style-type: none"> • If this marker is enabled, the stage transitions from <i>Available</i> to <i>Enabled</i> (after fulfilling enablement conditions if any). From here, a process participant can move the stage into <i>Active</i> or <i>Disabled</i> states. • If this marker is disabled, the stage automatically transitions from <i>Available</i> to <i>Active</i> (after fulfilling activation conditions if any). <p>To define enablement or activation conditions for a stage. See Define Conditions for Stage or Activity.</p>

Marker	Description
Auto Complete <input type="checkbox"/>	<ul style="list-style-type: none"> If the Auto Complete marker is enabled, a stage is automatically marked <i>Complete</i> if none of the activities within it are in the <i>Active</i> state and all required activities are <i>Completed</i>, <i>Terminated</i>, or <i>Disabled</i>. If the Auto Complete marker is disabled, a stage is marked <i>Complete</i> only if all activities are <i>Completed</i>, <i>Terminated</i>, or <i>Disabled</i>. <p>When this marker is disabled, you can also manually complete a stage if none of the activities is in the <i>Active</i> state, and all required activities are <i>Completed</i>, <i>Terminated</i>, or <i>Disabled</i>.</p> <p>Note: For a new stage, the Auto Complete marker is enabled by default.</p>

Define Activity Properties

You can customize an activity to suit your requirements by specifying markers, conditions, and other properties.

You set properties for an activity in the activity's Properties pane. Double click an activity or select an activity and click **Edit Properties**  that appears to open the Properties pane.

Optionally, select an activity and click the **Properties** icon  on the right of the process editor to open the Properties pane.

In the Properties pane, you can do the following:

- Edit the name of the activity. Click the field with the name to modify it.
- Enter a suitable description in the **Description** field. Click the field to enter a description.
- Enable markers for the activity, and if required set conditions for markers. See [Enable Markers for an Activity](#).
- For Human Task Activity, specify some additional details.
- For Process Activity, specify implementation details.
- For Service Activity, specify implementation details.
- For Connectors, specify implementation details.



Enable Markers for an Activity

Use markers to control transition of an activity between states.

Select the check box next to the marker to enable it. When a marker is enabled, a corresponding decorator appears on the activity.

When you enable a marker without conditions, it applies to a stage by default. However, you can also enable markers based on data conditions defined using process variables.

To enable a marker based on data conditions:

- Select the check box next to the marker.
- Click the Edit Property icon  that appears next to the selected marker.
- In the resulting pane, click **Create Data Condition**  to create a new condition.




4. Define the condition.

The following figure shows a data condition that has been defined.

5. Click **Create**.

The condition is set for the activity's marker. The marker gets enabled for the activity only when this condition is satisfied.

The following table lists all markers available for an activity.

Marker	Description
Repeatable 	<p>Enable the Repeatable marker to repeatedly execute an activity.</p> <ul style="list-style-type: none"> If this marker is enabled with no data condition, a new instance of the activity is created each time the execution of the activity is complete. If this marker is enabled with a data condition, a new instance of the activity is created only when the data condition is fulfilled and the activity transitions away from the <i>Available</i> state into the next state.
Required 	<p>Enable this marker to make execution of an activity obligatory for the stage/process to complete.</p> <p>For a new activity, this marker is enabled by default.</p>
Manually Activated 	<p>Enable the Manually Activated marker if you require an activity to be started by a process participant.</p> <ul style="list-style-type: none"> If this marker is enabled, the activity transitions from <i>Available</i> to <i>Enabled</i> (after fulfilling enablement conditions if any). From here, a process participant can move the activity into <i>Active</i> or <i>Disabled</i> states. If this marker is disabled, the activity automatically transitions from <i>Available</i> to <i>Active</i> (after fulfilling activation conditions if any). <p>To define enablement or activation conditions for a stage, see Define Conditions for Stage or Activity.</p>

Implement a Human Task Activity

For a Human Task Activity, in addition to the name, description, and markers, you have to configure some implementation details such as specify the task type, assign the task, associate a form with the activity, and so on in the Properties pane.

Select the Task Type

In the **Type** section:

- Select **Submit** to create a submit task where a user has to complete and submit a form.
- Select **Approve** to create an approve task where a user has to view or/and complete a form, and then perform actions such as approve or reject the form, or perform a custom action that you define.

Assign Tasks

In the **Assignees** section, click the **Assign to** field and assign the task to users (Members), roles (Roles) or users specified through expression (Expressions). You can also search for a user or role by typing the first few characters of the name in the **Search** field.

Specify how the Task appears to users in Workspace

In the **End User Display** section:

- Enter a title in the **Title** field, and a summary to describe it in the **Task Summary** field. These help users to identify the task in Workspace without having to view the details.
- Associate a form with the task to display information to users and perform certain actions such as submit, approve, or reject. Select the form from the **UI** field, and then select a presentation for the form in the **Presentation** field. Optionally, select an external UI connector from the **UI** field.

Select the **Bind to process data** check box to automatically bind form payload and outcome.

The **Action** field gets auto populated depending on the type of task. If the selected task type is Submit, then the field has a SUBMIT action. If the selected task type is Approve, then the field has APPROVE, REJECT actions.

Specify the Task's Priority

In the **Priority** section, you can specify a priority and the due date for the task. From the **Priority** drop-down field, select a priority level - High, Normal, and Low.

In the **Due Date** field, enter an interval to specify the amount of time an assignee has to complete the instance after the human task is triggered. You can select one of the following options:

- **Literal:** Use the format `##M##d##h##m`. For example:
 - One hour and thirty minutes: 1h30m
 - One day: 1d
 - Four months, two days, eleven hours and thirty minutes: 4M2d11h30m
- **Expression:** Specify an expression to calculate the due date of the human task.

These options enable you to sort the Workspace task list based on the task priority.

Implement a Process Activity

For a process activity, associate the structured process that you would like to invoke from within the dynamic process.

In the Implementation section of the Properties pane, select the structured process that you want to associate from the **Process** field.

Note that only structured processes with *None* start events are supported.

In Workspace, when a user starts a dynamic process containing a process activity, the structured process corresponding to this activity is automatically instantiated even if the user doesn't have permissions to instantiate it. However, in such cases, the user is only able to view the structured process instance and its associated tasks. To perform actions in the structured process instance, the user must have the correct permission level or has to be a Process Application Administrator.

Implement a Service Activity

For a service activity, associate a REST connector you'd like to invoke from within the dynamic process.


In the Implementation section of the Properties pane:


- Select the REST connector from the **Service** field.
- Specify the resource of the connector in the **Resource** field.
- Select the operation to perform for it in the **Operation** field.

Define Conditions for Stage or Activity




In the Conditions pane, you can specify conditions to enable, activate or terminate a stage or activity in workspace.

Enablement: If the Manually Activated marker is enabled, you can define an enablement condition to control transition of a stage from *Available* to *Enabled*.

Activation: If the Manually Activated marker is enabled, you can define an activation condition to control transition of a stage or activity from *Available* to *Active*. An activation decorator  appears on the stage or activity that has the condition defined.

Termination: Specify a termination condition to terminate an *Active* stage or activity without its execution being complete. A termination decorator  appears on the stage or activity that has the condition defined.

Conditions are triggered by events or by data or both. To create a condition:

1. Select the stage or activity for which you want to define an activation or termination condition.
2. Click the **Condition** icon  present on the right of the process editor canvas.
3. Click **Create Condition**  next to Activation or Termination.
4. In the resulting pane, provide a suitable label for the condition.
5. To create an event trigger, click **Create Event**  next to Events and define a trigger from the available drop-down fields.




For example, if you want a stage to be completed only after its previous stage is complete, select **Previous Stage** in the Stage or Activity drop-down field, and **Complete** in the Actions drop-down field.

The following table details the options you can select from the **Actions** drop-down field for a Stage or Activity.

Selection	Triggers the Condition When...
Complete	The selected plan item completes execution.
Auto-Start	The selected plan item is automatically started. (Auto transition between <i>Available</i> to <i>Active</i>)
Create	The selected plan item is instantiated.
Disable	The disable action occurs on the selected plan item.
Enable	The enable action occurs on the selected plan item.
Exit	The exit transition occurs on the selected plan item.
Fault	The selected plan item encounters a fault.
Manual Start	The selected plan item is manually started.
Reactivate	The selected plan item is reactivated.
Re-Enable	The selected plan item is re-enabled.
Terminate	The selected plan item's termination condition is met.

 **Note:**

- You can create event triggers for a stage or activity based only on its siblings. That is, for an activity within a stage, you may create an event trigger based on other activities within the same stage. Similarly, you can define event triggers for a stage based on other stages or global activities only.
- When you create multiple event triggers for the same stage or activity, a logical AND expression is formed which includes all triggers.

6. To create a data trigger, click **Create Data Condition**  next to Data Driven to define a condition based on a process variable's value in the payload.
7. Click **Create** to save the changes.
 - If you want to edit a defined data condition click **Edit** .
 - Click **Delete**  to delete a defined event trigger or data condition.

Define Process Input and Output

Define input and output arguments for a dynamic process to communicate or exchange data with other processes and services.


After defining the input arguments for your process, you must map them to data objects of your process or its activities. Similarly, data from an activity in the process can be mapped to output arguments as the output of the process. See [Configure Data Association](#).

In addition, you can use a form as the input to a process to start process execution when a user submits the form. The form is designed to get input from the user and present information relevant to the workflow.


Define process input

To define the process input:

1. On the process page, click **Start Process** .

Alternatively, click **Interface**  and click **Define Input Data** next to **Input**.

The Start the Dynamic Process dialog opens.

2. Choose one of the following options under Who can start the process?
 - Any user with *Use* or *Manage* permission
 - Any Authenticated User
3. Choose one of the following options under How to start the process?
 - **With Data Only**: Select to define input arguments to the process. Use this option to start a process instance with data from other processes or REST services.
 - **With Form**: Use this option to trigger a process instance using a form or REST service.
4. If you select the **With Data Only** option, click **Add Argument** to define new arguments.
 - a. Provide a name to identify the argument and select a data type for the same. You can use simple or pre-defined business data types.
 - b. Click **Delete**  to remove existing arguments from the list.
5. If you select the **With Form** option:
 - a. Provide a form title to display to the user at runtime.
 - b. In the **Form** field, select a form that you want to use as input.
 - c. In the **Presentation** field, select a presentation within the form to display to the user.

 **Note:**

When you use a form as input, a business object to store the form data is automatically created and displayed under the Interface Argument section within the dialog. This input business object is also displayed in the Data pane. See [Define Process Data Objects](#).

- d. Edit the name of the business object, if required.
6. Click **Define** to save changes and close the window.

The following figure shows a form defined as process input:

Start the Dynamic Process

Who can start the process?

Any user with *Use or Manage* permission
 Any Authenticated User

How to start the process?

With Data Only ?
 With Form ?

Form Presentation

Form Title *

Form *


Presentation *



Interface Argument

 MyForm

Define process output

To define the process output:

1. On the process page, click **End Process** .

Alternatively, click **Interface**  and then click **Define Output Data** next to **Output**.
The Define Output Data dialog opens.
2. In the Define Output Data dialog, click **Add Argument** to define new arguments.
 - a. Provide a name to identify the argument and select a data type for the same. You can use simple or pre-defined business data types.
 - b. Click **Delete**  to remove existing arguments from the list.
3. Click **Define** to save changes and close the window.

Define Process Data Objects

Data objects are variables used to store the information related to your process. They also define the type of information used by the process.


The value of a data object can be different for every instance of a process. However, the structure of the data object is the same for all process instances.



When you define a process you can define data objects to store information. In addition, you can also define in which part of the process you assign a value to these data objects. When you create an instance, the process engine assigns *Null* as the default value for all the data objects defined for that process. Thereafter, the activities in the process assign values to these variables.

You can create two types of data objects within a dynamic process:

- **Simple:** Simple data objects define the basic types of variables that you can use within your process. They can be used separately within your process or they can be combined into complex data objects.
- **Business:** Business objects enable you to create data structures based on simple data objects. For example, you can create a business data object called Patient that contains different data types for *patient name*, *age*, and *pincode*.

To create a new data object:

1. Click **Data Object**  on the process page.
2. In the Data Object pane, click **Create Data Object**.
3. In the Create Data Object dialog, enter a name to identify the data object.
4. Select a data type to specify the type of data that the variable can store.
 - a. If you select **Simple**, select a simple data type available in the drop-down list.
 - b. If you select **Business**, you can select from a set of pre-defined business data types available in the drop-down list.
5. Click **Create** to save changes and close the dialog.

Use **Edit**  or **Delete**  to modify an existing data object from the Data pane.



Note:



In addition to data objects, the input and output data that you define for the process also show up in the Data Object pane.

Define Process Completion and Termination


Determine how the dynamic process completes or terminates by specifying an Auto Complete marker, defining a termination condition, or editing its input and output.

Let us explore the ways to complete or terminate a dynamic process.



Enable the Auto Complete Marker


In the Properties pane, you can specify if you want the process to complete automatically by selecting the Auto Complete marker. Click **Properties**  to open the Properties pane. Select the check box next to the Auto Complete marker to enable it. An Auto Complete decorator  appears for the process above the dynamic process editor.

When you enable the marker without conditions, it applies to the process by default. You can also enable the marker based on data conditions defined using process variables.

Marker	Description
Auto Complete 	<ul style="list-style-type: none"> If the Auto Complete marker is enabled, a process is automatically marked <i>Complete</i> if none of the plan items (stages or activities) within it are in the <i>Active</i> state and all required plan items are <i>Completed</i>, <i>Terminated</i>, or <i>Disabled</i>. If this marker is disabled, a process is marked <i>Complete</i> only if all plan items are <i>Completed</i>, <i>Terminated</i>, or <i>Disabled</i>. <p>Note that when this marker is disabled, you can also manually complete a process if none of the plan items is in the <i>Active</i> state, and all required plan items are <i>Completed</i>, <i>Terminated</i>, or <i>Disabled</i>.</p>


Define a Condition for Process Termination

- Click **Condition**  to open the Condition pane
- Click **Create Condition**  next to Termination to create a new condition.
- In the resulting pane, give a suitable name for the condition in the **Label** field.
- Define event triggers or data conditions for the termination condition.
- Click **Create**.

A termination decorator  appears on the process, above the dynamic process canvas.

When a process is terminated, all *Active* stages and activities within the process are also terminated.

Edit Process Input or Output

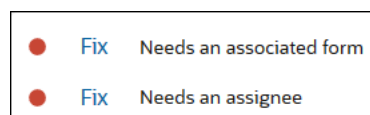
Click **Interface** , to open the Interface pane. Edit or define process input and output.

See [Define Process Input and Output](#).

Work with Inline Validations


Use inline validation to view and fix errors inline as you add and configure elements for your dynamic process. You can see validation issues in the stage, activity and process level. A validation icon in the stage, activity or process level indicates there are validation issues for that level.

- Click the error or warning validation icon to see details about the error or warning in the resulting pane.



Red circles indicate errors. Yellow triangles indicate warnings.

- Click **Fix** to open the related Properties pane where you can fix the issue.

- The validation icon's number displayed at the top of the dynamic process editor indicates the total number of issues found in the process.
- Use the Configuration pane to hide or display validation issues. Click the **Configuration**  icon to open the Configuration pane. Deselect the **Show Inline Errors** and **Show Inline Warnings** check boxes to hide inline validations.

5

Design Forms and User Interfaces

In Oracle Cloud Infrastructure Process Automation, you create web forms to interact with end users.

Topics:

- [Forms and Linked UIs at a Glance](#)
- [Work in the Forms Editor](#)
- [Position Controls on Forms](#)
- [Configure Basic Controls](#)
- [Configure Advanced Controls](#)
- [Implement Intelligent Document Processing in Forms](#)
- [Reuse Forms](#)
- [Work with Presentations](#)
- [Work with Styling](#)
- [Bind Form Data with Controls](#)
- [Create Computed Controls](#)
- [Specify Filters for Controls](#)
- [Add Dynamic Behavior to Forms](#)
- [Preview Forms and Their Payload](#)
- [Configure a Linked UI](#)

Forms and Linked UIs at a Glance

Start by familiarizing yourself with these form and linked UI concepts.

How are forms used in process applications?

Use forms to interact with end users. Many process application start with a form, although they can also start with data.

As part of creating a web form, add its controls, configure its data, change its style, and define form behavior.

What is a linked UI?

Instead of creating a form in Process Automation, you can use a linked UI (such as an external form) instead for displaying the information a user needs to view (or supply) in order to complete the task.

See [Linked UIs at a Glance](#).

Can I present alternate views of a form?

Use presentations to provide other views of a form. For example, present different form views to a user and a manager. You can even dynamically change a presentation during runtime, based on data such as values entered in controls.

See [Work with Presentations](#).

Are there two form editors?

The forms editor is available in two modes: Quick and Web Form.

- Use the **Quick editor** to lay out a basic form in seconds. If you decide you need additional features later, convert the form to use the full editor.
- Use the **Web Form editor** for full functionality, including presentations, advanced controls, dynamic behavior.

See [Work in the Forms Editor](#).

What controls can I add to a form?

The web form editor provides a wide variety of controls. Many of these controls, such as dropdown select fields, can make connector calls or use data to control their behavior.

See [Configure Basic Controls](#) and [Configure Advanced Controls](#).

How can I position and combine controls on a form?

In addition to dragging, you have many options for positioning controls in your form. You can group them in panels, sections, and tabs, and arrange them in tables and repeatable sections.

See [Position Controls on Forms](#).

Can I change the style of forms?

You can customize your form's appearance in many ways using stylesheets and styling, including CSS.

See [Work with Styling](#).

Can forms include dynamic behavior?

Using events, you can include dynamic behaviors in your forms. Combine them with actions, conditions, functions, and REST connector calls. Create filters, loops, and list of value fields.

When you add an event to a control that supports them, the event editor appears. When specifying functions, an embedded Help provides details on each function. View details and examples about selected functions.

See [Add Dynamic Behavior to Forms](#).

Can I preview a form?

At any point in configuring a web form, preview it to see its appearance at different device sizes, test its behavior, and view its payload.

See [Preview Forms and Their Payload](#).

Work in the Forms Editor

Use the forms editor to create and design forms.

The forms editor is available in two modes - Quick and Web Form. Create a simple form within minutes with the Quick editor. Or use the full range of options available in the Web Form editor to design a form.

You also have the option to quickly create a form in the Quick editor and then switch to the Web Form editor by clicking **Convert to Web Form**.

Access the Forms Editor

Creating or opening a form displays the editor.

- **From the Add component pane:** For creating a new form, click **Add** from the application's main page. In the Add component pane, under **UIs**, select either **Quick Form** to create and open the form in the quick editor, or **Web Form** to create and open the form in the web form editor.
- **On the Forms page:** Click **UIs** on the **Components** tabs. If there are no forms created then you get the options - Quick Form and Web Form, to create and open a new form. Forms that you create gets listed in the UIs page.

Use the Forms Editor

In the forms editor, drag, drop, and position controls onto the form canvas. Select from a wide variety of controls available. After adding controls, you can create multiple views or presentations of a form, and customize its styling and properties.

Want to learn more? See:

- [Position Controls on Forms](#)
- [Configure Basic Controls](#)
- [Configure Advanced Controls](#)
- [Work with Presentations](#)
- [Work with Styling](#)

Position Controls on Forms

You have many options for positioning controls in your form.

Rows and Columns

You can drop and position controls onto a form in rows and columns. Rows and columns are defined based on where you want to drop your control.

- Each row has a drop target on the top and bottom.
- Each column has a drop target on the left and right.

Each time you drop a control onto the canvas, the control is placed in a new row. If you drop a control beside another control, then that control is added to the same row but in a new column. The column size is based on the number of controls in a row.

The row is based on a 12-column grid system. For example, if the row has four controls, then each uses three columns. When more than 12 controls are added to a row, the remaining controls are displayed below the table.

Panels, Sections, and Tabs

You can group different controls inside a single control by using panels, sections, and tabs.

A **Panel** control allows you to group multiple controls inside it. Drag and drop controls inside a panel and position them as you would in the canvas.

A **Section** control is similar to a panel, but is collapsible. You can use the section control to create groups of controls that users can expand and collapse. To show the section expanded when the form loads, select **Expand** in the General tab. To show the section collapsed when the form loads, deselect **Expand**. You can drag and drop any other controls inside a section control, including panels and other section controls.

A **Tab** control allows you to group controls in different tabs, to create a tabbed view in your web form. By default, when you drag a tab control onto your web form, one tab is created. To add another tab, click **Add tab**. The new tab is added to the right of the existing one. To rearrange the tab order, drag a tab to the right or left of another tab. In small devices, such as mobile phones, tabs are shown as accordions to accommodate the content in the space. You can drag and drop other controls, such as panels, sections, and tables, into an individual tab. When you select a tab, only the controls in the currently selected tab can be seen.



Note:

If you delete a panel, a section, or a tab, all the controls grouped inside the panel, section, or tab also get deleted.

See [Place Controls in Panels, Sections, or Tabs](#)

Tables and Repeatable Sections

Use **Table** controls to conserve space in a form. The table control allows you to arrange the form controls in a grid pattern. You can add columns into the table and add one control per column. When you delete a column in the table, all the controls in the column get deleted.

If auto binding is selected when you drop a table onto the canvas, then a new entry in the data definition is generated. That attribute is an array of type object. Every time a control is dropped into a column, an attribute inside that list is added.

If auto binding isn't selected, you can select an attribute from the data definition. The controls inside a table can be bound to any attribute in the form data definition. When rendering the form, the data is fetched and shown for that control. However, modifying the value in one row doesn't update every row. Also, when submitting the form, the last control bound to that attribute is the one updating the payload.

Use **Repeatable Section** controls to display multiple copies of a set of controls in a form. Repeatable sections are useful when you want to allow users to enter the same type of input information multiple times in the form; for example, a phone number.

Unlike a table, which is structured, a repeatable section is a free-form container. You can add new controls above, below, or between existing controls. You can surround an individual control with a repeatable section control, or it can include an entire section control. The section control included in the repeatable section must contain all of the form controls that must be repeated.

Tables and repeatable sections provide a way to add information and content dynamically to forms. With these controls, you can run **action** and **if** triggers in the form to display dynamically changing information.

See [Configure Tables](#) and [Configure Repeatable Sections](#).

Configure Basic Controls

The Basic Palette contains a set of basic controls that you can drag onto the canvas.

Note that each time you select a control on the canvas, the **General** and **Styling** tabs become available in the Properties pane to configure settings specific to the selected control. An asterisk (*) indicates a required property.

- [Configure Text Input and Area Fields](#)
- [Configure Simple Text Fields](#)
- [Configure Buttons](#)
- [Configure Drop-down Select Fields](#)
- [Configure Check Lists and Check Boxes](#)
- [Configure Radio Buttons](#)
- [Configure Number Fields](#)
- [Configure Date and Time Fields](#)
- [Configure Email Fields](#)
- [Configure Web Address URL Fields](#)
- [Configure Message Fields](#)
- [Configure Links](#)

Configure Text Input and Area Fields

A text input control allows users to enter short, single line text entries. A text area control allows users to enter longer, multi-line text entries. In a text area control, as a user enters multi-line text, if needed, a scroll bar appears to scroll down and view the entire text.

To configure a text input or a text area control:

1. From the Basic Palette, drag the **Input text** control or the **Text Area** control onto the canvas.
2. Select the control and edit its properties on the Properties pane's **General** tab.

Field	Description
Name	An internal identifier that you will use to identify the control.
Binding	Defines a link between the control and a data attribute.
Computed Value	Set this property to apply a computation to the control. See Create Computed Controls .
Placeholder	Text that will appear in the control until any text is entered. If text is removed from the control, the text specified in this field reappears.
Hint	Hint text that will display to users when a user clicks into the control.

Field	Description
Help	Help text that will appear for the control. If you specify help text, a help icon appears next to the name of the control and displays the help text when a user hovers over it or clicks it.
Min Length	Defines the minimum number of characters a user must enter into the control. Leave this field blank if you don't want to define the minimum length.
Max Length	Defines the maximum number of characters a user can enter into the control.
Rows	Defines the number of text rows that will be visible to a user. A scroll bar appears automatically when the number of text rows entered by the user is greater than the value specified in this field.
Pattern	Allows you to define custom validations on the type of text a user enters into the control. Enter a pattern using regular expressions. When you specify a pattern for the control, you can also specify a message in the Pattern Validation Message field that will display if the validation fails.
Password	Allows you to create a secure text field that masks the characters entered into it.
Events	Allows you to configure events for the control. You can assign actions or If/Then/Else conditions to the control based on a selected event. See Add Dynamic Behavior to Forms .

3. On the **Styling** tab, edit the control's [styling properties](#).
4. Click **Preview** to try out using the control.

Configure Simple Text Fields

Use a simple text control to specify a label without binding for a form field.

To configure a simple text control:

1. From the Basic Palette, drag the **Simple Text** control onto the canvas.
2. Select the control and edit its properties on the Properties pane's **General** tab.

Field	Description
Name	An internal identifier that you will use to identify the control.
Text	Specifies the simple text value that displays as the field's label.
Hide	Select this check box to hide the control.
Events	Specifies events that trigger actions. You can change the simple text value of the control by using an action within Events.

3. On the **Styling** tab, edit the control's [styling properties](#).
4. Click **Preview** to try out using the control.

Configure Buttons

Use a button control to add a button to your web form.

To configure a button control:

1. From the Basic Palette, drag the **Button** control onto the canvas.
2. Select the control and edit its properties on the Properties pane's **General** tab.

Field	Description
Name	An internal identifier that you will use to identify the control.
Events	Allows you to configure events for the control. You can assign actions or If/Then/Else conditions to the control based on a selected event. See Add Dynamic Behavior to Forms .

3. On the **Styling** tab, edit the control's [styling properties](#).
4. Click **Preview** to try out using the control.

Configure Drop-down Select Fields

Use a drop-down select control to add a drop-down list to your web form.

To configure a drop-down select control:

1. From the Basic Palette, drag the **Select** control onto the canvas.
2. Select the control and edit its properties on the Properties pane's **General** tab.

Field	Description
Name	An internal identifier that you will use to identify the control.
Binding	Defines a link between the control and a data attribute.
Computed Value	Set this property to apply a computation to the control. See Create Computed Controls .
Hint	Hint text that will display to users when a user clicks into the control.
Help	Help text that will appear for the control. If you specify help text, a help icon appears next to the name of the control and displays the help text when a user hovers over it or clicks it.
Multiple	Determines whether users can select multiple values in the control. If this field is disabled, users can select only one option in the control.
Options Source	Select a source (Static, From Data, and Connector). <ul style="list-style-type: none"> • Static: Specify choices using Options Names and Options Values fields. Use Options Names to specify the label to display for an option and use Options Values to specify an internal value for an option. • From Data: In the Options List field, select a list of values options source from the data definitions available in the web form. If you selected a list of complex elements, then, in the Label Binding field, specify a data attribute that will display as the label and in the Value Binding field, specify a data attribute that will be the value. • Connector: Specify a REST connector, a resource, and an operation to use. Specify parameters to pass to the connector and define how the response should be mapped to the control properties. See Work with Connectors.
Filter	Select this check box to apply a filter to the control. See Specify Filters for Controls .
Default Value	If you selected Static in the Options Source field, then, specify a default option in this field. If you selected From Data or Connector in the Options Source field, then, select either the first or the last value as the default value.
Events	Allows you to configure events for the control. You can assign actions or If/Then/Else conditions to the control based on a selected event. See Add Dynamic Behavior to Forms .

3. On the **Styling** tab, edit the control's [styling properties](#).
4. Click **Preview** to try out using the control.

Some Useful Event Actions Related to Drop-down Select Fields

This section lists a few useful actions that you can define using a **Select** control. To create an event action, see [Specify Actions](#).

- You can fetch the selected object label using the **Selected Label** option in the **Property** field.
- When a **Select** control is populated by a connector or data attribute, you can fetch any property of the selected object using the **Selected Property** option in the **Property** field.

Configure Check Lists and Check Boxes

Use a check list control to add a list of options to your web form. A check list allows users to select one or more options. You can add a check box control to your web form to allow users to specify a true or false value.

Note that a check box control is set to false by default. You can edit the **Default Value** property on the **General** tab to change the default value.

To configure a check list or a check box control:

1. From the Basic Palette, drag the **Checklist** or the **Checkbox** control onto the canvas.
2. Select the control and edit its properties on the Properties pane's **General** tab.

Field	Description
Name	An internal identifier that you will use to identify the control.
Label	Specifies the control name that displays to the user. For check box controls, if you select the Inline HTML check box below the Label field, the value that you enter in the Label field is treated as an inline HTML.
Binding	Defines a link between the control and a data attribute.
Computed Value	Set this property to apply a computation to the control. See Create Computed Controls .
Help	Help text that will appear for the control. If you specify help text, a help icon appears next to the name of the control and displays the help text when a user hovers over it or clicks it.
Inline	Specify the layout of the options defined for the check list control. If enabled, the layout changes from vertical to horizontal.

Field	Description
Options Source	Select a source (Static, From Data, and Connector). <ul style="list-style-type: none"> • Static: Specify choices using Options Names and Options Values fields. Use Options Names to specify the label to display for an option and use Options Values to specify an internal value for an option. • From Data: In the Options List field, select a list of values options source from the data definitions available in the web form. If you selected a list of complex elements, then, in the Label Binding field, specify a data attribute that will display as the label and in the Value Binding field, specify a data attribute that will be the value. • Connector: Specify a REST connector, a resource, and an operation to use. Specify parameters to pass to the connector and define how the response should be mapped to the control properties. See Work with Connectors.
Filter	Select this check box to apply a filter to the control. See Specify Filters for Controls .
Default Value	If you selected Static in the Options Source field, then, specify a default option in this field. If you selected From Data or Connector in the Options Source field, then, select either the first or the last value as the default value. For the check box control, select either True or False as the default value.
Events	Allows you to configure events for the control. You can assign actions or If/Then/Else conditions to the control based on a selected event. See Add Dynamic Behavior to Forms .

3. On the **Styling** tab, edit the control's [styling properties](#).
4. Click **Preview** to try out using the control.

Configure Radio Buttons

Add mutually exclusive radio buttons to your web form. Radio buttons allow users to select an option from a set of available options.

To configure a radio button control:

1. From the Basic Palette, drag the **Radio Button** control onto the canvas.
2. Select the control and edit its properties on the Properties pane's **General** tab.

Field	Description
Name	An internal identifier that you will use to identify the control.
Binding	Defines a link between the control and a data attribute.
Computed Value	Set this property to apply a computation to the control. See Create Computed Controls .
Help	Help text that will appear for the control. If you specify help text, a help icon appears next to the name of the control and displays the help text when a user hovers over it or clicks it.
Inline	Specify the layout of the options defined for the control. If enabled, the layout changes from vertical to horizontal.

Field	Description
Options Source	Select a source (Static, From Data, and Connector). <ul style="list-style-type: none"> • Static: Specify choices using Options Names and Options Values fields. Use Options Names to specify the label to display for an option and use Options Values to specify an internal value for an option. • From Data: In the Options List field, select a list of values options source from the data definitions available in the web form. If you selected a list of complex elements, then, in the Label Binding field, specify a data attribute that will display as the label and in the Value Binding field, specify a data attribute that will be the value. • Connector: Specify a REST connector, a resource, and an operation to use. Specify parameters to pass to the connector and define how the response should be mapped to the control properties. See Work with Connectors.
Filter	Select this check box to apply a filter to the control. See Specify Filters for Controls .
Default Value	If you selected Static in the Options Source field, then, specify a default option in this field. If you selected From Data or Connector in the Options Source field, then, select either the first or the last value as the default value.
Events	Allows you to configure events for the control. You can assign actions or If/Then/Else conditions to the control based on a selected event. See Add Dynamic Behavior to Forms .

3. On the **Styling** tab, edit the control's [styling properties](#).
4. Click **Preview** to try out using the control.


Configure Number Fields

Add number controls to your web form. Number controls allow users to enter decimal numbers. By default, number controls display 0 but you can set this to any decimal number using the **Default Value** field in the **General** tab.

To configure a number control:

1. From the Basic Palette, drag and drop a **Number** control onto the canvas.
2. Select the control and edit its properties on the Properties pane's **General** tab.

Field	Description
Name	An internal identifier that you will use to identify the control.
Binding	Defines a link between the control and a data attribute.
Computed Value	Set this property to apply a computation to the control. See Create Computed Controls .
Default Value	Sets a value to display to users when the form loads.
Placeholder	Hint text that describes the expected value. This hint text will display in the control before users enter a value.
Max	Sets a maximum value that users can enter into the control.
Min	Sets a minimum value that users can enter into the control
Show increment/decrement buttons	Select this check box to display the up and down arrow buttons used to increment or decrement the control value.

Field	Description
Step	<p>Specifies a value based on which the number will increase or decrease when users increment or decrement the value with the up or down arrow.</p> <p>For example, if the step value is set to 4, and the initial value in the number field is 0, then when users increment the value for the first time, the value will be 4. When users increment the value the second time, it will be 8 and so on.</p> <p>By default the step value is set to 1. Step must be always greater than 0 and you can't enter a value lower than 1.</p>
<div style="border: 1px solid #0070C0; padding: 10px; background-color: #E6F2FF;">  Note: This property is displayed only when you select the Show increment/decrement buttons check box. </div>	
Events	<p>Specifies events that trigger actions and conditions. You can customize the control's behavior by configuring events. See Add Dynamic Behavior to Forms.</p>

3. On the **Styling** tab, edit the control's [styling properties](#).
4. Click **Preview** to try out using the control.

Configure Date and Time Fields

A Date Time control allow users to enter date and time together. Optionally, add a Date and a Time control to your web form to allow users to enter date and time separately. The Date Time, Date and Time controls are available in the Basic Palette.

To configure a Date Time control:

1. From the Basic Palette, drag and drop a **Date Time** control onto the canvas.
2. Select the control and edit its properties on the Properties pane's **General** tab.

Field	Description
Name	An internal identifier that you will use to identify the control.
Binding	Defines a link between the control and a data attribute.
Computed Value	Set this property to apply a computation to the control. See Create Computed Controls .
Placeholder	Hint text that describes the expected value. This hint text will display in the control before users enter a value.
Help	Help text to display to users when they hover over or click the control's help icon.
Default Value	Sets the default date and time that appears in the control when the form loads and the current control value is empty.
Max Time	Sets a maximum date and time users can enter in the control.
Min Time	Sets a minimum date and time users can enter in the control.
Format	Specifies a date format for the control. You can select from the available date formats such as yyyy-MM-dd, MM/dd/yyyy, dd-MM-yyyy and so on.

Field	Description
Events	Specifies events that trigger actions and conditions. You can customize the control's behavior by configuring events. See Add Dynamic Behavior to Forms .

3. On the **Styling** tab, edit the control's [styling properties](#).
4. Click **Preview** to try out using the control.

 **Note:**

When a user enters data into a Date Time control, the timezone information of the user is saved (in UTC). When other users access this data, the date-time information is displayed in timezones specific to them. This timezone conversion capability is available only in the Date Time control and not in the separate Date and Time controls.

Configure Email Fields

Add an email control to allow users to enter a valid email address to your web form.

1. From the Basic Palette, drag and drop an email control onto the canvas.
2. Select the control and edit its properties on the Properties pane's **General** tab.

Field	Description
Name	An internal identifier that you will use to identify the control.
Binding	Defines a link between the control and a data attribute.
Computed Value	Set this property to apply a computation to the control. See Create Computed Controls .
Default Value	Specifies an email to display to users when the form loads and the current control value is empty. You must ensure that you enter a valid email format in the field, otherwise you will get a message indicating that the email format entered isn't valid.
Placeholder	Hint text that describes the expected email format. This hint text will display in the control before users enter a value.
Hint	Useful hint text that displays to users when they select the control.
Help	Help text to display to users when they hover over or click the control's help icon.
Max Length	Sets the maximum number of characters users can enter before the @ symbol for the email.
Min Length	Sets the minimum number of characters users can enter before the @ symbol for the email.
Events	Specifies events that trigger actions and conditions. You can customize the control's behavior by configuring events. See Add Dynamic Behavior to Forms .

3. On the **Styling** tab, edit the control's [styling properties](#).
4. Click **Preview** to try out using the control.

Configure Web Address URL Fields

Add a URL control to allow users to enter a web address URL to your web form.

1. From the Basic Palette, drag and drop a **URL** control onto the canvas.
2. Select the control and edit its properties on the Properties pane's **General** tab.

Field	Description
Name	An internal identifier that you will use to identify the control.
Binding	Defines a link between the control and a data attribute.
Computed Value	Set this property to apply a computation to the control. See Create Computed Controls .
Default Value	Sets a value to display to users when the form loads and the current control value is empty. You must ensure that you enter a correct web address URL format in the field, otherwise you will get a message indicating that the URL format entered isn't valid.
Placeholder	Hint text that describes the expected value. This hint text will display in the control before users enter a value.
Hint	Useful hint text to display to users when they select the control.
Help	Help text to display to users when they hover over or click the control's help icon.
Max Length	Specifies the maximum number of characters users can enter into the control.
Min Length	Specifies the minimum number of characters users must enter into the control.
Events	Specifies events that trigger actions and conditions. You can customize the control's behavior by configuring events. See Add Dynamic Behavior to Forms .

3. On the **Styling** tab, edit the control's [styling properties](#).
4. Click **Preview** to try out using the control.

Configure Message Fields

Use a message control to allow users to enter a simple message to your web form.

1. From the Basic Palette, drag and drop a **Message** control onto the canvas.
2. Select the control and edit its properties on the Properties pane's **General** tab.

Field	Description
Name	An internal identifier that you will use to identify the control.
Binding	Defines a link between the control and a data attribute.
Computed Value	Set this property to apply a computation to the control. See Create Computed Controls .
Default Text	The default text message that will appear to users when the form loads. You can use inline HTML tags to format the default text. Before using, you have to select Inline HTML from the Type drop-down list.

Field	Description
Type	<p>Sets the style and format in which the message displays. For example, as a bold heading or as a paragraph text.</p> <p>Note that predefined message types such as Error, Info, Success, and Warning are available for selection. These message types have their unique style and format, so that the user can easily identify if the message is an error, info, success, or warning.</p> <p>Select Inline HTML if you want to use inline HTML tags to format the default text message that displays when the form loads.</p>
Events	<p>Specifies events that trigger actions and conditions. You can customize the control's behavior by configuring events. See Add Dynamic Behavior to Forms.</p>

3. On the **Styling** tab, edit the control's [styling properties](#).
4. Click **Preview** to try out using the control.

Configure Links

Use the link control to insert a URL into a form. You can specify the value for the link URL or configure the URL value to change dynamically based on the payload. For example, a URL link that contains an order item could change based on the order ID, which could be in the payload.


1. From the Basic Palette, drag and drop a **Link** control onto the canvas.
2. Select the control and edit its properties on the Properties pane's **General** tab.

Field	Description
Name	Defines an internal name for the control. It's an internal identifier that you'll use to identify the control.
Label	Specifies the control name that displays to a user.
Static	Select this option to use the value in the Default Label field as the control name when the form loads. This is the default selection.
Dynamic	Select this option to assign the control name dynamically when the form loads. If you select this option, you must create a data attribute in the Data pane and link it to the control using the Label Binding field. When the form loads, the value of the data attribute is fetched from the payload and assigned as the control name.

Note:

If the attribute value is not available in the payload, then the value in the **Default Label** field is used as the control name.

Label Binding	Defines a link between the control's label and a data attribute. Specify an attribute for this field by selecting an option from the autocomplete list or entering a valid binding.
Default Label	<p>Sets a label to display to users when the form loads. The value in this field is used as the control name in the following contexts:</p> <ul style="list-style-type: none"> • When you select the Static option in the Label field. • When you select Dynamic option in the Label field but the binding value is not available.

Field	Description
Value Binding	Defines a link between the control and a data attribute. Data attribute bound to the control, either automatically when Auto Binding is enabled or manually using autocompletion.
Computed Value	Set this property to apply a computation to the control. See Create Computed Controls .
Default Value	Sets a value to display to users when the form loads and the current control value is empty.
Open	Specifies whether you want the link to open in the current tab or in a new tab.
Anchor	Select this check box to enable linking to specific controls in the current form. When you select this check box, the Default Value field changes to a drop-down menu that lists names of all controls present in the form. Select a control name to which you want to link. All Basic Palette, Money, Phone, Image, and Video controls, if present in the current form, appear in the Default Value field. However, controls within repeatable sections, tables, or imported from other forms (unless detached) do not appear in the drop-down list.
<div style="border: 1px solid #0070c0; padding: 10px; background-color: #e6f2ff;">  Note: If the URL value for the link is set from binding or events, the value selected in the Default Value field is overridden. </div>	
Hide	Select this check box to hide the control. For example, you might hide a control by default, but configure another control that when selected triggers an event that displays the hidden control.
Events	Specifies events that trigger actions and conditions. You can customize the control's behavior by configuring events. See Add Dynamic Behavior to Forms .

3. On the **Styling** tab, edit the control's [styling properties](#).
4. Click **Preview** to try out using the control.

Configure Advanced Controls

The Advanced Palette contains a set of advanced controls that you can drag onto the canvas. When you select a control on the canvas, the **General** tab and the **Styling** tab become available in the Properties pane so you can configure settings specific to the selected control. An asterisk (*) indicates a required property.

- [Configure Currency \(Money\) Fields](#)
- [Configure Phone Number Fields](#)
- [Include Images](#)
- [Include Videos](#)
- [Configure Identity Browser Fields](#)
- [Place Controls in Panels, Sections, or Tabs](#)
- [Configure Static and Dynamic List of Values Fields](#)
- [Configure Tables](#)

- [Configure Repeatable Sections](#)
- [Configure Train Controls](#)
- [Configure Divider Controls](#)
- [Configure File Manager Controls](#)

Configure Currency (Money) Fields

Add a money control to your web form to allow users to enter money amounts (USD, EUR, JPY, GBP, and INR). By default, the currency type is set to **USD**. The user can change the currency type using the **Currency** property on the **General** tab. The corresponding currency symbol displays next to the currency amount. Users can use commas or decimal point to enter different amounts into the control, such as £ 30,700.00. If users don't specify commas or decimal point, the web form automatically displays these symbols. The web form rounds all currency amounts to two decimal places.

To configure a money control:

1. From the Advanced Palette, drag the **Money** control onto the canvas.
2. Select the control and edit its properties on the Properties pane's **General** tab.

Field	Description
Name	An internal identifier that you will use to identify the control.
Binding	Defines a link between the control and a data attribute.
Computed Value	Set this property to apply a computation to the control. See Create Computed Controls .
Currency	Allows you to change the currency type.
Placeholder	Text that will appear in the control until any text is entered. If text is removed from the control, the text specified in this field reappears.
Hint	Hint text that will display to users when a user clicks into the control.
Help	Help text that will appear for the control. If you specify help text, a help icon appears next to the name of the control and displays the help text when a user hovers over it or clicks it.
Min	Specify the minimum amount that users need to enter into the control.
Max	Specify the maximum amount that users can enter into the control.
Show increment/decrement buttons	Select this check box to display the up and down arrow buttons used to increment or decrement the control value.
Step	Specify a step value based on which the amount will be incremented or decremented correspondingly when a user increments or decrements the amount in the control. By default, the step value in this field is set to 1. For example, if the step value specified is 3 and the initial amount is \$ 0.00, then, when a user increments the amount in the control for the first time, the amount is updated to \$ 3.00. When the user increments the amount again for the second time, the amount is updated to \$ 6.00 and so on.

Note:

This property is displayed only when you select the **Show increment/decrement buttons** check box.

Field	Description
Show increment/decrement buttons	Select this check box to display the up and down arrow buttons used to increment or decrement the control value.
Events	Allows you to configure events for the control. You can assign actions or If/Then/Else conditions to the control based on a selected event. See Add Dynamic Behavior to Forms .

3. On the **Styling** tab, edit the control's [styling properties](#).
4. Click **Preview** to try out using the control.

Configure Phone Number Fields

Add a phone control to your web form to allow users to enter phone numbers in International or US formats. The phone control uses the US format by default and displays the expected phone number pattern (xxx-xxx-xxxx). You can change the format using the **Format** property on the **General** tab.

To configure a phone control:

1. From the Advanced Palette, drag the **Phone** control onto the canvas.
2. Select the control and edit its properties on the Properties pane's **General** tab.

Field	Description
Name	An internal identifier that you will use to identify the control.
Binding	Defines a link between the control and a data attribute.
Computed Value	Set this property to apply a computation to the control. See Create Computed Controls .
Format	Specify a phone number format. By default, the US format is selected.
Placeholder	Text that will appear in the control until any text is entered. If text is removed from the control, the text specified in this field reappears.
Hint	Hint text that will display to users when a user clicks into the control.
Help	Help text that will appear for the control. If you specify help text, a help icon appears next to the name of the control and displays the help text when a user hovers over it or clicks it.
Max Length	Defines the maximum number of characters a user can enter into the control.
Min Length	Defines the minimum number of characters a user must enter into the control. Leave this field blank if you don't want to define the minimum length.
Events	Allows you to configure events for the control. You can assign actions or If/Then/Else conditions to the control based on a selected event. See Add Dynamic Behavior to Forms .

3. On the **Styling** tab, edit the control's [styling properties](#).
4. Click **Preview** to try out using the control.

Include Images

Use an image control to include an image in your web form. Under image source properties, you can identify the image using an absolute or a relative image URL or a Base64 format string.

To configure an image control:

1. From the Advanced Palette, drag the **Image** control onto the canvas.
2. Select the control and edit its properties on the Properties pane's **General** tab.

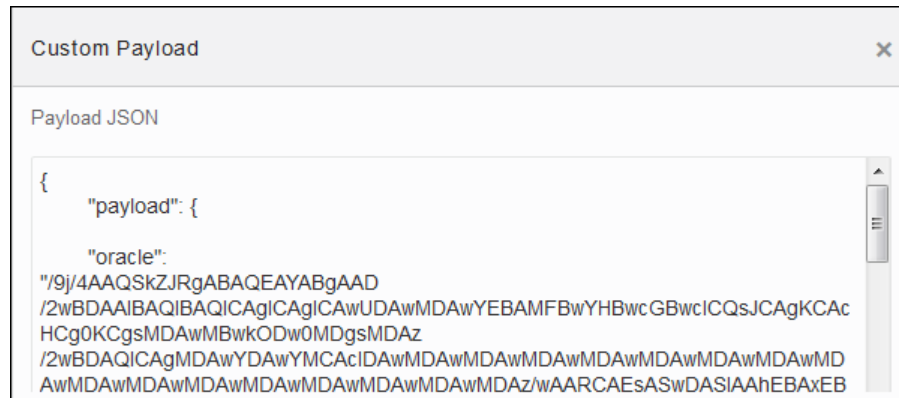
Field	Description
Name	An internal identifier that you will use to identify the control.
Binding	Defines a link between the control and a data attribute.
Computed Value	Set this property to apply a computation to the control. See Create Computed Controls .
From URL	To specify either an absolute or relative URL, select this option and enter the image URL in the field below it.
From Base64	To specify an image converted to base64 format, select this option, then identify the image's format in the Image Format field and its binding in the Base64 Binding field. Base64 images can be viewed in preview mode only.
Alternative text	Enter text to display if the image can't be loaded.
Events	Allows you to configure events for the control. You can assign actions or If/Then/Else conditions to the control based on a selected event. See Add Dynamic Behavior to Forms .

3. On the **Styling** tab, edit the control's [styling properties](#).
4. Click **Preview** to try out using the control.

Upload and Preview a Base64 Image

You can embed Base64 encoded images in forms. They load quickly and are useful for email signatures.

1. Drag and drop an Image control onto the canvas and select it.
2. On the General tab, specify Base64 settings in the **Image Source** fields:
 - a. Select **From Base64**.
 - b. In the **Image Format** field, specify the format you chose when converting the image to base64.
 - c. In the **Base64 Binding** field, enter a name for the image (for example, a data string called *oracle*).
3. If the encoded string of your base64 image starts with `data:image/jpeg;base64`, remove it. The base64 input you need begins with: `"/9j/4AAQ...."`, not `"data:image/jpg;base64,/9j/4AAQ...."`
4. Click **Preview**, then **Reload With Payload**. In the Payload JSON field, enter the binding name and copied code string to the payload. Enclose the binding name and code string within quotation marks.



5. Click **OK** and the image displays in the preview window.

Note that the **Reload with Payload** button enables you to test the form's functionality. In a typical use case, the encoded string would be passed as part of the payload.

Include Videos

Use a video control to add a video, such as a YouTube or Vimeo video, to your web form. You can specify full video URLs, embedded URLs, or shortened URLs using the **Source Url** property on the **General** tab. You can optionally loop the video or specify to automatically start playing the video when loaded.

To configure a video control:

1. From the Advanced Palette, drag the **Video** control onto the canvas.
2. Select the control and edit its properties on the Properties pane's **General** tab.

Field	Description
Name	An internal identifier that you will use to identify the control.
Binding	Defines a link between the control and a data attribute.
Computed Value	Set this property to apply a computation to the control. See Create Computed Controls .
Source Url	Specify a valid source URL for the video.
Allow Fullscreen	Set this property to allow users to play the video in full screen mode. By default, this field is enabled.
Loop	Set this property to loop the video continuously.
Auto Play	Set this property to automatically start playing the video when the web form loads.
Show Controls	Specify whether to display play or pause controls for the video. By default, this field is enabled.
Events	Allows you to configure events for the control. You can assign actions or If/Then/Else conditions to the control based on a selected event. See Add Dynamic Behavior to Forms .

3. On the **Styling** tab, edit the control's [styling properties](#).
4. Click **Preview** to try out using the control.

Configure Identity Browser Fields

Add an Identity Browser control to allow users to search and select individuals to be notified or assigned tasks downstream in the process.

Use identities to search for users, groups, or roles. You can add single or multiple entries to the Identity Browser based on the configuration you select. Use options in the search filter to restrict the scope of the search.

To configure an Identity Browser control:

1. From the Advanced Palette, drag the **Identity Browser** control onto the canvas.
2. Select the control and edit its properties on the Properties pane's **General** tab.

Field	Description
Name	An internal identifier that you will use to identify the control.
Label	Specifies the control name that displays to a user.
Binding	Defines a link between the control and a data attribute.
Placeholder	Text that will appear in the control until any text is entered. If text is removed from the control, the text specified in this field reappears.
Hint	Hint text that will display to users when a user clicks into the control.
Help	Help text that will appear for the control. If you specify help text, a help icon appears next to the name of the control and displays the help text when a user hovers over it or clicks it.
Default Scope	Specifies the default scope for the identity search. Available options in the drop-down menu are All , User , Group , and Role . The value of this field is set to User by default.
Required	Set this property to require users to complete the control in order to successfully submit the form.
Multiple	Set this property to allow multiple entries to the control.
Disabled	Set this property to display the control as inactive.
Hide	Set this property to hide the control. For example, you might hide a control by default, but configure another control that when selected triggers an event that displays the hidden control.
Scope Filter	Set this property to allow filtering of results in the control.
Auto Focus	Set this property to automatically select the control when the web form loads.
Events	Allows you to configure events for the control. You can assign actions or If/Then/Else conditions to the control based on a selected event. See Add Dynamic Behavior to Forms .

Note:

Except for assigning a constant value, you cannot trigger any changes to the Identity Browser using other controls. Actions such as assigning function results, connector call values, or data from the payload or another control are not applicable in the context of Identity Browser.


3. On the **Styling** tab, edit the control's [styling properties](#).
4. Click **Preview** to try out using the control.

Retrieving a List of User Identities

An identity object contains several fields, such as title, ID (user identity), type, e-mail, first name, last name, and contact number.

Using a combination of Identity Browser and Text controls, you can retrieve a comma-separated list of user identities (IDs), which can be used by other process activities.

To fetch a list of IDs:

1. Add an **Identity Browser** control to the form canvas.
2. Edit the control's properties as required. Make sure to select the **Multiple** checkbox to allow multiple entries to the control.
3. Drag the **Input text** control onto the canvas. You may also use the **Text Area** control.
4. Configure an event to populate data in the **Input text** control field based on the data entered into the **Identity Browser** control. See [Add Dynamic Behavior to Forms](#) for details on creating events.
 - a. Select **Identity Browser**, and in the Properties pane's **General** tab, scroll down until you see **Events**.
 - b. Click **Add**  to define an event.
 - c. Select an event option, for example, **On Change**; this event is activated when an end user changes the control's value.
 - d. After selecting the event option, click the editing icon next to the events drop-down menu to specify the action. An event window appears.
 - e. In the event window, click **+Action** to add an action.

A solid red **Action** indicator appears. Below it, configure the action.
 - f. In the **Control Name** field, select the control the action will affect, in this case select **InputText**.
 - g. In the **Action** field, select **Value**.
 - h. In the **Type** field select **Control**, and in the resulting **Control Name** field, select **IdentityBrowser**.
 - i. Finally, select **Identities** in the **Property** field. This fetches the data in the ID fields of entries in the **Identity Browser** control to the **Input text** field.
 - j. Click **OK**, then **Save**.
5. Click **Preview** to test the controls.
6. On the **Preview** page, search for and select multiple identity objects in the **Identity Browser** control.
7. The **Input text** field is populated automatically with the list of user identities, that is, the data in the ID fields of these entries.



Place Controls in Panels, Sections, or Tabs

You can add panels, sections, and tabs to your web form and use them to group multiple controls under a single control.

To configure a panel, section, or tab control:

1. Expand the Advanced Palette.
2. Drag and drop a **Panel**, **Section**, or **Tab** control onto the canvas.
 - A text *Drop Elements to this panel!* indicates that you can drop controls into the panel.
 - A text *Drop Elements to this section!* indicates that you can drop controls into the section.
 - A text *Drop Elements to this tab!* indicates that you can drop controls into the tab. By default, the tab control displays one tab (Tab1). Click **Add tab** to insert additional tabs into the control.
3. Select the control, and then edit its properties on the **General** tab in the Properties pane.

Field	Description
Name	An internal identifier that you will use to identify the control.
Label	The title of the panel, section, or tab that the user will see in the form.
Description	For a panel control, provides additional information or instructions for the user.
Type	For a section control, sets the style and format in which the section label displays.
Hide	Set this property to hide the control.
Read Only	Set this property to make the (panel, section, or tab) control read-only. Note that when this property is set, all the controls inside a panel, section, or tab control become read-only (irrespective of the individual property of each of the controls). The user can view but is not allowed to edit the control.
Lazy Loading	For a section control, implements lazy loading for the controls inside it. The lazy loading controls are collapsed by default.

Field	Description
Expanded	For a section control, specifies if the section control is expanded when the form loads. By default, this field is checked.
Event	Specifies events that trigger actions and conditions. You can customize the control's behavior by configuring events. See Add Dynamic Behavior to Forms .

- On the Styling tab, edit the control's styling properties.
- Drag and drop individual controls from the Basic Palette or the Advanced Palette into the panel, section, or tab control.

See [Position Controls on Forms](#).

- Configure general and styling properties for the controls inside your panel, section, or tab control.

See [Configure Basic Controls](#) and [Configure Advanced Controls](#).

 **Note:**

If you delete the panel, section, or tab control, all the controls grouped under it get deleted.

- Click **Preview** to try out using the control.

 **Note:**

If the data in a field within a section or tab is invalid, a validation error is displayed for the field and the entire section or tab is marked invalid.

Format the Title and Description of a Panel

Use a theme to format the title and description for a panel in your form.

To select a theme for a panel:

- From the Advanced Palette, drag and drop the **Panel** control onto the canvas.
- Select the **Panel** control on the canvas.
- In the Properties pane, select the **General** tab.
 - In the **Label** field, enter a title for the panel. The user will see this title on the form.
 - In the **Description** field, add information or instructions for the user.
- In the Properties pane, select the **Styling** tab.
- In the **Theme** field, select **Standard**.

The Standard theme automatically:

- Increases the font size of the label to 24 pixels (font-size: 24px)
- Applies the bold style to the label (font-weight: bold)

Selecting a theme overrides the value defined in the **Label Size** field on the **Styling** tab. If you want to use the **Label Size** field to format the panel label, then make sure the **Theme** field is set to **None**.

You can use the **Label Color** field to specify a color for the label and the descriptive text in the panel.

Indent Sections

Use a theme to indent the sections in your form.

To indent one or more sections in a form:

1. From the Advanced Palette, drag and drop the **Section** control onto the canvas.
2. Select the **Section** control on the canvas.
3. In the **Properties** pane, select the **Styling** tab.
4. In the **Theme** field, select **Indent**.

Configure Static and Dynamic List of Values Fields

Use checklists, radio button or select controls to create static or dynamic list of values in your web form. Static list of values are defined in the control properties. Dynamic list of values are defined from a collection of available attributes or are based on a REST call.

You can configure static and dynamic list of values for checklists, radio buttons and select controls from **Option Source** in the **General** tab. In **Option Source** specify whether the control's options should come from static values you enter, from a list of values data attribute, or externally from a REST connector created for the application.

To configure static and dynamic list of values (for checklist, radio button and select controls):

1. From the Basic Palette, drag and drop a checklist, radio button or select control onto the canvas.
2. Select the control and configure its static list of values on the Properties pane **General** tab.
 - a. Go to **Option Source** and click **Static**.
 - b. In the **Option Names** field, enter a label to display for each option.
 - c. In the **Option Values** field, enter an internal value for each option.
 - d. Specify a default value from the option values available in the **Default Value** field.
 - e. Select an option in the **Autofocus** field to make that option the selected option when the form loads.
3. Optionally, configure dynamic list of values for the checklist, radio button or select control. There are two options to configure dynamic list of values.
 - Specify that the control's options should come from a list of value data attributes defined in the Data definition pane.
 - a. Click **From Data**.
 - b. Select the list of values options source from available attributes in the **Options List** field.
 - c. Specify a default value from the options available in the **Default Value** field.
 - d. Select an option in the **Autofocus** field to make that option the selected option when the form loads.
 - Specify that the control's options should come from a REST connector created for the application.
 - a. Click **Connector**.

- b. Specify the connector settings in the Connector, Resource and Operation fields and map response settings. See [Work with Connectors](#).
 - c. Select an option in the **Autofocus** field to make that option the selected option when the form loads.
4. Click **Preview** to try out using the control.


Add an Option to a List of Values Field

You can add an option to a list of values field (checklist, radio button, or select) by specifying an action **Append Option** to it. The new option gets added to already present static or dynamic values in the list of values field.

See [Specify Actions](#).

Configure a control in your form such that you can use it for adding an option to a list of values field. For example, you can configure a button control with an On Click event and use it to add an option to a list of values field.

To add an option to a list of values (LOV) control:

1. Drag and drop a control, such as a button control, from the Basic Palette onto your form with the LOV control. Change the label and name of the control, for example **Add**.
2. Configure the control by specifying an event and defining an action for the event.
 - a. Select the control, and scroll down the **General** tab of the Properties pane until you find **Events**. Click  to add and define an event.
 - b. Select **On Click** from the drop-down list and then click the editing icon to specify the action for the event.
 - c. In the On Click event window, click **+Actions** to define an action.
 - d. Specify the **Append Option** action to the LOV control. Select the LOV control from the **Control Name** drop-down list, and then select **Append Option** from the **Action** drop-down list.
 - e. In the **Option Name Type** and **Option Value Type** drop-down lists, select a type, for example **Constant**.
 - f. Enter the value that you want to add to the default options for the LOV control in the **Value** fields.
 - g. Click **OK**.
3. Preview the form and see how it works.

Click **Preview**. When the form loads, view the options in the LOV field.

Click the control for which you configured an On Click event, and view the options in the LOV field again. Note that the value you specified in step 2f gets added to the list of values.


Clear All Options From a List of Values Field

You can delete all options from a list of values field (checklist, radio button, or select) by specifying an action **Clear Options** to it.

See [Specify Actions](#).

Configure a control in your form such that you can use it for clearing all options from a list of values field. For example, you can configure a button control with an On Click event and use it to clear all options from a list of values field.

To delete all options from a list of value (LOV) control:

1. Drag and drop a control, such as a button control, from the Basic Palette onto your form with the LOV control. Change the name and label of the control, for example **Clear**.
2. Configure the control by specifying an event and defining an action for the event.
 - a. Select the control, and scroll down the **General** tab of the Properties pane until you find **Events**. Click  to add and define an event.
 - b. Select **On Click** from the drop-down list and then click the editing icon to specify the action for the event.
 - c. In the On Click event window, click **+Actions** to define an action.
 - d. Specify the **Clear Options** action to the LOV control. Select the LOV control from the **Control Name** drop-down list, and then select **Clear Options** from the **Action** drop-down list.
 - e. Click **OK**.
3. Preview the form and see how it works.

Click **Preview**. When the form loads, view the options in the LOV field. Now click the control for which you configured an On Click event, and view the LOV field again.



Notice that the LOV field is empty and all options have been cleared.

Configure Tables

Use table controls to group multiple controls in a grid pattern into your form. You can use table controls to create dynamic content for your form.

To configure a table control:

1. From the Advanced Palette, drag and drop a table control onto the canvas.
By default the table control contains one column.
2. Select the control and edit its properties on the Properties pane's **General** tab.

Field	Description
Name	An internal identifier that you will use to identify the control.
Binding	Defines a link between the control and a data attribute.
Columns	In the Columns field, click Add  to add columns. Edit the labels of each column in their respective label fields. To hide a particular column from users, click Hide  beside its label field; you may require to hide a column initially and show it on occurrence of an event. Click the Delete icon to delete a column.
Users can Add/Remove Rows	Allows users to add or remove rows to the table.

Field	Description
Multiple Selection	<p>Allows users to select multiple rows of the table.</p> <p>While using a table with multiple-selection enabled to define dynamic behavior or computed values, the options available (specific to multiple selection) in the Event or Computed Value window under the Which? field are listed here:</p> <ul style="list-style-type: none"> • For Each Selected: Use to retrieve a value from another control to apply to each selected row of a table. • All Selected: Use to retrieve values from all selected rows of a table to apply to another control. <p>Similarly, in a table with two columns, you can use these options to apply values from one column to another on occurrence of an event. In this case, when you chose to apply an action to each selected row (by selecting the For Each Selected option), an additional option is available to chose the value source:</p> <ul style="list-style-type: none"> • Current Iteration Row: Use to retrieve the value from one cell of the current selected row and apply to another cell in the same row.
Use Data from Connector	<p>Allows you to populate the table from a REST connector defined for the application.</p> <p>Specify the connector settings in the Connector, Resource and Operation fields and map response settings. See Work with Connectors.</p>
Events	<p>Specifies events that trigger actions and conditions. You can customize the control's behavior by configuring events. See Add Dynamic Behavior to Forms.</p>

 **Note:**

While configuring events for a table, you can now add a condition based on number of rows present within the table using the **Row Count** property.

3. On the **Styling** tab, edit the control's [styling properties](#). You can specify column width in several ways:
 - Leave the **Automatic column size** checkbox selected so the table's columns automatically resize to fit the device on which the form is viewed.
 - Deselect the **Automatic column size** checkbox and specify column sizes for different device sizes by entering a number from 1 to 12 in the column size fields that display for different devices.
 - In the **Table Columns Width** fields, specify an absolute width for each column such as 2in (inches), 5cm (centimeters), 100px (pixels), or 25% (percent). If the column widths together exceed the table width, a scroll bar displays, unless columns are defined with percentages. If percentages are specified but exceed 100%, column widths are displayed proportionally across the table.
4. Drag and drop individual controls from the Basic or Advanced Palette into the columns. Each column can have one control.
5. Configure general and styling properties for the controls inside your table control. See [Configure Basic Controls](#) and [Configure Advanced Controls](#).

 **Note:**

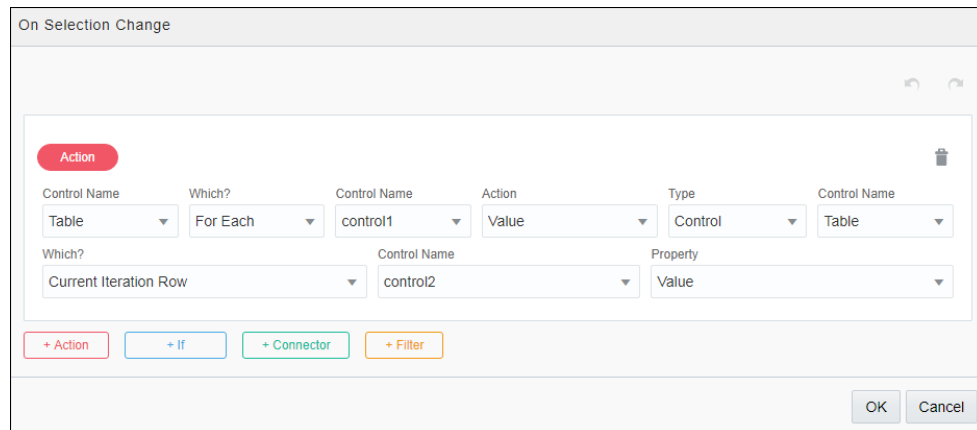
If you delete the table control, all the controls grouped under it get deleted.

- Click **Preview** to try out using the control.

Some Useful Event Actions for Tables

This section lists a few useful actions that you can apply to a table through the event window. To create an event action, see [Specify Actions](#).

- To retrieve a value from another control and apply it to each row of the table, use the **For Each** option under the **Which?** field.
- In a table with two columns, to copy values from one column to another (for each row) on occurrence of an event, use the **For Each** option along with an additional option called **Current Iteration Row** that is available to choose the value source. The **Current Iteration Row** option allows you to retrieve the value from one cell of a row and apply to another cell in the same row. The following figure shows an event action configuration for a table with two columns. The value of **control2** (the control in table column 2) is copied into **control1** (the control in table column 1) for each row of the table on occurrence of the specified event:



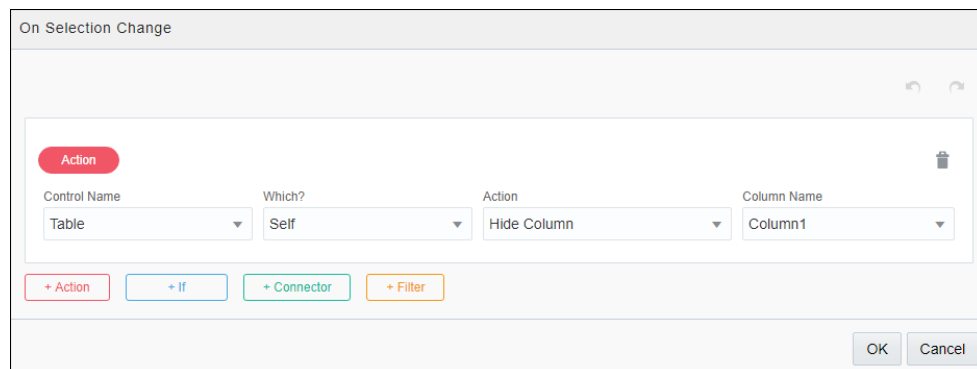
The screenshot shows the 'On Selection Change' event window. The 'Action' tab is active. The configuration is as follows:

Control Name	Which?	Control Name	Action	Type	Control Name
Table	For Each	control1	Value	Control	Table

Below this, the 'Which?' field is set to 'Current Iteration Row', the 'Control Name' is 'control2', and the 'Property' is 'Value'.

Buttons at the bottom include '+ Action', '+ If', '+ Connector', '+ Filter', 'OK', and 'Cancel'.

- To hide a column within a table on occurrence of an event, select **Self** under the **Which?** field, choose **Hide Column** under **Action**, and specify the column to hide. The following figure shows an event action configuration to hide a table column:



The screenshot shows the 'On Selection Change' event window. The 'Action' tab is active. The configuration is as follows:

Control Name	Which?	Action	Column Name
Table	Self	Hide Column	Column1

Buttons at the bottom include '+ Action', '+ If', '+ Connector', '+ Filter', 'OK', and 'Cancel'.

- Similarly, to show a previously hidden column—hidden either through Properties pane's **General** tab or through an event action, select **Self** under the **Which?** field, choose **Show Column** under **Action**, and specify the column to show.

Configure Repeatable Sections

Use repeatable section controls to display multiple copies of a set of controls in your web form. You can use repeatable section controls to create dynamic content for your web form.

To configure a repeatable section control:

1. From the Advanced Palette, drag and drop a repeatable section control onto the canvas.
2. Select the control and edit its properties on the Properties pane's **General** tab.

Field	Description
Name	An internal identifier that you will use to identify the control.
Label	Specifies the control name that displays to a user.
Binding	Defines a link between the control and a data attribute.
Label Binding	Allows you to define a dynamic label. To do this, you can use a data attribute listed in the Data definition pane.
Users can Add/Remove Rows	Allows users to add or remove rows to the repeatable section.
Multiple Selection	<p>Allows users to select multiple rows of the repeatable section.</p> <p>While using a repeatable section with multiple-selection enabled to define dynamic behaviors or computed values, the options available (specific to multiple selection) in the Event or Computed Value window under the Which? field are listed here:</p> <ul style="list-style-type: none"> • For Each Selected: Use to retrieve a value from another control to apply to each selected row of a repeatable section. • All Selected: Use to retrieve values from all selected rows of a repeatable section to apply to another control. <p>Similarly, in a repeatable section with two controls in a row, you can use these options to apply values from one control to another on occurrence of an event. In this case, when you chose to apply an action to each selected row (by selecting the For Each Selected option), an additional option is available to chose the value source:</p> <ul style="list-style-type: none"> • Current Iteration Row: Use to retrieve the value from one control of the current selected row and apply to another control in the same row.
Use Data from Connector	<p>Allows you to populate the repeatable section from a REST connector defined for the application.</p> <p>Specify the connector settings in the Connector, Resource and Operation fields and map response settings. See Work with Connectors.</p>
Events	Specifies events that trigger actions and conditions. You can customize the repeatable section control's behavior by configuring events. See Add Dynamic Behavior to Forms .

Note:

While configuring events for a repeatable section, you can now add a condition based on number of rows present within the section using the **Row Count** property.

3. On the **Styling** tab, edit the control's [styling properties](#).
4. Drag and drop individual controls from the Basic or Advanced Palette into the repeatable section control.
5. Configure general and styling properties for the controls inside your repeatable section control.

See [Configure Basic Controls](#) and [Configure Advanced Controls](#).

 **Note:**

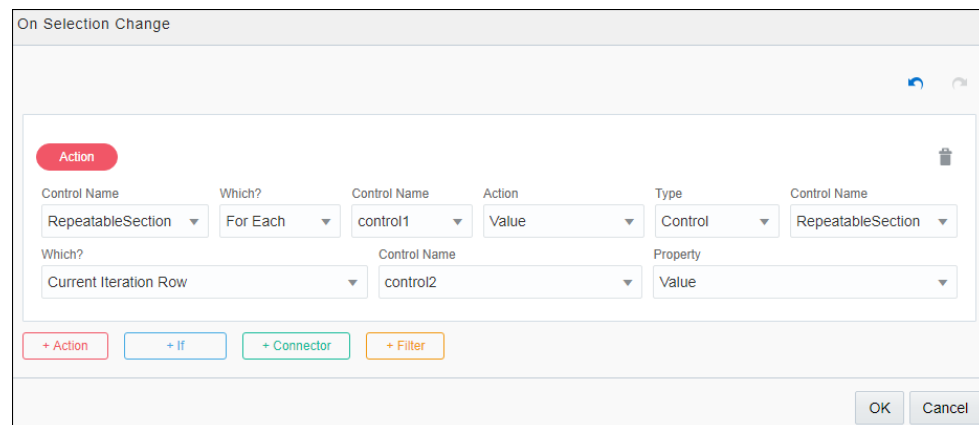
If you delete the repeatable section control, all the controls grouped under it get deleted.

6. Click **Preview** to try out using the control.

Some Useful Event Actions for Repeatable Sections

This section lists a few useful actions that you can apply to a repeatable section through the event window. To create an event action, see [Specify Actions](#).

- To retrieve a value from another control and apply it to each row of the repeatable section, use the **For Each** option under the **Which?** field.
- In a repeatable section with two controls in a row, to copy values from one control to another (for each row) on occurrence of an event, use the **For Each** option along with an additional option called **Current Iteration Row** that is available to choose the value source. The **Current Iteration Row** option allows you to retrieve the value from one control of a row and apply to another control in the same row. The following figure shows an event action configuration for a repeatable section with two controls in a row. The value of **control2** is copied into **control1** for each row of the repeatable section on occurrence of the specified event:



The screenshot shows a configuration window titled "On Selection Change". It contains an "Action" configuration table with the following details:

Control Name	Which?	Control Name	Action	Type	Control Name
RepeatableSection	For Each	control1	Value	Control	RepeatableSection

Below the table, there is a secondary configuration:

Which?	Control Name	Property
Current Iteration Row	control2	Value

At the bottom of the window, there are buttons for "+ Action", "+ if", "+ Connector", "+ Filter", "OK", and "Cancel".

Configure Train Controls

If you have a multistep web form activity that users should approach in a particular sequence, then add a Train control to guide them through this activity. The Train control displays the

number of steps a user must complete before submitting the form, and it also indicates the user's current place within a multistep activity.

In a multistep form (form with a set of presentations), you can represent each step using a train stop. In other words, you can map each presentation to a stop. Users can navigate between steps by clicking on the train stops; the current stop of the user is highlighted in blue and the visited stops bear a check mark. However, to display a different presentation for each stop, you must define an event with Change Presentation actions for the Train control.

To configure a Train control:

1. From the Advanced Palette, drag the **Train** control onto the canvas.
2. Select the control and edit its properties on the Properties pane's **General** tab.

Field	Description
Name	An internal identifier that you will use to identify the control.
Label	Specifies the control name that displays to a user.
Binding	Defines a link between the control and a data attribute.
Computed Value	Set this property to apply a computation to the control. See Create Computed Controls .
Options Source	Select a source (Static, From Data, and Connector). <ul style="list-style-type: none"> • Static: Specify choices using Options Names and Options Values fields. Use Options Names to specify the label to display for an option and use Options Values to specify an internal value for an option. • From Data: In the Options List field, select a list of values options source from the data definitions available in the web form. If you selected a list of complex elements, then, in the Label Binding field, specify a data attribute that will display as the label and in the Value Binding field, specify a data attribute that will be the value. • Connector: Specify a REST connector, a resource, and an operation to use. Specify parameters to pass to the connector and define how the response should be mapped to the control properties. See Work with Connectors.
Default Value	If you selected Static in the Options Source field, then, specify a default option in this field. If you selected From Data or Connector in the Options Source field, then, select either the first or the last value as the default value.
Events	Allows you to configure events for the control. You can assign actions or If/Then/Else conditions to the control based on a selected event. See Add Dynamic Behavior to Forms .

3. On the **Styling** tab, edit the control's [styling properties](#).
4. Click **Preview** to try out using the control.

 **Note:**

- You can add a maximum of 15 stops for your form using the Train control. When the form loads, users can scroll to navigate the stops or the control can be designed to fit into a single view by selecting **Stretch** from the Properties pane's **General** tab.
- To avoid redesigning the Train control each time you make changes (especially, addition or deletion of presentations) to the multistep form, add the control after you've completely configured the web form and all of its steps.

Configure Divider Controls

Use a divider control to separate related content within your form or to create logical breaks in your form's layout. For example, you might want to break up your form into sections or separate a related group of controls from the rest in the form.

To configure a divider control:

1. From the Advanced Palette, drag the **Divider** control onto the canvas.
2. Select the control.
3. In the Properties pane, select the **General** tab and edit the properties.

Field	Description
Name	An internal identifier that you use to identify the control.
Hide	Set this property to hide the control. For example, you might hide a control by default, but configure another control that, when selected, triggers an event that displays the hidden control.
Events	Allows you to configure events for the control. You can assign actions or If/Then/Else conditions to the control based on a selected event. See Add Dynamic Behavior to Forms .

4. On the **Styling** tab, edit the control's [styling properties](#).
 - To specify the color of the divider control, use the **Stroke Color** property.
 - To specify the width of the divider control, use the **Stroke Width** property. You can enter standard values such as 0.05in, 0.2em, or 5px.
5. To try out the control, click **Preview**.

Configure File Manager Controls

Use a file manager control to upload multiple files to an external document store and list the uploaded files in your web form.

The file manager control has two controls - *file upload* and *file list*. If you want to delete either the file upload or file list control, click the delete icon on the top-right of the control. But keep in mind that deleting the file upload or file list control also deletes all the associated configurations, which will not be restored when you add the control back to the file manager control.

Note that you *cannot* do the following in a file manager control:

- Move the file upload or file list control outside the file manager control in your web form.
- Add any other control inside the file manager control.

To configure file manager control:

1. From the Advanced Palette, drag the **File Manager** control onto the canvas.
2. Select the control and edit its properties on the Properties pane's **General** tab. Notice that the **Container** tab is selected by default for the file manager control in the Properties pane.

Field	Description
Name	An internal identifier that you will use to identify the control.
Label	The title of the file manager control that the user will see in the form.
Description	For a file manager control, provides additional information or instructions for the user.
Hide	Set this property to hide the control.
Connector and Resource	Allows you to choose the connector that uploads the files to the external document store. After choosing the Connector , choose the corresponding Resource . The connector and resource chosen here will apply for both file upload and file list controls. See Work with Connectors .
Events	Allows you to configure On Load event for the control. The event is triggered when the file manager control gets loaded. See Add Dynamic Behavior to Forms .

3. If you want to configure the properties of the file upload control, see [Configure File Upload Control](#).
4. If you want to configure the properties of the file list control, see [Configure File List Control](#).
5. On the **Styling** tab, edit the control's [styling properties](#).
6. Click **Preview** to try out using the control.

Configure File Upload Control

A file upload control allows users to upload multiple files to an external document store. The base64 string of the file content is uploaded via a POST API call.

Since the files are controlled by external APIs and are not stored in the process, the file upload control will not perform any anti-virus scans or checks.

Note:

- The file upload control is a part of the file manager control. It cannot be added separately outside the file manager control in the web form.
- Binary file upload is not supported.
- The maximum aggregate file size allowed is 2 MB.
- You can upload up to 5 files simultaneously.

To configure file upload control:

1. Select the file upload control and edit its properties on the Properties pane's **General** tab. Notice that the **Upload** tab is selected by default for the file upload control in the Properties pane.

Field	Description
Name	An internal identifier that you will use to identify the control.
Label	The title of the file upload control that the user will see in the form.
Accept only these file types	Defines the accepted file types as comma-separated values. If no file types are specified, the control accepts all the files.
Maximum Upload Size	Defines the maximum file size allowed in megabytes. The default value is 2 MB.
Help	Help text that will appear for the control. If you specify help text, a help icon appears next to the name of the control and displays the help text when a user hovers over it or clicks it.
Primary Text	Text that will appear in the drop-zone of the file picker.
Secondary Text	Subtext that will appear in the drop-zone of the file picker.
Disabled	Set this property to display the control as inactive.
Hide	Set this property to hide the control.
Auto Focus	Set this property to automatically select the control when the web form loads.
Events	Allows you to configure events for the control. You can choose one of the following events: <ul style="list-style-type: none"> • On Load: The event is triggered when file upload control gets loaded. • On Upload: The event is triggered when file is uploaded successfully. • On Error: The event is triggered when file upload is unsuccessful.

 **Note:**

- The **Last Uploaded File Name** property of the file upload control allows you to get the name of the most recently uploaded file while configuring events and actions.
- The **Upload Response** property stores the result of the **On Upload** event. You can use the same to auto populate values for other controls in the form.


See [Add Dynamic Behavior to Forms](#).

2. On the **Binding** tab, update the following properties.

 **Note:**

If you haven't configured the connector and resource for the file manager control, click **Go to Container tab** and configure the same. See [Configure File Manager Controls](#).

Field	Description
Operation	Choose the file upload control's operation based on the connector and resource you specified when configuring the file manager control.

Field	Description
Supports multiple uploads	Set this property to allow users to upload multiple files simultaneously.
<div style="border-left: 2px solid #0070C0; padding-left: 10px; margin: 10px 0;"> <p> Note:</p> <p>Please note that the API must also support multiple file uploads and have one or more array type request body parameters. If this property is set and the API does not support array type request body parameters, only the first file from the list will be uploaded.</p> </div>	
Parameters (Header Parameters, Template Parameters, Request Body)	Configure the values for header parameters, template parameters, and the request body based on the operation you specified.
Element with list of files	If the Supports multiple uploads property is set, you must select the connector element in which the user can upload multiple files. You can choose from the array type request body parameters that are listed in the drop-down. Further, configure the sub-properties of the selected field.

3. On the **Styling** tab, edit the control's [styling properties](#).
4. Click **Preview** to try out using the control.

Configure File List Control

The file list control allows users to see the list of documents that are successfully uploaded to the external document store.

To configure file list control:

1. Select the file list control and edit its properties on the Properties pane's **General** tab. Notice that the **List** tab is selected by default for a file list control in the Properties pane.

Field	Description
Name	An internal identifier that you will use to identify the file list control.
Label	The title of the file list control that the user will see in the form.
Help	Help text that will appear for the control. If you specify help text, a help icon appears next to the name of the control and displays the help text when a user hovers over it or clicks it.
Empty State Message	The text value that is displayed in the form when there are no files to list.
Disabled	Set this property to display the control as inactive.
Hide	Set this property to hide the control.
Read Only	Set this property to disable edit permissions.
Show Refresh Button	Set this property to display refresh button that refreshes the file list connector on click.

Field	Description
Events	<p>Allows you to configure events for the control. You can choose one of the following events:</p> <ul style="list-style-type: none"> • On Load: The event is triggered when file list control gets loaded. • On List Refresh: The event is triggered when the file list is refreshed. • On Download: The event is triggered when a file is downloaded successfully. • On Delete: The event is triggered when a file is deleted successfully. • On Download Error: The event is triggered when a file download is unsuccessful. • On Delete Error: The event is triggered when a file delete is unsuccessful. <p>See Add Dynamic Behavior to Forms.</p>

2. On the **Binding** tab, update the following properties.

Field	Description
Operation, Header Parameters, and Template Parameters	Choose the file list control's operation based on the connector and resource you specified when configuring the file manager control. Similarly, configure the values for header parameters, and template parameters.
Element that contains the list of files	Select the file attributes to be displayed in the list. Further, configure the sub-properties of the selected field.
Enable Download	Set this property to allow users to download a file from the file list by clicking the file name.
Operation, Header Parameters, and Template Parameters	Choose the file download control's operation based on the connector and resource you specified when configuring the file manager control. Similarly, configure the values for header parameters, and template parameters.
Enable Delete	Set this property to display delete button that allows the users to delete a file from the file list on click.
Show Confirmation Message	Set this property to display a confirmation dialog before deleting a file from the file list.
Operation, Header Parameters, and Template Parameters	Choose the file delete control's operation based on the connector and resource you specified when configuring the file manager control. Similarly, configure the values for header parameters, and template parameters.

3. On the **Styling** tab, edit the control's [styling properties](#).
4. Click **Preview** to try out using the control.

Implement Intelligent Document Processing in Forms

Intelligent document processing applies artificial intelligence (AI) and machine learning techniques to capture, extract, and process structured, semi-structured, and unstructured data from a variety of document formats. In Process Automation forms, intelligent document processing is implemented with the **Document Understanding** control that uses out of the box pretrained AI models from Oracle Cloud Infrastructure (OCI) Document Understanding AI service to automatically detect, classify and extract texts and objects from uploaded documents.

To know more about OCI Document Understanding AI service, see [Oracle Cloud Infrastructure Document Understanding documentation](#).

Process Automation uses two pretrained AI models from OCI Document Understanding AI service: Document Classification and Key Value Extraction.

- The **Document Classification** AI model is used to classify the document uploaded to the document understanding control. See [Document Classification](#).
- The **Key Value Extraction** AI model is used to extract values for predefined key value pairs in the uploaded document. For example, if a passport includes nationality and date of issue, Document Understanding can identify these values and return them as a key value pair. See [Key Value Extraction](#).

 **Note:**

Process Automation is preregistered with OCI Document Understanding AI service for tenancies in the OC1 realm.

To learn how to configure the document understanding control in forms, see [Configure Document Understanding Controls](#).

Configure Document Understanding Controls

Use the **Document Understanding** control to implement intelligent document processing in forms to automatically detect, classify, and extract texts and objects from files uploaded to it. The extracted text can be used for process routing, approvals or can be sent to a downstream system.

 **Note:**

The document understanding control is available in the Forms palette *only* if your tenancy is in the OC1 realm. To know more about Oracle Cloud Infrastructure realms, see [Regions and Availability Domains](#) in *Oracle Cloud Infrastructure Documentation*.

To configure a document understanding control:

1. In the Forms palette, expand AI Services and then drag and drop a document understanding control onto the canvas.
2. Select the control and edit its properties on the Properties pane's **General** tab.

Field	Description
Name	An internal identifier that you will use to identify the control.
Label	The title of the document understanding control that the user will see in the form.
Accept only these file types	Defines the accepted file types as comma-separated values. If no file types are specified, the control accepts all the files.
Maximum Upload Size	Defines the maximum file size allowed in megabytes. The default value is 2 MB which is also the maximum upload size allowed.
Help	Help text that will appear for the control. If you specify help text, a help icon appears next to the name of the control and displays the help text when a user hovers over it or clicks it.
Primary Text	Text that will appear in the drop-zone of the file picker.
Secondary Text	Subtext that will appear in the drop-zone of the file picker.

Field	Description
Disabled	Set this property to display the control as inactive.
Hide	Set this property to hide the control.
Events	<p>Allows you to configure events for the control. You can choose one of the following events:</p> <ul style="list-style-type: none"> • On Load: The event is triggered when the document understanding control gets loaded. • On Upload: The event is triggered when the file is uploaded successfully. • On Error: The event is triggered when file upload or detection/ extraction of text in the uploaded file are unsuccessful. <p>You can assign actions or If/Then/Else conditions to the control based on a selected event. See Add Dynamic Behavior to Forms.</p>

 **Note:**

- The **Last Uploaded File Name** property of the document understanding control allows you to get the name of the most recently uploaded file while configuring events and actions.
- The **Last Uploaded File Content** property of the document understanding control allows you to get the content of the most recently uploaded file while configuring events and actions.

3. On the **Document** tab, edit the following properties.

Field	Description
Document type	<p>Specify a document type from the drop-down list. The following document types are supported:</p> <ul style="list-style-type: none"> • Driver license • Passport • Receipt • Invoice
Create data definition	<p>Select this property to automatically create data definitions for the selected document type.</p> <p>For example, if you chose Passport as the document type then on selecting Create data definition, a <code>passport</code> data object is created.</p> <p>The data object and data definitions are displayed under the Data section. In this section, the document object consists of document and fields sub-objects. The document sub-object is a document reference type and consists of <code>documentId</code> and <code>documentName</code> attributes. You can use these attributes to reference the document.</p> <p>The fields sub-object contains the related data definitions such as <code>firstName</code>, <code>lastName</code>, <code>country</code>, <code>nationality</code> and so on that are automatically created.</p>
Generate controls	<p>Select this property to automatically generate controls for the created data definitions.</p> <p>The generated controls are like any other form controls. If required, you can change their position, edit, or delete them.</p>

Field	Description
Minimum confidence score	Either enter a value, or set a specific value between 0 and 100 using the increment/decrement arrows in the field. Note that higher the confidence score, higher is the chance of the AI model returning accurate results. For example, if you set the minimum confidence score as 96, then the pretrained models must return results that are at least 96 percent accurate.
Show warnings for low confidence score fields	Select this property to show warning messages for fields with confidence scores that are lower than the minimum threshold.
Hide fields with no data	Select this property to hide empty fields.
Hide image preview	Select this property to hide the preview of the uploaded document's image.
Hide document upload	Select this property to hide the document upload.
Document display size	Set this property to specify if the image rendered for the uploaded document is Small, Medium or Large.
Persist document reference in payload	Select this property to include the <code>documentId</code> and <code>documentName</code> details in the payload. These document details are used to reference the document in subsequent user tasks or business flows.

 **Note:**

This checkbox does not appear for data definition schemas that do not contain the document reference attributes.

Download document on demand	Select this property if you want the user to click on image to download and display the document. Use this option in scenarios where approvers want to see images on demand, or if there are multiple images and the user wants to avoid longer load times.
Regenerate data definition	This property appears when you import an application that uses an older schema that does not contain document referencing attributes. Click this link if you want to regenerate the data definition schema so that the new schema includes document reference information.

 **Note:**

Regenerating is an irreversible action and may introduce breaking changes.

Messages	Expand this section to view the default messages that will be displayed for the following: <ul style="list-style-type: none"> • Success message • Failure message • When uploaded document is of different type • When uploaded document cannot be recognized • When confidence score is too low <p>If required, you can edit the default messages and create messages customized to your business needs.</p>
----------	--

4. On the **Styling** tab, edit the control's [styling properties](#).
5. Click **Preview** to try out using the control.

Reuse Forms

You can reuse forms from the Forms Palette by embedding them in other forms that you build. Once you have embedded a form within another form, you can make changes to the form like styling changes or remove unwanted fields. You can also make the reused form as a read only form.



Note:

Process Automation does not allow you to:

- Nest forms beyond three levels.
- Deploy a project if there is cyclic dependency among the forms.

To reuse forms:

1. In the Forms Palette, expand Forms.

If the form that you want to reuse doesn't appear, search for it. Click the search icon and type the form's name in the search field. If no forms in your application match the search, you will get a message stating this. Otherwise, the form gets loaded into the Forms Palette.

2. Drag and drop the required form onto the canvas.

You can drop the form onto any part of the canvas. The layout of the reused form is preserved but depending on where you drop the form some panels may get added to it to maintain the form's layout.

3. Make styling and label changes to the reused form if required.
4. Optionally, click **Detach** on top of the reused form.

When you detach an embedded (referenced) form on another form's UI, the controls corresponding to the embedded form appear as separate entities on the UI, but the data attributes related to these controls continue to be mapped under the data attribute of the embedded form. However, if you add a new control to the detached embedded form's section, the new control and its data attribute are treated as a part of the referencing form. Additionally, once an embedded form is detached on another form's UI, new controls added at the source of the embedded form aren't reflected on the referencing form's UI. However, the data attribute of the embedded form reflects this change in the referencing form's data definition section. The referencing form and its associated process task use this updated data attribute.

Detaching can be helpful when you want only certain parts and don't want some parts of an embedded form. You can delete the unwanted controls and reuse only the controls that you want in building your web form.

Work with Presentations

A presentation is a single view of the form.

Presentations are useful:

- When forms are rendered in different devices
- To present different views of the same data for specific users


When a form is created, it uses a default presentation. When you add a control to the form, the new control is added to the currently selected presentation.

Selecting, adding, or deleting a presentation in the web forms editor

1. On the **Form** tab, notice that the **Presentations** table lists the available presentations. When a form is created, it uses the **Main** presentation by default.

The **Presentation** tab displays the properties of the currently selected presentation. An asterisk (*) indicates a required property. You can edit the properties of the currently selected presentation and specify new values:

Field	Description
Name	Specify a name for the presentation.
Description	Specify a suitable description for the presentation.
Border Color	Select a border color from the color palette or enter a hexadecimal value in the adjacent text field.
Border Width	Specify a border width in pixels or points. For example, 9px.
Border Style	Select a border style (Solid, Dotted, or Dashed) for the presentation.
Background Color	Specify a background color for the presentation. Select a background color from the color palette or enter a hexadecimal value in the adjacent text field.
Events	Configure events for the presentation. See Add Dynamic Behavior to Forms .
Reusable Snippets	<p>Create a global event snippet to reuse across the form's presentation. Use any event block, such as action, condition, or connector, to define a snippet.</p> <p>In addition, all the event snippets you've extracted while defining events for form controls appear here. See Extract a Snippet. Click Edit to view or update a previously extracted snippet. Changes you make at the presentation level reflect in all instances of the snippet across different controls.</p> <p>Note that deleting a global snippet converts all usages of the snippet into local copies.</p>

2. To add a presentation, click **Create Presentation**  in the **Presentations** table of the **Form** tab, and make an appropriate selection in the Select Presentation Type dialog box.
 - **From Scratch:** Select to create a fresh presentation. In the Create Presentation window, enter a name for the presentation and an optional description. Select **Make default** to make this presentation the default presentation. Select **Switch to this presentation** to switch to this new presentation immediately after creation. Click **Create**.
 - **Clone from Previous Presentation:** Select to create a presentation based on an existing presentation. In the resulting window, select an existing presentation from the list.
 - **Customize from Previous Presentation:** Select to create a customized presentation based on an existing presentation. In the resulting window, select an existing presentation from the list.

 **Note:**

A customized presentation is identical to the base presentation from which it is created. However, unlike a cloned presentation, you cannot add/delete controls or modify control properties within a customized presentation. You can only perform the followings actions on existing controls: hide, show, and modify read-only status. To edit a customized presentation after creating it, use the **Edit** button next to the presentation switcher.

3. To delete an existing presentation, click the **Delete Presentation** icon in the **Presentations** table. If you try to delete the default presentation, the Delete Default Presentation Confirmation dialog box appears. Select a new default presentation from the list. Click **Delete**.
4. To change the default presentation, in the **Presentations** table, select the **Make this the default presentation** icon next to the presentation you want to change to the default.

Notice that you can switch to a presentation using the drop-down list of available presentations on the top of the canvas.

Selecting the Right Presentation Type

This section lists all the differences between the available presentation types and helps you make the right selection.

Scratch	Clone	Customize
Created with no controls.	Created by copying controls from the base presentation.	Created by linking to controls in the base presentation.
Adding or editing new controls is allowed.	Adding or editing new controls is allowed.	Adding or editing new controls isn't allowed. Only visibility of the controls can be edited.
No base presentation is used.	<ul style="list-style-type: none"> • A base presentation is used; however, deleting the base presentation doesn't affect the cloned presentation. • In addition, a cloned presentation is not in sync with its base presentation. 	<ul style="list-style-type: none"> • A base presentation is used and deleting the base presentation deletes all the customized presentations created using it. • A customized presentation is always in sync with its base presentation.
Used to design a new presentation.	Used to create an independent presentation with all content from a previous presentation.	Used to create a presentation that is linked and identical to a previous presentation, which only allows for modifying the visibility of controls.

Some Useful Event Actions for Presentations

This section lists a few useful actions that you can apply to a presentation through the event window. To create an event action, see [specifying actions](#).

- Use the **Set Data** action to set the value of a particular data attribute used in the form. The value is added to the payload. When the event executes, all controls using this data attribute, either directly or indirectly (through the Computed Value option), are updated with this value.

- Use the **Change Presentation** action to switch the presentation of a form or an embedded form on occurrence of an event.
- Use the **Print** action to print a presentation of a form on occurrence of an event. For example, you can define an **On Click** event with the **Print** action for a Button control.
- Use the following actions to manipulate an array or its elements:
 - Use the **Array Value** action to set the value of a particular element in an array on occurrence of an event. Specify the array, index of the element, and value to set—in that order—in the subsequent fields of the event window. The following image shows a value of **option1** set to an element with index **0** in the array named **checklist**.

Control Name	Action	Value
Presentation	Array Value	checklist
Type	Value	Type
Constant	0	Constant
Value	option1	

- Use the **Insert Array Value** action to insert an element into an array. Specify the array, the index at which to insert the element, and the value to insert—in that order—in the subsequent fields of the event window.
- Use the **Append Array Value** action to insert an element at the end of an array. Specify the array and the element to append in the subsequent fields of the event window.
- Use the **Clean Array** action to clear an entire array. Specify the array in the **Value** field.
- Use the **Delete Array Value** action to delete a particular element in an array. Specify the array and the index of the element to delete in the subsequent fields of the event window.

Change Form Presentations Dynamically

Let's consider an example of a travel form to understand how you can use the **Change Presentation** action to dynamically switch a form's presentation in runtime.

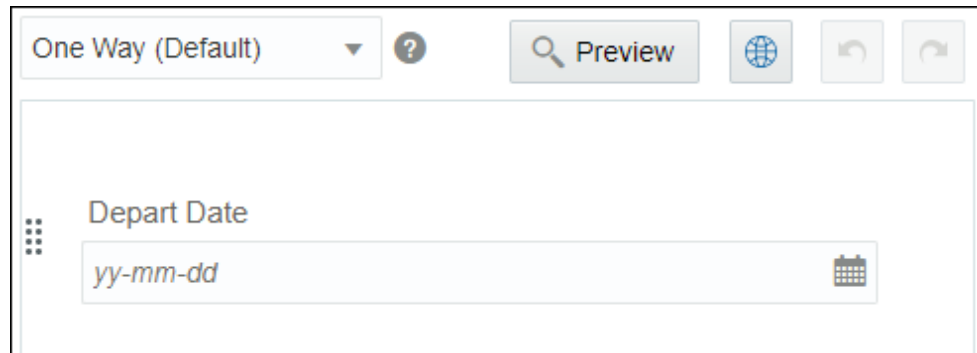
Note:

You can also use dynamic data, such as values entered in controls, payload data, or data from connectors to trigger the Change Presentation action.

Here, we create a simple travel form in which a user enters details, such as name, origin, destination, and so on. The user also mentions whether the intended journey is a one-way trip or a round trip using radio buttons, and the date (departure and return) fields are displayed according to the radio button selection. The date fields are supplied through an embedded web form.

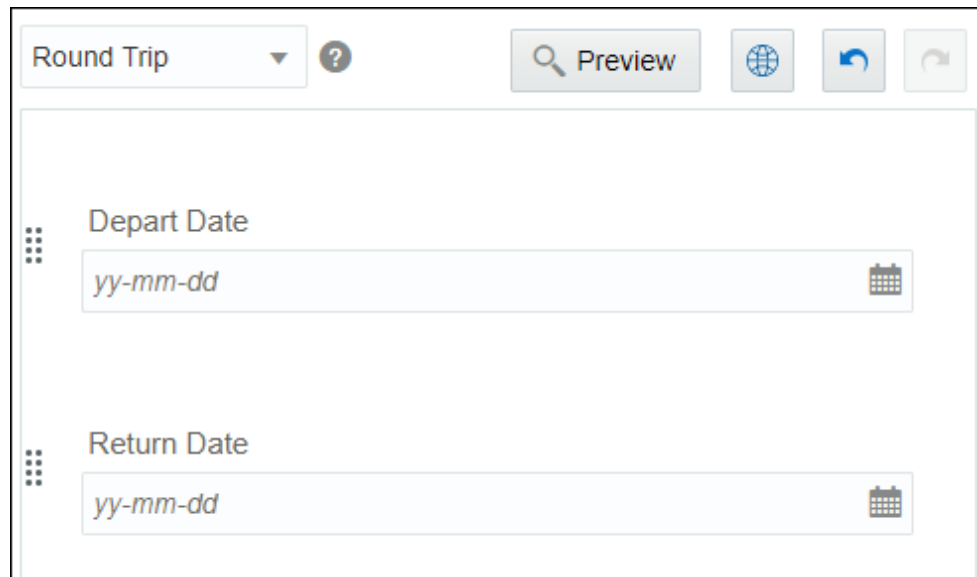
We start by creating this sub-form with date fields:

1. Within the form, create a presentation called **One Way**, make it default, and add a date control named **Depart Date** to it.



The screenshot shows a web form titled "One Way (Default)". At the top, there is a dropdown menu with "One Way (Default)" selected, a help icon (question mark), a "Preview" button, a globe icon, and two circular arrows. Below the title bar, there is a "Depart Date" label followed by a text input field containing the placeholder text "yy-mm-dd" and a calendar icon on the right side.

2. Create another presentation called **Round Trip** and add two date controls as follows.



The screenshot shows a web form titled "Round Trip". At the top, there is a dropdown menu with "Round Trip" selected, a help icon (question mark), a "Preview" button, a globe icon, and two circular arrows. Below the title bar, there are two date controls. The first is labeled "Depart Date" and the second is labeled "Return Date". Each label is followed by a text input field containing the placeholder text "yy-mm-dd" and a calendar icon on the right side.

3. Now, create the main travel form with the following controls:
 - a. Input text controls for name, origin, and destination.
 - b. Radio buttons to specify if it's a one-way journey or round trip.
 - c. Embed the web form created in the previous steps to provide the date controls.

The screenshot shows a web form with the following elements:

- Full Name**: A text input field.
- Origin City**: A text input field.
- Destination City**: A text input field.
- One Way**: A radio button that is selected (indicated by a blue dot).
- Round Trip**: A radio button that is not selected.
- Detach**: A blue text link located to the right of the radio buttons.
- Depart Date**: A date picker control with a calendar icon and the placeholder text `yy-mm-dd`.

4. Set up option names and values for the radio button control as follows; specify the **One Way** option as default.

The screenshot shows the configuration panel for the radio button control. It includes the following settings:

- Options Source**: A dropdown menu with a question mark icon, currently set to **Static**.
- Options Names**: A list box containing the text `One Way` and `Round Trip`.
- Options Values**: A list box containing the text `One Way` and `Round Trip`.

5. Now, define an **On Change** event for the radio button control to change the presentation of the inner web form according to the user's selection. Use the **Change Presentation** action available for the inner form as shown in the following figure:

On Change

If + Condition

Type: Control Control Name: RadioButton Property: Value

is equal to

Type: Constant Value: Round Trip

Then + Action

Action

Control Name: InnerForm Action: Change Presentation Presentation: Round Trip

Else If + Condition

Type: Control Control Name: RadioButton Property: Value

is equal to

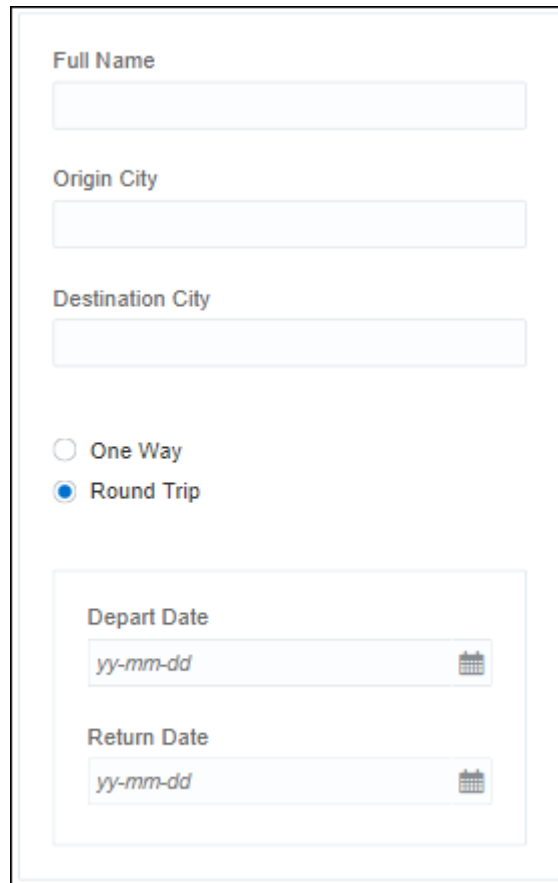
Type: Constant Value: One Way

Then + Action

Action

Control Name: InnerForm Action: Change Presentation Presentation: One Way

6. Click **Preview** to test the travel form. When the form loads, the **One Way** option is selected and the default presentation of the inner form loads with only the depart date control.
7. Populate all the controls and select **Round Trip**. The inner form's presentation changes to display two date controls; if you change your selection back to **One Way**, the embedded form displays its default presentation.



The screenshot shows a web form with the following elements:

- Full Name**: A text input field.
- Origin City**: A text input field.
- Destination City**: A text input field.
- One Way**: A radio button.
- Round Trip**: A radio button, which is selected.
- Depart Date**: A date input field with a calendar icon and the placeholder text `yy-mm-dd`.
- Return Date**: A date input field with a calendar icon and the placeholder text `yy-mm-dd`.

Work with Styling

Styling enables you to change a selected control's appearance. Use styling to define display-specific properties of a currently selected control in your web form.

Changing the Style of a Form's Control

To change the styling of a control:

1. Select the control for which you want to change the styling. The Properties pane changes and displays the **General** and **Styling** tabs.
2. Select the **Styling** tab. You can see a number of styling options displayed for that particular control.
3. Select the styling property you want to apply to your control and specify the styling.

See [Styling Properties](#).

Styling Properties

You can change a control's appearance by defining its styling properties on the **Styling** tab in the Properties pane. Use the **Styling** tab to define display-specific properties of the currently selected control.

The following table provides an alphabetical list of the properties available on the **Styling** tab.

**Note:**

Styling properties are control-specific. Not all of the properties listed below are available for every control.

Property	Description
Automatic column size	<p>Calculates the column size for the control based on the amount of visible controls in the row. Automatic column size is selected by default.</p> <p>Note: In small devices like a phone, each control is displayed in one row when automatic column size is enabled.</p> <p>You can specify absolute column sizes for different device sizes. To do this, deselect Automatic column size and enter a number from 1 to 12 in the four available options: Small, Medium, Large, and Extra large column sizes.</p>
Background Color	Specifies the background color of the content area in a control.
Border (Color, Width, Style, Radius)	<p>Determines the appearance of the border in the content area of your control.</p> <ul style="list-style-type: none"> • Border Color: Defines the color of the border. • Border Width: Defines the width of the border. Use standard values such as 1in, 5em, or 20px. • Border Style: Defines the style of the border—Solid, Dotted, or Dashed. • Border Radius: Defines the value of rounded border corners. Use standard values such as 1in, 5em, or 20px.
Color	Specifies the color of the text in the content area of a control.
Control Alignment	Specifies the alignment (left, right, or center) of a control in the form.
Font Size	Defines the font size of the text in the content area of a control. The available values are x-small, small, normal, large, and x-large.
Height	Sets an absolute height for the control. Use standard values such as 1in, 5em, or 20px.
Label Color	Specifies the color of a control's label.
Label Size	<p>Specifies the size of a control's label. The available values are x-small, small, normal, large, and x-large.</p> <p>Note: For a Panel control, using the Standard theme overrides the value in the Label Size property.</p>
Reset Inline Styles to Default	Discards all styling selections made on the Styling tab and restore settings to their default values.
Text Alignment	Specifies the alignment (left, right, or center) of the content in a control. This property applies to controls where the user can type in text such as the Input Text control.
Theme	<p>Uses a CSS to define the format of a control. This property applies to Panel and Section controls.</p> <p>For a Panel control:</p> <ul style="list-style-type: none"> • Standard: Automatically increases the size of the control's label to 24 pixels (font size) and makes the label bold (font weight). The Standard theme overrides the value in the Label Size property. • None: Applies no formatting to the control. None is the default value. <p>For a Section control:</p> <ul style="list-style-type: none"> • Indent: Automatically indents the section. With the Indent theme, you can easily nest sections within sections. • None: Applies no formatting to the control. None is the default value.
Width	Sets an absolute width for the control. Use standard values such as 5in, 20px, or 5%.

Work with CSS

Use cascading style sheets (CSS) to customize the look and feel of your web form elements. Modify visual attributes such as color, font, or margin, of an element using custom CSS code.

You can manage and control the appearance of the web form by applying an available CSS stylesheet to the form or by importing an existing CSS stylesheet.

While importing a CSS, Process Automation validates the stylesheet and highlights any possible errors. Process Automation allows you to make changes to the imported stylesheet in the CSS editor before creating it.

 **Note:**

Process Automation does not support javascript references in CSS. Oracle does not recommend using the CSS to position elements.

You can apply any uploaded CSS to a form. To apply a CSS to a form, select the required CSS and click **Apply**. Once you have applied a CSS to a form, you can preview the changes on the form. However, you cannot maintain more than 20 stylesheets in each process automation application. The stylesheets are specific to an application.

 **Note:**

The size of the stylesheet must not exceed 100 KB.

Apply a Stylesheet to a Form

To apply a stylesheet to a form, do the following steps:

1. In the form's properties pane, scroll down to the Stylesheet section.
2. Click the **Lookup stylesheet** icon.
The Css Browser page opens.
3. Click **Create** to upload a stylesheet that is stored on your computer.
The File Open dialog box opens.
4. Select the stylesheet that you want to apply from your computer and click **Open**.
5. The **Create stylesheet** editor appears and displays the code of the stylesheet.
Process automation validates the stylesheet and shows if there are any errors.
6. If the stylesheet is valid, click **Create**.
Process automation imports the stylesheet. The imported stylesheet is now available in the Css browser and can be applied to forms.
7. Select the **Form** tab in the Preview pane to preview your form with the selected stylesheet applied to it.
You can make changes to the code in the stylesheet and preview the change on the form.

Style Form Controls

You can style different components of a specific control using corresponding CSS selectors.

The best practice to styling a control is to define a control class name for it. The control class name serves to identify the control in the CSS file, and you can use this name as a class name and modify the control's selectors within that class.

The following table lists all customizable controls and associated selectors that you can style using CSS.

UI Control	Selectors	Description
Identity Browser	<code>.oj-opa-cc-identity-browser-container</code>	Specify a style for the Identity container.
	<code>.oj-opa-cc-identity-browser__input-search .oj-combobox-input</code>	Specify a style for the identity input.
	<code>.oj-opa-cc-identity-browser__input-search.oj-invalid .oj-combobox-input</code>	Specify a style for the invalid identity input.
	<code>.oj-opa-cc-identity-browser__input-search.oj-disabled .oj-combobox-input</code>	Specify a style for the identity input when the control is disabled.
	<code>.oj-combobox-open-icon</code>	Specify a style for the Drop-Down Icon container.
	<code>.oj-combobox-open-icon:before</code>	Specify a new signifier instead of the default Drop-Down icon/signifier.
	<code>.oj-opa-cc-identity-browser-container</code>	Specify a style for the Identity container when multiple selection is enabled.
	<code>.oj-opa-cc-identity-browser-container .oj-combobox-selected-choice</code>	Specify a style for selected options when multiple selection is enabled.
	<code>.oj-opa-cc-identity-browser-container .oj-combobox-selected-choice-label</code>	Specify a style for the selected option label.
Tab	<code>.tab-container-control</code>	Specify a style for the Tabs container.
	<code>.tab-container-control .oj-tabbar-item</code>	Specify a style for the tab header.
	<code>.tab-container-control .oj-tabbar-item.oj-selected</code>	Specify a style for the selected tab header.
	<code>.tab-container-control .oj-tabbar-item.oj-disabled</code>	Specify a style for the disabled tab header.
	<code>.tab-container-control .oj-tabbar-item-label</code>	Specify a style for the tab title.
Table	<code>.table-container .add-row</code>	Specify style for add button.
	<code>.table-container .remove-row</code>	Specify style for remove/delete button.
Repeatable section	<code>.repeatable-control-renderer-buttons .add-row</code>	Specify style for add button.
	<code>.repeatable-control-renderer-buttons .remove-row</code>	Specify style for remove/delete button.
File manager	<code>.file-upload-control</code>	Specify a style file picker.

UI Control	Selectors	Description
	.opa-forms-file-list-renderer__label-container .oj-label	Specify a style for file list label.
	.opa-forms-file-list-renderer__refresh-icon	Specify a style for file list refresh button.
	.opa-forms-file-list-renderer__list	Specify a style for file list view.
	.opa-forms-file-list-renderer__list-item .opa-forms-file-list-renderer__file-icon	Specify a style for list item file icon.
	.opa-forms-file-list-renderer__list-item .opa-forms-file-list-renderer__list-file-name	Specify a style for list item file name.
	.opa-forms-file-list-renderer__list-item .opa-forms-file-list-renderer__list-file-create-details	Specify a style for file list item created by and date details.
	.opa-forms-file-list-renderer__list-item .opa-forms-file-list-renderer__delete-icon	Specify a style for list item delete button.
Document understanding	.document-understanding-file-picker	Specify a style the file picker part of document understanding.
	.document-understanding-container	Specify a style for container having image and controls of document understanding.
	.document-understanding-image-container	Specify a style image preview of document understanding.
	.document-understanding-controls-container	Specify a style for document understanding controls container.
Rich text editor	proc-form-rich-text-editor .tox-tinymce	Specify style for styling the rich text editor like height and border.
Train	t.rain-overflow-wrapper .oj-train	Specify style for train container.
	.oj-train .oj-train-step-list-item	Specify style for train item.
	.oj-train-step-list-item.oj-selected	Specify style for selected train item.
	.oj-train .oj-train-button-connector	Specify style for train item button.
	.oj-train .oj-train-label	Specify style for train item label.

Bind Form Data with Controls

In forms, data and controls are decoupled. You can define your controls and data independently and then connect the controls to the data using the binding property.

In the forms editor, on the **Forms** tab, notice that the **Auto Binding** field is selected by default. When the auto binding property is enabled, the forms editor creates data attributes linking them to the controls as you drag and drop controls from a controls palette onto the canvas. The data attributes are listed in the Data pane. If you disable the **Auto Binding** field, then, you

must create data attributes in the Data pane and link them to every unbound control using the **General** tab's **Binding** field.

On the **General** tab, notice that the forms editor displays the **Binding** field for every control on the canvas. In the **Binding** field, you can specify an attribute for a control by selecting an option from the autocomplete list or entering a valid binding.

 **Note:**

- The name of the binding in the **Binding** field is the same as the name of the control to which it is linked when you use auto-binding.
- When a data attribute is bound—either directly or indirectly (through computed values or events)—to multiple controls, any change to the data attribute is immediately reflected in all associated controls. Repeatable sections or tables bound by the same data attribute are also in sync, that is, addition or deletion of rows in one control is automatically reflected in the other control.

The background colors displayed for the autocomplete options are:

- Green for valid binding names
- Red for invalid binding names
- Blue for complex binding names (their children may contain valid binding names)

If a control isn't linked to an attribute or if a control is linked to an invalid attribute, an error message is displayed in the **Binding** field and an error icon is displayed in the canvas. When a control doesn't have a valid binding, any value entered into the control in runtime is not passed on to the form payload.

To create a new attribute, click the **Add** icon in the Data pane. In the Create Attribute window, enter a name and specify whether the new attribute is a simple type (String, Number, Boolean, Date, Date Time, and Time) or a business type. If you selected **Business** type, then, select an option from the available list of business types. Optionally, enable or disable the list of values. Click **Create**.

 **Note:**

- An attribute name can only start with a lower case letter and can only contain letters, digits, and underscores. Also, an attribute name can't start with XML.
- In the Create Attribute window, if you select **Business** in the **Type** field, then, the business types defined for your application are available in the drop-down list. Notice that the business types defined for your application are listed in the **Business Types Palette**.

To delete an attribute, click the **Delete** icon adjacent to the attribute you want to delete in the Data pane. If the attribute you're trying to delete is bound to at least one control, click **Delete** in the Delete Attribute Confirmation window. Note that the forms editor clears the binding when you delete an attribute which is bound to a control.

To edit an attribute's name, click the **Edit** icon adjacent to the attribute you want to edit in the Data pane. Enter a new name and click **OK**.

Create a Form Based on a Business Type

A business type is used to organize and store related information used in your process, such as employee data. You can create business type and list of values for your application and use it in forms.

The forms palette lists all business types defined for your application under Types.

Drag a business type under Types in the forms palette onto the canvas, and the forms editor automatically creates a form based on the business type's data attributes.

A business type of Enum or List of values is represented as a customized select control when you drop it onto the form's canvas.

See [Work with Business Types](#).

Create Computed Controls

Within a web form, you can define a control as a computed field that uses a data definition, value of an existing control, result of a custom formula, or connector data as its value.

You can define complex formulas using different functions based on data definitions, connector data, and existing controls to compute the control value. All controls that support binding also support computed values.

To convert an editable control to a computed control in a web form:

1. Click on the control in the form's canvas and select the **Computed Value** check box on the Properties pane's **General** tab. An **Edit** button appears.
2. Click **Edit** to define how the control's value is calculated.
3. In the Configure Computed Value window, select from the following options in the **Type** field:
 - a. **Constant** lets you specify a value for the selected control. Be sure to specify the corresponding value in the **Value** field.
 - b. **Data Definition** lets you use the data from the payload. When you click the adjacent **Value** field that displays, available items are listed.
 - c. **Control** lets you use the data from a control. Be sure to select the control name.
 - d. **Function** lets you define custom formulas using common operations with strings, values and arrays, as described in [Specify Functions](#).
 - e. **Connector Data** lets you assign a value from a global connector call.
4. After configuring the value, click **OK**, then **Save**.

Note:

- If a control has a computed value and also has a value assigned through payload, the computed value has the priority.
- If a control has a computed value but is edited, either through user input or event actions, the new value remains. However, when a dependency (data definition, control value, or connector data) is updated, the new computed value is set.

Example of a Computed Control

Let's create a web form to collect sales and costs data through user input and calculate total profits.

When the form loads, you should be able to enter sales and costs data in editable controls, and a computed control should calculate the difference between data entered to arrive at total profits.

Now, let's add and configure controls in the web form to suit this example.

1. Add and configure two editable number fields onto the form's canvas.
 - a. From the Basic Palette, drag and drop a **Number** control onto the canvas.
 - b. Drag and drop another **Number** control below the first control.
 - c. Select the first **Number** control. On the **General** tab, change the **Name** field to *TotalSales*, and the **Label** field to *Total Sales (in USD)*.
 - d. Repeat the previous step to change the second control's name to *TotalCosts* and its label to *Total Costs (in USD)*.
2. Add a third number field and configure it as a computed control.
 - a. Drag and drop a **Number** control below the previously added controls.
 - b. Select the control. On the **General** tab, change the **Name** field to *TotalProfits*, and the **Label** field to *Total Profits*. Select the **Computed Value** check box and click the **Edit** button.
 - c. In the Configure Computed Value window, define the formula to calculate profits. Select **Function** in the **Type** field, and choose the **Subtract (-)** function in the **Function** field.

As illustrated in the following figure, specify parameters for additional fields to use values from editable controls.

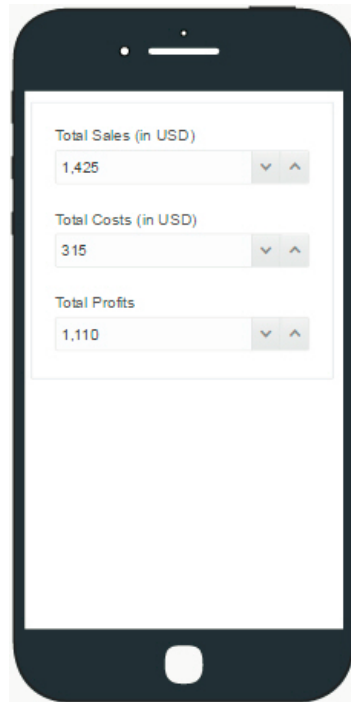
The screenshot shows the 'Configure Computed Value' dialog box. It contains a formula editor with the following fields:

- Type:** Function
- Function:** -
- Type:** (
- Control Name:** TotalSales
- Type:** Control
- Control Name:** TotalCosts
- Type:**)

At the bottom right, there are **OK** and **Cancel** buttons.

Click **OK** to close the window.

3. Click **Preview** to test the computed control. The following figure shows sample data entered and profit calculated automatically.



Specify Filters for Controls

You can specify filters for controls that accept list of values (LOV) as input, such as drop-down select, checklist, checkbox, and radio button controls. You can use a filter to selectively extract data from a data source (such as payload or an external connector) and display the filtered data as options of the control.

To specify a filter for an LOV control (for example, drop-down select):

1. Click on the control in the form's canvas, and in the Properties pane's **General** tab, navigate to the Option Source section.
2. Select the data source as either **From Data** or **Connector**. Configure subsequent fields to match your selection.
3. A **Filter** check box appears after you complete configuring the Option Source section.
4. Select the check box, and click the **Edit** button that appears next to it.
5. In the Filter window, specify how the data is to be filtered from the source. You can also use an If block to specify the same.

Note:

If you specify more than one way to filter data from the source, the results are ORed.

6. Click **OK**, then **Save**.

Filtering in Controls – An Example

This section demonstrates filtering in controls using a web form with a drop-down select control.

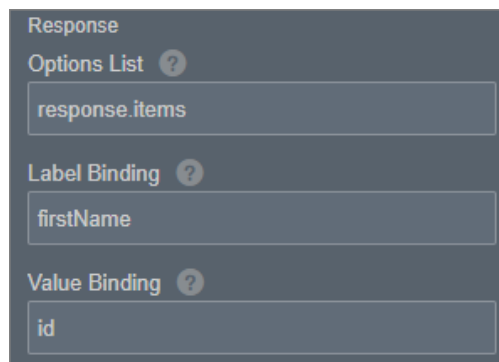
Based on the filtering condition specified for the control, you will be able to see relevant data as options of the control when the form loads.

In this example, we use a previously created REST connector to fetch data from a web server, which contains a list of items. Each item, in this case, is a complex data object that contains details of employees of an organization, such as first name, last name, unique ID, phone number, `isActive` (this Boolean attribute specifies if the individual is a current employee or not), city, and so on. To create a REST connector from scratch, see [Add a REST Connector](#).

We use this REST connector as the input to the select control.

Now, let's configure the select control to suit this example.

1. Click on the control, and on the **General** tab, change the **Label** field to *First Name*.
2. Under Options Source, select the **Connector** option and populate the **Resource** and **Operation** fields with respect to the REST connector defined within the application. See [Add REST Connector Resources and Operations](#).
3. Under Response, specify the items list to display as options of the control.
 - a. For this particular example, we map the first name attribute to the **Label Binding** field, so that only first names of employees appear as options in the select control. Also, the unique ID attribute is mapped to the **Value Binding** field as follows:



The screenshot shows a configuration window for a select control. The 'Response' section is expanded, showing three fields:

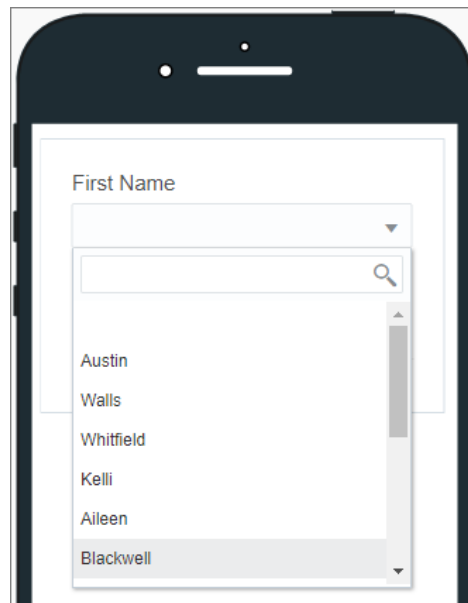
- Options List**: `response.items`
- Label Binding**: `firstName`
- Value Binding**: `id`

4. Select the **Filter** check box, and click the **Edit** button that appears next to it.
5. In the Filter window, specify a condition to filter specific data from the source. The filter configuration in the following figure extracts only active items (that is, current employees of the organization) from the data source to display as options of the select control:

The screenshot shows a 'Filter' dialog box with the following configuration:

- Filter** (orange button)
- Options List**: items
- Value**: isActive
- is equal to** (dropdown menu)
- Type**: Constant
- Value**: 1
- + Filter** (orange button)
- + If** (blue button)
- OK** and **Cancel** buttons at the bottom right.

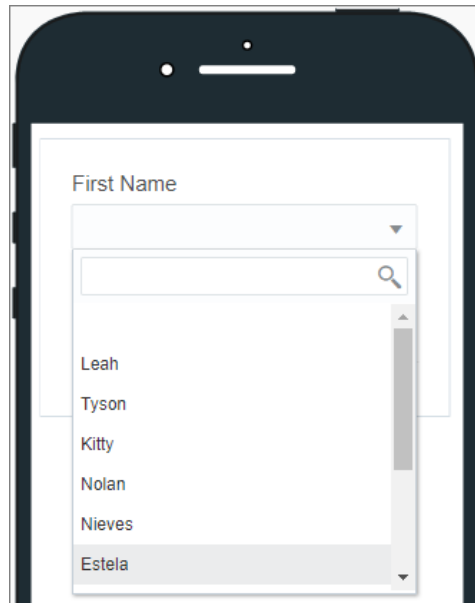
6. When the form loads, the control displays first names of all current employees as follows:



7. You can change the filter configuration on the select control to display only former employees (*isActive=false*) as follows:

The screenshot shows a 'Filter' configuration window. It has a title bar 'Filter' and a trash icon. Inside, there's a 'Filter' button, a 'Value' field containing 'isActive', and a dropdown menu showing 'is equal to'. Below that, there's a 'Type' dropdown set to 'Constant' and another 'Value' field containing '0'. At the bottom, there are '+ Filter' and '+ If' buttons, and 'OK' and 'Cancel' buttons at the very bottom right.

8. The options in the drop-down select change accordingly:



Add Dynamic Behavior to Forms

Use events to introduce dynamic behaviors into your forms, and combine them with actions, conditions, functions, and REST connector calls.

For example, you can introduce the following behaviors into your forms:

- Populate data in a control field based on another control field in the form. For example, a Country select field will impact the State select field and the State select field will impact the City select field.
- Enable control field validation based on another form control field. For example, if Start Date is given then End Date is mandatory or a Full Name gets its value from the First Name and Last Name.

- Make a REST call on demand, store the call's response, and use response data in an event action or condition.

 **Note:**

To use logged-in user's credentials when loading a form to execute a REST operation, define a REST connector without credentials. To use the same operation as a service call in the process, define another identical REST connector with credentials. This applies to internal REST API calls only.

- [Configure Events](#)
- [Specify Actions](#)
- [Specify Conditions](#)
- [Specify Functions](#)
- [Specify Filters in Events](#)
- [Reuse Event Snippets](#)
- [Execute Loops in Events](#)
- [Execute REST Connector Calls in Events](#)
- [Populate Controls Using REST Calls](#)
- [Link and Refresh List of Value Fields](#)

Configure Events

By configuring one or more events on a control or presentation, you change the control or presentation's behavior. Configuring events in forms enables you to trigger connector calls, actions, conditions, and functions.

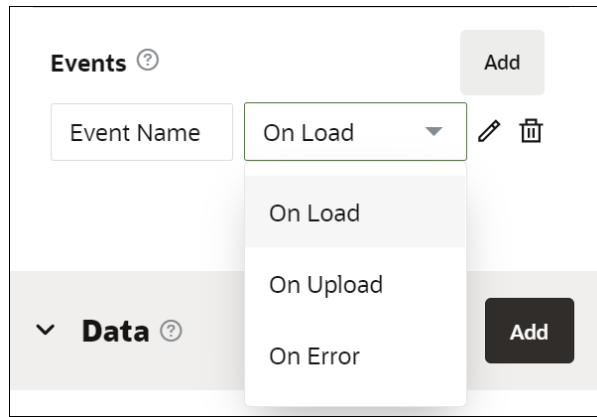
To configure events for a control or presentation:

1. In the web forms editor, select a control or presentation, and locate the **Events** options.
 - a. For a control, select the control and scroll down the **General** tab of the Properties pane until you see **Events**.
 - b. For a presentation, click a blank area of the form so that the Form and Presentation tabs display in the Properties pane. On the **Form** tab, select a presentation, then select the **Presentation** tab. Scroll down until you see **Events**.

Notice placeholder text displayed in the Events field *No events defined*.

2. Click **Add** to define an event.

The Events section now displays two fields; an **Event Name** field and a field with a drop-down menu listing event options available for the control or presentation.



3. Select an event option from the list and, optionally, enter a name for the event. Available event options are specific to the selected control or presentation.

Event	Description	Control or Presentation
On Load	Fires when a form or presentation loads.	Input Text, Text Area, Date, Time, Date Time, Email, URL, Money, Phone, Checklist, Checkbox, Radio Button, Button, Select, Section, Table, Repeatable Section, Message, Image, Video, Tab and Panel. Presentation
On Change	Fires when an end user changes the control's value.	Input Text, Text Area, Date, Time, Date Time, Email, URL, Money, Phone, Checklist, Checkbox, Radio Button and Select.
On Focus	Fires when the cursor is in the control.	Input text, Text Area, Number, Date, Time, Date Time, Email, URL, Money and Phone.
On Blur	Fires when the control loses focus.	Input text, Text Area, Number, Date, Time, Date Time, Email, URL, Money and Phone.
On Submit	Fires before the form or presentation is submitted.	Input Text, Text Area, Date, Time, Date Time, Email, URL, Money, Phone, Checklist, Checkbox, Radio Button, Button and Select. Presentation
On Click	Fires when an end user clicks the button.	Button and Link
On Expand	Fires when a section is expanded.	Section
On Collapse	Fires when a section is collapsed.	Section
On Expand Toggle	Fires when an expanded section is toggled.	Section
On Row Add	Fires when an end user adds a row to a table or repeatable section.	Table and Repeatable Section
On Row Remove	Fires when an end user removes a row from a table or repeatable section.	Table and Repeatable Section
On Selection Change	Fires when an end user changes the row selection in a table or repeatable section.	Table and Repeatable Section
On Upload	Fires when a file upload is successful.	File Upload in the File Manager
On Error	Fires when a file upload is unsuccessful.	File Upload in the File Manager


Event	Description	Control or Presentation
On List Refresh	Fires when the file list is refreshed.	File List in the File Manager
On Download	Fires when a file is downloaded successfully.	File List in the File Manager
On Delete	Fires when a file is deleted successfully.	File List in the File Manager
On Download Error	Fires when a file download is unsuccessful.	File List in the File Manager
On Delete Error	Fires when a file delete is unsuccessful.	File List in the File Manager

- After selecting an event option, click the editing icon next to the events drop-down menu to specify the actions, conditions, connectors, or functions for the event. Click **OK** to complete configuring the event.

See [Specify Actions](#), [Specify Conditions](#), and [Execute REST Connector Calls in Events](#).

Note:

To cancel an event, click its delete icon. Click **Add** to define an additional event. You can define multiple events for a control.

You can configure multiple events in a control. The configured events are executed sequentially in runtime. Note that, by default, the newest configured event is added last in the event sequence, and subsequently gets executed last in runtime. But you can re-order events to control their execution sequence. Use the event re-order handler  to re-order events and ensure that they get executed in the correct sequence in runtime. The event re-order handler gets displayed in front of each event name field only when there is more than one event configured on a control.

If you want to disable all events configured for a control, select the control, and specify the action **Disable Events**. See [Specify Actions](#). This disables all events configured for the control and the events do not get executed in runtime. But note that On Load type of events are exceptions. On Load events get executed even if you specify the **Disable Events** action.

To enable events in controls where they have been disabled by the **Disable Events** action, configure the **Enable Events** action. This ensures that all those events get executed in runtime.

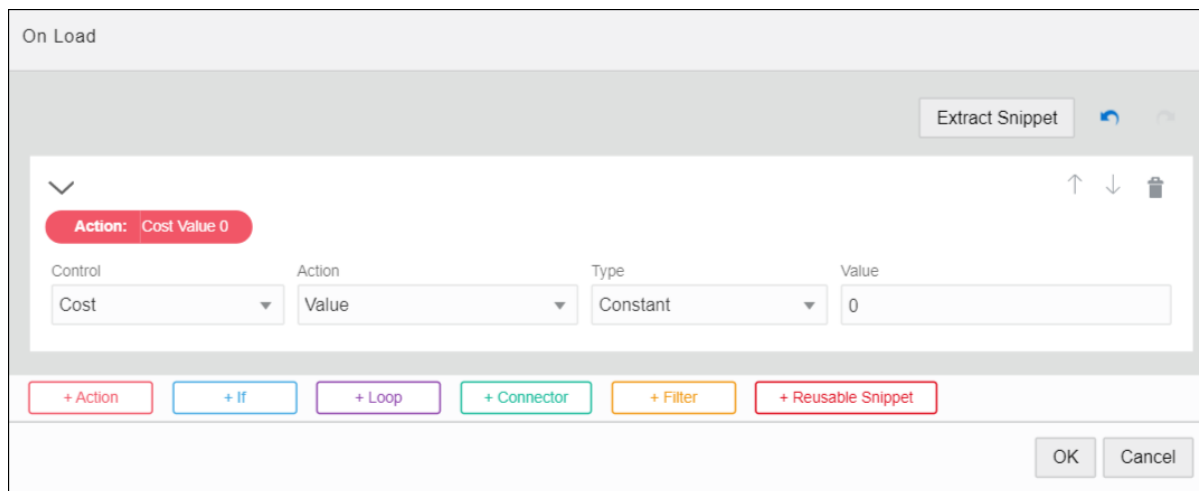
If the **Disable Events** action is configured for container controls such as tables, repeatable sections, sections, panels, and tabs, then all events in child controls that reside inside them are disabled. This also applies to any rows that you add to container controls after the **Disable Events** action has been configured in the container controls. To enable events in container controls, and also in all child controls inside the container controls, configure the **Enable Events** action in the container controls.

Specify Actions

Actions let you trigger changes to a control. You can choose from a variety of control and style actions. For example, you might configure a Clear button to clear the values of other form controls.

To specify actions for a control:

1. Select a control on the form canvas and specify an event for it. For example, choose **On Click** for a button control. See [Configure Events](#).
2. Click the **Event** editing icon adjacent to the **Event** field.
The event window displays with the selected event option (for example, On Load) at the top, and color coded buttons for adding actions, conditions, or connectors to the event.
3. Click **+Action** to add an action.
A solid red **Action** indicator appears. The **Summary** field below it displays a compact version of the action you'll define. Use the button at the end of this field to collapse or expand the actions editor.
4. In the **Control Name** field, select the control the action will affect.
5. In the **Action** field, select the action to take place. For example, select **Clear Value** to clear the selected control's value.
You can select from a variety of actions grouped in **Control** and **Style** categories. The options vary based on the control.
6. If configuring a control action, complete any additional fields that display. Control actions such as Value, Label, Help, Hint, Placeholder, Min Value, Max Value, and Pattern require further configuration for **Type** and **Value** fields.



If a **Type** field displays, select from these options:

- **Constant** lets you specify a value for the selected control. Be sure to specify the corresponding value in the **Value** field.
- **Data Definition** lets you use the data from the payload. When you click the adjacent **Value** field that displays, available items are listed.
- **Control** lets you use the data from a control. Be sure to select the control name and property to use.

Properties are control specific. For example, some controls, such as the number control, have many properties as options. Other controls, such as the image control, only have one property: Hidden. Email and phone controls have a property Placeholder but don't have the Min Value and Max Value properties.

- **Function** lets you perform common operations with strings, values and arrays, as described in [Specify Functions](#).

- **Connector Data** lets you assign a value from a connector call made from the same event.

 **Note:**

You can modify multiple properties of controls such as their value or label. You can also use the data, such as connector data, to modify the control, but you cannot modify data.

7. If configuring a style action, complete additional fields that display for the selected action. For example, you might combine a condition with a style action so that if a certain value is exceeded, a control's color, label, or class changes to alert users.
8. If needed, use the buttons at the bottom of the event window to configure additional actions or specify a condition or connector call. See [Specify Conditions](#) and [Execute REST Connector Calls in Events](#).

Connector calls cannot be executed based on a condition. Connectors are always executed when their event is executed. You can later use the connector response values in any condition or action.

 **Note:**

In the event window, use the **Undo** or **Redo** buttons to remove or restore recent changes to actions. Use the delete icon to delete an action. Click **Cancel** to exit the event window without saving changes.

9. After completing the event, click **OK**, then **Save**.

Specify Conditions

Conditions let you configure an If/Then/Else condition to trigger an action or connector call for a control's selected event or a control's specified action.

To specify conditions for a control:

1. Select a control on the form canvas and specify an event for it. See [Configure Events](#).
2. Click the event editing icon adjacent to the **Event** field.

The event window displays with the selected event option (for example, On Load) at the top, and color coded buttons for adding actions, conditions, or connectors to the event.

3. Click **+If** to add a condition.

Multiple items display to help you construct and complete the condition.

- Solid blue **If**, **Then**, and **Else** indicators signal each portion of the condition to complete.

The **Summary** field below the **If** indicator displays a compact version of the condition you'll define. Use the button at the end of this field to collapse or expand the conditions editor.

- Use the **+Condition**, **+Action**, **+Else If**, and **+Else Action** buttons to add additional conditions or actions to the condition.
- Use the fields displayed under the **If** or **Else If** indicators to define conditional behavior.

4. Complete the If portion of the condition.
 - a. Select a type from the **Type** field and complete fields that display. Select from these options:

- **Constant** lets you specify a value for the selected control. Be sure to specify the corresponding value in the **Value** field.
- **Data Definition** lets you use the data from the payload. When you click the adjacent **Value** field that displays, available items are listed.
- **Control** lets you use the data from a control. Be sure to select the control name and property to use.

Properties are control specific. For example, some controls, such as the number control, have many properties as options. Other controls, such as the image control, only have one property: Hidden. Email and phone controls have a property Placeholder but don't have the Min Value and Max Value properties.

- **Function** lets you perform common operations with strings, values and arrays, as described in [Specify Functions](#).
- **Connector Data** lets you assign a value from a connector call made from the same event.

- b. Select an operation. For example, choose **is True** from the drop-down field.

 **Tip:**

To check for a null value, choose **is False**; if a control has no value, the condition result returns true. Conversely, choose **is True** to check that a property is present.

 **Note:**

You can also check if a particular property, such as value, of a control is empty or not using the **is empty** operation. Controls of array type (repeatable sections or tables) are considered empty if the array is empty. A check box or a checklist is considered empty if no selection is made.

- c. Based on the selected operation, complete a second **Type** field and additional fields that display. You must specify a type and additional fields for all operations except when **is True** or **is False** are selected.
- d. To add another condition, click **+Condition**, specify if it's an **And** or an **Or** condition and configure it.

For example, as shown in the figure, you can configure a condition for the Cost control field of a travel request form such that if cost is greater than 5000 and the duration of travel is less than 14 days then you can set further Then or Else conditions to be triggered.

5. Complete the Then portion of the condition by clicking **+Action** and completing the action. See [Specify Actions](#).

For example, for the Cost control of a travel request form if the cost is greater than 5000 and the number of days of travel is less than 14 then the Justification field is required in the form.

6. Complete the Else portion of the condition by clicking **+Else Action** and completing the action.

For example, for the Cost control of a travel request form if the cost is greater than 5000 and the number of days of travel is less than 14 then the Justification field is required in the form else it's optional.

The screenshot shows the Oracle Forms Event Window interface. It features two main sections for defining conditions. The top section is titled 'Then' and contains a red button '+ Action' and a blue button '+ Then if'. Below this, a condition is defined with the action 'Justification Required'. The 'Control' dropdown is set to 'Justification' and the 'Action' dropdown is set to 'Required'. The bottom section is titled 'Else' and contains a red button '+ Else Action' and a blue button '+ Else If'. Below this, a second condition is defined with the action 'Justification Optional'. The 'Control' dropdown is set to 'Justification' and the 'Action' dropdown is set to 'Optional'. Each condition section includes a collapse arrow, up/down arrows for reordering, and a delete icon.

7. If needed, add and configure additional If/Then/Else conditions.

Note:

In the event window, use the **Undo** or **Redo** buttons to remove or restore recent changes to conditions. Use the delete icon to delete a condition. Use up and down arrows to reorder conditions within the window. Click **Cancel** to exit the event window without saving changes.

8. After completing the event, click **OK**, then **Save**.

Specify Functions

Use functions in event actions and conditions to perform common operations with strings, values, and arrays. For example, use a function to add two values, concatenate two strings, or sum items in table rows.

Important points about functions:

- You can specify parameters for selected functions. The parameter value can be a constant, data definition value, control value, other function, or connector data value.
- Some functions support selecting arrays of data or repeatable controls.
- You can nest functions, such as concatenate multiple strings or the results of multiple rows. You might select a concatenate function and concatenate a data value, connector data value, or control value with another value such as a constant.

Note that the event window where you specify functions has an embedded Help that gives details on each function. Choose a category in the **Category** column, the **Function** column on the right displays the related functions of the chosen category. Select a function to view details about that function with examples in the area below.

Help

Category	Function
Date	And
Logic	Or
Relational	Not
Math	If
Array	
Text	
Interval	
Objects	

And(statement1, statement2, ..., statementN)

Evaluates to true if all the statements are true, otherwise evaluates to false.

- **Boolean statement:** Statement to evaluate. There can be multiple statements.

Return Type: Boolean

Examples

- A store offers a discount for members of a VIP program when they buy three items or more. They want to verify that both conditions are true. Example: `And(vip_program, items >= 3)`

Function Category	Function Name	Parameters	Description
Date	Current Date	None	Returns the current date in the format yyyy-mm-dd. For example: 1998-06-12.
Date	Current Time	None	Returns the current time in 24-hour time using the format T00:00:00. For example: T21:07:28.
Date	Current Date Time	None	Returns the current date and time in the format yyyy-mm-dd T00:00:00. For example: 1977-06-16 T21:07:28.

Function Category	Function Name	Parameters	Description
Date	Add Seconds	(Date, Number)	<p>Adds a specified number of seconds to a specified time.</p> <p>AddSeconds(originalTime, secondsToAdd)</p> <ul style="list-style-type: none"> Time originalTime: the time to which you want to add the seconds. Number secondsToAdd: the number of seconds that you want to add to the specified time.
Date	Add Minutes	(Date, Number)	<p>Adds a specified number of minutes to a specified time.</p> <p>AddMinutes(originalTime, minutesToAdd)</p> <ul style="list-style-type: none"> Time originalTime: the time to which you want to add the minutes. Number minutesToAdd: the number of minutes that you want to add to the specified time.
Date	Add Hours	(Date, Number)	<p>Adds a specified number of hours to a specified time.</p> <p>AddHours(originalTime, hoursToAdd)</p> <ul style="list-style-type: none"> Time originalTime: the time to which you want to add the specified hours. Number hoursToAdd: the number of hours that you want to add to the specified time.
Date	Add Days	(Date, Number)	<p>Adds a specified number of days to a specified date.</p> <p>AddDays(originalDate, daysToAdd)</p> <ul style="list-style-type: none"> Date originalDate: the date to which you want to add the days. Date daysToAdd: the number of days you want to add to the specified date.
Date	Add Months	(Date, Number)	<p>Adds a specified number of months to a specified date.</p> <p>AddMonths(originalDate, monthsToAdd)</p> <ul style="list-style-type: none"> Date originalDate: the date to which you want to add the months. Number monthsToAdd: the number of months to add to the specified date.
Date	Add Years	(Date, Number)	<p>Adds a specified number of years to a specified date.</p> <p>AddYears(originalDate, yearsToAdd)</p> <ul style="list-style-type: none"> Date originalDate: the date to which you want to add the years. Number yearsToAdd: the number of years to add to the specified date.
Logic	And	(statement1, statement2, ..., statementN)	<p>Evaluates to <i>true</i> if all the statements are <i>true</i>, otherwise evaluates to <i>false</i>.</p> <p>And(statement1, statement2, ..., statementN)</p> <ul style="list-style-type: none"> Boolean statement: Statement to evaluate. There can be multiple statements.
Logic	Or	(statement1, statement2, ..., statementN)	<p>Evaluates to <i>true</i> if any of the statements is <i>true</i>, otherwise evaluates to <i>false</i>.</p> <p>Or(statement1, statement2, ..., statementN)</p> <ul style="list-style-type: none"> Boolean statement: Statement to evaluate. There can be multiple statements.
Logic	Not	(Condition)	<p>Negates the value of the specified statement.</p> <p>Not(statement)</p> <ul style="list-style-type: none"> Boolean statement: statement to negate.

Function Category	Function Name	Parameters	Description
Logic	Inline If	(Condition, ValueIfTrue, ValueIfFalse)	<p>Inline if condition. If the condition is <i>true</i>, it returns the first value. If the condition is <i>false</i>, it returns the second value.</p> <p>If(condition, valueIfTrue, valueIfFalse)</p> <ul style="list-style-type: none"> Boolean condition: Condition to evaluate. Any valueIfTrue: The value to return if the condition evaluates to <i>true</i>. Any valueIfFalse: The value to return if the condition evaluates to <i>false</i>.
Relational	Equal (=)	(Input1, Input2)	<p>Returns <i>true</i> if the two values are the same, otherwise returns <i>false</i>.</p> <p>=(input1, input2)</p> <ul style="list-style-type: none"> Input input1: Statement to evaluate. Input input2: Statement to evaluate.
Relational	Greater than (>)	(Input1, Input2)	<p>Returns <i>true</i> if the first statement is greater than the second statement, otherwise returns <i>false</i>.</p> <p>>(input1, input2)</p> <ul style="list-style-type: none"> Input input1: Statement to evaluate. Input input2: Statement to evaluate.
Relational	Less than (<)	(Input1, Input2)	<p>Returns <i>true</i> if the first statement is less than the second statement, otherwise returns <i>false</i>.</p> <p><(input1, input2)</p> <ul style="list-style-type: none"> Input input1: Statement to evaluate. Input input2: Statement to evaluate.
Relational	Greater than or equal to (>=)	(Input1, Input2)	<p>Returns <i>true</i> if the first statement is greater than or equal to the second statement, otherwise returns <i>false</i>.</p> <p>>=(input1, input2)</p> <ul style="list-style-type: none"> Input input1: Statement to evaluate. Input input2: Statement to evaluate.
Relational	Less than or equal to (<=)	(Input1, Input2)	<p>Returns <i>true</i> if the first statement is less than or equal to the second statement, otherwise returns <i>false</i>.</p> <p><=(input1, input2)</p> <ul style="list-style-type: none"> Input input1: Statement to evaluate. Input input2: Statement to evaluate.
Math	Sum (+)	(Number, Number)	<p>Adds two values.</p> <p>+(value1, value2)</p> <ul style="list-style-type: none"> Number value1: Value to add. Number value2: Value to add.
Math	Summation	([Numbers])	<p>Adds the values of multiple numbers.</p> <p>Summation(numbers)</p> <ul style="list-style-type: none"> Number/[Number] numbers: The values to add. You can add multiple Number parameters, arrays, or a combination of both. If you add only one Number parameter, then it returns that same value.
Math	Subtract (-)	(Number, Number)	<p>Subtracts two values.</p> <p>-(value1, value2)</p> <ul style="list-style-type: none"> Number value1: The minuend. Number value2: The subtrahend.

Function Category	Function Name	Parameters	Description
Math	Multiply (*)	([Number])	Multiplies two or more values. *(factors) <ul style="list-style-type: none"> ([Number]) factors: The factors to multiply. You can multiply multiple Number parameters, arrays, or a combination of both. If you add only one Number parameter, then it returns that same value.
Math	Divide (/)	(Number, Number)	Divides a value. The result has a scale of 10. For example: 355/113 = 3.1415929203 /(dividend, divisor) <ul style="list-style-type: none"> Number dividend: The value being divided. Number divisor: The value to divide by.
Math	Integer Division	(Number, Number)	Divides numbers and truncates the result (for example, 5/2=2, -5/2=-2).
Math	Modulo (%)	(Number, Number)	Finds the remainder after division of one number by another.
Array	Min	([Number])	Returns the minimum value in a set of numbers. Min(number1, number2, ..., numberN) <ul style="list-style-type: none"> [Number]/Number number: The arrays or numbers where to search for the minimum. You can add multiple Number parameters, arrays, or a combination of both.
Array	Max	([Number])	Returns the maximum value in a set of numbers. Max(number1, number2, ..., numberN) <ul style="list-style-type: none"> [Number]/Number number: The arrays or numbers where to search for the maximum. You can add multiple Number parameters, arrays, or a combination of both.
Array	Count	([Any])	Counts the number of elements in one or more arrays. Count(array1, array2, ..., arrayN) <ul style="list-style-type: none"> [Any] array: The arrays that contain the elements that you want to count. You can add multiple arrays.
Array	Avg	([Number])	Calculates the average value of the elements in one or more arrays. Avg(array1, array2, ..., arrayN) <ul style="list-style-type: none"> [Number] array: The arrays for which you want to calculate the average. You can add multiple arrays.
Array	Concat	([Any])	Concatenates the specified arrays by appending them at the end of the first array. ConcatArrays(array1, array2, ..., arrayN) <ul style="list-style-type: none"> [Any] array: The arrays to concatenate. You can add multiple arrays.
Array	Index Of	([Array], Element)	Returns the index of an element in an array. For example, IndexOf([1,2,3], 1) returns 0. Returns -1 if the element specified is not found in the array.
Array	At Index	([Any], Index)	Finds the value at a particular index of an array or string.
Text	Concat	([String])	Concatenates multiple strings into one single string. Concat(array1, array2, ..., arrayN) <ul style="list-style-type: none"> [String]/String array: The strings or array of strings to concatenate. You can add multiple strings, arrays, or a combination of both.

Function Category	Function Name	Parameters	Description
Text	Split	(String, String)	Splits a string into multiple parts based on a separator. The separators delimit the different parts in the string, and aren't part of the result. Split(text, separator) <ul style="list-style-type: none"> String text: The string to split. String separator: The separator used to split the string.
Text	Join	([String], String)	Combines multiple strings into one using a separator between the different strings. Join(strings, separator) <ul style="list-style-type: none"> [String] strings: The strings to combine. String separator: The separator used to separate the strings in the resulting string.
Text	To Lowercase	(String)	Converts all the characters in a string to lowercase. ToLowercase(text) <ul style="list-style-type: none"> String text: The string to convert to lowercase.
Text	To Uppercase	(String or Array, Element)	Converts all the characters in a string to uppercase. ToUppercase(text) <ul style="list-style-type: none"> String text: The string to convert to uppercase.
Text	Length	(String)	Returns the number of characters in a string.
Text	Trim	(String)	Removes leading or trailing spaces in a string.
Text	Contains	(String or Array, Element)	Checks if a string or array (within a control) contains a specific element and returns a Boolean value.
Text	Replace	(String, String, String)	Replaces a text string. Uses three parameters, where: <ul style="list-style-type: none"> The first string is the original text The second string is the value or regular expression that will be replaced by the new value The third text is the replacement For example, REPLACE("Hello World!", "Hello", "World") returns "World World!"


 **Note:**

In regular expressions there are 12 characters with special meanings: backslash \, caret ^, dollar sign \$, period or dot ., vertical bar or pipe symbol |, question mark ?, asterisk or star *, plus sign +, opening parenthesis (, the closing parenthesis), opening square bracket [, and opening curly brace {. If you want to use any of them as regular characters, you will need to escape them with backslash \.

Function Category	Function Name	Parameters	Description
Text	Substring	(String, StartingIndex, Length)	Returns a substring of a string, beginning at the specified starting index up to the length indicated. For example, Substring ('apple', 0, 3) returns <i>app</i> .
Interval	Duration in Minutes	(Date, Date)	Calculates the number of minutes between two dates. DurationinMinutes(date1, date2) <ul style="list-style-type: none"> • Datetime date1: The first date. • Datetime date2: The second date.
Interval	Duration in Hours	(Date, Date)	Calculates the number of hours between two dates. DurationinHours(date1, date2) <ul style="list-style-type: none"> • Datetime date1: The first date. • Datetime date2: The second date
Interval	Duration in Days	(Date, Date)	Calculates the number of days between two dates. DurationinDays(date1, date2). <ul style="list-style-type: none"> • Date date1: The first date. • Date date2: The second date.
Interval	Duration in Months	(Date, Date)	Calculates the number of months between two dates. DurationinMonths(date1, date2) <ul style="list-style-type: none"> • Date date1: The first date. • Date date2: The second date.
Interval	Duration in Years	(Date, Date)	Calculates the number of years between two dates. DurationinYears(date1, date2) <ul style="list-style-type: none"> • Date date1: The first date. • Date date2: The second date.
Objects	Convert to JSON	Any	Converts an object into a String using the JSON format. ConverttoJSON(objectToConvert) <ul style="list-style-type: none"> • Any objectToConvert: The object to convert to JSON format.
Objects	Parse from JSON	(String)	Parses a String that contains content in JSON format and returns an Object. ParsefromJSON(JSONtoParse) <ul style="list-style-type: none"> • String JSONtoParse: The JSON content to parse.
Objects	Get Property	(Object, String)	Gets a specified property from an Object. GetProperty(object, property) <ul style="list-style-type: none"> • Object object: The object from which you want to get the property from. • String property: The property to retrieve.
Current User	Current user's ID	None	Gets the ID for the current logged in user.
Current User	Current user's first name	None	Gets the first name for the current logged in user.
Current User	Current user's middle name	None	Gets the middle name for the current logged in user.

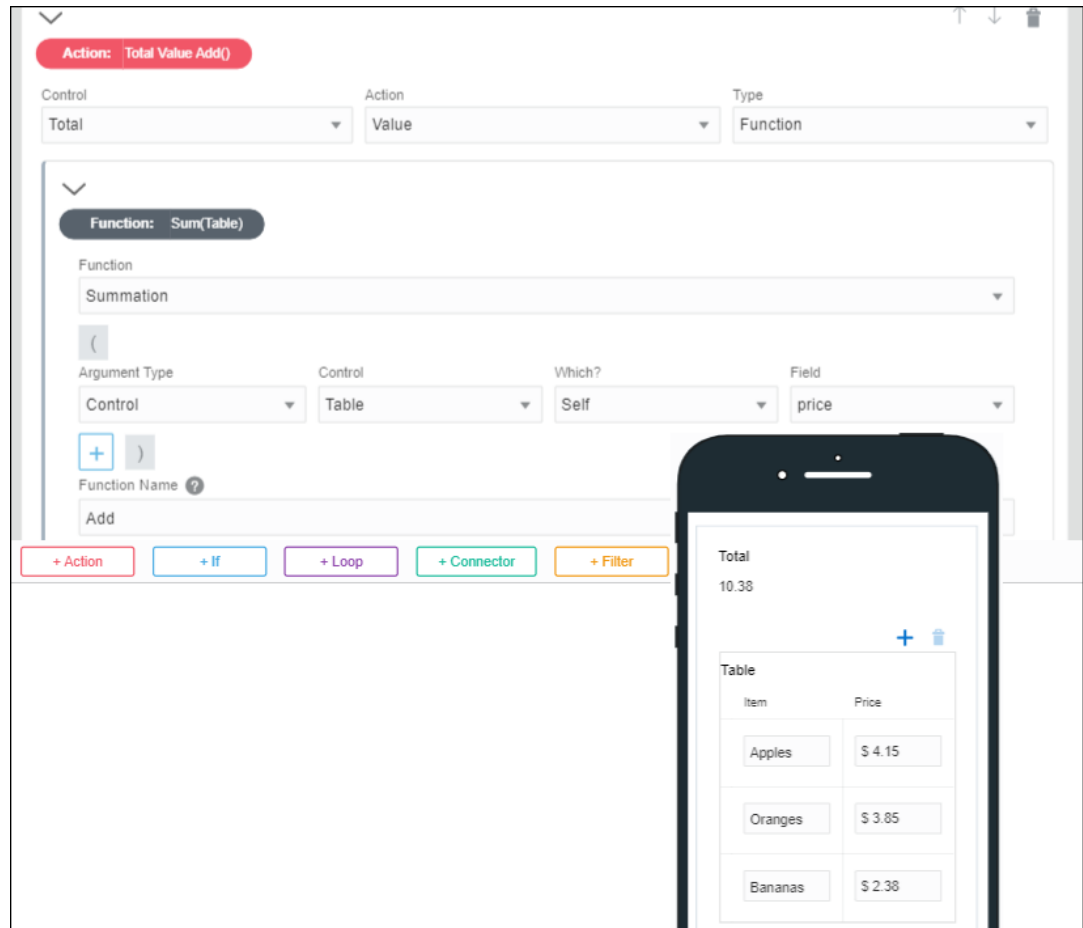
Function Category	Function Name	Parameters	Description
Current User	Current user's last name	None	Gets the last name for the current logged in user.
Current User	Current user's email	None	Gets the email for the current logged in user.
Other	Current UUID	None	Generates a universally unique identifier.
Other	Current browser language	None	Gets the locale that the user's browser is using.
Other	Get application name	None	Returns the name of the process application.

To use a function:

- In the web forms editor, select a control on the canvas. Note that you can also apply events and functions to presentations, as described in [Configure Events](#).
- On the Properties pane, scroll down to the **Events** field. Click **Add** to add an event and select its type in the drop-down field (for example, On Change).
- Click **Edit**  next to the event you just added.
- In the event window, add an action or condition by clicking the **+Action** or **+If** button.
- Configure the action or condition to use a function.
 - For an action, select a control on which to apply the function in the **Control Name** field. In the **Action** field that displays, select **Value** in the **Action** field. In the **Type** field that displays, select **Function**, then choose a function from the functions listed by category in the **Function** field. Optionally, provide a name for the function in the **Function Name** field to easily identify it (this name displays in the **Action Summary** field).
 - For an if condition, select **Function** in the **Type** field that displays, and choose a function from the functions listed by category in the **Function** field. Optionally, provide a name for the function in the **Function Name** field to easily identify it (this name displays in the **Condition Summary** field). Complete the condition in the remaining condition fields.
- Specify parameters within parentheses for functions that allow them.

Additional fields display within parentheses for functions that include parameters, such as math functions.

The **Compact Function** field displays a summary of the function you define. Use the button at the end of this field to collapse or expand the functions editor.
- Click **OK** to close the event window. Click **Preview** to test the function. For example, a simple application uses a function set on the Value field to sum values each time a user enters or changes a value in the table, and displays the calculated value in the Total field. If the total exceeds a set amount (constant), a style action displays the total in red letters.



Example of Current User Function

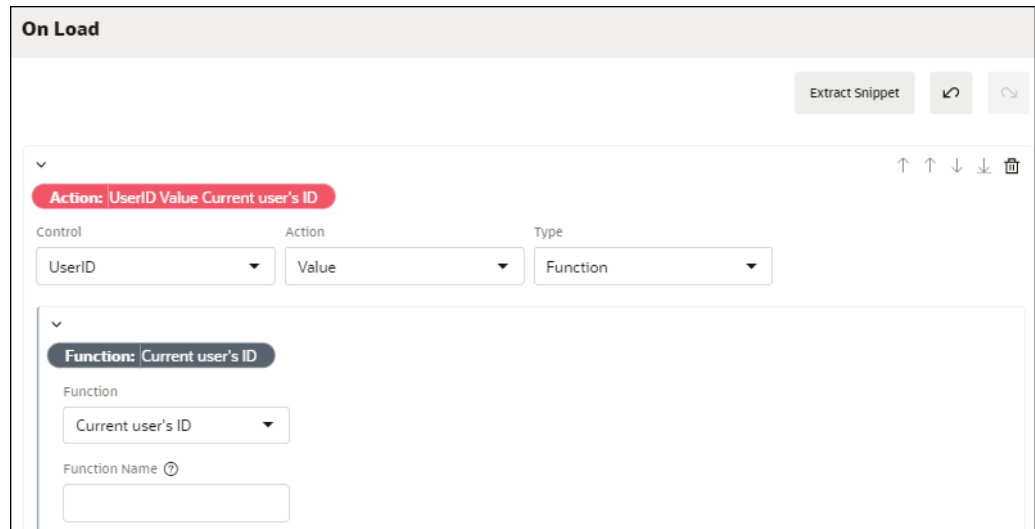
Use the Current User function within actions in form controls to get information about users in forms.

The Current User function allows you to dynamically populate information about the current user's id, email, and the first, middle, and last name of the user in forms.

Let us explore how to use the Current User function with an example. In this example, let us configure a form with three fields (user id, first, and last name) such that they automatically populate when the form loads. Then configure another field (email) such that it gets populated when the user clicks a button **Get Email**.

1. Drag and drop an Input Text control onto the form's canvas. Then drag and drop two more Input Text controls and place them side by side.
 - a. Change the name of the first control to `UserID` and its label to `User ID`.
 - b. Change the names of the controls that you placed side-by-side to `FirstName` and `LastName`. Change their labels to `First Name` and `Last Name`.
2. Configure the User ID control.
 - a. Select the control, and in the properties pane scroll down till you find **Events**.
 - b. Click **Add** to configure an event. Select **On Load** from the drop-down list and then click the edit icon.

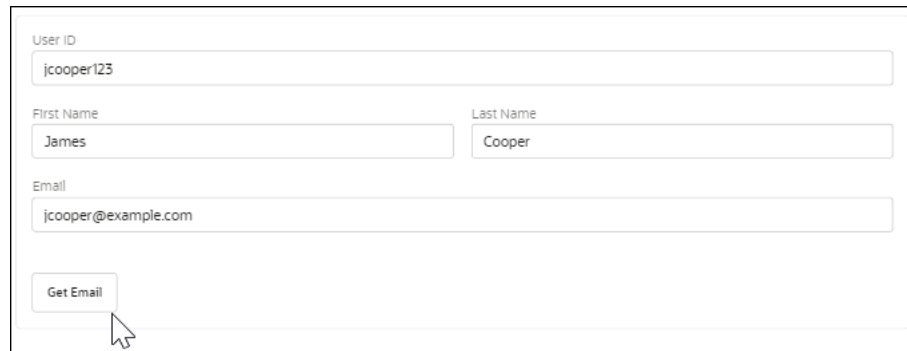
- c. In the event On Load window, click **+ Action**. In the **Control Name** field select **UserID** and then select **Value** from the **Action** drop-down list. In the **Type** field that displays, select **Function**.
- d. In the **Function** drop-down list that displays, scroll down till you find **Current User**, and then select **Current user's ID**.



- e. Click **OK**.
3. Configure the First Name and Last Name controls.
 - a. For each control, select the control and repeat steps 2b to 2d. Ensure to select the correct control (First Name or Last Name) in the **Control Name** field.
 - b. For the First Name control, in the **Function** drop-down list, select **Current user's first name** under **Current User**.
 - c. For the Last Name control, in the **Function** drop-down list, select **Current user's last name** under **Current User**.
 4. Drag and drop an Email control. Then drag and drop a Button control under it. Change the name of the button control to `GetEmail` and its label to `Get Email`.
 5. Configure the Get Email button.
 - a. Select the button, and in the properties pane scroll down till you find **Events**.
 - b. Click **Add** to configure an event. Select **On Click** from the drop-down list and then click the edit icon.
 - c. In the event On Click window, select **Email** from the **Control Name** drop-down list, and then select **Value** from the **Action** drop-down list. In the **Type** drop-down list that displays, select **Function**.
 - d. In the **Function** drop-down list that displays, scroll down till you find **Current User**, and then select **Current user's email**.
 - e. Click **OK**.
 6. Click **Preview**.

When the form loads, the user's id, first, and last name gets auto populated into the **User ID**, **First Name** and **Last Name** fields. The Email field is empty.

Click the **GetEmail** button. Note that the email gets loaded into the **Email** field.



A screenshot of a web form. It contains four input fields: 'User ID' with the value 'jcooper123', 'First Name' with the value 'James', 'Last Name' with the value 'Cooper', and 'Email' with the value 'jcooper@example.com'. Below the fields is a button labeled 'Get Email'. A mouse cursor is pointing at the button.

Specify Filters in Events

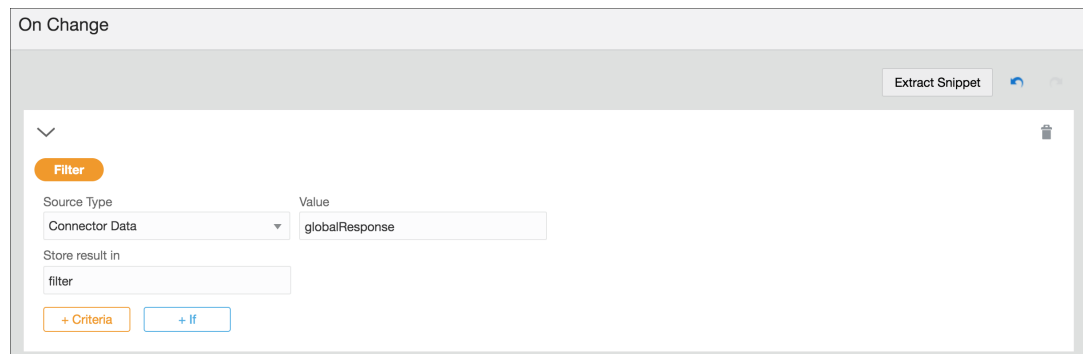
Filters let you selectively use a subset of data from a larger set. For a form control, you can define a filter within an event and use the filter data in other controls through event actions and conditions.

To specify filters within an event:

1. Select a control on the form canvas and specify an event for it. See [Configure Events](#).
2. Click the event editing icon adjacent to the **Event** field.

The event window displays with the selected event option (for example, On Change) at the top and color coded buttons for adding actions, conditions, connectors, or filters to the event.

3. Click **+Filter** to add a filter, and specify the data source from which you want to filter specific information. You can choose data attributes, another control, or a REST connector.
4. If the data source selected is valid (of the type: array), additional **+Criteria** and **+If** buttons appear as shown in the following figure:



A screenshot of the 'On Change' event configuration window. The window title is 'On Change'. At the top right, there is an 'Extract Snippet' button and a refresh icon. Below the title bar, there is a dropdown menu with a downward arrow. A 'Filter' button is highlighted in orange. Below the 'Filter' button, there are two input fields: 'Source Type' with a dropdown menu showing 'Connector Data', and 'Value' with the text 'globalResponse'. Below these fields, there is a 'Store result in' label and an input field containing the text 'filter'. At the bottom of the window, there are two buttons: '+ Criteria' (highlighted in orange) and '+ If'.

5. Click **+Criteria** to specify how data is filtered from the data source. You can also use an If condition to do the same.

 **Note:**

You can specify more than one way to filter data from the source. If you specify an Or condition, the results are combined or unioned. If you specify an And condition, the result is the intersection of that filter condition with the other filter conditions.

You can also specify another Filter (specified before) as the source type for your filter. This gives you the ability to filter data from another filter.

6. Provide a suitable name for the filter; the information extracted from the data source is stored within this filter.
7. In the same event window, you can use the data in the filter to populate other controls dynamically through actions or conditions.
8. After completing the event, click **OK**, then **Save**.

Filtering Using Events – An Example

This section demonstrates filtering through events using a web form with a drop-down select control and a table.

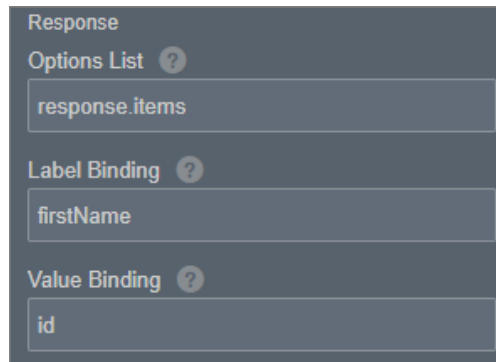
Based on the option you choose in the select control, you will be able to filter data from a data source and selectively use the extracted data to populate the table.

In this example, we use a previously created REST connector to fetch data from a web server, which contains a list of items. Each item, in this case, is a complex data object that contains personal details of individuals, such as first name, last name, unique ID, company name, email, phone number, city, and so on. To create a REST connector from scratch, see [Add a REST Connector](#).

We use this REST connector to define a global connector call, *globalResponse*, under Presentation properties of the form. Want to know more about global connectors? See [Work with Presentations](#). In addition, this connector also serves as the input to the select control.

Now, let's configure controls and events in the web form to suit this example.

1. Add and configure a drop-down select field on the form's canvas.
 - a. From the Basic Palette, drag and drop a **Select** control onto the canvas.
 - b. Select the control, and on the **General** tab, change the **Label** field to *First Name*.
 - c. Under Options Source, select the **Connector** option and populate the **Resource** and **Operation** fields with respect to the REST connector defined within the application. See [Populate Controls Using REST Calls](#).
 - d. Under Response, specify the items list to display as options of the control.
 - i. For this particular example, we map the first name attribute to the **Label Binding** field, so that only first names of all individuals appear as options in the select control. Also, the unique ID attribute is mapped to the **Value Binding** field as follows:



2. Add and configure a table control on the form's canvas.
 - a. From the Advanced Palette, drag and drop a **Table** control onto the canvas.
 - b. Add two additional columns, and edit column labels as *Last Name*, *Organization*, and *Email*.
 - c. Drag and drop an input text control into each column.
3. Click on the select control, and define an **On Change** event for it in the **General** tab.
4. Click the event editing icon to open the event window. In this window, click **+Filter** to add a filter.
5. The following figure shows the event configuration for this example:
 - a. Connector data is set as the data source and its value comes from the global connector call, *globalResponse*, defined previously.
 - b. The subsequent Filter section shows how the data is filtered from the source, that is, when the ID of an item in the source list matches with the ID of the option selected by the user, all data associated with that item is stored in the filter, *filter*.
 - c. An event action selectively uses the data contained in this filter to populate the table. Although each item may have several attributes associated with it, here we use only last name, company, and email attributes. We map each of these attributes a corresponding input text control or table column.

Note:

While defining event actions, if you choose to map the filter data to a control of simple data type (for example, input text), you have to specify an index value in addition to the attribute that you would like to map to the control.

- When the form loads, the select control displays first names of all individuals as follows:

- After you make a selection, all data associated with the selected option is stored within the filter, and this data is selectively used to populate the table as shown in the following figure:

First Name
Aileen

Last Name	Organization	Email
Welch	CUBIX	aileenwelch@cubix.com

Reuse Event Snippets

You can extract event snippets from an event editing window and reuse them across the form's presentation.

Define a complex event snippet once and reuse it across different events, thereby saving time and minimizing errors. You can manage all the extracted snippets from the form's **Presentation** tab. Additionally, updating a global snippet definition updates all instances of the snippet within other events.

- [Extract a Snippet](#)
- [Use a Snippet](#)

Extract a Snippet

Extract any event block, such as action, condition, loop, connector, or filter, as a global snippet and reuse it in other event definitions.

To extract an event snippet:

1. In an event definition window, click **Extract Snippet**.
A name field appears at the top of the window and all the available blocks are selected for extraction.

snippetName OK Cancel

Get started creating reusable snippets

Choose the actions, conditions, loops, connectors, and filters you want to extract, name your snippet, and click OK. Extracted actions, loops, and if blocks must be adjacent. Manage created reusable snippets from the presentations property panel.

Action: InputText Hide

Control: InputText Action: Hide



Action: Checklist Disable

Control: Checklist Action: Disable

2. Turn a block's toggle button off to exclude it from extraction.
3. Enter a suitable name for the snippet and click **OK**.
The snippet is now available for reuse in any event within the same presentation. You can access and edit the extracted snippet from the form's **Presentation** tab. See [Work with Presentations](#).

Use a Snippet

Use previously extracted event snippets in any event definition.

1. In the event definition window, select **+ Reusable Snippet** at any point in your definition.
2. Select an extracted snippet from the drop-down list to add it to the window.
3. Additionally, you can click **Detach**  to independently retain the blocks within a global snippet, but not the snippet as a whole.
Note that changes made to the global snippet won't reflect in the blocks you've retained.
4. Click **Delete**  to remove the snippet and all its constituents.

Execute Loops in Events

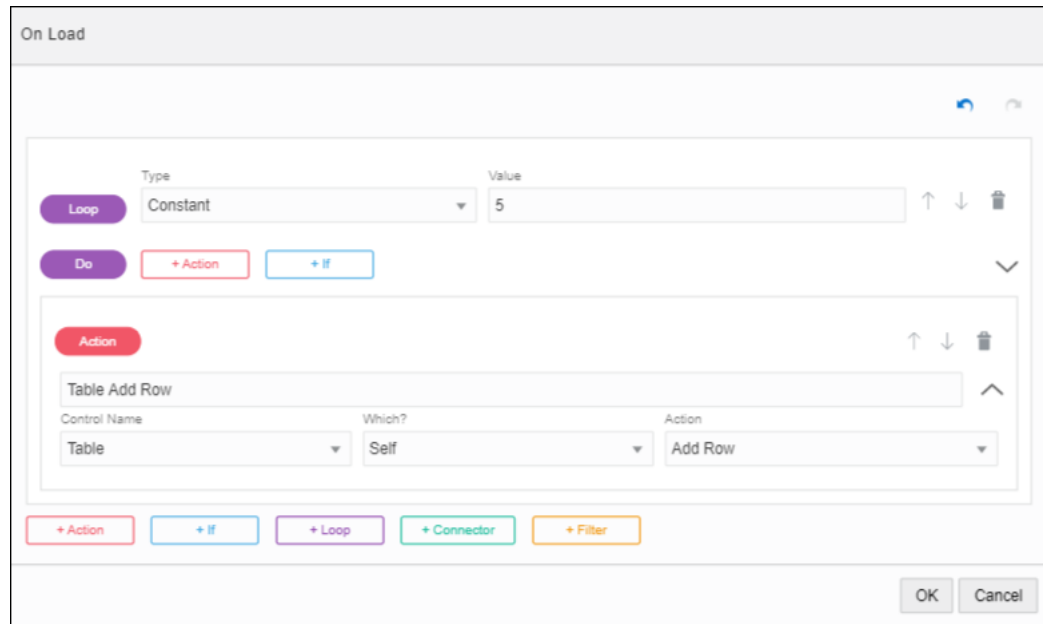
Use loops within Events in form controls to execute actions that need to iterate. The iterator can be a number or an array. Loops are a way to execute the same action multiple times in a form.

Let us explore with some examples how loops work in forms.

Example of a Simple Loop

In this example, let us add multiple rows to a table by configuring a simple loop with a number iterator.

1. Drag and drop a table control onto your form's canvas. Then drag and drop an InputText control in the table's column.
2. Select the table control, and in the properties pane that displays scroll down till you find **Events**. Click **Add** to configure an event for the control.
3. Select **On Load** from the drop-down list, and then click the editing icon.
4. In the event On Load window, click **Loop**. Select **Constant** in the **Type** drop-down list, and enter a number in the **Value** field, for example: 5. This value determines the number of times looping occurs when the form loads.
5. Click **+Actions**. In the **Control** drop-down list select **Table**, and in the **Action** drop-down list select **Add Row**.



When the loop executes (on form load), the Add Row action gets executed 5 times.

6. Click **OK**.
7. Preview the form and see how it works.

Note that 5 rows get added to the table as the form loads.

To see how the loop executed, click **Log** in the Preview window.

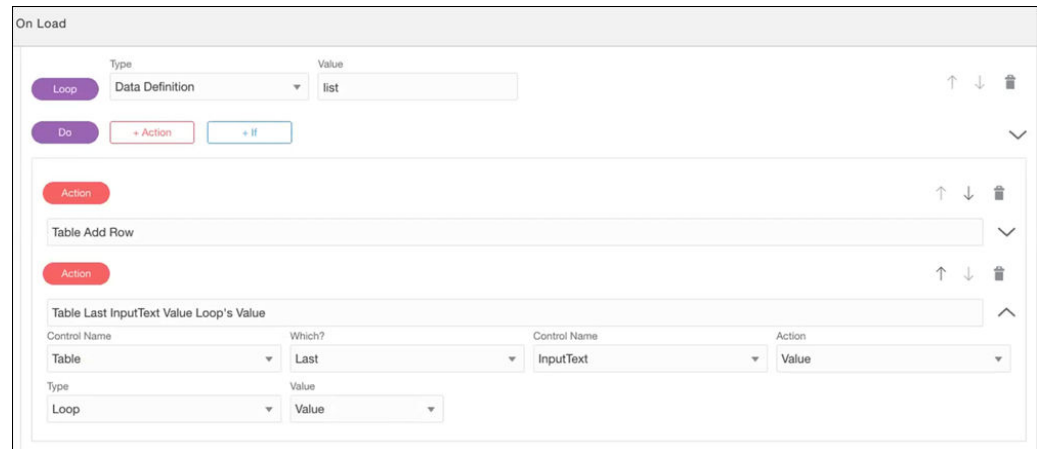
Example of a Loop with an Array

In this example, let us configure a loop with an array of strings.

1. Drag and drop a table control onto your form's canvas. Then drag and drop an InputText control into the table's column.
2. Select the table control, and in the properties pane scroll down till you find **Events**. Click **Add** to configure an event.
3. Select **On Load** from the drop-down list, and then click the editing icon.
4. In the event On Load window, click **Loop**. Select the type as **Data Definition** in the **Type** drop-down list, and enter the value as `list` in the **Value** field.

Note that `list` is already defined as an array of strings.

5. Click **+Actions**. From the **Control** drop-down list, select **Table**. From the **Action** drop-down list, select **Add Row**.
6. Configure another action. Click **+Actions**.
 - a. Select **Table** in the **Control Name** drop-down list, and specify **Last** in the **Which?** field.
 - b. In the **Control Name** that displays, select **InputText**, and then select **Value** in the **Action** drop-down list.
 - c. In the **Type** field that displays, select **Loop**, and then select **Value** from the **Value** drop-down list.



Note that the Loop type can have three options as value.

- Index: 0-based index of the iteration (0, 1, 2, 3...)
- Position: 1-based index of the iteration (1, 2, 3, 4...)
- Value: The value of the current iteration. If the iterator is a number, this value will be same as Index.

7. Click **OK**.
8. Preview the form and see how the loop works.
 - a. In the Preview window, click **Reload with Payload**.
 - b. In the Custom Payload window, specify an array, and click **OK**.



- c. You can see that the loop executes with the specified array.

On form load, the table displays two rows with the values (red, blue) specified in the array.

Execute REST Connector Calls in Events

Executing a REST call in an event enables you to store the call's response and use it in an event action or condition.

Examples

- Configure a web form that prompts users to enter a zip code. Add an event that calls a weather site to query weather values. Store the response data, then add actions to the event that display temperature-related values.
- Configure a web form that prompts users to enter a company name and click a **Get Quote** button. Add an event to the button that calls a stock service site to query stock values for the specified company. Store the response data, then add event actions that display read only stock value fields.

Before configuring a REST connector call, create the REST connector you want the event to call. See [Configure REST Connectors in Designer](#).

To add and configure the connector call:

1. In the web form editor, select a control, add an event, and select its type.
For example, include an On Change event for a text input control or an On Click event for a button control. See [Configure Events](#).
2. Click the event's **Edit** icon to display the **Events** window.
3. Click **+Connector** to add the connector call.
4. Configure the connector call.
 - a. In the **Connector** field, select a REST connector from those defined for the application.
 - b. In the **Resource** and **Operation** fields, select the resource to access and operation to perform as configured in the REST connection.

 **Note:**

You can also perform a POST operation from the event window. Currently, for POST operations, arrays aren't supported for requests but can be successfully received and handled in responses.

- c. Complete the connector call parameters (query, header, or body) that display based on the operation you choose. For example, the illustration below shows a connector call with a GET operation and a template parameter called `message`.

You can configure query parameters either from static texts or from form expressions (as shown in the example). Use **Form Expressions** to specify complex values in query parameters based on functions, payload data, and control values. Get values from payload data directly without dependency on controls by using Data Definition type. Configure functions by using the Function type, for example, specify a concat function to combine two or more values.

- d. Enter a variable name in which to store the response data (for example, `quoteResult` or `weatherInfo`).

Connector: StockService

Connector: StockService Resource: Quote Operation: getQuote

Template Parameters

message: Form Expression Type: Control Control: Company

Store response in: quoteResult

+ Error Handler

What do you want to add?

+ Action + If + Loop + Connector + Filter + Reusable Snippet

OK Cancel

5. Optionally, configure error handling. You can set actions to address all errors or a selected error detected in the call.
 - a. Click **+Error Handler** to add an error handler.
 - b. In the **Run on Error** field, select an error (or all errors) to detect.
 - c. In the **Control Name** field, select the control on which to identify the error. In the **Action** field, select the control or style action to take.
6. Collapse the connector call.
7. Below the connector call, configure actions, conditions, or additional connectors, and click **OK**.

A common use case is to add an action to display response data from the connector call. Click **+Action**, select a control, specify **Value** as its action, **Connector Data** as its type, and the response variable data to display in the **Value** field.

Connector: StockService Store response in: quoteResult

Action: LastPrice Value quoteResult.lastPrice

Control	Action	Type	Value
LastPrice	Value	Connector Data	quoteResult.lastPrice

Action: LowPrice Value quoteResult.low

Control	Action	Type	Value
LowPrice	Value	Connector Data	quoteResult.low

Action: High Value quoteResult.high

Control	Action	Type	Value
High	Value	Connector Data	quoteResult.high

+ Action + If + Loop + Connector + Filter + Reusable Snippet

OK Cancel

 **Note:**

- If the connector data is an array of elements, you can specify a particular array item in the response **Value** field by entering the corresponding array index, for example, `quoteResult[0].lastPrice[1].price`.
- If the REST response consists of a complex data object, you cannot map the data object directly into a control (such as Input Text or Text Area) through an event. Instead, you must specify which details of the complex data object should be mapped to the control, along with an appropriate function such as *Concat* (if required), in the event window. For example in a REST response, if a complex data object called **name** contains two elements, **firstname** and **lastname**, you must specify which one of these elements should be populated into the control. If you require both of these elements to be populated, use a function such as *Concat* with a suitable separator in the event window.

Populate Controls Using REST Calls

Dynamically populate controls such as drop-down select, check list, radio button, table, and repeatable section controls with data using REST connectors.

1. Drag and drop a drop-down select, a check list, a radio button, a table, or a repeatable section control onto the canvas to populate it with data using a REST call.
2. Select the control and optionally, on the **General** tab in the Properties pane, edit the name, binding, label, and other fields.
3. For a drop-down select, check list, or radio button control, in the **Options Source** field, select **Connector** and then configure the fields under the **Options Source** field to define the values based on a REST call. For a repeatable section or table control, select the **Use Data From Connector** field and configure the fields under this field.
 - a. **Connector:** Select a REST connector from the defined list of connectors for the application.

A REST connector lets you fetch data and perform tasks based on that data. See [Add a REST Connector](#).
 - b. **Resource:** Select a resource from which data should be fetched.
 - c. **Operation:** Specify an operation to call.

An operation indicates the task you want to perform. For example, a user may use an operation to fetch all the countries or to fetch a list of states based on a country code.

4. Based on the call, the system will display a list of required parameters (*payload values*) and a response below the **Options Source** field or the **Use Data From Connector** field.
 - a. Payload values: Specify the information to pass to the REST connector (parameters). You can specify query, header, or template parameters.

If you selected **Text**, enter the parameter information. Or, if you selected **Control value**, select a control value from the available options.

 **Note:**

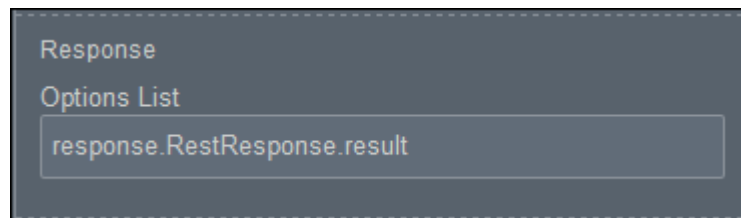
Text parameters are secure and remain on the server.

- b. **Response:** Define how the response will be mapped to the control properties:
 - **Options List:** Define the mapping by specifying an attribute list from the response that contains the items to display as the options in the control.
 - If the list is a simple type of list such as a list of strings or numbers, then no label and value mappings are needed. The value of the item in the list will be used as both the label and the value of the option.

If the list consists of complex elements, then you need to specify a mapping using the **Label Binding** and **Value Binding** fields to identify the label and value.

 **Note:**

- For a table control, define how the response will be mapped to each column in the table. For a repeatable section, define which property from the response will be mapped to which control inside the repeatable section.
- If the connector data is an array of elements, you can map a particular array item to a **Response** field by entering the corresponding array index, for example, `response[0].RestResponse[1].result`.



5. You may also use the **Skip Upon Load** property to determine when the connector data loads into the control.
 - Deselect the checkbox (default state) to allow connector data to be populated into the control when the form loads.
 - Select the checkbox to prevent connector data being populated into the control when the form loads. When you use this option, you must explicitly execute a connector refresh for the data to load into the control.
6. Optionally, in the **Events** field, configure the events for the control. See [Configure Events](#).

Test the dynamic fetching of data by testing the application. And, verify if the mappings you defined work correctly as expected.


Link and Refresh List of Value Fields

Use the Refresh Connector control action in an event so that each time an end user makes a selection in a List of Value (LOV) field, REST call data is refreshed and reflected in another LOV field. For example, in an order application form, after a user selects from a Category drop-down field, the list of products for that category only displays in a Product drop-down field.

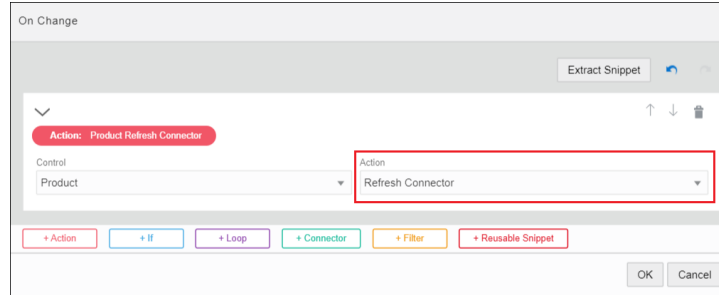
To refresh an LOV field:

1. In a web form, configure two LOV controls (select, checklist, or checkbox) to use the same REST connector to populate and return values such as categories and products. See [Populate Controls Using REST Calls](#).

In this LOV use case, the Product control has query parameters set to search for the Category control's value.

2. Select the first control (for example, Category field). On the Properties pane, scroll down to the **Events** field and add or edit an event. For example, add an **On Change** event and click **Edit**  to edit the event.
3. In the event window, add an action by clicking the **+Action** button.

- In the **Action** field that displays, select **Refresh Connector**. This setting means that each time users make a selection in the Category field, the Product field's REST call is refreshed to return product values for the selected category.



- Click **OK** to close the event window. Click **Preview** to test the function.

Preview Forms and Their Payload

At any point in configuring a web form, preview it to see its appearance at different device sizes, test its behavior, and view its payload.

- In the web form editor, click **Preview**.
- In the Preview window, switch to device sizes users might use to see your web form's layout. The web forms automatically adjust their layout according to the screen size of the device that users use to display them. Use the drop-down fields to change the language and display percent (for example, choose 150% to zoom in on a form).
- Enter sample values in the form's controls, and click **Submit**.

Notice that the field at the bottom of the Preview window displays the payload values. The payload values are the values that you specified in the web form's controls. These values are stored as data attributes for use in a process.

Note:

REST connector calls are not supported in the Preview window. You cannot make REST calls in a form's preview mode.

- Click the **Trigger Custom Outcome** button to test the form with different outcome values. In the resulting window, enter a custom outcome value and click **Submit**. Observe the form's behavior in the payload field.
- Click **Close**.

Monitoring Form Logs

Click **Log** at the bottom of the Preview window to view activity logs for a form. Monitoring these logs helps you identify and address issues with your form.

Under the **Log** tab, all activities (such as events, actions, connector calls, computed value population, and errors) occurring in your form are recorded chronologically. Expand an activity log to view the details under it, including the control on which the activity occurred, the time at which the activity occurred, and so on.

Narrow down to specific activity logs using the **Filter Logs** button. You can filter based on actions, events, errors, and so on. You can use more than one filter at a time.

Click **Clear Logs** to delete all log messages.

Reloading a Form with a Custom Payload

Click the **Reload With Payload** button to test your form with custom payload. In the resulting dialog box, enter the payload JSON. Edit the JSON to include your custom data and click **OK**. The form loads with the custom data.



Note:

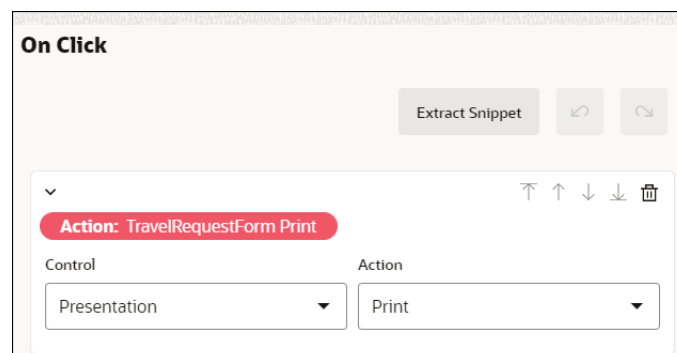
You can obtain the payload JSON from the **Outcome** tab.

Export Forms as PDF

Forms associated with tasks can be exported as PDF in runtime. To enable end users to export a form as PDF, while designing the form you've to add a button control and assign the Print action to it by configuring an event.

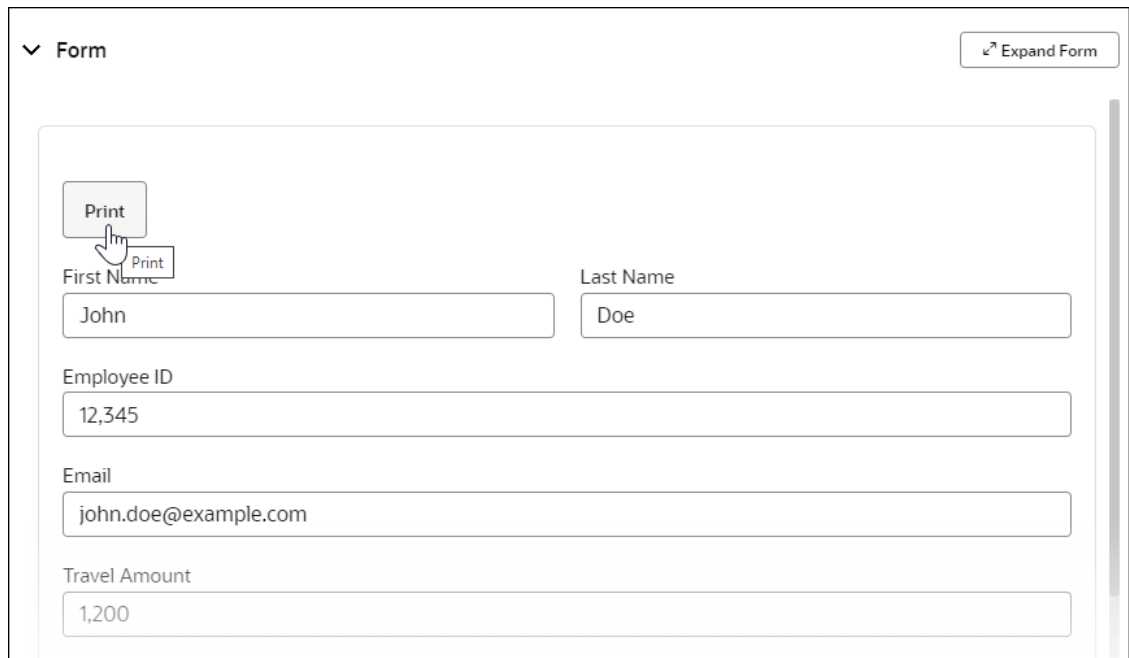
Let us explore with a simple example how you can design a form that can be exported as PDF in runtime.

1. Design a process with human task activities.
2. Design a form with different presentations to associate with the human task activities.
3. In one of the presentations of the form, configure a button control as follows:
 - a. From the Basic palette, drag and drop a **Button** control onto the form.
 - b. Select the control and edit its properties on the Properties pane's **General** tab.
 - c. Change the button control's label to `Print` so that end users can identify that it is a button to print the form.
 - d. Scroll down to the Events section, click **Add** to add an event to the **Print** button.
 - e. Select the **On Click** option from the event drop-down menu and optionally enter a name for the event.
 - f. Click the editing icon next to the events drop-down menu to define the event.
 - g. On the On Click window, click **+Action**.
 - h. In the **Control** drop-down field, select **Presentation**.
 - i. In the **Action** drop-down field, select **Print**. See [Specify Actions](#).



- j. Click **OK**.
4. Implement the form and its presentation (with the Print button) in a human task activity.
See [Implement a Human Task Activity](#).
5. Activate the process application to test the form in runtime.

Once the activated process application is instantiated in runtime, tasks get assigned to respective users. As a user assigned the task that has the form with the **Print** button (configured earlier in design-time), click the **Print** button on the form. The form gets automatically downloaded as a PDF to your local system.



The screenshot shows a web form titled "Form" with a "Print" button in the top left corner. A hand cursor is pointing at the "Print" button. The form contains several input fields: "First Name" with the value "John", "Last Name" with the value "Doe", "Employee ID" with the value "12,345", "Email" with the value "john.doe@example.com", and "Travel Amount" with the value "1,200". There is also a small "Print" button next to the "First Name" field. In the top right corner, there is an "Expand Form" button with a double arrow icon.

Configure a Linked UI

You can configure a human task activity to use a linked UI (such as an external form) for displaying the information a user needs to view (or supply) in order to complete the task.

- [Linked UIs at a Glance](#)
- [Add a Linked UI Connection](#)
- [Use a Linked UI in a Human Task](#)
- [Integrate a Linked UI with the Task List](#)

Linked UIs at a Glance

Start by familiarizing yourself with these concepts.

Why use a linked UI instead of a web form?

In a process application, you can configure human task activities to use an external, linked UI instead of a web form for interacting with end users. The linked UI feature enables you to:

- Design a complex web application that uses features not available in Process Automation forms and integrate it with Process Automation tasks
- Develop web applications using applications such as Oracle Visual Builder and Oracle APEX and integrate them with process applications

Where can I use a linked UI?

You can create and use linked UI connections in dynamic and structured process tasks.

Are there identity domain restrictions with linked UIs?

To associate a linked UI with a process application, the linked application interface must share the same identity domain as Oracle Cloud Infrastructure Process Automation.

How might I use a linked UI?

Here are some use cases:

- Associate an external web application URL as the form for a task activity in a dynamic or structured process.
- Pass process data into an external web application as a URL parameter.
- Handle business logic on actions such as task approval or saving by configuring two way javascript client side messaging between Workspace and an external application.

What's the overall process for configuring a linked UI?

1. Define a linked UI connection.
See [Add a Linked UI Connection](#).
2. Implement the linked UI connection in a human task. You can use them in structured and dynamic processes.
See [Use a Linked UI in a Human Task](#).
3. Integrate the linked UI with the runtime process task list.
See [Integrate a Linked UI with the Task List](#).

Add a Linked UI Connection

A linked UI connection identifies the URL to connect to, and its parameters so you can use the connection in a human task.

1. In a process application, click **Add**.
2. In the Add component pane, expand **UIs**, and select **Linked UI**.
3. Enter a title for the linked UI connection in the **Title** field, and click **Create**.
4. Open the new connection.
5. In the **Endpoint** field, provide the base URL of the UI you want to connect to.

Note:

You can also provide path or template parameters while specifying the linked URL, for example, `https://www.linkedui.com/{instanceId}`.

6. Under **Connection parameters**, add required query parameters for the URL.
In addition to these parameters, the task ID parameter is passed with the value corresponding to the task ID.
7. Under **What data is going to be exchanged?**, add custom payload attributes that you'll use to send/receive data from the linked application. Unlike query parameters, here you can add complex business data types as well.

 **Note:**

Through these attributes, you can use the data from a linked UI in any task within a process. However, you must perform a POST operation using the REST API to save the changes made to the payload attributes in the linked UI.

8. Exit the linked UI components using the breadcrumbs.
9. Implement the linked UI in a structured or dynamic process.

See [Use a Linked UI in a Human Task](#).

As part of implementation, configure data associations and assign values to path parameters, query parameters, and payload attributes from within a process. The final URL invoked will have the following format:

```
https://www.linkedui.com/{instanceId}?permitId={permitid}&taskId={taskid}
```


 **Note:**



You can edit the base URL of the linked UI while activating your application.

Use a Linked UI in a Human Task

You can configure a human task activity to use an external form to display information to users, so that they can provide approvals or enter data for the task.


Link in a Structured Process

1. In the structured process editor, select a human task, click **Menu**  and select **Open Properties**.
2. Select a linked UI connection.
 - a. In the Properties pane, scroll to the **UI** field under **End User Display**. All forms and linked UI connection in the application appear in this dropdown field.
 - b. Select a connection listed under **Linked UIs**.

Tip: To edit a linked UI connection, click the **Edit** icon next to the **UI** field, or choose **Open UI** from the **Menu**  with a human task selected.
3. Configure data association for the human task activity.
Select the human task activity, click , and choose **Data Association**.

From the Data Association window, assign necessary values to path parameters, query parameters, and custom payload attributes that you need to pass to the external application.

Link in a Dynamic Process

1. In the dynamic process editor, select a human task activity, and click **Edit Properties** .
2. Under End User Display on the Properties pane, select an existing external UI connector from the Form drop-down list. All external UI connectors within the application appear in the list.
3. Configure data association for the human task activity.

Select the human task activity, click , and choose **Data Association**.

From the Data Association window, assign necessary values to path parameters, query parameters, and custom payload attributes that you need to pass to the external application.

Integrate a Linked UI with the Task List

After configuring a human task activity to use a linked UI instead of a web form, complete configuration on the runtime side.

General guidelines

- As expected, the linked form is displayed when you select the task in the runtime task list.
- If a linked form is used for a human task activity, the associated external application must provide the action buttons for the task.
- When an action is performed within the task, the linked UI must make the corresponding REST API call to the Process Automation server and also fire an appropriate JavaScript event to update the runtime task list. The process task list listens to these events and refreshes accordingly.

JavaScript events a linked UI must implement

The following table lists all the JavaScript events an external UI must implement for successful integration with the process task list:

Event	Description	Example Syntax
loaded	The external form sends a loaded event, so that the parent can calculate formHeight.	<code>{source:externalForm, event:loaded, data: {formHeight : 600,showCloseIcon : true,showResizeIcon : true} }</code>
reload	The external form requests to reload itself, without refreshing the task list.	<code>{source:externalForm, event:reload, data: FUTURE}</code>
close	The external form requests to close itself, without refreshing task the list.	<code>{source:externalForm, event:close, data: FUTURE}</code>
done	The external form requests to close itself after an action, and it also requests to refresh the task list.	<code>{source:externalForm, event:done, data: {actionId : ACTION_ID}}</code>

6

Work with Human Task Activities

Human task activities enable you to model how users interact with the application. Understand how end users interact with tasks in Workspace and then learn how to create and implement human task activities in Designer.

Topics:

- [Understand End User Tasks](#)
- [Create Human Task Activities](#)
- [Implement Human Task Activities](#)
- [Customize Notification Emails](#)

Understand End User Tasks

As a end user, you have two main actions in Oracle Cloud Infrastructure Process Automation Workspace.

- Start an application, and then complete and submit a form. For example, you may submit a travel request, a sales opportunity, or an order. See [Start an Application](#).
- View and complete your assigned tasks. See [Work on Tasks](#).

Create Human Task Activities

Human task activities represents part of your process where an end user (process participant) is required to perform an action. The task can be a simple interaction, such as entering a form, or part of a more complicated workflow that requires input from multiple process participants.

Different types of human task activities enables you to model different types of interactions:

- **Submit Tasks:** Enable you to display a form that the user must submit to create a request or to provide information about a certain subject.
- **Approval Tasks:** Enable you to display a form that the user must review or complete and then perform a certain action. The user might approve or reject the request. You can also define custom actions for the user to perform. Approval tasks enable you to define an approval pattern. Generally you use the outcome of the approval task to drive the rest of the process flow.

Note that human task activities typically have a form associated with them but not always. You can configure a human task activity without an associated form.

Want to learn how to create human task activities? See:

- [Typical Workflow for Creating a Human Task Activity](#)
- [Create Submit and Approve Tasks](#)

Typical Workflow for Creating a Human Task Activity

Human task activities enable you to model how a user interacts with the application. When you add a human task activity to your process, you must also configure its implementation, define a form, and configure data associations.

When you create a human task activity you must:

1. Add a human task activity to an existing business process.
2. Configure the human task activity's implementation.
3. Create a form to display the human task activity information.
4. Configure the human task activity to use the created form.
5. Configure the human task activity's input and output arguments using data associations.

Create Submit and Approve Tasks

You can create human task activities to model user interaction with the application. Use a submit task to display a form for the user to complete and submit. Use an approve task to display a form for the user to view and/or complete, and then perform an action such as approve, reject, or a custom action that you define.

To create a submit or approve task:

1. Edit an existing business process or create a new one.
2. For a structured process - expand the **Human** category in the Activities palette, and select the human task activity (Submit or Approve). Then drag it to the process editor, and drop it on the swimlane that represents role that you want to have access to this human task activity. If you drop the task over a transition, it gets automatically added to the process flow.

The Submit or Approve task activity is added to the structured process.

3. For a dynamic process - choose a **Human Task Activity** from the **Activities** drop-down field, provide a suitable name, and click **Add**.

You can configure the human task activity as either Submit or Approve. Click **Edit Properties** to open the Properties pane for the activity. Under **Types** select either **Submit** or **Approve**.

A Submit or Approve task activity is added to the dynamic process.

4. Configure the human task activity by defining its properties from the Properties pane.

For example, select the form and presentation associated with the task. The selected form with the presentation will be available to users assigned the task in Workspace.

Implement Human Task Activities

After you create a human task activity, you have to configure its implementation details such its title and summary to identify it, its due date and priority, action rules, the UI to associate with it and so on, from the task's Properties pane

- [Assign Human Task Activities](#)
- [Configure the Title and Task Summary](#)
- [Configure Action Rules](#)

- [Use Forms or Linked UIs to Display Task Information.](#)
- [Define an Approval Pattern](#)
- [Bypass the Approval Chain](#)
- [Exclude Participants](#)
- [Configure the Due Date and Priority](#)
- [Configure Task Escalation or Expiration](#)
- [Configure Task Reminders](#)

Note that for dynamic processes, there are additional properties to configure. See [Implement a Human Task Activity](#).

Assign Human Task Activities

You can assign a human task activity to a specific user/users, or to users in a certain role.

To assign a human task activity in a *structured* process:


1. Select the human task activity in the structured process, and open its Properties pane.
2. In the Properties pane, scroll to the **Assignees** section.
3. In the **Select Participants** drop-down field, select one of the following:
 - **Current Lane Participants:** Choose this to assign based on the swimlane's role. When a task is completed, to automatically assign to the same approver in the swimlane, choose the **Assign to previous task approver in current lane** checkbox.
 - **Individual Assignees:** Choose this to assign based on Users, Roles, and Expressions. If you selected Users or Roles, a search field appears where you can search for users or roles.

If you selected Expressions, enter an expression in the **Create Expression** field that determines a user or role to assign the task.

 - You can create an expression with user name, user ID, role name or role ID.
 - Both application and global roles are supported for creating the expression. However, note that if you create an expression with a role name and if the *same* role name is used for both an application and a global role, then by default the task will be assigned to the application role in runtime.
 - If you want to dynamically assign a task to a number of users, you can do so by configuring an expression that resolves to a comma separated list of user IDs. The task gets assigned to the users - all at once, in sequence or in parallel, depending on the policy selected. If instead of a list of user IDs, you want to use a list of user names, you can use the `IdentityService.getUserIds` function in the expression. This function converts a list of user names to user IDs. See details about the function under the Identity Service category in [Work with Expressions](#).
4. For an approval type human task activity, a **Policy** field is displayed. From the **Policy** drop-down list, select one of the following.
 - **Any Single Assignee**
 - **All Assignees in Parallel**
 - **All Assignees in Sequence**
 - **Management Chain in Sequence**

5. If you selected **All Assignees in Parallel**, then you've to specify values for the **Percentage Required** and the **Default Outcome** fields.
For example, if you set the percentage required as 55% and the default outcome is APPROVE. This means that a minimum of 55% of task assignees must approve the task for the task's outcome to result in APPROVE. Otherwise, the outcome is REJECT.
6. If you selected **Management Chain in Sequence**, specify how to seek approval: going up one level, going up a specified number of levels, or going up until a specified title is reached.
 - Choose **Immediate Manager** to go up one level from the assignee.
 - Choose **Until Level or Title Reached** to continue seeking approval until the specified level or title is reached.
 - Specify a number between 2 and 10 in the **Levels Up to Reach** field.
 - (Optional) Specify a title to reach by selecting **Title to Reach** and entering a title to match. If the title isn't found, approval continues until the level specified in the **Levels Up to Reach** field is reached.

To assign a human task activity in a *dynamic* process:

- Select the human task activity in the dynamic process, and open its Properties pane.
- Under **Assignees**, click .
- If you selected Users or Roles, a search field appears where you can search for users or roles.

If you selected Expressions, enter an expression in the **Create Expression** field that determines a user or role to assign the task.

Configure the Title and Task Summary

You can specify a title to identify the human task activity, and a summary to describe it. The title and the summary appear in the user's task list, so that they can easily identify the task they're looking for without having to view the details. You can specify the title and the summary using plain text, or generate it using expressions.

To configure the title and task summary:

1. Select a human task activity in a process, and open its Properties pane.
2. Under **End User Display** use one of these options to enter a title and summary for the **Title** and **Task Summary** fields:
 - **Literal mode**: Enter a literal value.
 - **Expression mode**: Enter a simple expression.

Note:

Your plain text string and your expression are maintained so that you can toggle between the two options without losing any data.

Configure Action Rules

Process designers can control the system actions that assignees can perform on a task. Use this functionality when you want to control end-user behavior to prevent confusions or deviations from an intended business flow. For example, you can prevent an assignee from reassigning the task. Process designers can also define custom actions for the user to perform and also enforce that the user provides a comment while performing a custom action. Comments on the custom action allows users to understand the context or the reason why a particular action was taken.

By default, all the predefined system actions are enabled. Users with Manage permissions can always access all the predefined system actions. You can't restrict any system action to a user with Manage permissions.

To configure an action rule in a *structured* or *dynamic* process:

1. Select the human task activity in the structured process, and open its Properties pane.
2. In the Properties pane, scroll to the **Actions** section.
3. To configure custom actions, in the **Custom actions** field, enter one or more custom actions as per the requirement. For example, you can type Submit, Reject, or Approve as the custom actions.
4. To enforce \comments for the custom actions, in the **Mandate comments on action** field, select the custom action for which you want to enforce a comment.
5. In the **System actions** field, by default the following actions are selected:
 - **Escalate**
 - **Reassign**
 - **Release**
 - **Withdraw**
 - **Request Info**

You can retain the required system actions and remove the unwanted actions.


Use Forms or Linked UIs to Display Task Information

You can configure the human task activity to use a specific form or a linked user interface (such as an external form) to display the information the user needs to view or complete to perform the task assigned to them.

Using Forms to Display Task Information

To associate a form with a human task activity:

1. Select a human task activity in a process, and open its Properties pane.
2. In the Properties pane, scroll to the **End User Display** section.
3. In the **UI** drop-down field, select a form to associate with the task.

Note that once you've selected a form, you can click  next to the **UI** field, and open the form in the form editor to edit it.

4. From the **Presentation** drop-down field, select a presentation for the form.

5. Select **Bind to process data** if you want to automatically bind form data. Notice that the Task Payload and Task Outcome fields get populated.

Keep in mind that when you implement a human task activity with a form, data association is automatically performed when a form is selected:

- If the data object already exists, then that one is used.
- If the data object with the same name doesn't exist, then the first data object of the same type is used.
- If there's no data object of the same type, then a new data object is automatically created. New data objects use this naming convention:

```
<form-name-starting-in-lower-case>DataObject<n>
```

where *n* is a number added to avoid duplicate names.

- Name restrictions for data objects and forms are similar, so no special treatment is required.
- After data association is performed, you're notified with a message below the form input box.

Using linked UIs to display task information

To use a linked user interface like an externally designed form to display task information, see [Use a Linked UI in a Human Task](#).

Define an Approval Pattern

Approval human task activities let you define an approval pattern by specifying actions. By default, the actions APPROVE and REJECT are already specified. However, you can also define custom actions, such as HOLD and MOREINFO.

To define an approval pattern:

1. Select an approval human task activity in a process, and open its Properties pane.
2. In the Properties pane, scroll to the **End User Display** section.
3. In the **Action** field, enter custom actions if required. Notice that the actions APPROVE and REJECT appear by default.

Custom actions must be in all uppercase letters. Each action must be separated by a comma.

Bypass the Approval Chain

You can bypass the approval chain for a specified action on approval type human task activities. For example, if you set the policy to be **All Assignees in Sequence** and the second of four approvers rejects the task, you can use this feature to bypass the remaining two approvers.

To bypass the approval chain:

1. Select the approval human task activity, and open its Properties pane.
2. In the Properties pane, scroll to the **Assignees** section.
3. Select the **Skip Approval** check box to activate the drop-down menu.
4. Select the action that you want to skip approval on.

If you add or remove an action in the **Action** field, the drop-down menu for skipping approvals automatically updates to reflect the addition or deletion. See [Define an Approval Pattern](#).

Exclude Participants

Excluding participants lets you control who can work on tasks such as approving the tasks.

You can control the users that can approve a task by excluding any user from acting on a task. For example, you can ensure that requestors cannot approve their own requests, and prevent unauthorized users from acting on a task. If you want to exclude multiple users, you can specify an expression. You can enter comma separated usernames and user IDs in the expression.


To exclude participants:

1. Select a human task activity in a process, and open its Properties pane.
2. In the Properties pane, scroll to the **Assignees** section.
3. In the **Exclude the following** section, select the required checkboxes:
 - **Creator**
 - **Expression**
If you select the **Expression** checkbox, the **Enter an expression** field appears where you can type the required expression.

Configure the Due Date and Priority

You can specify a due date and a priority for a human task activity. After the due date is reached, the task is marked as overdue.

To configure a task's due date and priority:


1. Select a human task activity in a process, and open its Properties pane.
2. In the Properties pane, scroll to the **End User Display** section.
3. In the **Due Date** field enter an interval to specify the amount of time the assignee has to complete the instance after the task is triggered. Select one of the following options:
 - *Manually*: Use the format **##M##d##h##m**. For example:
 - One hour and thirty minutes: 1h30m
 - One day: 1d
 - Four months, two days, eleven hours and thirty minutes: 4M2d11h30m
 - *Expression Editor*: Enable the expression mode by clicking  and specify an expression to calculate due date of the task.

Note:

A task due date is different from a process due date. Setting a due date for a process does not automatically set a due date for the task.

4. Under **Priority**, select an option from the **Level** drop-down list.
 - High (1)

- Normal (2)
- Low (3)



In addition, you can use expressions to dynamically set a value for the priority during human task activity creation. Click the literal mode icon in the **Level** field once to enable the expression mode . Use the inline expression builder to specify an expression. The expression should evaluate to an integer. This is suggested by a hint text on the top-right of the field.

Note that when you use an expression to set a task's priority, if the expression returns a value that doesn't fall under the predefined range, then by default the task priority will be set to Normal (2).



In Workspace, you can sort task list based on the task priority. High priority tasks are marked with a red exclamation mark.

Configure Task Escalation or Expiration

You can configure a human task in a structured or dynamic process to never expire, to expire after a certain time, or to escalate after a certain time passes.

1. Select the human task activity in a structured or dynamic process. Notice that the **Escalation and Expiration** icon  displays on the sidebar menu of the process editor when you select a human task.
2. Click **Escalation and Expiration**  to open the Escalation and Expiration pane.
3. Use the radio buttons to specify if you want the human task to **Never expire**, **Expire**, or **Escalate**.
 - **Never expire**: the human task doesn't expire and if no user completes it, it remains in the users task list for an indeterminate period of time. Note that by default, a human task is set to never expire.
 - **Expire**: the human task expires after the specified time and is no longer accessible from the task list.
 - **Escalate**: when the specified time passes, the human task is escalated to the specified users and/or roles.
4. If you chose the human task to expire or escalate, specify the interval to wait before performing this action.

You can specify the duration of the interval to wait using one of the following options:

- **Manually** : Use the format ##M##d##h##m. For example:
 - One hour and thirty minutes: 1h30m
 - One day: 1d
 - Three months, two days, eleven hours and thirty minutes: 3M2d11h30m
 - **Expression Editor** : Enable the expression mode and specify an expression to dynamically calculate the duration.
5. If you escalate the human task, additionally you can do the following:
 - Assign the task to users and/or roles. In runtime, after the task is escalated, it will be assigned to the users and/or roles that you specify in this field. You can also dynamically assign the task using the expression editor.


- If required, you can update the default text in the mandatory **Comment** field to a customized text or configure it as a dynamic comment using the expression mode.
6. Close the Escalation and Expiration pane to save the changes.
- **Escalation:** In Workspace (after the process application has been activated), if the task hasn't been acted upon till the specified interval duration, the task will be assigned to the users and/or roles that you specified in the Escalation and Expiration pane. If it is escalated to a single user, then the task's state changes to *Assigned* with the sub-state *Escalated*; and the process waits for the completion of the task. If it is escalated to one or more users and/or roles, then the task's state changes to *Unassigned* with the sub-state *Escalated*; and the process waits for the completion of the task.
 - **Expiration:** In Workspace (after the process application has been activated), if the task hasn't been acted upon till the specified interval duration, the task's state changes to *Completed* with the sub-state *Expired*; and the process moves to the next activity in the process flow. The task will no longer be available under *In Progress* tasks, and will be available under *Completed* tasks. If you open the audit details of the task from the Workspace [Tracking](#) page, you can see that the state of the task is marked as *Expired*, along with other details such as the expiration date and time.

A notification email is sent to the task assignee that the task has expired.

Configure Task Reminders

You can specify reminders for a human task activity in a structured or dynamic process.



To configure reminders for a task:

1. Select the human task activity in a structured or dynamic process. Notice that the **Reminders** icon  is displayed on the right of the process editor when you select a human task.
2. Click **Reminders** to open the Reminders pane.
3. From the **Reminder** drop-down list, select the number of times to send reminders to the task assignees to complete the assigned task.

Note that by default the **No Reminder** option is selected.

- **Remind Once**
- **Remind Twice**
- **Remind Three Times**

When you select one of the reminder options other than the default No Reminder option, the **Interval** and **When** fields are displayed

4. Click the **Interval** field to specify the reminder duration. Select one of the following options:
 - **Manually** : Use the format `##M##d##h##m##s`. For example:
 - ten minutes: 10m
 - two minutes and thirty seconds: 2m30s
 - **Expression Editor** : Enable the expression mode and specify an expression to calculate the reminder duration of the task.

Note that the expression needs to evaluate to the format `##M##d##h##m##s`.

5. From the **When** field, select when you want the reminders to be sent for the task.

- Choose **Before Expiration** if you want the reminders to be sent before the task expires.
- Choose **After Assignment** if you want the reminders to be sent right after the task has been assigned.
- Choose **Before Due Date** if you want the reminders to be sent before the due date of the task.

The screenshot shows a 'Reminders' configuration pane for 'Activity: User task 1'. It includes a 'Reminder' dropdown set to 'Remind Twice', an 'Interval' field set to '30m' with a 'Duration' button, and a 'When' dropdown set to 'Before Due Date'.

6. Close the Reminders pane to save the changes.

If the task hasn't been acted upon after assignment or till the specified duration of time, notification emails are sent to the task assignees to remind them about the action they need to perform on the task.

Customize Notification Emails

Keep your users informed about their task assignments and the progress of a process. You can easily configure notification emails for human task activities and create templates for those notifications.

- [About Notification Email and Templates](#)
- [Configure Email Notifications](#)
- [Manage Email Templates](#)
- [Configure Email Templates](#)

About Notification Email and Templates

You can configure Oracle Cloud Infrastructure Process Automation to send a notification email to the task assignee when an event such as assignment, approval, escalation, reminder, and reassignment occurs. A notification email contains outcomes defined for a task, such as Approve or Reject. Clicking on the outcomes will take the assignees to Workspace where they can complete their assigned tasks.

You can also configure and use customized notification emails for human task activities. You have an option to create customized email templates that include the following information:

- Payload
- Task
- Links to perform an action on a task
- Task comments

About Actionable Emails

You can configure a notification email with actions that contains outcomes defined for a task, such as Approve or Reject. Clicking on these outcomes takes you to Workspace where you can complete the assigned task. In addition, you can add comments and attachments to the task.


About Default Email Templates

If you enable email notification for a task, you have the option of selecting the default email template for your use. The default email template contains outcomes defined for a task, such as Approve or Reject. You can also add comments and attachments to the task.

Configure Email Notifications

Configure email notifications and specify when an email is sent to the task assignee. You can also customize email notifications so they contain all the information recipients need to easily recognize and act on them.

To configure email notifications:

1. Open a process in the process editor.
2. Select a human task activity for which you want to configure email notification, and then click **Notifications**  on the right side of the process editor.
3. In the Notification pane, disable or re-enable email notifications.

Note that notifications are disabled by default. Note that disabling the notification for a task won't disable reminder or expiration emails.

Note:

If you've configured the human task activity to use a linked UI instead of a web form, the options to include the form as image, PDF, or inline are not available on the **Notification** tab.

4. In the **Email Template** field, select an email template that you want to use for the task-related emails.
5. Specify when the email notification is sent, for example when a task is assigned or completed, or when an error occurs. By default, **On Assign** and **On Complete** are selected. Use the drop down to determine the template for each notification email. The **Use Parent Template** is selected by default.


The parent template uses the template you selected before. You can either use the same template or select any other template. To manage email templates, see [Manage Email Templates](#).

- Optionally, you can select **Task Attachments** to include files that users have uploaded to tasks.



Manage Email Templates

Use a customized notification email for human task activities that includes basic task information, form payload values, and comments, and that optionally lets end users perform a task action without having to sign in to the application. Create customized notification emails from the Notification pane of a selected task.

To assign an email template to a human task:

- Open a process in the process editor.
- Select a human task activity, and click **Notifications** .
- In the Notification pane, scroll down to the **Manage Email Templates** section.
- Select an email template that you want to use for the task-related emails.

You can do either of the following:

- Create a new email template for your use. Click **Add**  to the right of **Manage Email Templates**. See [Configure Email Templates](#).
- Copy, preview, or download an existing email template for your use. Click  next to the email template that you want to use, and choose from the available options.

Option	Description
Copy	Copy and save the email template with another name.
Preview	Open the email template in the Preview Email Template dialog. You can view how the email appears to end users. The payload information isn't displayed in the Preview mode.
Download	Download and save the email template in HTML format to your local drive or select an application to open the file.

Configure Email Templates

Create customized email templates and use them to send notification emails for human task and notify task activities. You can copy an existing email template and modify it as per requirements.

In this topic, you can learn:



- [Create an Email Template for Human Tasks](#)
- [Create an Email Template for Notify Tasks](#)

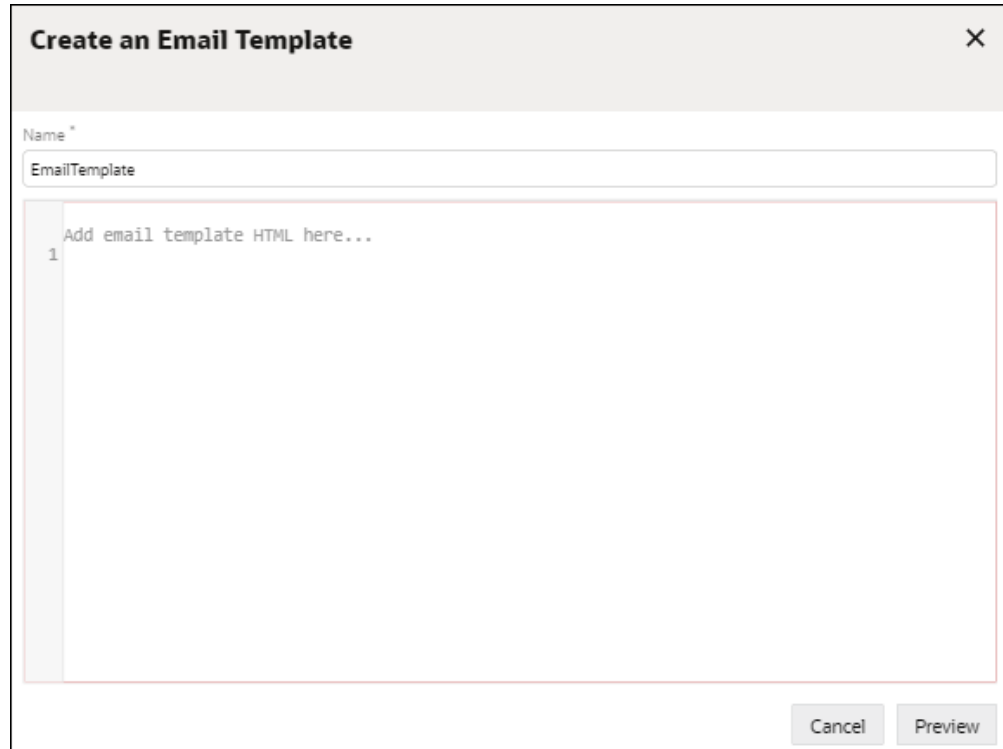
Create an Email Template for Human Tasks

You can create an email template by manually entering the HTML markup directly into the code editor provided in the Create an Email Template pane.

To create a new email template:

- Open a process in the process editor.

2. Select a human task activity, and click **Notifications**  on the right side of the process editor.
3. In the Notification pane, go to **Manage Email Templates**, and click . The Create an Email Template pane opens.



4. Enter the HTML markup for your email template in the space provided for it. To include values from process variables and form arguments, you'll use mustache templates. Mustache templates provide a way to include [predefined](#) task variables, and values in an HTML markup. The following example shows how to reference the first name and last name with a data object using the mustache template:

Template:

```
{{#payload}} {{#DOName}} {{AttributeName}} {{/DOName}} {{/payload}}
```

Object:

```
<div class="container">
  <div class="label">Hello {{#payload}} {{#DOName}} {{firstName}} {{/DOName}}
  {{/payload}} {{#payload}} {{#DOName}} {{lastName}} {{/DOName}} {{/payload}}
  </div>
</div>
```

 **Note:**

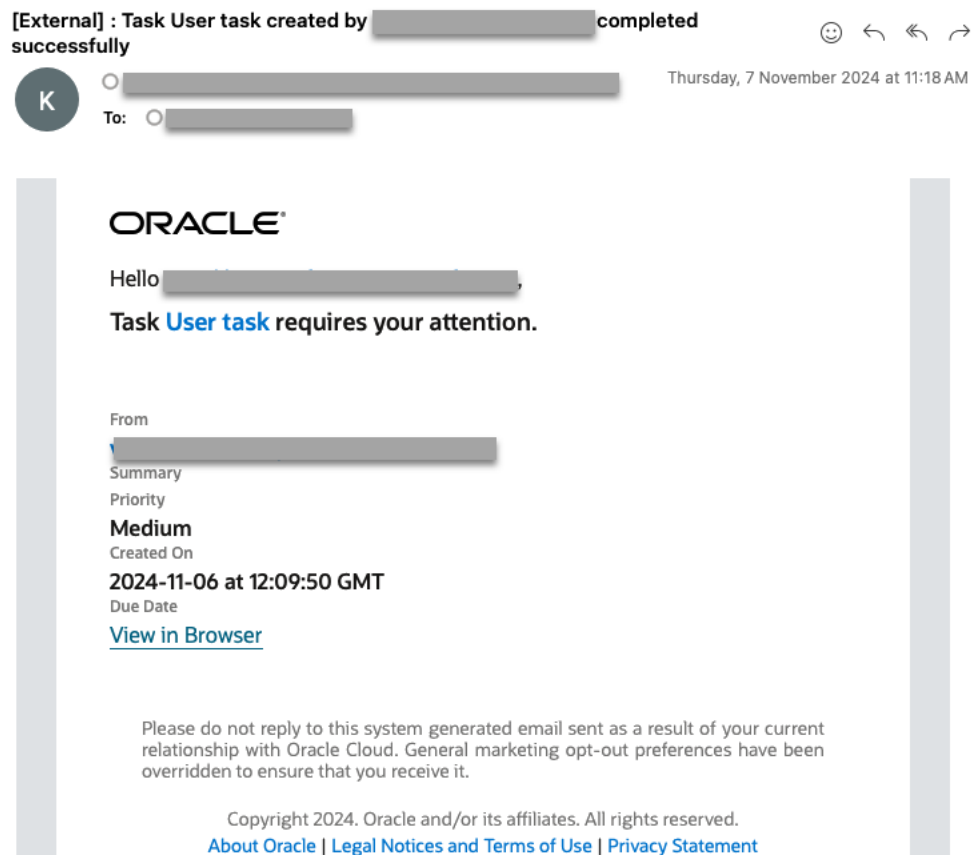
To use values from process or task variables that are not **predefined**, associate the variables with a data object before referencing them as shown in the example.

Output:

Hello Joe Smith

See [Defining tags using mustache templates](#).

5. Click **Preview** to view your template before finalizing it.



[External] : Task User task created by [redacted] completed successfully

Thursday, 7 November 2024 at 11:18 AM

ORACLE

Hello [redacted],

Task **User task** requires your attention.

From [redacted]

Summary [redacted]

Priority

Medium

Created On

2024-11-06 at 12:09:50 GMT

Due Date

[View in Browser](#)

Please do not reply to this system generated email sent as a result of your current relationship with Oracle Cloud. General marketing opt-out preferences have been overridden to ensure that you receive it.

Copyright 2024. Oracle and/or its affiliates. All rights reserved.
[About Oracle](#) | [Legal Notices and Terms of Use](#) | [Privacy Statement](#)

 **Note:**

In preview mode, the payload and action map sections don't show the data.

Predefined variables of human task email notification

The following table provides a description of the predefined variables that you can use in a human task email notification template. Note that the variable names are case-sensitive.

You can refer the variables using the following format:

```
{{name_of_the_predefined_variable}}
```

For example:

```
{{title}}, {{assignee}}
```

Variable	Description	S y n t a x
assignee	Username of the task assignee.	{ { a s s i g n e e } }

Variable	Description	S y n t a x
actions	List of actions.	Y o u c a n d e f i n e t h e a c t i o n s a s s h o w n i n t h e f o l l o w i n g s y n t a

Variable	Description	S
		y n t a x
		x :
		{ { #
		h a s A c t i o n s
		} }
		< d i v
		c l a s s = " b u t t o n - c o n t

Variable	Description	S y n t a x
		a i n e r " >
		< d i v
		c l a s s = " l a b e l " >
		A c t i o n s < / d i v >

Variable	Description	S y n t a x
----------	-------------	----------------------------

<
d
i
v

c
l
a
s
s
=
"
b
u
t
t
o
n
s
"
>

{
{

a
c

Variable	Description	S y n t a x
		t i o n s } }
		< a c l a s s = " a c t i o n - b u t t o n

Variable	Description	S y n t a x
----------	-------------	----------------------------

"

h
r
e
f
=
"
{
{
w
o
r
k
s
p
a
c
e
U
R
L
}
}"
>

<

Variable	Description	S y n t a x
		s p a n
		t i t l e
		=
		"
		{
		{
		a
		c
		t
		i
		o
		n
		D
		i
		s
		p
		l
		a
		Y
		N
		a
		m
		e
		}
		}"
		>{
		{
		a
		c
		t
		i
		o
		n
		D
		i
		s
		p
		l

Variable

Description

**S
y
n
t
a
x**

a
y
N
a
m
e
}
}<
/
s
p
a
n
>

<
/
a
>

{

Variable	Description	S y n t a x
		{ / a c t i o n s }
		< / d i v >
		< / d i v > { { / h a s A c t

Variable	Description	S y n t a x
		i o n s } }

Variable	Description	S y n t a x
actionDisplayName	Display name of the action. For example, Approve, Reject.	Y o u c a n d e f i n e t h e a c t i o n d i s p l a y n a m e a s s h o w n i n t h e f o l l o w i n g s

Variable	Description	S y n t a x
		y n t a x : { { # h a s A c t i o n s } } < d i v c l a s s = " b u t t o n - c

Variable	Description	S y n t a x
		o n t a i n e r " >
		< d i v
		c l a s s = " l a b e l " >
		L i s t
		o f
		T a

Variable	Description	S y n t a x
----------	-------------	----------------------------

s
k

A
c
t
i
o
n
s
<
/
d
i
v
>

<
d
i
v

c
l
a
s
s
=
"
b
u
t
t
o
n
s
"
>

Variable	Description	S y n t a x

{
{

a
c
t
i
o
n
s

}
}

<
a

c
l
a
s

Variable	Description	S y n t a x
		s = " a c t i o n - b u t t o n " h r e f = " { { w o r k s p a c e U R L } }" >

Variable	Description	S y n t a x
----------	-------------	----------------------------

<
s
p
a
n

t
i
t
l
e
=
"
{
{
a
c
t
i
o
n
D
i
s
p
l
a
y
N
a
m
e

Variable	Description	S y n t a x
		} } " >{ { a c t i o n D i s p l a y N a m e } }< / s p a n >

<

Variable	Description	S y n t a x
----------	-------------	----------------------------

/
a
>

{
{
/

a
c
t
i
o
n
s

}
}

<
/
d
i
v
>

Variable	Description	S y n t a x
createdDate	Date of creation of the task	<pre>{ createdDate }</pre>
creator	Stores the username of the creator.	<pre>{ creator }</pre>

Variable	Description	S y n t a x
dueDate	Due date for completing the task	{ { d u e D a t e } }
expirationDate	Expiry date of a task.	{ { e x p i r a t i o n D a t e } }
hasActions	A Boolean value that indicates if an action is configured for a task.	{ { h a s A c t i o n s } }
logo	Logo	{ { l o g o } }

Variable	Description	S y n t a x
priority	Task priority	{ { p r i o r i t y } }
payload	Map of task payload	{ { p a y l o a d } }
shortSummary	Task summary	{ { s h o r t S u m m a r y } }
taskId	Task ID	{ { t a s k I d } }

Variable	Description	S y n t a x
title	Title of the task	{ { t i t l e } }
url	URL for accessing the task details in runtime.	{ { u r l } }
updatedBy	Username of the user who updated the task.	{ { u p d a t e B y } }
updatedAt	Date the task was updated.	{ { u p d a t e D a t e } }

Predefined payload data and variables in human task notifications

Process Automation uses Mustache templates to display data. See [Mustache](#) and [Mustache 5](#).

To define a...	Example
Variable	<p data-bbox="696 247 1425 275">Reference the predefined variables as shown in the following format:</p> <pre data-bbox="696 285 821 312">{{title}}</pre> <p data-bbox="696 323 1425 380">To use multiple predefined variables, use comma as a separator. For example:</p> <pre data-bbox="696 422 1016 449">{{title}}, {{assignee}}</pre> <p data-bbox="696 501 1187 529">See the list of supported predefined variables.</p>

To define a...	Example
Payload	<p>To get the structure of the payload, see the incoming webform dataObject in the Data Association page of the human task activity. You can reference the following:</p> <ul style="list-style-type: none"> • Process level data objects • Form argument data objects • User defined data objects <p>.</p> <p>Payload Example 1: Format of a Form with form and business data objects</p> <pre> {{#payload}}{{#formDO}}{{inputText1}}{/formDO}}{/payload}} {{#payload}}{{#myFormArgs1}}{{inputText2}}{/myFormArgs1}}{/payload}} {{#payload}}{{#myFormArgs2}}{{inputText3}}{/myFormArgs2}}{/payload}} </pre> <p>Payload Example 2: Form with form and business data objects using HTML tags</p> <pre> <div class="info"> <div class="container"> <div class="label">Loan Application 1</div> <div class="value">{{#payload}} {{#appDataObject}}{{appName}}{/appDataObject}}{/payload}}</div> </div> <div class="container"> <div class="label">Loan Application 2</div> <div class="value">{{#payload}} {{#myFormArgs}}{{appStatus}}{/myFormArgs}}{/payload}}</div> </div> <div class="container"> <div class="label">secondLevel</div> <div class="value">{{#payload}} {{#accessSecondLevelValue}}{{#firstLevel}} {{secondLevel}}{/firstLevel}}{/accessSecondLevelValue}}{/payload}}</div> </div> <div class="container col"> <div class="created"> <div class="label">Created On</div> <div class="value">{{createdDate}}</div> </div> <div> <div class="label">Due Date</div> <div class="value" </pre>

To define a...**Example**

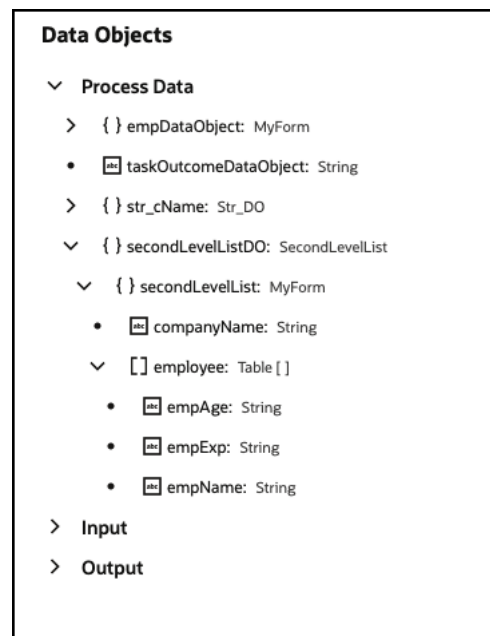
```

amber">{{dueDate}}</div>
    </div>
</div>
<div class="container link">
    <a class="view-link" href="{{url}}">
View in Browser</a>
    <div class="underline"></div>
</div>
</div>

```

Array loops

You can define arrays containing information in the form of elements. You can define the elements at different levels. For example the following screenshot shows how to define employee information at the second level.



Sample code to define array at second level. In this sample code, the employee name, age and experience details are defined as elements at the second level.

```

{{#payload}}{{#secondLevelListDO}}
{{#secondLevelList}}{{#employee}}
<li>Employee Name:{{empName}}<br>Employee
experience: {{empExp}}<br>Employee age:
{{empAge}}</li>
{{/employee}}{{/secondLevelList}}{{/
secondLevelListDO}}{{/payload}}

```

Create an Email Template for Notify Tasks

If you have an email template, you can manually enter the HTML markup directly into the code editor provided in the Create an Email Template pane and modify the markup as per requirements.

To reference data objects in the email template, use the following code format:

```
${DOName.prop('AttributeName')}
```

For example:

```
${loanApplicationDataObject.prop('loanAmount')}
```

 **Note:**

Email templates for notify tasks do not support:

- Referencing process variables that are not part of a data object.

 **Note:**

To use values from process or task variables that are not predefined, associate the variable values with a user defined data object before referencing.

- Predefined variables
- Array data in data objects

7

Model Decisions

Create decisions to automate the decision logic inherent in your business process. As part of modeling decisions, define their input, model their logic using simple and complex types, test them, and expose them as services.

Topics:

- [Decisions at a Glance](#)
- [Understand Local and External Decisions](#)
- [Create a Local or External Decision](#)
- [Understand the Graph View](#)
- [Add a Decision Logic Type](#)
- [Define Expressions with the Friendly Enough Expression Language \(FEEL\)](#)
- [Define Decision Input and Type](#)
- [Model Decision Logic](#)
- [Test Decision Models](#)
- [Expose Decisions as Services](#)
- [Add Decision Services to Processes](#)

Decisions at a Glance

Start by familiarizing yourself with these decision concepts.

How do decisions work in process applications?

In business, decisions are everywhere. Should this loan application or document change be approved? Should emergency vehicles be dispatched to this incident? How many bonus shopping points is this shopping cart worth? Use decisions to automate and reuse decisions throughout your business processes.

Use the Decision editor as a canvas to define, test, and output decision logic for process applications. The editor supports the Decision Modeling and Notation (DMN) standard, version 1.1, and uses FEEL (Friendly Enough Expression Language) to make decision modeling easier and more intuitive.

To learn more about creating decision tables or complex logic structures such as functions, relations, or contexts, see [Model Decision Logic](#).

To learn how to use FEEL to define expressions within all notations of decision logic, see [Define Expressions with the Friendly Enough Expression Language \(FEEL\)](#).

What are the types of decisions that you can use in your processes?

You can use two types of decisions in processes - local decisions and external decisions.

Local decisions can only be used in the context of the process application in which they're created. They are typically used for simple use cases.

External decisions can be used to automate complex decision logic in business processes. External decisions are standalone decisions that are created and activated in decision applications. They can be called in more than one process applications as external decisions. In addition, you can also use these type of decisions in external applications by calling the active REST API endpoints exposed by the decision services.

See [Understand Local and External Decisions](#).

How do I add a decision component?

Add a decision as you would add any other Process Automation component. See [Create a Local or External Decision](#).

What do DMN and FEEL stand for?

The editor supports the Decision Modeling and Notation (DMN) standard, version 1.1, and uses FEEL (Friendly Enough Expression Language) to make decision modeling easier and more intuitive.

See [Define Expressions with the Friendly Enough Expression Language \(FEEL\)](#).

How do I model the actual decision?

You'll select from a variety of decision elements, define their input and types, and model your decision logic.

- [Add a Decision Logic Type](#)
- [Define Decision Input and Type](#)
- [Model Decision Logic](#)

Can I test a configured decision?

While modeling a decision, quickly test its logic by entering sample input values and viewing test outcomes.

See [Test Decision Models](#).

Why do I need a decision service?

Creating a service exposes one or more output decisions as public REST APIs. A decision service consists of a set of input data and a set of output decisions from the containing decision model. You then use the decision service in your structured process to implement the decision.

See [Expose Decisions as Services](#).

How do I add a decision service to a structured process?

After exposing your decision as a decision service, add the decision service in your structured process where it's needed.

See [Add Decision Services to Processes](#).

What should I know before I create a decision?

- Decisions are supported for structured processes.

- Nested or subdecisions are not supported.
- You can create multiple decision services for a decision.

Understand Local and External Decisions

Local decisions have to be modeled from scratch and are created and activated in the context of process applications. **External decisions** are ready to be used, already modeled decisions that are created and activated in decision applications. They can be called and used across multiple process applications as well as in other external applications.

Here's a comparison of local and external decisions.

Local decision	External decision
Created as a component of a process application.	Created as a component of a decision application.
Can be used only in the context of the process application in which it is created. Local decisions support simple use cases.	Gives more flexibility to designers as external decisions can be used in multiple process applications and also can be called by external applications. External decisions support complex use cases.
Only one decision can be created per process application.	More than one decisions can be used per process application.
Decision services of a local decision become active only when the process application is activated. The REST API endpoints of the exposed decision services are not readily active. They become active only when the process application is activated.	As only activated decision applications and their versions are available as external decisions, so decision services of external decisions are always active. The REST API endpoints of the exposed decision services are already <i>active</i> and can be readily called from other process applications and external clients.
Versions aren't supported	Versions are supported. If multiple versions exist for a particular decision application, you can only use one version per process application. If required, you can change to a higher or lower version but cannot use more than one version of the same decision application in a process application.

Some external decisions may have badges to indicate their status.

- **New version available:** There is a new version available for the current decision.
Under the **Versions** section of the external decision details window, the current and new versions are listed. If you want to use the new version, click the **Use** button. You'll be prompted to confirm. On confirmation, the decision gets updated to the new version.
You'll get a warning if the new version that you are trying to use is incompatible and hence can cause validation errors. You can choose to cancel the update or update anyway and fix the errors later.
- **Update model:** The linked decision application (current version) has been updated and activated.
Click and open the external decision details window. Under the **Versions** section, the currently used decision is listed with a **Update** button. Click **Update** and then confirm when prompted to update to the latest version.
- **Missing decision:** The linked decision application has been deactivated or deleted.

As a workaround, you can delete the decision and use an available active decision application.

Create a Local or External Decision



To use decisions in business processes, first you have to create a decision component in your process application. A decision is a container for decision logic, input data, and decision services.

You can either create a **local decision** that have to be modeled from scratch or create an **external decision** by using an already modeled decision from a decision application.

Before adding a decision component to a process application, understand the differences between local and external decisions. See [Understand Local and External Decisions](#).

1. In the process application's main page, click **Add**.
2. In the Add component pane, expand **Decisions** and choose one of the following:
 - **Local decision**: Select this if you want to create a decision from scratch for your process application. This type of decision can only be used in the context of the current application.
 - **External decision**: Select this if you want to use an already modeled external decision from an activated decision application.
3. If you select **Local decision**, enter the following information in the resulting pane:
 - a. **Title**: Enter a meaningful name for the decision.
 - b. **Identifier Name**: This is the unique identifier of the decision. By default, this value is same as the title. But you can edit it if required.
 - c. **Description**: Enter a suitable description for the decision.
4. If you select **External decision**, then all activated decision applications and their versions are displayed. Select an activated decision application from the list.
5. Click **Create**.

The decision component gets listed in the Decisions page of your process application. You can differentiate between local and external decisions by their icons.

- Local decision: 
- External decision: 

Local decision: For a local decision, an empty decision component is added. Clicking it will open the decision editor canvas. You can start modeling the decision as described in the following topics:

- [Add a Decision Logic Type](#)
- [Define Expressions with the Friendly Enough Expression Language \(FEEL\)](#)
- [Model Decision Logic](#)
- [Test Decision Models](#)
- [Expose Decisions as Services](#)
- [Add Decision Services to Processes](#)




External decision: For an external decision, an already modeled, ready to be used, and activated decision is added. You can directly implement it in the processes of your process application. See [Add Decision Services to Processes](#).

If required, you can also edit an external decision like any other decision by opening it in the decision editor canvas.

Understand the Graph View

In Oracle Cloud Infrastructure Process Automation, decisions are created in the graph view. In the graph view, you can create decision requirement diagrams (DRD), in accordance with DMN standards, to visually represent your decisions.

Oracle Cloud Infrastructure Process Automation currently supports one DRD per decision. The following table lists all the available DRD components, which you can use to create your decision.







DRD Component		Description	Icons
Elements	Decision	Denotes a node that determines an output based on its inputs and the logic it contains.	
	Input Data	Denotes information used as input by one or more decision elements.	
Requirements	Information Requirement	Denotes the flow of information from an input data or a decision element to another decision element.	

The graph view is divided into the following areas:

- Toolbar
- Canvas
- Diagram palette

Toolbar

The decision toolbar provides quick access controls to manage the decision canvas.

Toolbar Icon	Name	Description
	Pan Mode	Pan the entire canvas. Note that this is the default mode.
	Undo	Revert the last action.
	Redo	Repeat the last action.
	Maximize view	Open the editor in an expanded view.
	Minimize view	Open the editor in a minimized view.
	Decision service	Open the Services pane.

Canvas

The decision canvas is the central area where you can create a diagram that represents your decision, using the elements in the diagram palette. The zoom controls at the bottom of the canvas allow you to zoom in or fit the canvas in your browser.

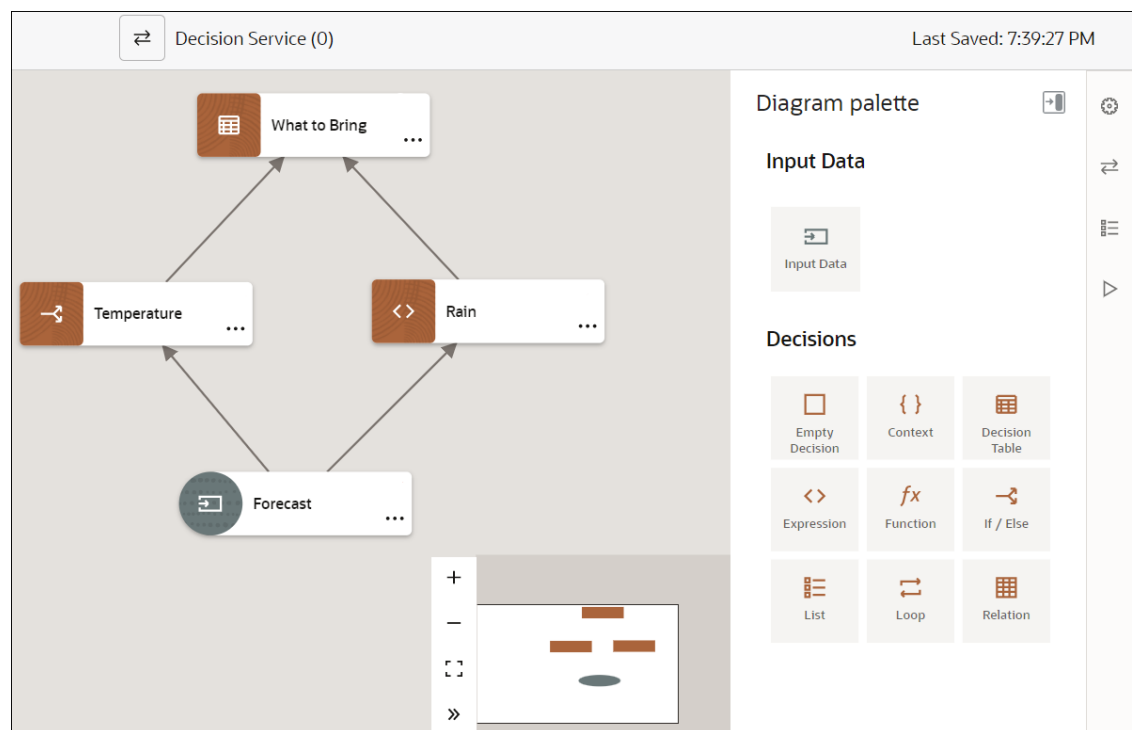
Diagram palette

From the diagram palette, drag and drop DRD components onto the canvas to use them in your diagram.

Under **Decisions** on the palette view all the decision types available for use. You can drag a particular decision type or add an empty decision element and then associate a decision notation to it.

Click  to hide or view the diagram palette.

The following figure shows a decision model in the graph view:







Add a Decision Logic Type


A decision has logical notations such as decision tables, expressions, if-then-else rules, and so on. The first step in modeling a decision is to select the logic type you plan to use in order to obtain a specific output.

1. On the diagram palette, choose one of the following decision types under **Decisions**.
 - Empty Decision
 - Context
 - Decision Table

- Expression
 - Function
 - If-Then-Else
 - List
 - Loop
 - Relation
2. Select and drag a decision type onto the canvas.
 3. Position the cursor at the point you want to add the decision and release the mouse click.
 4. To edit decision properties, click the three dots in the decision, and click **Edit**. The Decision Properties pane opens. Note that when you add any decision element to the canvas, a new decision logic with a default name (for example, Decision1) is created and associated with the element. You can edit the information related to the logic from the properties pane.
 5. Enter the necessary information about the decision logic in the properties pane.

Field	Description
Name	<p>Enter a unique name for the decision logic you want to create and associate with the element. You can also select a previously defined decision logic for the element. Click Expand  and select an implemented logic from the available options.</p> <div style="border: 1px solid #0070C0; padding: 10px; margin-top: 10px;"> <p> Note:</p> <p>If you add a particular decision type to the canvas and select a different type of logic from the Name drop-down menu, the element's logic type changes.</p> </div> <p>Click Edit  to edit the decision element's logic. You can also double-click element on the canvas to view or edit its logic.</p> <p>Click Delete to delete the decision logic associated with the element.</p>
Logic	<p>Change the decision logic type if required. For example, you can add an empty decision to the canvas and change its logic from the properties pane.</p> <div style="border: 1px solid #0070C0; padding: 10px; margin-top: 10px;"> <p> Note:</p> <p>If you select a previously implemented decision logic in the Name field and change its type in the Logic field, the content of the previous implementation is overwritten.</p> </div>
Description	Provide additional information, if any, for the decision logic.
Question	Add the questions allowed for the decision logic.
Allowed Answers	Add the possible answers for the decision logic.

6. Click **Close** to save the information and close the pane.

7. To delete the decision element you added to the canvas, select the element and click  and then click **Delete**.

Define Expressions with the Friendly Enough Expression Language (FEEL)

Decision Modeling and Notation (DMN) defines Friendly Enough Expression Language (FEEL) to provide standard executable semantics to all expressions used within a decision model.

In Oracle Cloud Infrastructure Process Automation, you use FEEL to define expressions within all notations of decision logic, including decision tables.

- [Data Types](#)
- [Grammar Rules](#)
- [Built-In Functions](#)
- [List Iteration Expressions](#)
- [Date, Time, and Duration Functions](#)

Data Types

Oracle Cloud Infrastructure Process Automation supports the following FEEL data types that you can use for input data, expression values, function arguments and return values.

FEEL Data Type	Notation in Processes	Description
number	Number	FEEL Numbers are based on IEEE 754-2008 Decimal128 format, with 34 decimal digits of precision and rounding toward the nearest neighbor with ties favoring the even neighbor. Numbers are a restriction of the XML Schema type <code>precisionDecimal</code> , and are equivalent to Java <code>BigDecimal</code> with <code>MathContext DECIMAL128</code> .
string	Text	Variable-length sequence of characters italicized or encapsulated in double quotes.
boolean	True or False	Logical Boolean (true/false).
date-time	Date and Time	Calendar date and time combination.
duration	Duration	Duration in ISO 8601 date time format.



Note:

You can extend these basic data types by defining custom data types. See [Define Custom Data Types](#).

Grammar Rules

Learn about the syntax for commonly-used FEEL expressions through simple examples. For the complete definition of FEEL syntax, see *Decision Model and Notation (DMN), v1.1*.

Arithmetic Expressions

Name	FEEL Expression	Return Value
Addition (+)	0.15+30	30.15
Subtraction (-)	15-30	-15
Multiplication (*)	.20*40.02	8.004
Division (/)	1/50	0.02
Exponentiation (**)	2**3	8

Interval Expressions

Start Value	End Value	FEEL Expression	Return Value
Inclusive	Inclusive	15 in [15..30]	true
Exclusive	Exclusive	15 in (15..30)	false
Exclusive	Inclusive	30 in (15..30]	true
Inclusive	Exclusive	30 in [15..30)	false



Note:

In decision table input entry and input/output allowed value cells, you can use intervals or list of intervals to test against the input data.

Comparison Expressions

Name	FEEL Expression	Return Value
Less than (<)	8<2**3	false
Less than or equal to (<=)	15 in (<=15)	true
Equal (=)	8=2**3	true
Greater than (>)	30 in (>30)	false
Greater than or equal to (>=)	1/5>=0.20	true
Not equal to (!=)	8!=2**3	false



Note:

In decision table input entry and input/output allowed value cells, you can use comparison operators to define unary expressions.

Other Expressions

Name	FEEL Expression	Return Value
Disjunction	$(2*2=2**2)$ or $(3*2=3**2)$	true
Conjunction	$(2*2=2**2)$ and $(3*2=3**2)$	false
Negation	not($2*2=2**2$)	false



Note:

In decision table input entry and input/output allowed value cells, you can use comma-separated list of values to specify disjunction.

Built-In Functions

FEEL includes a library of built-in functions that you can use to define expressions.

Oracle Cloud Infrastructure Process Automation supports the following types of built-in functions:

- [Conversion Functions](#)
- [Boolean Functions](#)
- [String Functions](#)
- [List Functions](#)
- [Numeric Functions](#)

Conversion Functions

Name(parameters)	Parameter Domain	Description	Example
date(<i>from</i>)	date string	convert <i>from</i> to a date	date("2012-12-25") - date("2012-12-24") = duration("P1D")
date(<i>from</i>)	date and time	convert <i>from</i> to a date (set time components to null)	date(date and time("2012-12-25T11:00:00Z")) = date("2012-12-25")
date and time(<i>from</i>)	date time string	convert <i>from</i> to a date and time	date and time("2012-12-24T23:59:00") + duration("PT1M") = date and time("2012-12-25T00:00:00")
time(<i>from</i>)	time string	convert <i>from</i> to time	time("23:59:00") + duration("PT2M") = time("00:01:00")
time(<i>from</i>)	time, date and time	convert <i>from</i> to time (ignoring date components)	time(date and time("2012-12-25T11:00:00Z")) = time("11:00:00")
number(<i>from</i> , <i>grouping separator</i> , <i>separator</i> , <i>decimal separator</i>)	number, separator, decimal notation	convert <i>from</i> to a number	number("1 000,0", " ", ",", ".") = number("1,000.0", ",", ".")
string(<i>from</i>)	non-null	convert <i>from</i> to a string	string(1.1) = "1.1" string(null) = null

Name(parameters)	Parameter Domain	Description	Example
duration(<i>from</i>)	duration string	convert <i>from</i> to a days and time or years and months duration	<ul style="list-style-type: none"> date and time("2012-12-24T23:59:00") - date and time("2012-12-22T03:45:00") = duration("P2DT20H14M") duration("P2Y2M") = duration("P26M")
years and months duration(<i>from</i> , <i>to</i>)	both are date and time	return years and months duration between <i>from</i> and <i>to</i>	years and months duration(date("2011-12-22"), date("2013-08-24")) = duration("P1Y8M")

Boolean Functions

Name(parameters)	Parameter Domain	Description	Example
not(<i>negand</i>)	boolean	logical negation	<ul style="list-style-type: none"> not(true) = false not(null) = null

String Functions

Name(parameters)	Parameter Domain	Description	Example
substring(<i>string</i> , <i>start position</i> , <i>length?</i>)	string, number1	return <i>length</i> (or all) characters in <i>string</i> , starting at <i>start position</i> . 1 st position is 1, last position is -1	<ul style="list-style-type: none"> substring("redwood",3) = "dwood" substring("redwood",3,3) = "dwo" substring("redwood", -2, 1) = "o"
string length(<i>string</i>)	string	return length of <i>string</i>	string length("red") = 3
upper case(<i>string</i>)	string	return uppercased <i>string</i>	upper case("aBc4") = "ABC4"
lower case(<i>string</i>)	string	return lowercased <i>string</i>	lower case("aBc4") = "abc4"
substring before (<i>string</i> , <i>match</i>)	string, string	return substring of <i>string</i> before the <i>match</i> in <i>string</i>	<ul style="list-style-type: none"> substring before("redwood", "wood") = "red" substring before("redwood", "xyz") = ""
substring after (<i>string</i> , <i>match</i>)	string, string	return substring of <i>string</i> after the <i>match</i> in <i>string</i>	<ul style="list-style-type: none"> substring after("redwood", "dw") = "ood" substring after("", "o") = ""
replace(<i>input</i> , <i>pattern</i> , <i>replacement</i> , <i>flags?</i>)	string2	regular expression pattern matching and replacement	replace("abcd", "(ab) (a)", "[1=\$1][2=\$2]") = "[1=ab][2=]cd"
contains(<i>string</i> , <i>match</i>)	string	does the <i>string</i> contain the <i>match</i> ?	contains("redwood", "de") = false
starts with(<i>string</i> , <i>match</i>)	string	does the <i>string</i> start with the <i>match</i> ?	starts with("redwood", "re") = true
ends with(<i>string</i> , <i>match</i>)	string	does the <i>string</i> end with the <i>match</i> ?	ends with("redwood", "d") = true
matches(<i>input</i> , <i>pattern</i> , <i>flags?</i>)	string2	does the <i>input</i> match the regexp <i>pattern</i> ?	matches("redwood", "^re*w") = true

List Functions

Name(parameters)	Parameter Domain	Description	Example
list contains(<i>list</i> , <i>element</i>)	list, any element of the semantic domain including null	does the <i>list</i> contain the <i>element</i> ?	list contains([1,2,3], 2) = true
count(<i>list</i>)	list	return size of <i>list</i>	count([1,2,3]) = 3
minimum(<i>list</i>)	(list of) comparable items	return minimum item	minimum([1,2,3]) = 1
maximum(<i>list</i>)	(list of) comparable items	return maximum item	maximum([1,2,3]) = 3
sum(<i>list</i>)	(list of) numbers	return sum of numbers	sum([1,2,3]) = 6
mean(<i>list</i>)	(list of) numbers	return arithmetic mean (average) of numbers	mean([1,2,3]) = 2
sublist(<i>list</i> , <i>start position</i> , <i>length</i> ?)	list, number1, number2	return list of <i>length</i> (or all) elements of <i>list</i> , starting with <i>list</i> [<i>start position</i>]. 1 st position is 1, last position is -1	sublist([1,2,3], 1, 2) = [2]
append(<i>list</i> , <i>item</i> ...)	list, any element including null	return new <i>list</i> with <i>items</i> appended	append([1], 2, 3) = [1,2,3]
concatenate(<i>list</i> ...)	list	return new <i>list</i> that is a concatenation of the arguments	concatenate([1,2],[3]) = [1,2,3] concatenate(1,2,3) = [1,2,3] concatenate([1,2],3) = [1,2,3]
insert before(<i>list</i> , <i>position</i> , <i>newItem</i>)	list, number1, any element including null	return new <i>list</i> with <i>newItem</i> inserted at <i>position</i>	insert before([1,3],1,2) = [1,2,3]
remove(<i>list</i> , <i>position</i>)	list, number1	<i>list</i> with item at <i>position</i> removed	remove([1,2,3], 2) = [1,3]
reverse(<i>list</i>)	list	reverse the <i>list</i>	reverse([1,2,3]) = [3,2,1]
index of(<i>list</i> , <i>match</i>)	list, any element including null	return ascending list of <i>list</i> positions containing <i>match</i>	index of([1,2,3,2],2) = [2,4]
union(<i>list</i> ...)	list	concatenate with duplicate removal	union([1,2],[2,3]) = [1,2,3]
distinct values(<i>list</i>)	list	duplicate removal	distinct values([1,2,3,2,1]) = [1,2,3]
flatten(<i>list</i>)	list	flatten nested lists	flatten([[1,2],[3]], 4) = [1,2,3,4]

Numeric Functions

Name(parameters)	Parameter Domain	Description	Example
decimal(<i>n</i> , <i>scale</i>)	number, number1	return <i>n</i> with given <i>scale</i>	<ul style="list-style-type: none"> decimal(1/3, 2) = .33 decimal(1.5, 0) = 2 decimal(2.5, 0) = 2
floor(<i>n</i>)	number	return greatest integer <= <i>n</i>	<ul style="list-style-type: none"> floor(1.5) = 1 floor(-1.5) = -2
ceiling(<i>n</i>)	number	return smallest integer >= <i>n</i>	<ul style="list-style-type: none"> ceiling(1.5) = 2 ceiling(-1.5) = -1

List Iteration Expressions

This sections lists a few frequently-used list iteration expressions.

Name(parameters)	Description	Example
for [item] in [list] return [expression]	iterate over a list	for i in [1,2,3,4] return i^i = [1,4,9,16]
sum (for [item] in [list] return [expression])	iterate over a list and return the sum of iterations	sum(for i in [1,2,3,4] return i^i) = 30
every [item] in [list] satisfies [expression]	test if every item in the list satisfies the test condition described by the expression.	<ul style="list-style-type: none"> every n in [12,50,51] satisfies $n > 5$ = true every n in [12,50,51] satisfies $n < 50$ = false
some [item] in [list] satisfies [expression]	test if at least one of the items in the list satisfies the test condition described by the expression.	<ul style="list-style-type: none"> some n in [12,50,51] satisfies $n > 50$ = true some n in [12,50,51] satisfies $n > 51$ = false

Date, Time, and Duration Functions

Because FEEL does not support literal representation of date, time, or duration values, you must use a combination of built-in functions and string/number literals to express these values.

Built-in functions extract date, time, and duration data from strings or numbers. The following sections provide examples of a few frequently-used date and time operations using built-in functions:

- [Conversion Examples](#)
- [Arithmetic Operation Examples](#)
- [Comparison Operation Examples](#)

Conversion Examples

Examples in this section demonstrate conversion of number or string literals into date, time, or duration data types.

Date and Time Data Type Examples

The following examples convert string literals to date or date-time values:

Example	Description
date("2012-12-25")	Represents the date 2012-12-25 in the YYYY-MM-DD format
time("23:59:00")	Represents the time 23:59:00 in the hh:mm:ss format
time("23:59:00-08:00")	Represents the local time 23:59:00 in the hh:mm:ss format; the local time is 8 hours behind UTC.
date and time("2012-12-25T11:00:00")	Represents the date 2012-12-25 and time 11:00:00 in YYYY-MM-DD and hh:mm:ss formats respectively
date and time("2012-12-25T11:00:00Z")	Represents the date 2012-12-25 and time 11:00:00 in YYYY-MM-DD and hh:mm:ss formats respectively; "Z" represents UTC time

Duration Data Type Examples

A few conversion examples using the duration function are listed in the following table:

Example	Description
duration("P1DT12H30M")	Represents a duration of 1 day, 12 hours, and 30 minutes
duration("-P120D")	Represents a duration of minus 120 days
duration("PT2000H")	Represents a duration of 2000 hours
duration("P1Y2M3DT10H30M")	Represents a duration of 1 year, 2 months, 3 days, 10 hours, and 30 minutes

Arithmetic Operation Examples

Examples in this section demonstrate arithmetic operations that can be performed on date, time, or duration data types.

Operator	Example	Result	Description
(+)	date("2016-12-25") + duration("P3Y2M6D")	date("2020-03-02")	Returns the date 3 years 2 months and 6 days after 2016-12-25
(+)	date and time("2016-12-25T12:30:00Z") + duration("P1Y2M3DT10H30M")	date and time("2018-02-28T23:00:00Z")	Returns the date and time 1 year 2 months 3 days and 10 hours 30 minutes after the date 2016-12-25 and time 12:30:00
(+)	date("2016-12-25") + duration("-P3Y2M6D")	date("2013-10-19")	Returns the date 3 years 2 months and 6 days before 2016-12-25
(-)	date("2015-12-25") - date("2012-12-25")	duration("P1095DT0H0M0S")	Returns a duration indicating number of days and time
(-)	date and time("2012-12-25T12:00:00") - date and time("2015-12-25T11:12:00")	duration("-P1094DT23H12M0S")	Returns a duration indicating number of days and time
(-)	date and time("2012-12-25T12:00:00-08:00") - date and time("2015-12-25T11:12:00Z")	duration("-P1094DT15H12M0S")	Returns a duration indicating the number of days and time between date and time of two different time zones.
(-)	date("2016-12-25") - duration("P3Y2M6D")	date("2013-10-19")	Returns the date 3 years 2 months and 6 days before 2016-12-25
(/)	(date("2015-12-25") - date("2012-12-25"))/ duration("P1D")	1095	Returns the number of days between two dates
(/)	(date and time("2015-12-25T17:00:00") - date and time("2015-12-25T09:12:00"))/ duration("PT1H")	7.8	Returns the number of hours between two date and time values
(/)	(date("2015-12-25") - date("2015-12-24"))/ duration("P1Y")	0.0027397260273972603	Returns the number of years between two dates
None	years and months duration(date("2012-12-23"), date("2015-12-25"))	duration("-P3Y0M")	Returns the years and months duration between two dates

Comparison Operation Examples

Examples in this section demonstrate comparison operations that can be performed on date or time data types.

Operator	Example	Result	Description
>	<code>date("2012-12-25") > date("2015-12-25")</code>	false	Determines if Date A occurs after Date B
>	<code>((date("2015-12-25") - date("2015-11-25")) > duration("P1Y")) ></code> <code>duration("P1Y")</code>	false	Determines if Duration A is greater than Duration B

Define Decision Input and Type

An input data variable is a placeholder for information that is to be supplied to a decision model when the model is invoked.

Supported data types for input data are text, number, boolean (true or false), and date and time. See [Data Types](#). You can extend the built-in data types to define custom complex data types.

- [Create Input Data](#)
- [Define Custom Data Types](#)



Create Input Data

Use built-in or custom data types to create input variables for the decision.

To create a input data variable:

1. On the Diagram palette, select the **Input Data** element and drag it onto the canvas.
2. Position the cursor at the point you want to add the element and release the mouse click.
3. To edit the element's properties double click the element to open the Input Properties pane. Optionally, click the three dots in the element and click **Edit** to open the Input Properties pane.


Note that when you add an input data element to the canvas, a new input variable with a default name (for example, inputData1) and type is created and associated with the element. You can edit the information related to the variable from the properties pane.

4. In the Input Properties pane:
 - a. Enter a suitable name for the variable you want to create and associate with the element. You can also select a previously defined variable for the element. Click  and select a variable from the available options.
Click **Delete**  to delete the variable associated with the element.
 - b. Change the **Mode** (that is, the data type) for the variable. You can select one of the built-in data types or choose a custom data type (**Other Type**).
 - If you select a built-in type, you can specify if the variable is a list by selecting the **Make a list** check box. Additionally, you can restrict the variable to enumerated values or ranges.
 - If you select **Other Type**, you can either define a new data type or use a previously defined custom data type. See [Define Custom Data Types](#).

Note:

If you select a previously defined variable in the **Name** field, the mode and other details are auto-populated.

Click **Close** to save the details and close the pane.


5. Now, create outgoing connections from the input data element to the decisions.
 - a. Select the element and click **Create New Connection**  and keep the mouse depressed.
 - b. Move the cursor to the decision you want to connect to, and then release the click.




You can create multiple outgoing connections from an input data element. To delete a connection, hover your cursor over the arrow and then double-click.
6. To delete the input element from the canvas, select it and click **Delete**.

Define Custom Data Types

If built-in data types aren't suitable for an input variable in your decision model, you can create a custom data type.

To define a custom data type:

1. Click the  **Types** icon to open the Type Definition pane.
2. In the Type Definition pane, click **Add New Type**.
In the resulting New Type pane that opens for editing, you see that a new type definition is created with a default name.
3. In the New Type pane:
 - a. Edit the default name in the **Name** field, if necessary.

- b. Select the **Mode** (that is, the data type) for the definition. You can select one of the built-in data types to create aliases or choose to define a complex data type.
- c. To identify the data type as a list, select the **Make a list** check box.
- d. If you select the mode as **Text**, **Number**, or **Date and Time**, you can optionally define allowed values to restrict the data type definition to enumerated values or ranges.
- e. If you select the mode as **Complex**, define attributes within the type definition.
 - i. Click  to add a new attribute. The New Type Attribute pane opens and an attribute is created with a default name.
 - ii. In the pane, edit the attribute's name, select the mode, and enter allowed values or a range for the attribute.
 - iii. Select the **Make a list** check box to identify the attribute as a list.
 - iv. Click **Close**  to return to the Type Definition pane, and repeat steps i-iii to add another attribute to the complex type definition.
 - v. Click on any type attribute to modify the type attribute.
 - vi. To delete a type attribute, click the row and then click **Delete** .


Applicant Data ✕

Edit Type

Name


Mode
 ▼

Make a list

Define Type Attributes + 

Name	Type
Name	Text
Phone	Number
Account Number	Number

4. Now select the input data element on the canvas for which you want to use the custom variable type.

5. Double click the element, or click **Open Properties** to open the Input Properties pane.
6. From the **Mode** drop-down list, select **Other Type**.
7. In the resulting **Other Type** drop-down list, select the type definition you just created.
8. Click **Close**  to save the changes and close the Input Properties pane.

Create Custom Data Types in Decision Applications

A custom data type can be a new complex data type or an alias of a built-in data type. Further, to create a complex data type, you can use a combination of built-in data types or other complex data types.

While creating a decision application, you can define a custom data type in one of the following ways:

- Manually define each custom data type.
- Import a JSON file or schema containing the custom data types.

Limitations of Importing a JSON Schema

The following are the limitations of importing a JSON schema:

- The maximum size of the JSON schema that you can import is 200 KB.
- The JSON schema must not have fields that exceed 20 levels of nesting.
- In the JSON schema, data is extracted to create types from the following keywords:
 - type
 - format
 - items
 - enum
 - definitions
 - properties
 - \$ref

Hence, Process Automation does not support keywords that are not included in this list. For example, anyOf, allOf, oneOf, pattern, required, minLength, maxLength, and regex.

- Field names in the schema must:
 - Start with a letter, underscore, or colon.
 - Contain only letters, digits, underscores, hyphens, or periods.

To import a JSON file for creating custom data types in decision applications:

1. In the decision application, click the  **Types** icon to open the Type Definition pane.
2. In the Type pane, click



A drop-down appears and shows the options **Import from sample** and **Import from schema**.

3. Select one the following options:
 - To import a JSON sample, select **Import from sample**.
 - To import a JSON schema, select **Import from schema**.
4. On the **Create new type** pane, in the **Name** field, enter the name for the custom data type.
5. If you have selected **Import from sample**, the **Sample** tab appears where you paste the JSON sample containing the data types. Click **Next** to view the schema created. If there are no changes required, click **Create**.
6. If you have selected **Import from schema**, the **Schema** tab appears where you paste the JSON schema containing the data types. Click **Create** to create custom data type.

Model Decision Logic

Model the decision logic by defining how each decision's output is derived from its inputs.

In Oracle Cloud Infrastructure Process Automation, you have the following notations to model the logic within a decision:

- [Create Empty Logic Decisions](#)
- [Create Decision Tables](#)
- [Create Expressions](#)
- [Create If-Then-Else Statements](#)
- [Create Functions](#)
- [Create Contexts](#)
- [Create Lists](#)
- [Create Relations](#)
- [Create Loops](#)

Create Empty Logic Decisions

While outlining the framework of a decision model, you can create a decision with the Empty Logic notation as a placeholder for actual logic.

To create a decision with the Empty Logic notation:

1. On the Diagram palette, under **Decisions** select **Empty Decision**.
2. Drag it onto the canvas.

The empty logic decision gets added to the canvas.

Create Decision Tables

Decision tables are the notation of choice to model complex logic. Their tabular layout helps you effectively document all the possible conditions and results of a problem.

To create a decision table:

1. On the Diagram palette, under **Decisions** select **Decision Table** and drag it onto the canvas.
2. Double-click the element to edit its logic.

You can also edit the table's logic from the Decision Properties pane. Click the three dots on the element and select **Edit** to open the Decision Properties pane of the element. Click

Edit  next to the **Name** field to edit the element.

3. Click **Close** to return to the canvas.

Learn about each decision table element in detail in the following topics:

- [Define Decision Table Input](#)
- [Define Decision Table Output](#)
- [Configure Rules](#)
- [Configure a Hit Policy](#)

The following figure shows an example decision table with all its elements noted:

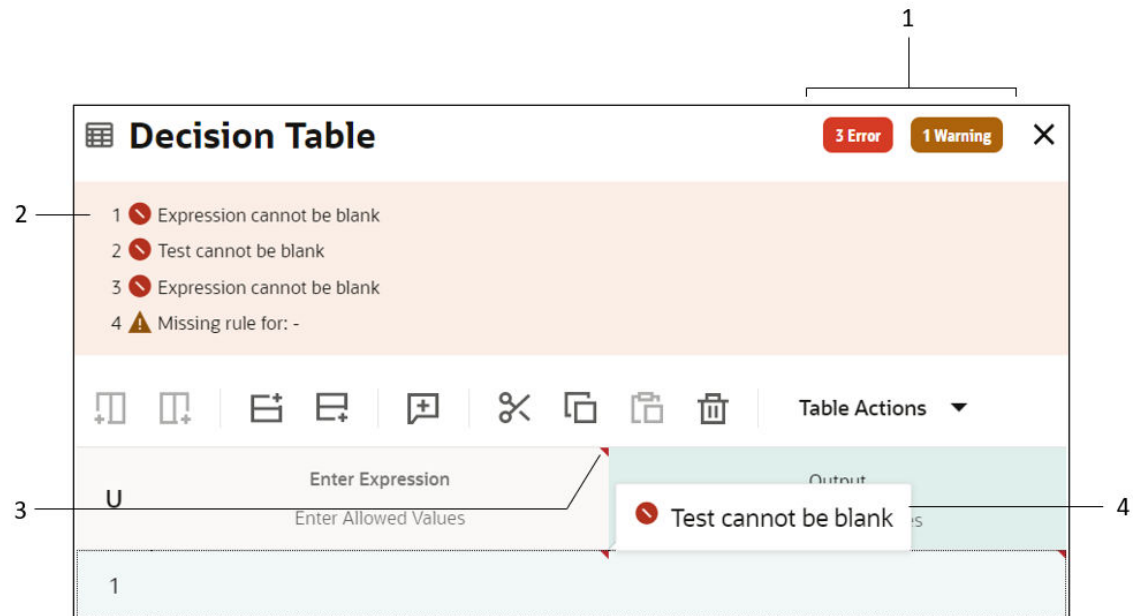
	Age	Work_Experience	Bonus Percent
A	-	-	Enter Allowed Values
1	-	>=20	10
2	>=50	-	10
3	<50	<20	5

1. Row and column controls – Add or remove rows and columns
2. Input Expression cell – Contains the expression associated with a particular input column or clause.
3. Hit Policy Indicator cell – Displays the hit policy selected for the table.
4. Rules – Rows within the table.
5. Input Entry cell – Contains an input entry.
6. Output Entry cell – Contains an output entry.
7. Output Label cell – Contains the name of the decision.
8. Add Annotation button – Adds a column for documenting or annotating decision rules. Annotations aren't considered as part of the decision logic; they serve as explanatory notes for designers.
9. Cut, Copy, and Paste button – Cut, copy, and paste rows and columns within decision tables. However, you cannot use this functionality to:
 - Copy a row into a column, and vice versa.
 - Copy an input column into an output column or an annotation column, and vice versa.

10. Table Actions - Copy and paste a decision table.

Note that when you edit a decision table, the total number of resultant errors and warnings are displayed on the top-right side area above the decision table. Click the error or warning to view more details. Additionally, you can also see where the errors or warnings occur in the decision table cells.

The following image shows the errors and warnings that are displayed when you create an empty decision table:



1. The total count of errors and the total count of warnings.
2. Details of the errors and warnings that are displayed when you click the errors and warning counts.
3. The upper-right corner of a cell is marked in red color to indicate erroneous cells.
4. The error message appears when you hover the mouse on the red marked upper-right corner.

Define Decision Table Input

An input (also referred to as input clause) to a decision table consists of an input expression and several input entries. It is represented as a column within a table. A decision table may have multiple inputs.

Specify Input Expressions

In a decision table, you enter an input expression in the header cell of an input column. In combination with input entries, an input expression determines the value of a particular input column. It can be a simple test expression, for example, `Age>50`. You can use input variables, outputs of other decisions, or built-in functions to define input expressions.

You can enter a built-in function in any cell of the decision table. For example, you can use built-in functions such as `contains`, `starts with`, and `substring` that can calculate or filter a specific value from the input values defined in the header cell of the input column. Built-in functions used in input conditions must evaluate to a boolean value.

In an Input Expression cell, press **Ctrl+Space** to open a drop-down menu. Use the options available in the menu to define your expression. You can also click on the Enter Allowed Values cell to enter or edit the input expression.

In the Decision Model editor, the expression language used for all expressions, including input expressions, is friendly enough expression language (FEEL). Want to learn more about the expression language syntax? See [Define Expressions with the Friendly Enough Expression Language \(FEEL\)](#).

Allowed Values

Click on the Enter Allowed Values cell within an Input Expression or Output Label cell to specify permitted values for cell entries of that column. However, for an input column, the type of allowed values (or specific allowed values) are automatically populated when you specify the input expression for the column. For example, if an input expression returns Boolean values, allowed values for that column are populated as *true*, *false*. The options you can choose from for changing the type of allowed values are listed in the following table. Note that after you specify the input expression for a column, the type of allowed values you can toggle between are limited.

Allowed Value Type	Description
Auto	This is the default selection for a new input column.
Any	Use this option to specify that there are no restrictions on the data type of entries. Optionally, restrict entries to a given value or list of values. This is the default selection for the output column.
Text	Use this option to restrict entries to text strings. Optionally, specify a particular string or list of strings.
Number	Use this option to restrict entries to numbers. Optionally, specify a particular number, limit, or range.
Date and Time	Use this option to restrict entries to date and time values. Optionally, specify a particular date and time value, limit, or range.
True or False	Use this option to permit only Boolean entries.
Other Type	Use this option to restrict entries to a given custom data type.
Advanced	Use this option to permit FEEL expressions and <i>null</i> values as entries. Optionally, specify constraints using FEEL expressions.

Note:

The data type of input entry cells is determined by the data type of the input expression. Make sure that the type of allowed values you supply is consistent with the data type of the input clause.

Specify Input Entries

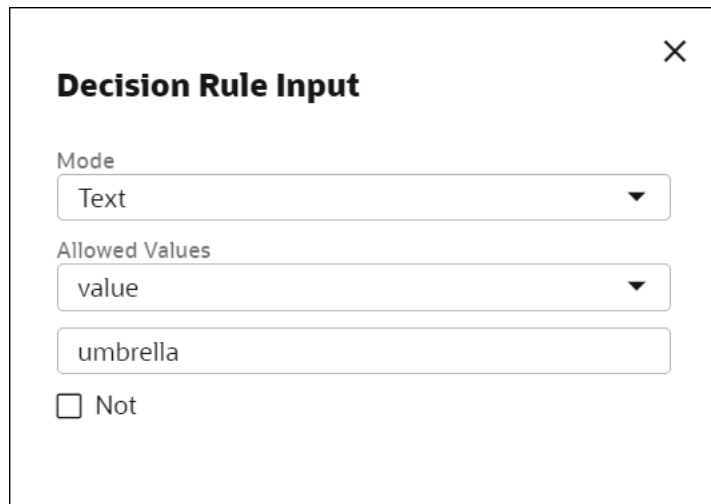
You can enter strings, numbers, Boolean values, date and time values, or FEEL expressions as input entries based on the mode selected. If you've provided specific allowed values or if your input expression returns a finite set of values, an auto-suggest menu appears when you click on an input entry cell.

 **Note:**

If the data type of an input entry does not match the data type of the column, or if the input entry is not among the allowed values, an error is displayed within the decision.

Modes

Based on the input expression and/or allowed values you specify, the mode for entry cells is automatically selected. If required, click on the **Mode** icons to switch to a different mode. The following figure shows the Mode Editor window with Text mode selected:



The mode options available to you in the editor is dependent on the data type of the input expression or allowed values you've specified. The following table details all the available modes:

Mode	Description
Any	Use this mode to mark an entry as irrelevant (-).
Text	Use this mode to enter strings. In this mode, you can enter a string as a plain literal without double quotes.
Number	Use this mode to enter numbers. Optionally, use the constraint options available in the Mode editor.
True or False	Use this mode to enter Boolean values.
Date and Time	Use this mode to enter date and time values. Optionally, use the constraint options available in the Mode editor.
Advanced	Use this mode to enter advanced FEEL expressions and <i>null</i> values. You can use the constraint options available in the Mode editor to define an expression. Want to learn more about the syntax and examples? See Grammar Rules .

Define Decision Table Output

An output (also referred to as output clause) of a decision table consists of an output label and several output entries. When you create a new decision table, a table with a single output

column appears by default. To add additional outputs, select the existing output column and use the column add controls.

Specify Output Labels

Generally, for single-output tables, the output label is the name of the decision table. In the Output Label cell, you can specify the allowed values for output entries. See Allowed Values in [Define Decision Table Input](#).

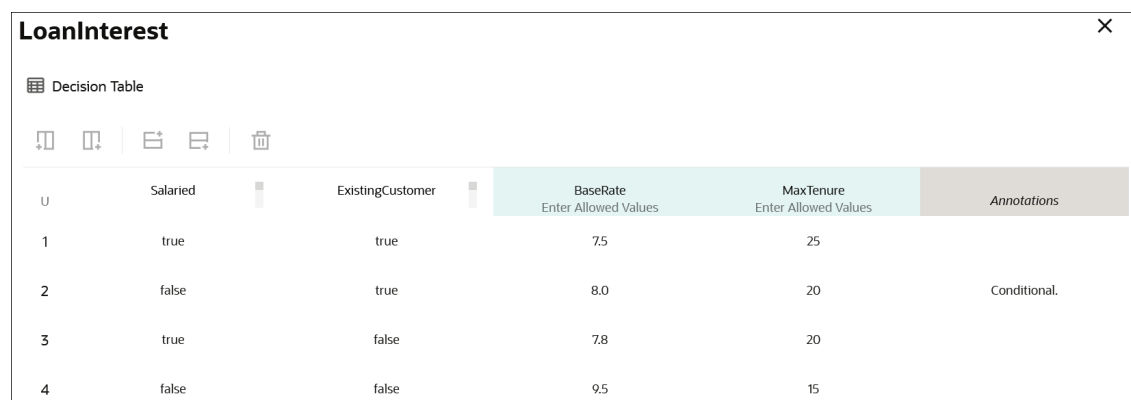
Specify Output Entries

You can enter text, numbers, boolean values (true or false), date and time values, or FEEL expressions as output entries based on the mode selected. See Modes in [Define Decision Table Input](#).

Based on the allowed values you specify, an auto-suggest menu appears when you click on an output entry cell.

You can use input variables, outputs of other decisions, or built-in functions to specify FEEL expressions for output entries.

The following figure shows an example decision table with two output columns:



U	Salaried	ExistingCustomer	BaseRate Enter Allowed Values	MaxTenure Enter Allowed Values	Annotations
1	true	true	7.5	25	
2	false	true	8.0	20	Conditional.
3	true	false	7.8	20	
4	false	false	9.5	15	

To reference a particular output of this multi-output table from another decision, use the following format: *DecisionName.OutputLabel*; for example, *LoanInterest.BaseRate*.

Configure Rules

Rules are expressed as rows within a table. Every rule consists of one or more input entries and a corresponding output entry.

Generally, a decision table consists of multiple rules. When the input data matches the input entries of a rule, the result of the decision table contains the output entry of the rule.

Configure a Hit Policy

The Hit Policy Indicator cell displays the hit policy selected for the table. The hit policy determines the output of a decision table from the output cells of matched rules. A rule is matched when all of its condition cells match the inputs of the decision table.

Based on the hit policy, Decision Model Notation (DMN) broadly groups decision tables into the following categories:

1. **Single Hit:** A single-hit table returns the output of only one rule. Under the single-hit category, Process Automation supports the following hit policies:
 - **Unique (U)** – Only one of the rules can match.
 - **Any (A)** – Multiple rules can match, but all matching rules must have the same output.
 - **First (F)** – Multiple rules can match; the output of the first rule that matches is returned.
 - **Priority (P)** – Multiple rules can match; the output value that has the highest priority is returned.
2. **Multiple Hit:** A multiple-hit table returns the output of multiple rules. Under this category, the following hit policies are supported in Process Automation:
 - **Collect (C)** – Multiple rules can match; outputs are returned as an arbitrarily-ordered list.
 - **Collect Sum (C+)** – Multiple rules can match; the sum of outputs is returned.
 - **Collect Min (C<)** – Multiple rules can match; the smallest output value is returned.
 - **Collect Max (C>)** – Multiple rules can match; the largest output value is returned.
 - **Collect Count (C#)** – Multiple rules can match; the count is returned.

When you create a new table, the Unique (U) hit policy is selected by default. To change the policy, click the Hit Policy Indicator cell and choose from the available options in the Hit Policy drop-down list. If rules within the table violate the selected hit policy, a warning is displayed within the decision.



Hit Policy Examples

Here are examples for all the hit policies.

- **Single Hit Unique**
In a decision table with Unique hit policy, only one rule can match. All rules are independent of each other, and no overlap is permitted. The decision table returns the output of the rule that matches.

Here is a decision table created with the Unique hit policy:

What to Wear		What to Wear
U	Temperature	Enter Allowed Values
	<100	
1	<25	Wool Coat
2	25	Jacket
3	>25	Casuals

In this example, for any input value of temperature, only one rule can match.

- **Single Hit Any**

In a decision table with Any hit policy, multiple rules can match. The overlap is permitted only if the matching rules have the same output. The decision table returns the output of any one of the matching rules. The hit policy is breached if matching rules have different outputs.

Here is a decision table created with the Any hit policy:

Bonus Percentage			
A	Age	Work_Experience	Bonus Percent
	-	-	Enter Allowed Values
1	-	>=20	10
2	>=50	-	10
3	<50	<20	5

In this example, for an input age of 50 and work experience of 20, the first and second rules match. This overlap is allowed because these rules have the same output. The decision table returns the output of any one of these rules.

- **Single Hit First**

In a decision table with First hit policy, multiple rules with different output entries can match. The output of the lowest-numbered matching rule is the result of the table.

Here is a decision table created with the First hit policy:

Vacation Days		
F	Service_Years	Vacation Days
	-	Enter Allowed Values
1	<5	5
2	>=5	10
3	>10	15

In this example, if service years are 11, the second and third rules match. The decision table returns only the second rule's output.

- **Single Hit Priority**

In a decision table with Priority hit policy, multiple rules with different output entries can match. The priority of output values (in descending order) is specified as a list in the Allowed Values cell of the output column. The decision table returns the output value that has the highest priority among outputs of all matching rules.

Here is a decision table created with the Priority hit policy:

Discount Percent		
P	Age	Discount Percentage
	-	5,10,15
1	<18	5
2	[18..45]	5
3	>45	10
4	>60	15

In this example, the last two rules match for an input age of 61. The decision table returns the output value that has the highest priority among of these rules, that is, 15; the priority order is defined in the Allowed Values cell.

- **Multiple Hit Collect**

In a decision table with Collect hit policy, multiple rules with different output entries can match. The decision table returns outputs of all matching rules in an arbitrarily ordered list.

Here is a decision table created with the Collect hit policy:

Interest Rates		
C	Age	Interest Rates
	-	Enter Allowed Values
1	>18	10
2	>60	8
3	<18	5

In this example, two rules match for an input age of 61. The decision table returns output values of these rules in a list, that is, 10 and 8.

- **Multiple Hit Collect (Sum)**

In a decision table with Collect (Sum) hit policy, multiple rules with different output entries can match. The decision table returns the sum of outputs of all matching rules.

Here is a decision table created with the Collect (Sum) hit policy:

Discount Percent		
C+	Age	Discount Percentage
-	-	Enter Allowed Values
1	<18	5
2	[18..45]	5
3	>45	10
4	>60	15

In this example, the last two rules match for an input age of 61. The decision table returns the sum of output values of these rules, that is, 25.

- **Multiple Hit Collect (Min)**

In a decision table with Collect (Min) hit policy, multiple rules with different output entries can match. The decision table returns the smallest output value among all matching rules.

Here is a decision table created with the Collect (Min) hit policy:

Discount Percent		
C<	Age	Discount Percentage
-	-	Enter Allowed Values
1	<18	5
2	[18..45]	5
3	>45	10
4	>60	15

In this example, the last two rules match for an input age of 61. The decision table returns the smallest output value among these rules, that is, 10.

- **Multiple Hit Collect (Max)**

In a decision table with Collect (Max) hit policy, multiple rules with different output entries can match. The decision table returns the largest output value among all matching rules.

Here is a decision table created with the Collect (Max) hit policy:

Discount Percent		
C#	Age	Discount Percentage
	-	Enter Allowed Values
1	<18	5
2	[18..45]	5
3	>45	10
4	>60	15

In this example, the last two rules match for an input age of 61. The decision table returns the largest output value among these rules, that is, 15.

- **Multiple Hit Collect (Count)**

In a decision table with Collect (Count) hit policy, multiple rules with different output entries can match. The decision table returns the count of matching rules.

Here is a decision table created with the Collect (Count) hit policy:


Discount Percent		
C#	Age	Discount Percentage
	-	Enter Allowed Values
1	<18	5
2	[18..45]	5
3	>45	10
4	>60	15


In this example, the last two rules match for an input age of 61. The decision table returns the count of matching rules, that is, 2.

Create Expressions

An expression is a logical notation, defined according to the syntax of FEEL, that evaluates to a single value. It may consist of one or more entities, such as a literal, constant, or variable, interconnected by zero or more operators. In Oracle Cloud Infrastructure Process Automation, you can also use outputs of other decisions or built-in functions to define an expression.

To create a decision with the Expression notation:

1. Add a new decision element to the model.
 - a. From the Diagram palette, under **Decisions** select **Expression**, and drag it onto the canvas.
 - b. Double-click the element to edit its logic. You can also edit the expression's logic from the Decision Properties pane. Click the three dots on the element and select **Edit** to open the Decision Properties pane of the element. Click **Edit**  next to the **Name** field to edit its logic.

Click **Close**  to return to the canvas.

2. In the **Expression** field, press **Ctrl+Space** to view a suggestion list. You can use any decision outputs, variables, functions, and keywords to form your expression.
3. Changes you make within the decision model are automatically saved and validated from time to time. Errors and warnings, if any, are displayed within the decision panel.

Following are some examples of decisions created using simple expressions:

- If the *age* property of the variable *applicant* is less than 70, then the decision returns *true* else it returns *false*.




- The decision calculates the area using a constant and an input variable, *radius*.





Create If-Then-Else Statements

An If-Then-Else expression is a logical notation that evaluates a test statement. It executes a primary expression if the test is *true* and a secondary expression if the test is not *true*. You can also introduce additional test statements using the **Add Else If** button.

To create a decision with the If-Then-Else logic:

1. Add a new decision element.
 - a. On the Diagram palette, under **Decisions** select **If-Then-Else**, and drag it onto the canvas.
 - b. Double-click the element to edit its logic. You can also edit the element's logic from the Decision Properties pane. Click the three dots on the element and select **Edit** to open the Decision Properties pane of the element. In the Decision Properties pane, click **Edit**  next to the **Name** field.



Click **Close**  to return to the canvas.
2. In the **if** expression field, press **Ctrl+Space** to view a suggestion list. You can use any decision outputs, variables, functions, and keywords from the list to define expressions in **if**, **then**, and **else** fields. Use the FEEL syntax to define expressions. See [Define Expressions with the Friendly Enough Expression Language \(FEEL\)](#).
3. If necessary, change the logical notations for then and else fields to create a nested logic.

All fields have the expression notation selected by default. Click **Change Value**  and select a different notation from the available options for a particular field. Configure logic for the selected notation.

- Changes you make within the decision model are automatically saved and validated from time to time. Errors and warnings, if any, are displayed within the decision panel.

In the following example, the input value of temperature determines the output of the If-Then-Else decision:





Weather


Operation	Expression	
if	Temperature>25	
then	"Warm"	
else	"Cool"	

 Add Else If

The following example uses an additional test statement, precipitation > 50, through Else If to determine the final output:

Weather

Operation	Expression	
if	Temperature>25	
then	"Warm"	
else if	Precipitation>50	
then	"Rain"	
else	"Cool, dry"	

 Add Else If

Create Functions

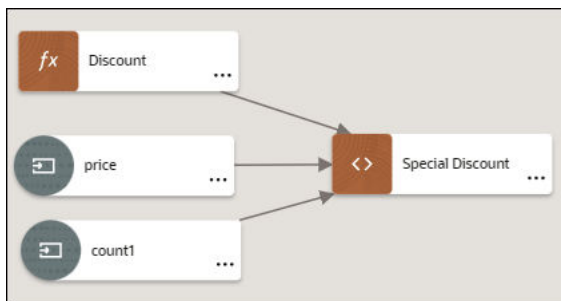
You can create functions to define specific operations that aren't available through built-in functions. In Oracle Cloud Infrastructure Process Automation, decisions created using the Function notation return a value only when invoked from another decision.

To successfully invoke a function from another decision, the number and type of parameters in the function invocation must match those in the function definition.

 **Note:**

Because a function decision by itself doesn't return a result, it's not an output decision. Therefore, you can't add function decisions to a decision service.

The following example demonstrates a function implementation in Process Automation. Here, the output decision invokes a function logic decision to calculate the seasonal discount percentage. The function decision contains the logic for regular discounts in the form of a decision table.



fx Discount

Parameters (c, p)

Parameter Name: c, Type: Number; Parameter Name: p, Type: Number

+ Add Parameter


Body

Decision Table



	c	p	Discount
A	Enter Allowed Values	Enter Allowed Values	Enter Allowed Values
1	-	>100	10
2	>=50	-	10
3	<=50	<=100	5

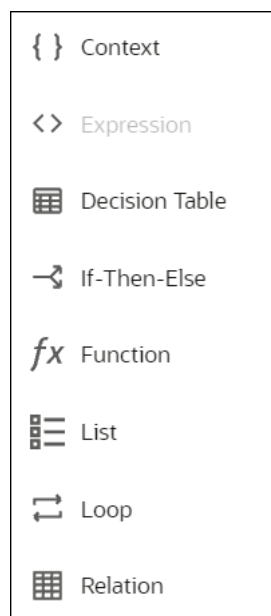
To create a decision with the Function notation:

1. Add a new decision element to the model.

- a. On the Diagram palette, under **Decisions** select **Function**, and drag it onto the canvas.
- b. Double-click the element to edit its logic. You can also edit the function's logic from the Decision Properties pane. Click the three dots on the element and select **Edit** to open the Decision Properties pane of the element. In the Decision Properties pane, click **Edit**  next to the **Name** field.

A function with empty Parameter and Body fields is created, with the Expression notation selected by default for the **Body** field.

2. Click the Parameter field, and then click **Add Parameter**  to add a new parameter. Enter a name for the parameter and select a data type for it. See [Data Types](#).
3. In the **Body** field, click **Change Value**  to change the logical notation. Select the required notation from available options.




4. Configure the logic for the selected notation. Press **Ctrl + Space** to view a suggestion list. You can use input variables or built-in functions from the list to define the logic.
5. Changes you make within the decision model are automatically saved and validated from time to time. Errors and warnings, if any, are displayed within the decision panel.

Create Contexts


A **context** is a collection of one or more key-value pairs with an optional result field. Each pair is called a context entry. The key attribute within a context entry acts as an identifier to its corresponding value attribute.

You can use a context to collectively document all decision logic related to a particular scenario or entity. Say you need to determine the loan eligibility of an applicant, based on the applicant's net monthly income and expense. For this purpose, you can create a decision named *Loan Eligibility* using the Context notation and add expressions or logic for gross monthly income, monthly expense, and net monthly income. Then, you can add a result field (within the context) that evaluates the net income and expense for loan eligibility. The result field can be a boxed expression that contains logic using the keys specified in the context.

To create a decision with the Context notation:

1. Add a new decision element to the model.
 - a. On the Diagram palette, under **Decisions** select **Context**, and drag it onto the canvas.
 - b. Double-click the element to edit its logic. You can also edit the context's logic from the Decision Properties pane; click the three dots on the element, then select **Edit** to open the Decision Properties pane. In the Decision Properties pane, click **Edit**  next to the **Name** field.

An empty Context is created.

2. Click **Add entry** to create a new context entry. A key-value pair is created with the expression notation selected by default in the Value field.
3. To change the logical notation for an entry, click **Change Value**  in the Key field to open the Change Value menu. Select a different notation from the available options.
4. In the Key field of a context entry, enter a unique name.
5. In the corresponding Value field, configure the logic for the selected notation. You can use input variables or built-in functions to define the logic.
6. Repeat steps 2 to 5 to add another entry into the context.
7. Drag and drop context entries to reorder them within the context.
8. Changes you make within the decision model are automatically saved and validated from time to time. Errors and warnings, if any, are displayed within the decision panel.

 **Note:**

If you add a function as one of the context entries, the context as a whole doesn't return a result. However, you will still be able to invoke results of individual context entries throughout the decision model.

The following image shows a context with a result field that determines the total income of applicants:

{ } Total Income ✕

Key	Expression								
<div style="border: 1px solid #ccc; padding: 5px;"> <h4 style="margin: 0;">{ } Context ☰ ⋮</h4> <table style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th style="width: 30%; text-align: left;">Key</th> <th style="text-align: left;">Expression</th> </tr> </thead> <tbody> <tr> <td>Salary</td> <td>10000 ☰ ⋮</td> </tr> <tr> <td>Other Income</td> <td>5000 ☰ ⋮</td> </tr> <tr style="background-color: #f0f0f0;"> <td>Salary+Other Income</td> <td>☰ 🗑</td> </tr> </tbody> </table> <div style="text-align: center; margin-top: 10px;"> + Add Entry </div> </div>		Key	Expression	Salary	10000 ☰ ⋮	Other Income	5000 ☰ ⋮	Salary+Other Income	☰ 🗑
Key	Expression								
Salary	10000 ☰ ⋮								
Other Income	5000 ☰ ⋮								
Salary+Other Income	☰ 🗑								
Total Income									

The output of the result field is the context's output. In this case, the context returns the result of the boxed expression. You can reference the context's result in other decisions within the model using the context name (for example, *Total Income*).

The following image shows contexts with and without a result field, and a boxed expression referencing a context and a context entry to determine the loan eligibility of applicants:

{ } Loan Eligibility X


Key	Expression													
	<div style="border: 1px solid #ccc; padding: 5px; margin-bottom: 10px;"> { } Context ☰ ⋮ </div>													
	<table style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th style="width: 20%; text-align: left;">Key</th> <th style="text-align: left;">Expression</th> <th></th> </tr> </thead> <tbody> <tr> <td>Salary</td> <td>10000</td> <td style="text-align: right;">☰ ⋮</td> </tr> <tr> <td>Other Income</td> <td>5000</td> <td style="text-align: right;">☰ ⋮</td> </tr> <tr> <td>Total Income</td> <td>Salary+Other Income</td> <td style="text-align: right;">☰ ⋮</td> </tr> </tbody> </table> <div style="display: flex; justify-content: space-around; margin-top: 10px;"> + Add Entry + Add Result </div>		Key	Expression		Salary	10000	☰ ⋮	Other Income	5000	☰ ⋮	Total Income	Salary+Other Income	☰ ⋮
Key	Expression													
Salary	10000	☰ ⋮												
Other Income	5000	☰ ⋮												
Total Income	Salary+Other Income	☰ ⋮												
Income														
	<div style="border: 1px solid #ccc; padding: 5px; margin-bottom: 10px;"> { } Context ☰ ⋮ </div>													
	<table style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th style="width: 20%; text-align: left;">Key</th> <th style="text-align: left;">Expression</th> <th></th> </tr> </thead> <tbody> <tr> <td>GeneralExpenses</td> <td>2000</td> <td style="text-align: right;">☰ ⋮</td> </tr> <tr> <td>RentExpenses</td> <td>1000</td> <td style="text-align: right;">☰ ⋮</td> </tr> <tr style="background-color: #f0f0f0;"> <td>GeneralExpenses+RentExpenses</td> <td></td> <td style="text-align: right;">☰ 🗑️</td> </tr> </tbody> </table> <div style="display: flex; justify-content: center; margin-top: 10px;"> + Add Entry </div>		Key	Expression		GeneralExpenses	2000	☰ ⋮	RentExpenses	1000	☰ ⋮	GeneralExpenses+RentExpenses		☰ 🗑️
Key	Expression													
GeneralExpenses	2000	☰ ⋮												
RentExpenses	1000	☰ ⋮												
GeneralExpenses+RentExpenses		☰ 🗑️												
Expenses														
Income.Total Income>Expenses		☰ 🗑️												
<div style="border: 1px solid #ccc; padding: 5px 10px; border-radius: 5px; display: inline-block;">+ Add Entry</div>														

The output of this context returns a *true* or *false* decision that is based on evaluating the contexts. To reference a particular context entry from another decision, use the format *ContextName.EntryKey* (for example, *Income.Salary*). Within a context, an entry can only reference entries that are above it.



Create Lists

A **list** notation is a vertical list of elements, where each element is an independent logical notation. The output of a list notation contains outputs of all its elements. You can also invoke the output of a particular list element from another decision.

To create a decision with the list notation:

1. Add a new decision element to the model.
 - a. On the Diagram palette, under **Decisions** select **List** and drag it onto the canvas.
 - b. Double-click the element to edit its logic. You can also edit the element's logic from the Decision Properties pane; click the three dots on the element, then select **Edit** to open the Decision Properties pane. In the Decision Properties pane, click **Edit**  next to the **Name** field.

An empty list is created.

2. Click **Add Item**  to create a new list entry. An entry is created with the expression notation selected by default.
3. To change the logical notation for an entry, click **Change Value**  for the particular entry. Select a different notation from the available options.
4. In the entry field, configure the logic for the selected notation. You can use input variables or built-in functions to define the logic.
5. Repeat steps 2 to 4 to add another entry into the list.
6. Changes you make within the decision model are automatically saved and validated from time to time. Errors and warnings, if any, are displayed within the decision panel.

To delete a list entry, click **Change Value**  for the particular entry and select **Delete**.

Note:

- If you add a function as one of the list entries, then the list notation as a whole doesn't return a result. However, you will still be able to invoke results of other list entries throughout the decision model.
- According to the FEEL syntax, you can also define horizontal lists in expression fields across all notations. For example, a list of all prime numbers less than 10 can be defined as [2,3,5,7].

The following example is a list of simple expressions containing prime numbers that are less than 10:



In a list of n elements, use $List_name[n]$ to invoke the n^{th} element from the beginning of the list, and use $List_name[-n]$ to invoke the n^{th} element from the end of the list. In this example, to invoke the list entry of 2, you can either use $Prime\ Numbers[1]$ or $Prime\ Numbers[-4]$.

You can also use suitable built-in list functions on a decision containing a List notation. For example, the following Expression decision returns the sum of all items in the *Prime Numbers* decision.



Create Relations


You can use a relation notation as convenient shorthand to represent multiple contexts.

A **relation** is a vertical list of similar contexts arranged horizontally. In other words, each row of a relation table is a context and each column consists of context entries, where the column name is the common key attribute for all cell entries under it that act as value attributes of respective contexts (rows). For details about contexts and key-value pairs, see [Create Contexts](#). In a relation, each cell entry is an independent logical notation.


In the output of a relation notation, outputs of all contexts within it are clearly distinguished. You can also invoke the output of a particular context or context entry from another decision.

To create a decision with the Relation notation:

1. Add a new decision element to the model.
 - a. On the Diagram palette, under **Decisions** select **Relation**, and drag it onto the canvas.
 - b. Double-click the element to edit its logic. You can also edit the element's logic from the Decision Properties pane; click the three dots on the element, then select **Edit** to open

the Decision Properties pane. In the Decision Properties pane, click **Edit**  next to the **Name** field.

An empty Relation table is created.

2. Use add row/column buttons above the table to add additional rows or columns. All cells have expression notation selected by default.
3. Enter a name for each column.
4. Select a cell and click **Change Value**  to change the logical notation for the cell. Select a different notation from the available options. Note that you cannot insert a decision table within a relation.
5. Within cells, configure the logic. You can use input variables or built-in functions to define the logic.
6. Changes you make within the decision model are automatically saved and validated from time to time. Errors and warnings, if any, are displayed within the decision panel.

 **Note:**

If you add a function as one of the cell entries, then the relation as a whole doesn't return a result. However, you will still be able to invoke results of other cell entries or contexts throughout the decision model.

The following relation contains the stock information for a particular brand of phone in the form of multiple contexts:

Color	Count	Price
"Black"	2	500
"White"	3	700
"Rose Gold"	5	600

Similar to list notations, use either *Phone[1]* or *Phone[-3]* to access the entire context related to black-colored phones. To access all cell entries of a particular column, use the relation name in combination with the column name, for example, *Phone.Price* returns all entries of the Price column in a list. To access a particular context entry (for example, "Rose Gold"), use *Phone.Color[3]* or *Phone.Color[-1]*.

The following image is the result of the entire relation, which has results of each context listed separately:


Results	
Color	Black
Count	2
Price	500
Color	White
Count	3
Price	700
Color	Rose Gold
Count	5
Price	600

Create Loops

Create loops to iterate over lists or arrays. Using the **loop** logical notation, you can create three different types of loops, namely For, Some, and Every.


- **For**: Iterates over a list and returns a new array/list containing calculated results in each index as per the expression defined in the result field.
- **Some**: Checks if at least one list item satisfies the test condition defined by an expression and returns a Boolean value.
- **Every**: Checks if every list item satisfies the test condition defined by an expression and returns a Boolean value.

To create a decision with the Loop logic:

1. Add a new decision element to the model.
 - a. On the Diagram palette, under **Decisions** select **Relation**, and drag it onto the canvas.
 - b. Double-click the element to edit its logic. You can also edit the element's logic from the Decision Properties pane; click the three dots on the element, then select **Edit** to open the Decision Properties pane. In the Decision Properties pane, click **Edit**  next to the **Name** field.


An empty loop decision is created.


2. In the **Operation** column, choose the type of loop to create from the drop-down menu. In the corresponding expression field, enter the loop variable.
3. In the **in** field, enter the list or array over which to iterate. In the **return** or **satisfies** fields, enter the expression to return (For loop) or the test expression (Some and Every loops), respectively. Press **Ctrl+Space** to view a suggestion list. You can use any decision outputs, variables, functions, and keywords in the list to define expressions in the **in**, **return**, and **satisfies** fields. Use the FEEL syntax to define expressions.
4. If necessary, change the logical notations for return and satisfies fields to create a nested logic. These fields have the expression notation selected by default. Click **Change Value**

 and select a different notation from the available options. Configure logic for the selected notation.

5. Furthermore, for a For loop, you can add an additional condition, **where**, using the **Add Condition** button.
6. Changes you make within the decision model are automatically saved and validated from time to time. Errors and warnings, if any, are displayed within the decision panel.


The following example shows a simple For loop that returns the squares of list items:


 **Square**
✕

Operation	Name or Expression
For	▼ n
in	[1,2,3,4]
return	n*n 

+ Add Condition

The following decision contains a Some loop that checks if at least one element in the list is greater than 50:

 **Greater Than**
✕

Operation	Name or Expression
Some	▼ n
in	[12,50,51]
satisfies	n>50 


The following decision contains an Every loop that checks if every element in the list is greater than 50:

Greater Than		×
Operation	Name or Expression	
Every ▼	n	
in	[12,50,51]	
satisfies	n>50	

Test Decision Models

You can verify that your decision model works the way you want by testing it.


To test the decision model:


1. Click **Test**  on the sidebar menu.
2. In the Test Decision Model pane, enter the input data to test your decision and click **Start Test**. The Decision Model Result pane displays decision results.
3. Click each green check mark to see the decision's result.
4. Click **Go Back**, and repeat steps 1 to 3 to test each decision rule's outcome.

Expose Decisions as Services

You must add at least one decision service to your decision before you can use it in processes. Decision services expose one or more output decisions of your decision model as public REST APIs. A decision service consists of a set of input data and a set of output decisions from the containing decision model.

Configure a decision service

1. On the top of the decision editor, click **Decision Service**  to open the Services pane. Optionally, you can click the Decision Service icon from the right sidebar menu of the decision editor.



Note that the number within brackets besides the Decision Service icon, indicates the number of services created for the decision.
2. In the Services pane, click **Add new service**  to open the Add Decision Service pane.
3. Enter a name for the decision service and click **OK**. The new service appears in the pane with two fields: **Output Decisions** and **Input Data**.
4. Click the **Output Decisions** field to select the output decision you want to expose through the service.
5. Click the **Input Data** field to select the input data to expose through the service.

**Note:**

You can add multiple values for the input data and the output decision.

Get the REST API endpoint URL, the request payload, and the response payload of the decision model


Once a decision service is set up, you can note down the REST API endpoint URL, the request payload, and the response payload of the decision model to call a decision service.

1. In the decision editor, click **Decision Service**  to display the Services pane.
2. Click **Options** next to the service name, and then click **Payload**.
3. In the Service Payload pane that displays:
 - a. Click **URL** and then click **Copy**  to note down the decision service REST API endpoint URL.

The following is an example of a decision service endpoint URL:

```
https://<<base_url>/decision/api/v1/decision-models/<application_name>/
versions/<application_version>/active/definition/decision-services/
<decision_service_name>
```

Where:

- `base_url`: base URL of your Process Automation instance
 - `application_name`: the name of your process application (in case of local decision) or the name of your decision application (in case of external decision)
 - `application_version`: the version of your process application or decision application
 - `decision_service_name`: the name of your decision service
- b. Click **Request Payload** and then click **Copy**  to note down the request payload JSON schema.

The JSON schema of the request payload provides the format in which you provide the input data to the decision service.

- c. Click **Response Payload** and then click **Copy**  to note down the response payload JSON schema.

The JSON schema of the response payload provides the format in which the decision service provides the output data.

Add Decision Services to Processes

Once decisions are exposed as services, you can use the decision services within your processes.

To add a decision service to a process:

1. Open the process where you want to add the decision service.
2. In the Activity palette, expand **Decision Service**. All the available decision services are listed under it.

Note that if no decision services are available then the **Decision Service** section won't be displayed in the Activity palette.

3. Drag a decision service and drop it onto the process flow.

The decision service gets added to the process.

Another way to add a decision service to a process is by adding a decision activity to the process flow. You can drag and drop a decision activity to the process flow from the **System** category in the Activity palette. But in this case, you would need to implement the decision activity.

To implement a decision activity:

1. Double click the decision activity in the process flow to open the Properties pane.
Optionally, select the decision activity in the process flow, and click **Open Properties** to open its Properties pane.
2. Select a decision service from the **Decision Service** drop-down list.
3. Close the Properties pane to save and implement the changes.

8

Manage Data

Almost all business processes require some type of data. For example, a purchase order application requires customer name, contact information, order number, items purchased, and payment details. Before you can use data in your application, you must define how it's structured and stored.

Topics:

- [About Managing Data](#)
- [Work with Data Objects](#)
- [Work with Business Types](#)
- [Configure Data Association](#)
- [Define Conditions for Data Associations](#)
- [Work with Transformations](#)
- [Filter Arrays](#)
- [Work with Expressions](#)
- [Use Business Keys](#)

About Managing Data

Most business applications require users to create and manipulate data. For example, in a travel request application, a user enters data related to the request which includes information about the employee, to and from destinations, and other types of data.

Data is stored within a data or a business object. Create simple data and business objects from scratch using the Data pane from the process editor. Complex business objects are also auto-generated when designing forms using the form-first design methodology and when importing services into an application. In these situations, business objects are referenced by forms and services, and can only be deleted if the artifacts referring them are deleted. Business objects can be hierarchical - attributes can be simple or point to other business objects.

Work with Data Objects

Data objects are the variables used to store the information used by your business processes. They're defined during the design and implementation stage of a process. To complete a process application, you must ensure that you have associated each activity with the data objects it requires.

At runtime, data objects contain information that might be altered as users interact with your business process. Running processes can store, access, and manipulate data. The values stored within a data object can also be used to determine the branch that a process takes.



Oracle Cloud Infrastructure Process Automation supports data objects that are based on generally simple data types. However, complex business objects can be created indirectly, by

either creating a form or importing a JSON schema/sample in a connector. Which type of data type you base your data object on depends on the type of data it must handle.

Create a Data Object

You can create a data object based on a simple data object type or on a business object (complex data type). If you're creating a data object on a business object, you must create the business object first.

To create a data object:

1. Open a process, and then click **Data** .
2. In the Data pane, click **Create Data Object** .
3. In the resulting Create Data Object pane, enter a name for your data object or use the default name in the **Name** field.
4. Under Data Type, select **Simple** to create a simple data object, or select **Business** to create a data object based on a business object.
5. Click **Create**.

Note:

A default process data object with the name `TaskOutcomeDataObject` is automatically created for the human task outcome when a human task is created. The **Edit** and **Delete** links are disabled for this data object.


In the Data pane, you can also edit or delete a data object. After editing or deleting a data object, validate your application to verify that there are no references to the changed or deleted data object.

If your process contains an inline/embedded subprocess, then to view the data objects of the subprocess in the Data pane of the main (parent) process, you have to toggle the **Show subprocess data objects** button. Note that you won't be able to edit or delete the subprocess data objects from the Data pane of the main process. To edit or delete the subprocess data objects, you have to open the Data pane in the context of the subprocess.

Edit or Delete a Data Object


You can edit or delete data objects.

To edit or delete a data object:

1. Open the process where you want to edit or delete a data object.
2. Click **Data**  and in the Data pane that opens, expand the list of data objects for your process.
3. Select the specific data object.

This step activates the **Edit** and **Delete** icons.

- To edit the data object, click **Edit Data Object** , then change the name or data type as necessary.

- To delete the data object, click **Delete Data Object** .

4. Click **Close**.

After editing or deleting data objects, validate your application to verify that there are no references to the changed or deleted data objects.

After editing a data object, you must ensure that all references to it are still valid. For example, if you change a data object type from an integer to a string, you must verify that all of the expressions that use the data object still function correctly. If they don't function correctly, the application doesn't validate.

After deleting a data object, you must ensure that all references to it are removed. This includes any data associations and expressions that use the data object. If you don't remove references to the deleted data object, the application doesn't validate.

Work with Business Types

Business Types are containers for application data, such as an invoice or purchase order. In Oracle Cloud Infrastructure Process Automation, you can create Business Types, List of Values, and Business exceptions. You use them to group related data types and to define the structure for data used in your process application.

- **Business Types** are created by adding attributes. See [Create a Business Type](#).
- **List of Values** are created by adding set of predefined values. For example, the days of the week. See [Create a List of Values](#).
- **Business exceptions** are created by adding attributes. See [Create a Business Exception](#).


Once created, the business types and the list of values in your application are listed in the Types page.

Create a Business Type

You can create a business type using attributes to define the data it contains. For example, you can create an employee business type that contains different types of data such as name (string), Id (integer), and address (string).

To create a business type:

1. Click **Add** on the top right of the application home page, expand **Types** and click **Business Type**.

Alternatively, if you don't have any business type or list of values created yet for your application, click **Types** on the Components tab to open the Types page. On the Types page, click .

The Add Component pane appears.

2. In the Add Component pane, enter the following and click **Create**.
 - Enter a suitable name in the **Title** field.
 - If required, you can update the default unique identifier name in the **Identifier** field.
 - Optionally, enter a suitable description in the **Description** field.

A confirmation dialog lets you know that the component is being created. After it gets created, the business type is listed in the Types page.

3. Click the business type to open it in the editor.

4. Add attributes to your business type.
 - a. Under Add New, enter the attribute name in the **Name** field. For example, `firstName`.
 Note that the attribute name should always start with a lowercase alphabet. An inline validation error message lets you know about this.
 The attribute gets added in the Attributes pane. By default, the data type for a newly added attribute will be String.
 - b. Select a data type from the **Type** drop-down list if you want to change the default data type.


The screenshot shows a user interface for defining business type attributes. It includes a title 'What data does your business type contain?', an 'Attributes' list with input fields for 'Name' and a 'Type' dropdown, and an 'Add New' section with a 'Name' input field. A dropdown menu is open for the 'Type' field, showing options: String (selected), Boolean, Integer, and Number. There is a 'Make it an array' checkbox.


The different data types that you can assign to an attribute are:

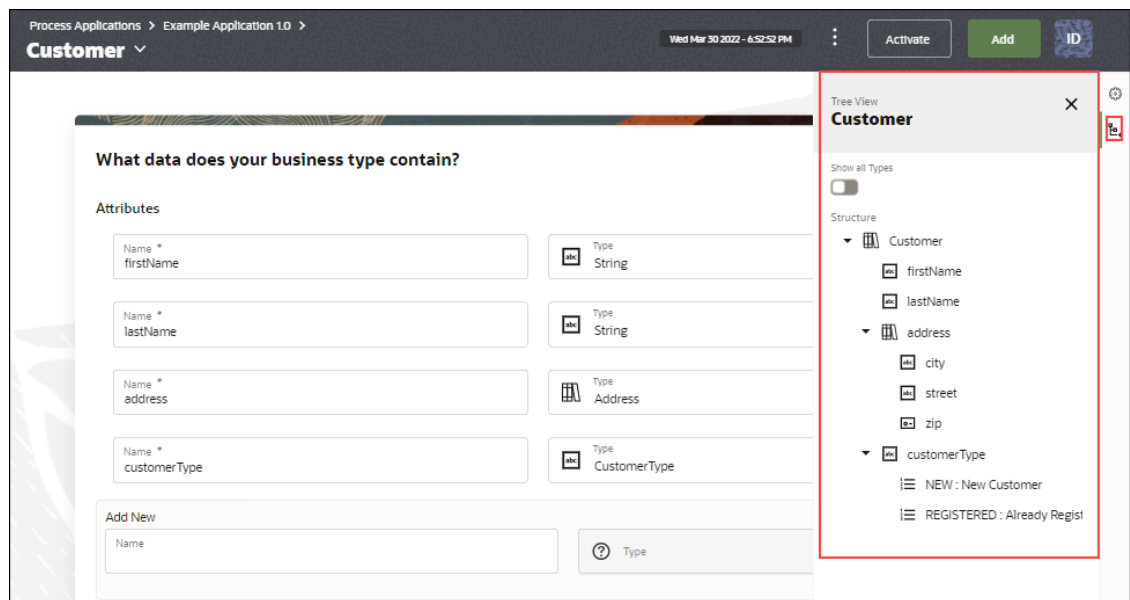
- Simple types
- List of values or Enums
- Form data objects
- Data types from connectors
- Data types from linked external UI
- Other business types

If you want to make the selected type as array, select the **Make it an array** check box.

To delete an attribute click the **Delete** icon next to the attribute.

Click the **Properties**  icon on the right side of the editor to open the Properties pane and edit the business type's properties. For example, you can edit the name and description of the business type.

Click the **Tree View**  icon on the right side of the editor to drill down and view the data structure of the business type. For example, the image below shows the tree view of a business type Customer. It is configured with simple data types (`firstName`, `lastName`), another business object (`address`), and a list of values (`customerType`).



You can toggle and enable **Show all Types** to view all types (business types and list of values) present in your application. Expand each type to view the details.

Create a List of Values

List of values (or Enums) are special types of business types that contains a set of predefined constants. For example, a list of values can contain the names of the days of a week (Sunday, Monday, Tuesday, and so on)

To create a list of values:

1. Click **Add** on the top right of the application home page, expand **Types** and click **List of Values**.

Alternatively, if you don't have any business types or list of values type created yet for your application, click **Types** on the components tab to open the Types page. On the Types page, click **+**.

The Add Component pane appears.

2. In the Add Component pane, enter the following information and click **Create**.
 - Enter a suitable name in the **Title** field.
 - If required, you can update the default unique identifier name in the **Identifier** field.
 - Optionally, enter a suitable description in the **Description** field.
 - Select a data type (String, Number, or Integer) from the **Data Type** drop-down list.

A confirmation dialog lets you know that the component is being created. After it gets created, the list of values is listed in the Types page.


3. Click the list of values to open it in the editor.
4. Under **Add New**, enter a label in the **Label** field and a corresponding value in the **Value** field.


The item gets added to the List of Values pane.

Add the required number of items to create the list of values.

You'll get inline validation error if you enter incorrect values. For example, you've created a list of values with *integer* data type but enter *string* while setting its values.

To delete an item click the **Delete** icon next to the item.

Click the **Properties**  icon on the right side of the editor to open the Properties pane and edit the list of values' properties. For example, you can edit the name, description or the data type of the list of values.

Click the **Tree View**  icon on the right side of the editor to drill down and view the data structure of the list of values.

You can toggle and enable **Show all Types** to view all types (business types and list of values) present in your application. Expand each type to view the details.

Use Business Types

You can use the business types and list of values that you have created in multiple ways in a process.

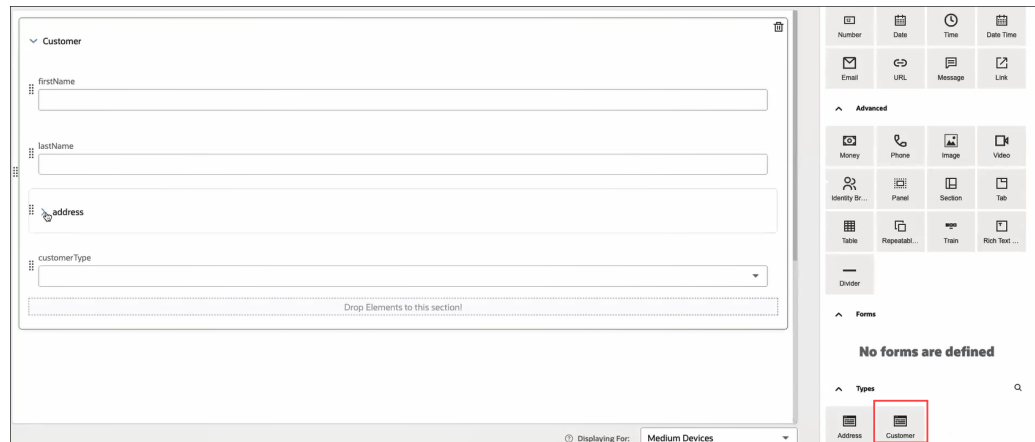
Let's explore through a simple use case how to use business types in a process.

1. Create a simple structured process to check customer onboarding.

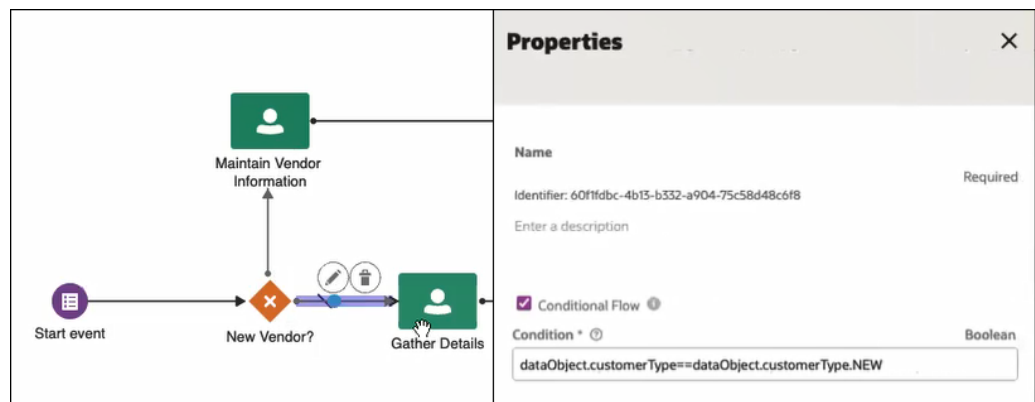
Process flow:

- The process starts with a start form that gathers customer information such as the first and last name, address, and also if the customer is new or already registered.
 - If the customer is new the process moves to the next task of gathering customer information.
 - Alternatively, if the customer is already registered, the process moves to the next task of fetching customer information.
 - When the task is submitted, required task notifications are sent and the process completes.
2. Create the following business types:
 - An *Address* business type for containing customer address data.

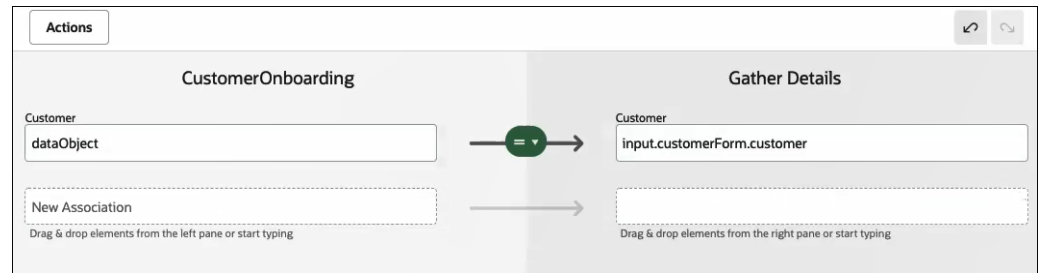
- A *CustomerType* list of values for containing the predefined values - New Customer and Registered Customer
 - A *Customer* business type for containing data such as customer's first and last name, address (using the Address business type), and the type of customer (using the CustomerType list of values).
3. Use the business types that you created in the following ways:
- Use business types in a form:** For the customer onboarding form, use the *Customer* business type. Business types that you've created are available under Types in the form palette. Drag and drop the business type from the form palette onto the form's canvas. You don't have to configure individual form fields but the form fields get automatically configured with the business type. Thus business types can be used to quickly build a complex form with its presentation.



- Use business types for configuring conditional flow:** You can use the *CustomerType* list of values to configure the conditional flow of the process.
 - If the customer is new then move to the next task to gather the new customer details.
 - Else, if the customer is already registered then move to the next task of maintaining the existing customer's information.



- Use business types in data association:** With business types, you can easily configure data associations. You can bind the process data objects directly to the form data objects that were created with the business type. You don't need to bind individual form data objects in your process, as you would have to do with normal form fields.



Create a Business Exception

Business exceptions are used to convey unexpected situations that can occur while running a business process. Define business exceptions for your application and use them in an error end event that is triggered under a certain condition, such as an invalid Id or poor credit rating.

A business exception has the following three attributes:


- code and type String. This attribute is not editable.
- message and type String. This attribute is not editable.
- details. This attribute is editable. The attribute created based on the type selected.

Data association displays these attributes for the following events:

- Boundary event
- Error event subprocess
- Error end event

To create a business exception:

1. Click **Add** on the top right of the application home page, expand **Types** and click **Business exception**.

Alternatively, if you don't have any types created yet for your application, click **Types** on the Components tab to open the Types page. On the Types page, click .

The Add Component pane appears.

2. In the Add Component pane, enter the following and click **Create**.
 - Enter a suitable name in the **Title** field.
 - If required, you can update the default unique identifier name in the **Identifier** field.
 - Optionally, enter a suitable description in the **Description** field.


A confirmation dialog lets you know that the component is being created. After it gets created, the business exception is listed in the Types page.


3. Click the business exception to open it in the editor.
4. Add the attribute details to your business exception.
 - Select a data type from the **Type** drop-down list if you want to change the default data type.

The different data types that you can assign to an attribute are:

- String
- Integer
- Boolean

- Date
- Time
- Number

Click the **Properties**  icon on the right side of the editor to open the Properties pane and edit the business exception's properties. For example, you can edit the name and description of the business exception.

Click the **Tree View**  icon on the right side of the editor to drill down and view the data structure of the business exception.


Configure Data Association

Data association refers to the flow of data within a process. Use the data association editor to define input and output for activities that need them.

A data association involves a source and a target, where the source provides a value or an expression to be assigned to the target. An approval human task, for example, needs both input and output data association.

- On the input side, it needs data input into the activity (referred to as its payload).
- On the output side, after the activity has just finished, it needs output from the activity to data objects, to store results for use elsewhere in the process.


To configure data association:

1. For a structured process, select an activity that needs data association, such as a human task or service task, click  and then select **Open Data Association**. (This option is active when the selected activity needs association.)

The data association editor opens. Use its center pane as a canvas for mapping associations, by dragging and dropping elements from the left and right panes. Any data associations already configured display below **Data Association**. You use the empty pair of fields to create a new association.

Note:

Oracle Cloud Infrastructure Process Automation creates data associations for you in some cases, such as when you create a human task or start event with a form.

- a. Notice that the **Input** tab is selected. The left pane displays source objects (**Data Objects**) in an expandable tree. The right pane displays the payload, or entry parameters the activity needs to perform its function.
 - b. Click the **Output** tab and notice how the left and right panes change. Now the left pane displays the task's payload, and the right pane displays the variables available in the process.
2. For a dynamic process, select a human task activity or a service activity. Click , select **Data Association** and then select **Input** to configure input data association. (This option is active when the selected activity needs association.)

You can configure output data association by selecting **Output**.

3. In the **Input** tab begin creating a new association by dragging an input object from the Data Objects tree and dropping it in the input field titled *New Association*. You can also begin typing and select from auto-complete options that appear using the inline expression builder. To specify attributes within objects, enter a . (period) and select from the list that displays.

See [Work with Expressions](#).

The input source provides the value or expression to assign to the target. You select variables related to the process (such as data objects) to map to the specific parameters the activity needs to perform its function (its entry parameters).

4. Complete the association by dragging an output object from the payload tree and dropping it in the output field. You can also begin typing and select from auto-complete options that appear using the inline expression builder.

Oracle Cloud Infrastructure Process Automation validates the new association and displays a green association icon if valid, or a red association icon and error if invalid:



Valid data association



Invalid data association

See [Work with Expressions](#).

See [Data Association Tips](#).

5. If needed, create additional associations and reorder them.

Drag and drop an association to move it up or down in the associations list. Associations are executed in the order in which you position them. For example, if multiple associations assign a value to the same object, the last assigned value is used.

As you edit, click the **Undo** or **Redo** buttons as needed.


6. Click the **Output** tab, and create output data associations. You can think of the output as though the activity has just finished, and likely contains results that you'd like to store for use elsewhere in the process.
7. Click **Apply** to save the data associations.



You can save invalid associations and fix them later. Note that you can't deploy applications that contain validation errors.

Data Association Tips

Use the Data Association Editor to add, edit, delete, and reorder data associations.

Below are tips for configuring data associations.

- The following activities need data association. Note that some activities have only output, such as a start form event, which captures the data end users enter into the form.
 - Human
 - System
 - Events, except timer catch and end events
- You can access the data association editor:
 - By clicking an activity in a structured process, clicking the **More**  icon, and choosing **Open Data Association**.

- By selecting an activity in a structured process, and then clicking the **Data Association**  icon that is present at the top right of the process editor.
- By clicking a human or service activity in a dynamic process, clicking the **More**  icon, and then selecting **Data Association**. Choose **Input** to configure the activity's input data, and choose **Output** to configure the activity's output data.
- In the Data Association Editor, click the **Input** tab to define data input into the activity, and the **Output** tab to define data output resulting from the activity.
- Dropping an object into an association field that already contains a value replaces its value. If you drop an object into an input field that ends with an operator, the object name is appended to the field's contents.
- You can use *predefined variables* in data associations for structured and dynamic processes. Predefined variables contain process execution information. They enable users to use process execution information such as `instanceId`, `creatorName`, `createdDate` and so on in data association. Note that predefined variables are read only, and their values can't be edited.

Human Task Exec Variables

The following human task exec variables are supported in output data associations:

Variable	Description	Type
<code>taskExec.taskId</code>	Task number	String
<code>taskExec.title</code>	Title	String
<code>taskExec.approverDisplayName</code>	First name and last name of latest approver.	String
<code>taskExec.summary</code>	Summary	String
<code>taskOutcome</code>	Outcome of a task. For example, APPROVE or REJECT.	String
<code>taskExec.approvers</code>	Details of each approver. Use when multiple approvers act on a task.	Array of objects, contained fields.
<code>taskExec.approvedDate</code>	Date of task action. For example, date of approval.	DateTime
<code>taskExec.approver</code>	System identifier of latest approver.	String
<code>taskExec.createdDate</code>	Created Date	DateTime
<code>taskExec.approvedDate</code>	Date on which action is taken. For example, the date when approval is taken for a task.	DateTime
<code>taskExec.assignedDate</code>	Assigned Date	DateTime
<code>taskExec.priority</code>	Specify the priorities as: <ul style="list-style-type: none"> • 1 for high • 2 for medium • 3 for low 	int
<code>taskExec.fromUser</code>	Creator or user who reassigned the task.	Object (Identity Type). Contains the following fields: <ul style="list-style-type: none"> • id • name • display name • type

Variable	Description	Type
taskExec.approverName	Username of the latest approver.	String
taskExec.dueDate	Due date	DateTime

Process Variables

The following process variables are supported in output data associations:

Variable	Description	Type
businessKey	Business key	String
creatorid	IDCS Id	String
creatorType	User applicant	String
createdDate	Date of creation	DateTime
documents	Document	Document
processExec.title	Process title	String
processExec.instanceId	Process Id	String
processExec.creatorName	Creator username	String
processExec.creatorDisplayName	Creator display name	String

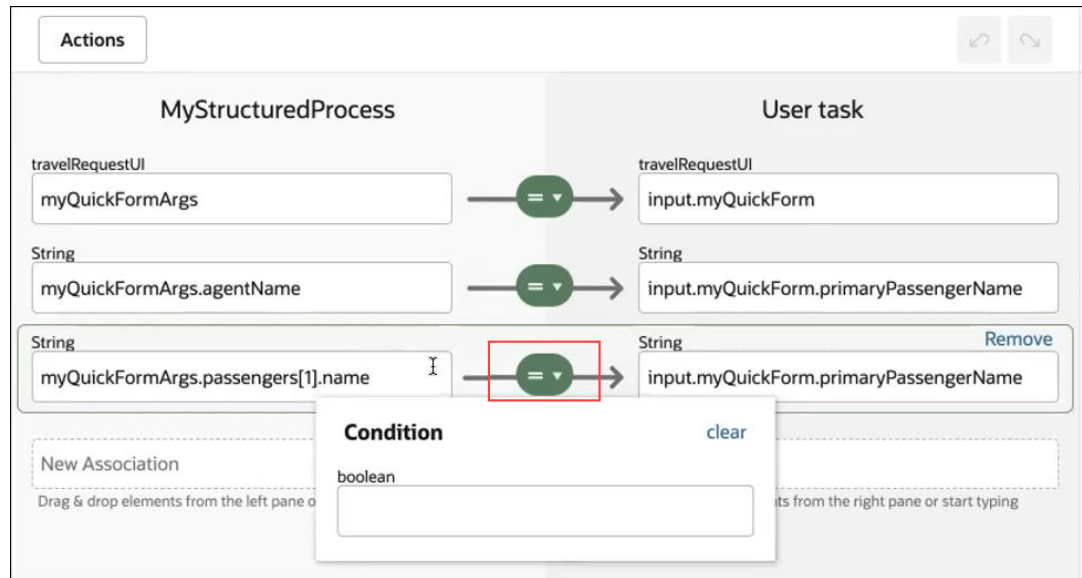
Define Conditions for Data Associations

There may be a business scenario where you want a data association to execute in runtime only when a particular condition fulfills. Control the execution of data associations in runtime by adding conditions to them in the data association editor.

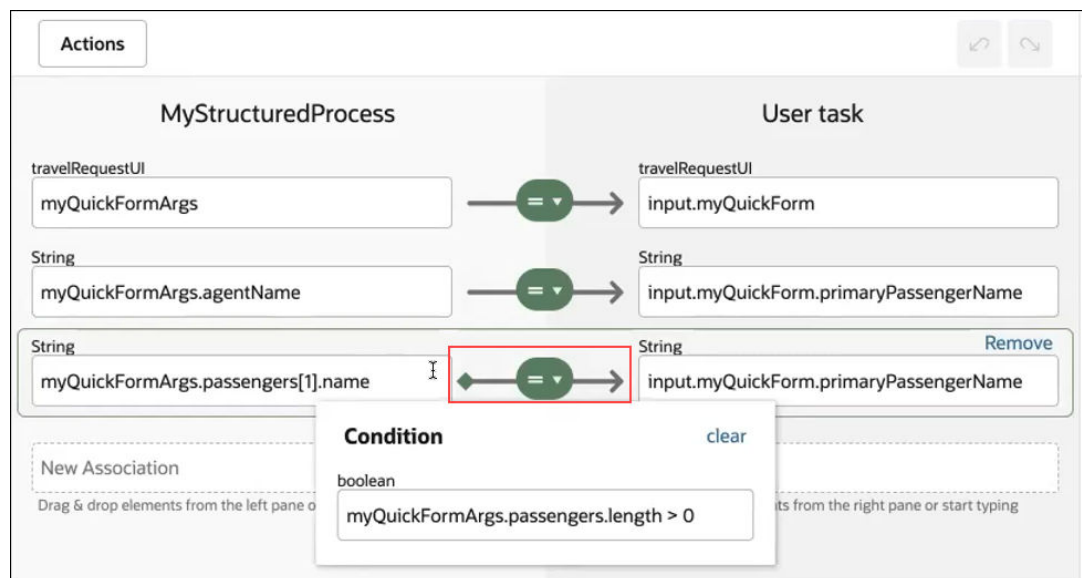
A data association that has been configured with conditional mapping executes in runtime only when the defined condition fulfills. Otherwise, it fails to execute. All other data associations that do not have conditional mapping defined execute normally in runtime.

To define a new condition:

1. Click the **Data Association** icon in the data association for which you want to configure a condition. You can define the new condition in the resulting dialog box.



2. Define the new condition by entering values directly in the given field.
 3. Alternatively, use the inline expression builder to define a condition using standard functions and operators.
- See [Work with Expressions](#).
4. If the condition is valid a green icon is displayed next to the **Data Association** icon.



5. If the configured condition is invalid, the icon turns red indicating an error



Note that you can also view the error details when you click the process validation error from the process editor canvas.

6. Finally, click **Apply** to save all data associations including the data associations with conditions set on them.

In runtime, the data associations for which you have defined conditions execute only if the conditions fulfill. In case of our example, if the first passenger name was entered in the Start activity form, then its value will be passed on as payload to the primary passenger field in the

next activity of the process. Otherwise, the travel agent name will appear in the primary passenger field.

Work with Transformations

A transformation is a special type of data association between data types that don't match. For example, the service that you're calling in your process may return an employee representation that's slightly different than the one you have modeled on a data object. Using transformations you can associate and map these different data types.

Transformations are reusable; once configured, you can use them wherever applicable in your process application.

Transformations have their own icon, which slightly differs from the data association icon. As

with data associations, the transformation icon  is green when valid. When invalid, the icon is red and its error displays.

You can create and apply data transformations in the following ways:

- Create and apply transformations while configuring data associations
- Create individual transformations and then apply them to data associations

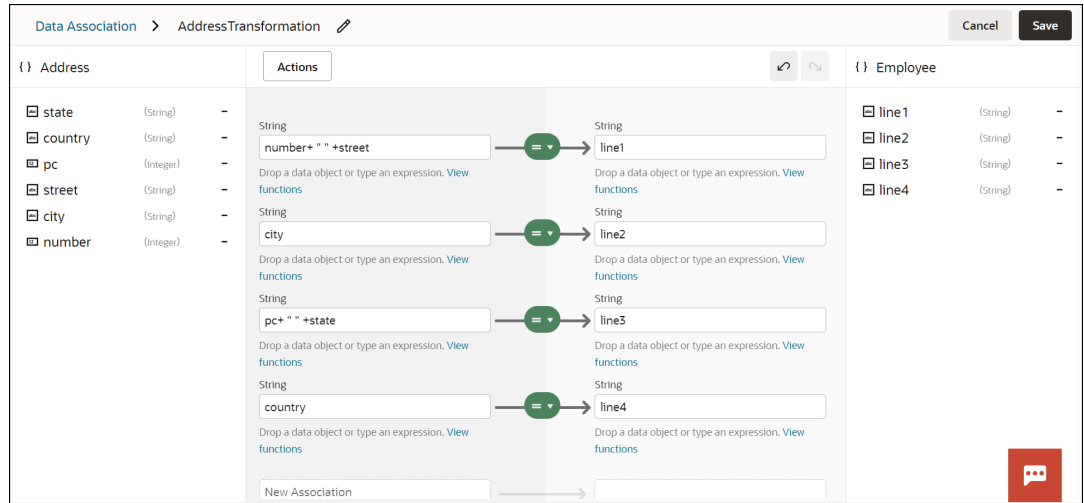
To create and apply transformations while configuring data associations:


1. Select an activity in a process that needs data association and click **More Options**.
2. Select **Data Association** and choose either **Input** or **Output**.
3. Drag and drop the data objects into the input and output fields to configure an association. You can also begin typing and select from auto complete entries that display.
4. Click the data association icon, and in the resulting dialog toggle the **Transform** button.
The data association can be valid or invalid.
5. The Transformations window (named based on the objects you selected) displays the list of transformations that match the source and target types.

Click **Create** to create a new transformation.

6. In the Transform dialog box, enter a name, and click **Create**.

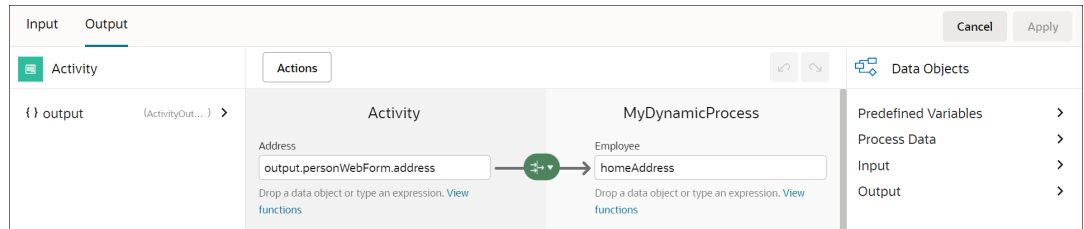
The Transform window displays. It looks and functions much like the data association editor, except that its left and right panes reflect the fields from the selected objects to transform.



 **Note:**
You can also create transformations within transformations, that is, nested transformations. To create nested transformations, repeat steps 4 to 6 in the Transform window, and click **Save**.

7. Click **Save** to save the transformation.

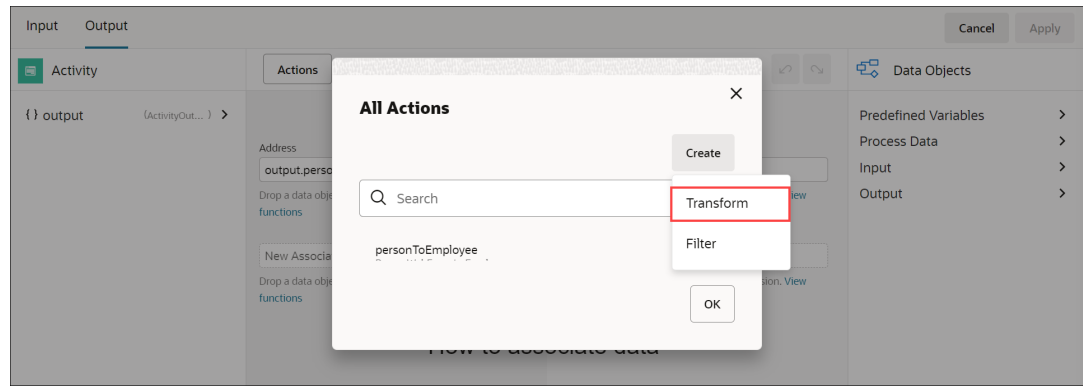
Process Automation validates the new transformation, displaying a green transformation icon if valid or a red transformation icon with error if invalid. Note that you can save an invalid transformation and fix it later.



To create individual transformations and then apply them to data associations:

1. Use the **Actions** button on top of the data association editor to open the All Actions window.

The All Actions window displays all transformations that are defined for the process. Click **Create**, and select **Transform**.



Note:

Select Filter to create a filter. See Filter Arrays.

Additionally, you can either edit or delete an existing transformation. To do so, click the edit or delete icon next to the transformation in the All Actions window.

2. In the Transform dialog, enter a name for the transformation and select the source and target data objects.
3. Click **Create**.
In the Transform window, configure the transformation.
4. Click **Save**.
You've now created a transformation.
5. In the data association editor, click the data association icon, and in the resulting dialog toggle the **Transform** button.
6. The Transformations window displays the list of data transformations in the process application. Select a transformation that is applicable for the association and click **OK**.
7. To apply a different transformation for an already configured transformation:
 - a. In the data association editor, click the data transformation icon. The resulting dialog lists the transformations applicable for the association. Click **change** next to the transformation you want to update.
 - b. Select another transformation from the Transformations window and click **OK**.

Append to Array

An array is a series or list of data elements of the same type. You can add a single data element to an array with the append action available in the data mapper. For example, you can append strings to a string array.

The append action is available in the Data Association editor for both structured and dynamic processes.

Ensure the following:

- The target expression type must be always an array.
- The source expression type must be of the same data type as the target array.

- Both the source and target expressions have to be valid.

To append to an array while configuring data associations:

1. Select an activity in a process that needs data association and click **More Options**.
 2. Select **Data Association** and choose either **Input** or **Output**.
 3. Drag and drop the data objects into the input and output fields to configure an association. You can also begin typing and select from auto complete entries that display.
 4. Click the data association icon, and in the resulting dialog toggle the **Append** button.
- The data association can be valid or invalid.

Filter Arrays

While defining data associations between process components, you can selectively filter elements from a source array and assign them to a target array object.

In the Data Association editor of a structured or dynamic process, an option to define a filter appears if the following conditions are met:

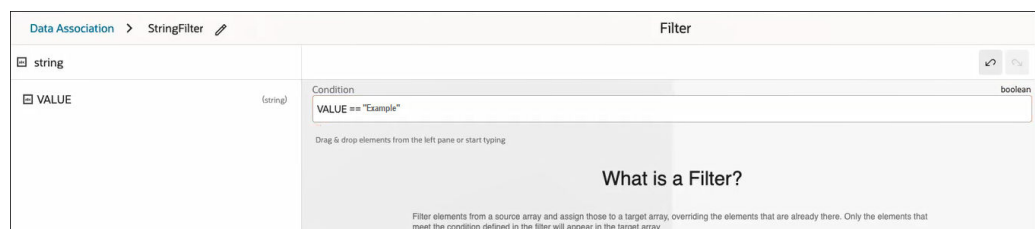
- The target expression type is an array.
- The source expression type is an array. The base type of the source is assignable to the base type of the target (for example, integer to double assignments).
- Both the source and target expressions are valid.

To define a new filter:

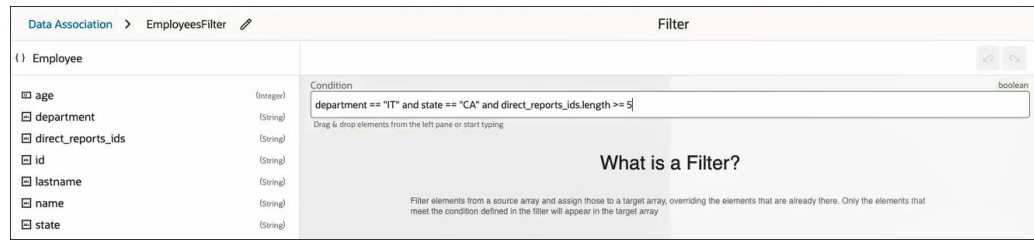
1. Click the Data Association icon.
2. In the Actions pop-up, toggle the **Filters** button.
3. In the resulting dialog, click **Create**.

You can also do the following:

- Select and use an already existing filter.
 - Search for an existing filter by entering its name in the Search field.
 - Edit a filter, by selecting it and then clicking the **Edit** icon.
 - Delete a filter, by selecting it and then clicking the **Delete** icon.
4. Enter a name for the filter in the **Name** field, and click **Create**. A window to define the filter condition appears.
 5. Based on the type of your array, define a filter condition.
 - a. For a simple array object (of the type integer, string, or double), an element named **VALUE** is present in the window, which represents each element of the source array. Drag and drop the **VALUE** element into the **Condition** field to define an expression.

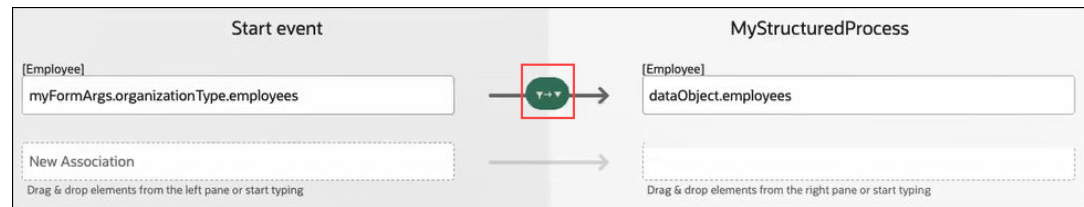


- b. For a complex array data object, you can define a condition comprising all attributes it contains.



You can use the inline expression builder to create and define expressions in the filter **Condition** field. See [Work with Expressions](#).

6. Click **Save** to save the filter condition.
7. Note that the Data Association icon changes to show that a filter was applied to the data association.



8. Click **Apply**.


In runtime, target arrays are populated according to the filter conditions you've defined.

Work with Expressions

Use expressions to evaluate and perform calculations on data stored in data objects, process pre-defined variables, process input, process output, and task outcomes.

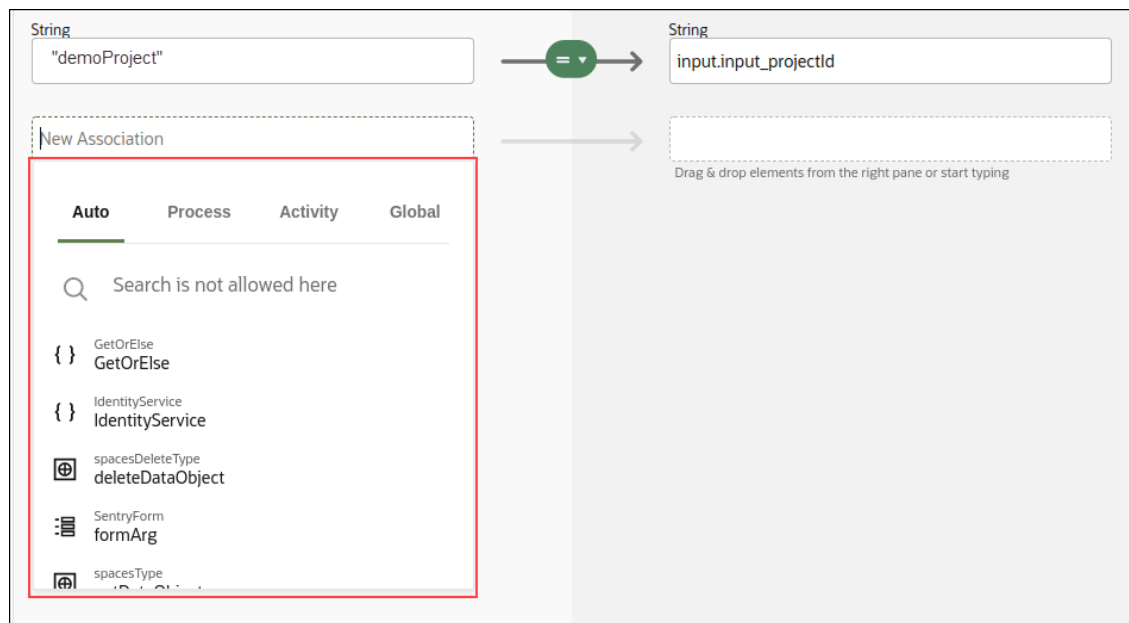
Use expressions to change values passed to and from a sequence flow. You can create expressions when configuring data associations, and when configuring properties for these activities:

- Human tasks: Use an expression to dynamically determine the task title, summary, due date, and assignees.
- Conditional sequence flows: Use an expression to define its condition.
- Conditions and markers: Use an expression to define activation conditions and markers.

To create an expression for a flow element, click  from the flow element's properties field. This enables the expression mode and you can use the inline expression builder which gives auto-complete options for you to create and build your expression.

To create an expression while configuring data associations, enter the expression directly in the input or output fields of the data association editor. You can use the inline expression builder which gives auto-complete options for you to create and build your expression.

The following is an example image of the inline expression builder being used while configuring data association.



- Create expressions with data objects, operators, and functions. As soon as you put the cursor in the expression field, you get a list of suggested objects using which you can build the expression.
- As you choose the object, related suggestions for the next level are displayed below the field. The suggestions are context sensitive. For example, suggestions displayed for a **Title** field will be different from that of a **Due Date** field.
- Use Ctrl + Space to get more tabs and suggestions. A search field also displays that you can use to search data objects under the various tabs such as **Process** or **Activity**.
- The type that is expected to be entered into the expression field is auto-suggested at the top right of the field.
- If an expression is invalid, for example, if you type a number in a string field, an error message indicating the details of the error is displayed.
- If the configured expression is too long, you can expand the field by using Ctrl + Enter keys.

About Simple Expressions

Simple expressions are defined using a basic expression language and support. You can use these operators to write expressions and conditions to define your process flow. Generally these expressions perform their calculations based on the data objects in your business process. You can write expressions and conditions using the value of the data objects, but you can't explicitly modify the value within the data object.

Here are some examples of expressions using operators:

- totalAmount - discount
- activationCount > 3
- unitsSold <= 1200

Operator Precedence

Operator precedence defines the order in which the compiler evaluates operators. You can change operator precedence in an expression by using parentheses. In Oracle Cloud Infrastructure Process Automation, the operator precedence is:

- Addition, Subtraction
- Multiplication, Division, Remainder
- Plus, Minus
- Less than, Greater than, Less than or equal to, Greater than or equal to
- Equals, Not equals
- Not
- Conditional And
- Conditional Or

The following sections lists Operators and Functions per type.

Unary

Operator	Name	Description
+	Plus	Has no effect on the value of the numeric operand. Use it to explicitly indicate that a certain value is positive.
-	Minus	Negates an arithmetic expression.
!	Not	Logical complement operator. Negates the value of a Boolean expression.

Equality and Relational

Operator	Name	Description
= or ==	Equal to	Returns true if the first operand is equal to the second operand
!=	Not equal to	Returns true if the first operand isn't equal to the second operand
>	Greater than	Returns true if the first operand is greater than the second operand
>=	Greater than or equal to	Returns true if the first operand is greater than or equal to the second operand
<	Less than	Returns true if the first operand is less than the second operand
<=	Less than or equal to	Returns true if the first operand is less than or equal to the second operand

Conditional

Operator	Name	Description
and	Conditional And	Returns true if both operands evaluate to true
or	Conditional Or	Returns true if either operand evaluates to true

String

Function/Operator	Description	Usage Expression	Usage Result
+	String concatenation	"pine" + "apple"	"pineapple"
==	Equals	"apples" == "apples"	true
!=	Not equals	"apples" != "oranges"	true
>	Greater than	"word" > "work"	false
>=	Greater than or equals	"work" >= "work"	true

Function/ Operator	Description	Usage Expression	Usage Result
<	Less than	"word" < "work"	true
<=	Less than or equals	"work" <= "work"	true
contains	Returns true if the first argument string contains the second argument string; otherwise returns false	"caramel".contains("ram")	true
endsWith	Returns true if the first argument string ends with the second argument string; otherwise returns false	"immutable".endsWith("table")	true
length	Returns the number of characters in a string	"house".length()	5
lowerCase	Returns a string with all the characters in the argument converted to lower-case representation	"Example".lowerCase()	"example"
startsWith	Returns true if the first argument string starts with the second argument string, otherwise returns false	"caramel".startsWith("car")	true
substring	Returns the substring of the first argument starting at the position specified in the second argument and continuing to the end of the string	"care".substring(2)	"are"
substring	Returns the substring of the first argument starting at the position specified in the second argument with length specified in the third argument	"care".substring(1,3)	"car"
upperCase	Returns a string with all the characters in the argument converted to upper-case representation	"Example".upperCase()	"EXAMPLE"
replaceAll	Replaces each substring of this string that matches the given pattern with the given replacement.	"care".replaceAll("e", "t")	"cart"
indexOf	Returns the index of the first occurrence of the specified substring, or -1 if there is no such occurrence.	"care".indexOf("a")	2

Numeric

The following table includes operators for both `Integer` and `Number`.

Operator	Description	Usage Expression	Usage Result
+	Addition	2 + 8	10
-	Subtraction	7 - 4	3
*	Multiplication	3 * 4	12

Operator	Description	Usage Expression	Usage Result
/	Division	3 / 2	1.5
%	Remainder	3 % 2	1
==	Equals	12 == 13	false
!=	Not equals	12 != 13	true
>	Greater than	15 > 16	false
>=	Greater than or equals	15 >= 15	true
<	Less than	12 < 10	false
<=	Less than or equals	12 <= 12	true
abs	Returns the absolute value of a number	abs(- 6)	6

In addition to the above, the following specific operators are also available for `Number`.

Operator	Description	Usage Expression	Usage Result
floor	Returns the largest (closest to positive infinity) number that isn't greater than the argument and is an integer	floor(5.60)	5
ceil	Returns the smallest (closest to negative infinity) number that isn't less than the argument and is an integer	ceil(5.60)	6
round	Returns the number that is closest to the argument and is an integer	round(5.60)	6

Date and Time

Date and Time types include `Date`, `Time`, and `DateTime`. All of them have the following operators.

Operator	Description
+	Addition (valid only when the second argument is a duration)
-	Subtraction (valid only when the second argument is a duration)
==	Equals
!=	Not equals
>	Greater than
>=	Greater than or equals
<	Less than
<=	Less than or equals
format	Returns the formatted string of date-time using the provided format picture

In addition, `Date`, `Time`, and `DateTime` have specific operators as listed under each of them.

Date

Operator	Description
year	Returns a number representing the year component of the date-time argument.

Operator	Description
month	Returns a number representing the month component of the date-time argument.
day	Returns a number representing the day component of the date-time argument.

Time

Operator	Description
hours	Returns a number between 0 and 23, both inclusive, representing the hours component of the date-time argument
minutes	Returns a number between 0 and 59, both inclusive, representing the minutes component of the date-time argument.
seconds	Returns a number between 0 and 59, both inclusive, representing the seconds component of the date-time argument.

DateTime

Operator	Description
year	Returns a number representing the year component of the date-time argument.
month	Returns a number representing the month component of the date-time argument.
day	Returns a number representing the day component of the date-time argument.
hours	Returns a number between 0 and 23, both inclusive, representing the hours component of the date-time argument.
minutes	Returns a number between 0 and 59, both inclusive, representing the minutes component of the date-time argument.
seconds	Returns a number between 0 and 59, both inclusive, representing the seconds component of the date-time argument.
timezone	Returns an interval value, representing the time offset from UTC.

Operator	Description
toTimezone	<p>Returns the date-time expressed in the time offset corresponding to the timezone ID provided.</p> <p>You have the following options for specifying a timezone ID:</p> <ul style="list-style-type: none"> • Provide a fixed offset from UTC/Greenwich, such as "-07:00". <p>For example:</p> <ul style="list-style-type: none"> – <code>dateTimeDO.toTimezone("-07:00")</code> returns 2002-11-30T17:20:00-07:00 when <code>dateTimeDO</code> is 2002-12-01T01:20:00+01:00 • Specify a geographical region, which is an area where a specific set of rules for finding the offset from UTC/Greenwich apply. A geographical region is usually represented in the format "{area}/{city}", such as "Europe/Amsterdam" or "America/New_York". <p>For example:</p> <ul style="list-style-type: none"> – <code>dateTimeDO.toTimezone("America/Los_Angeles")</code> returns 2002-11-30T16:20:00-08:00 when <code>dateTimeDO</code> is 2002-12-01T01:20:00+01:00

 **Note:**

You must include double quotation marks around the timezone ID value.

For more information about timezone IDs, see [Class ZoneId](#) in the Java Platform documentation.

Boolean

Operator	Description	Usage Expression	Usage Result
==	Equals	<code>true == true</code>	true
!=	Not Equals	<code>true != false</code>	true
and	Conditional — And	<code>true and false</code>	false
or	Conditional — Or	<code>true or false</code>	true
not	Logical complement operator, inverts the value of a Boolean expression.	<code>not true</code>	false

Duration

Operator	Description
==	Equals
!=	Not equals
>	Greater than
>=	Greater than or equals
<	Less than
<=	Less than or equals

Array

Operator	Description
[]	Access a particular element into the array
==	Equals
!=	Not equals
length	Returns the number of elements contained within the array

Other

Operator	Description
==	Equals
!=	Not equals

Special Constants

Constants	Description
null	Null value
true	Logical true
false	Logical false
'now'	Current dateTime

Casting

In some cases, it could be desirable to bypass the type validation in order to assign types that aren't necessarily compatible. For example, you may want to assign an 'int' value to a 'string' one and in order to do that you can use the conversion operation like this:

<conversionTypeName> (<valueToConvert>)

where the 'conversionTypeName' is the type you want to see as the value.

Here are some conversion examples:

- string(myIntDO)
- int(myStringDO)
- duration(mystringDO)

Note:

You can only cast to primitive types, so the 'conversionTypeName' will only accept those that have a valid value.

Assigning two values that are incompatible will result in a runtime error.

Additional Functions

Identity Service

Function	Description	Available	Usage Function Prototype	Usage Example
getUserId	Returns a string containing the id for a specific user name.	Process wide	IdentityService.getUserId(<userName: string>): string	IdentityService.getUserId("wfaulkner")
getUserIds	Returns a string containing the ids (separated by commas) for a list of user names (also separated by commas).	Process wide	IdentityService.getUserIds(<userNames:string>): string	IdentityService.getUserIds("wfaulkner, jstein")

Get or Else

Function	Description	Available	Usage Function Prototype	Usage Example
boolean	Request an object property along with a backup value. The backup value is used if	Process wide	GetOrElse.boolean(<expression:boolean>, <fallbackValue:boolean>): boolean	GetOrElse.boolean(myDO.boolAtt, true)
string	the first argument corresponds to a missing node or an uninitialized value. This	Process wide	GetOrElse.string(<expression:string>, <fallbackValue:string>): string	GetOrElse.string(myDO.stringAtt, "User")
number	function set is useful while dealing with an untrustworthy data source, where	Process wide	GetOrElse.number(<expression:number>, <fallbackValue:number>): number	GetOrElse.number(myDO.numberAtt, 25.3d)
integer	initialization isn't guaranteed.	Process wide	GetOrElse.integer(<expression:integer>, <fallbackValue:integer>): integer	GetOrElse.integer(myDO.integerAtt, 12)

Use Business Keys

Business keys are application specific identifiers that are used to tag related processes and tasks within an application. You can define business keys using the **businessKey** predefined variable under process data objects.

For example, a company can have multiple shipping orders tied to the same main order ID. When a process is started to ship an order, that process instance can be linked to the main order by using the main order ID as the business key. At a time, multiple shipping orders can occur. Using the business key, the company can track all shipping order process instances that are tied to the same order ID.

Set business key value in the start of a structured or dynamic process. In the Properties pane, specify the business key in the **Business key** field. You can use the in-built expression builder to set the business key. See [Work with Expressions](#).

- Set a business key to a structured process that starts with a form or a message.
- Set a business key to a dynamic process that starts with a form.
- Propagate the business key to any task or child processes called by the process through data association.

Let us see how you can set a business key to a process and propagate it to a child process that is called by the main process.

1. Set the business key to the process start.
 - a. Create a structured process that starts with a form.
 - b. Select the start activity of the process, and open its Properties pane.
 - c. Under the End User Display section, enter a name for the activity in the **Title** field.
 - d. In the **UI** field, select a form to associate with the activity from the drop-down list. We'll use a data object of this form to configure the business key.
 - e. In the **Business key** field, enter an expression using the in-built expression builder. For example, `bKStartFormArgs.orderID`, where `orderID` is the data object of the Start form.

Properties [X]

Any User with Use Permission

End User Display

Title *

Start form

UI *

BK-Start-Form

Presentation

BK-Start-Form

Data Binding Reference: bKStartFormArgs

Business key

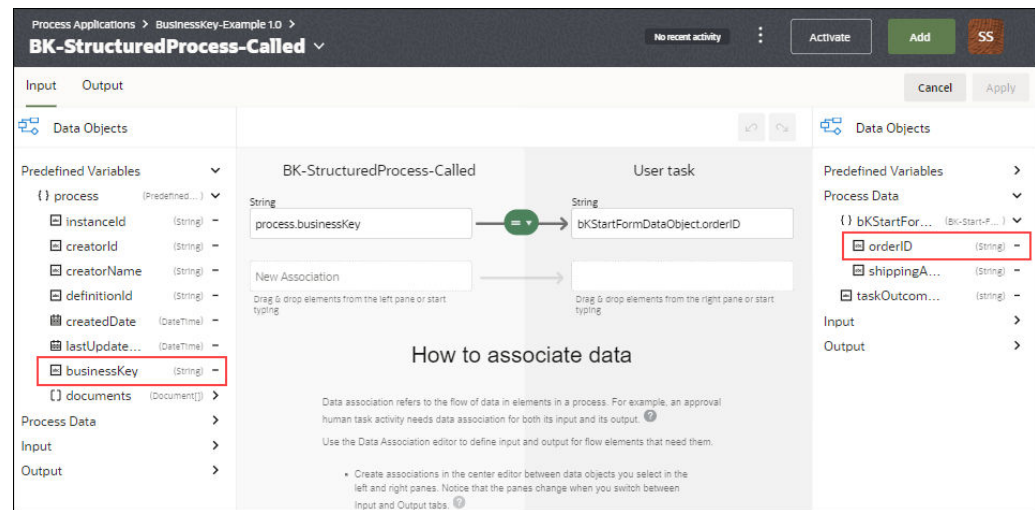
Expression ⓘ string

bKStartFormArgs.orderID

- f. Close the Properties pane to save the changes.
2. Create another structured process, and associate/tag the business key to it.
 - a. Create a structured process that starts with an empty start event.
 - b. Add a human task activity into the process flow.
 - c. Open the Data Association editor for the human task. Click the human task, and from the menu that displays select **Open Data Association**.
 - d. On the Data Association editor's **Input** tab, from the left Data Objects pane expand **Predefined Variables**, then drag and drop the `businessKey` variable into the **Input** field.

- e. From the right Data Objects pane, expand **Process Data**, then drag and drop the process data object you want to associate with the business key.

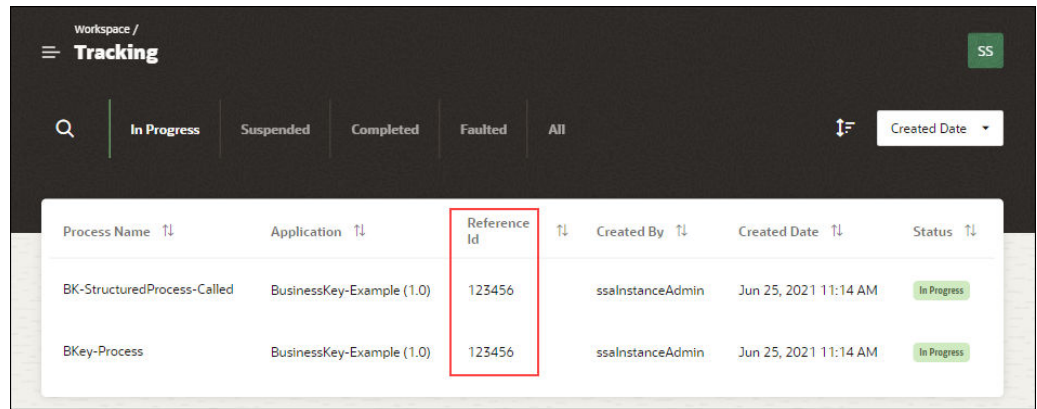
In the example below, the `orderId` data object is associated with the `businessKey` predefined variable.



- f. Click **Apply**.
3. Propagate the business key to the called process.
 - a. Open the first structured process where you defined the business key to the process start.
 - b. In the Activity palette, expand **System**, and then drag and drop a **Call Activity** into the process flow.
 - c. Select the call activity, then click **Open Properties** to open its Properties pane.
 - d. Under Implementation, select the second structured process you created in step 2 from the drop-down list in the **Process** field.
 - e. Select the check box **Propagate business key when process is called**.

The business key that is set at the start of the parent process is propagated to the child called process.

4. Activate and run the process application to see how it works in run time.
 - a. Click **Activate**.
 - b. In the Activate dialog, click **Test in Workspace**.
 - c. From the Start Requests page, select your application instance.
 - d. In the application start form that appears, enter a value, for example 123456, in the Order ID field that is configured for the business key.
 - e. Navigate to the Tracking page. Select **Tracking** from the workspace navigation menu options.
 - f. All the process instances associated with the same reference ID are displayed. The reference ID's value is same as the business key value you entered in the start form.



The screenshot shows the Oracle Tracking interface. At the top, there is a breadcrumb 'Workspace / Tracking' and a search icon. Below this are tabs for 'In Progress', 'Suspended', 'Completed', 'Faulted', and 'All'. A filter dropdown is set to 'Created Date'. The main content is a table with the following columns: Process Name, Application, Reference Id, Created By, Created Date, and Status. The 'Reference Id' column is highlighted with a red box. Two rows of data are visible, both with 'In Progress' status.

Process Name	Application	Reference Id	Created By	Created Date	Status
BK-StructuredProcess-Called	BusinessKey-Example (1.0)	123456	ssalInstanceAdmin	Jun 25, 2021 11:14 AM	In Progress
BKey-Process	BusinessKey-Example (1.0)	123456	ssalInstanceAdmin	Jun 25, 2021 11:14 AM	In Progress

The business key associated with a task is displayed in the My Tasks page. When you open a task, you can see if there is a business key associated with it under the More Information section.

9

Integrate with Applications and Services

Learn about the possibilities for integrating process applications with other applications and services.

Topics:

- [Work with Connectors](#)
- [Work with Integrations](#)
- [Embed a Start Form in an External Application](#)
- [Embed a Task List in an External Application](#)
- [Embed a Task Detail in an External Application](#)

Work with Connectors

Connectors enable process applications to interact with other Oracle Cloud applications via REST API.

Learn how to configure REST connectors in Designer, specify connector visibility in process editors, and use connectors in process applications

- [Connectors at a Glance](#)
- [Configure REST Connectors in Designer](#)
- [Specify Connector Visibility in Process Editors](#)
- [Use Connectors in Process Applications](#)
- [Delete a Connector](#)

If you're a Process Automation Administrator, learn additional details about managing credentials. See [Manage Credentials in Workspace](#).

You can also create connectors using active integrations from Oracle Integration designed with REST triggers. See [Work with Integrations](#).

Connectors at a Glance

Start by familiarizing yourself with these connector concepts.

How are connectors used in process applications?

Use an outbound connector to call a REST service to retrieve, create, update, or delete data on a web server that supports the REST architecture. The connector enables process applications to interact with other Oracle Cloud applications via REST including SaaS and PaaS applications running inside or outside Oracle Cloud.

You can create and use two kinds of connectors in Process Automation - REST connectors and Integration connectors. Use REST connectors to connect to external services and applications. Use integration connectors to connect to active integrations in Oracle Integration that are designed with REST triggers.

Once configured, connectors can be implemented in:

- **Service activities** in structured or dynamic processes
- **Integration activities** in structured processes
- **Forms** when a field needs values from a REST call

How are REST connectors configured?

Using the REST Connector editor, you define the resources and operations needed to connect to a REST service, regardless of the description language used to define the service. Configuring a connector involves defining its:

- base (or target) URL
- resources
- operations
- security

What service information do I need to know?

You'll need the following:

- URLs to the location of resources to call
- Access to resources to get a JSON sample or schema to create the types needed to send and receive data to and from the service
- List of operations to use on each resource
- List of parameters to pass to operations

How can I secure REST connector calls?

You can secure REST connector calls in one of several ways, depending on the resources you're calling and your organization's needs.

Security Type	Description	Supported Grant Type
No Security	REST calls are not secured.	Not applicable
Basic Auth	Provides basic username/password authentication credentials with each request.	Not applicable
OAuth 2.0	Uses OAuth security tokens for making outbound calls to external services. If you choose OAuth 2.0 security, you'll need to follow steps to generate and gather Oracle Identity Cloud Service-related information needed.	<ul style="list-style-type: none"> • JWT Assertion • Client Credential
OCI Signature Policy	Provides a secure way for an OCI Service such as Oracle Cloud Infrastructure Process Automation to interact with service consumers.	Not applicable

Which OAuth grant type credential should I create?

Your process needs determine the OAuth grant type to choose: JWT Assertion or Client Credential.

- Use **Client Credential** when invoking a system that requires secured communication but doesn't require a user identity to be present in it. The client credential identifies the OAuth client and not the user.
Example: You may want to invoke Oracle Identity Cloud Service or Oracle Integration APIs using client credential security, since they typically don't require a user identity.
- Use **JWT Assertion** for system-to-system communication that requires invoking on behalf of a specific user without taking consent from that user.
Example 1: Model a Process-to-customer system process (structured or dynamic) that invokes endpoints on behalf of the user who creates the process. Use a service task connector that makes a call on behalf of the initiating user, creating an OAuth token using that user and JWT Assertion, and invoke the target endpoint.
Example 2: The Process Automation forms connector also supports invoking endpoints on behalf of the user. Suppose you configure a form with a connector that fetches information from an HCM system. If the connector is configured to use a JWT assertion credential, it automatically uses the user identity of the current logged in user accessing the form to generate an OAuth token to make a call to the HCM system and get information.

How do global credentials work?

Process Automation Administrators can create global credentials (Basic, OAuth, OCI Signature Policy) in Workspace. In Designer, Process Automation Designers can find and apply a global credential to a selected connector.

Note that any changes an administrator makes to a global credential, are applied in all connectors in all process applications (activated or not) in which the global credential is used.

How are credentials stored?

Credentials are stored centrally. They are separate from a process application's metadata and aren't exported or imported with a process application. Process Automation Administrators can share credentials by creating global ones that Process Automation Designers can select for a process application.

Configure REST Connectors in Designer



Learn how to add, configure, and secure REST connectors.

- [Add a REST Connector](#)
- [Add REST Connector Resources and Operations](#)
- [Define REST Connector Input and Output Types](#)
- [Secure the REST Connector](#)

Add a REST Connector

Create a REST connector whenever you want to call an external REST service to retrieve, create, update, or delete data in the application.

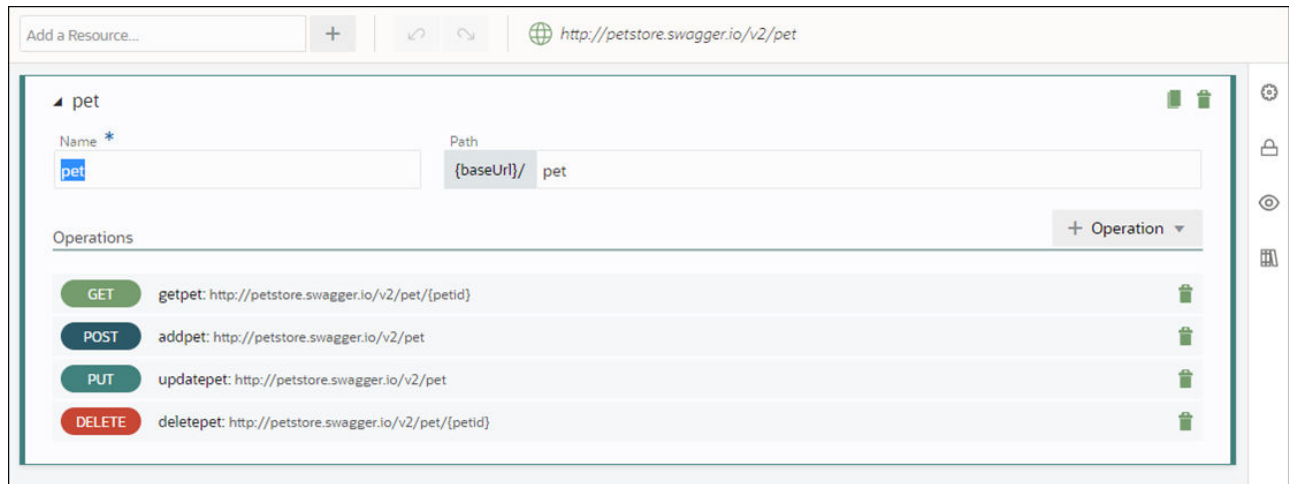
1. From a selected process application, click **Add** in the upper right corner of the page.
2. In the Add component pane, expand **Connectors** and choose **REST API**.
3. In the Add component pane, enter the following information:
 - a. **Title:** Enter a descriptive name for the connector.

- b. **Identifier Name:** Automatically populated to uniquely identify the connector. This value can't be changed later.
 - c. **Base URL:** Enter the base endpoint you want the connector to access in REST calls. By default, Process Automation appends `.com` to the connector's name and displays that as the default URL.
4. Click **Create**.
The REST connector is displayed on the Connectors page as well as on the process application's main page. You can identify it as a REST connector by the **REST** icon .
5. Open the new connector.
Click the **Open Now** link in the creation message that briefly appears or select the new connector from the connector list.
The REST connector editor opens.
6. If you want to edit the properties of the connector:
 - a. Click **Properties**  from the side menu.
 - b. On the Properties pane, you can edit the following:
 - **Name**
 - **Description**
 - **Base URL**
 - c. Close the Properties pane to save the changes.

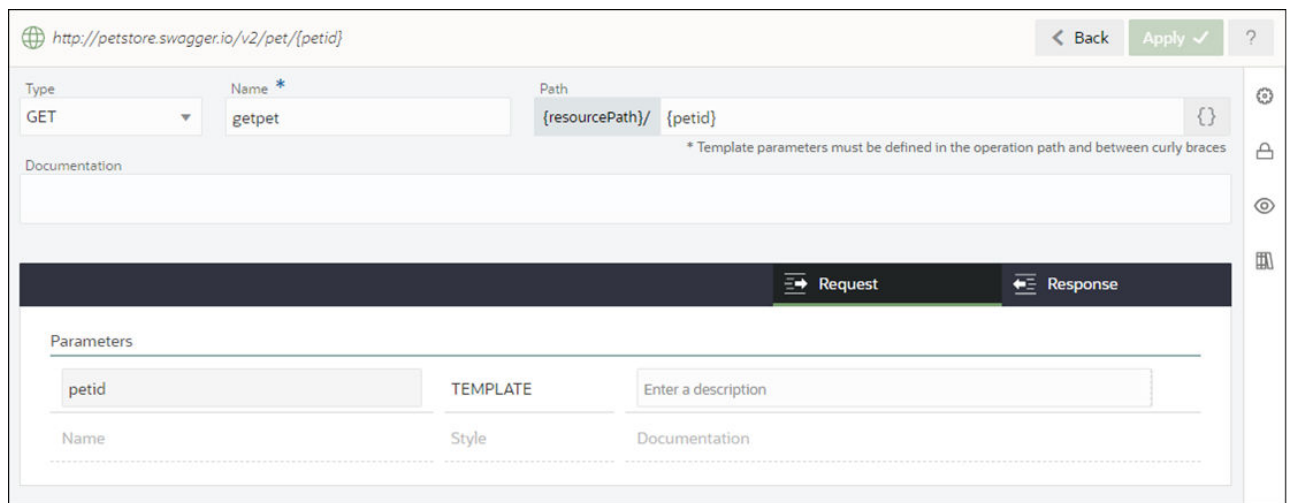
Add REST Connector Resources and Operations

Configure the REST connector's resources, operations, and their request and response parameters.

1. From a selected process application, click the **Connectors** tab and open a REST connector.
The Resources page is displayed for the connector.
2. Add resources to the connector.
A resource contains one or more operations that control data by performing basic create, read, update, and delete operations (CRUD) on resources using standard HTTP method requests.
 - a. In the **Add a Resource** field at the top of the section, enter a resource name and click **+**. You'll select this resource name when implementing a service task
The new resource is added to the Resources list and expanded.
 - b. In the **Path** field, identify the resource path within the base URL.
 - c. If needed, duplicate (clone) and delete resources, and undo and redo changes to them, using options in the Resources section. Duplicating a resource can be useful when configuring similar resources.
3. Add operations to resources.



- a. In the Operations section, click **Add**.
 - b. Select an HTTP method to add (GET, PUT, POST, PATCH, DELETE, or HEAD). The new operation is added to the Operations list, along with its name and path.
4. Specify request and response parameters for operations.



- a. In the Operations list, select an operation to expand it. If needed, make changes to the operation's **Type**, **Name**, and hierarchical **Path** fields. Optionally, enter a description in the **Documentation** field.
- b. Click the **Request** and **Response** tabs to view the operation's parameter and body fields. The HTTP method you selected determines the request and response message combination to complete for the operation. An asterisk displays if one or more required fields on the tab need to be completed.

Refer to the table below for required and optional parameters for HTTP methods. Information in this table is based on the Methods definition section of the HTTP/1.1 protocol definition.

HTTP Method	Description	Request Message-Body	Response Message-Body
GET	Retrieve information from a resource	Optional	Yes
POST	Create a resource	Yes	Optional
PUT	Completely update an existing resource	Yes	Optional
PATCH	Partially update an existing resource	Yes	Optional
DELETE	Delete a resource	Optional	Optional
HEAD	Identical to a GET except that no message body is returned in the response	Optional	No


- c. In the **Body** field, add or select a business object to store the message data. You can create business objects based on JSON files for REST operation payloads or on a schema. To create a business object by importing or pasting a JSON sample, and upload or paste JSON sample text. The JSON text or schema is validated, and if invalid, an error is displayed.

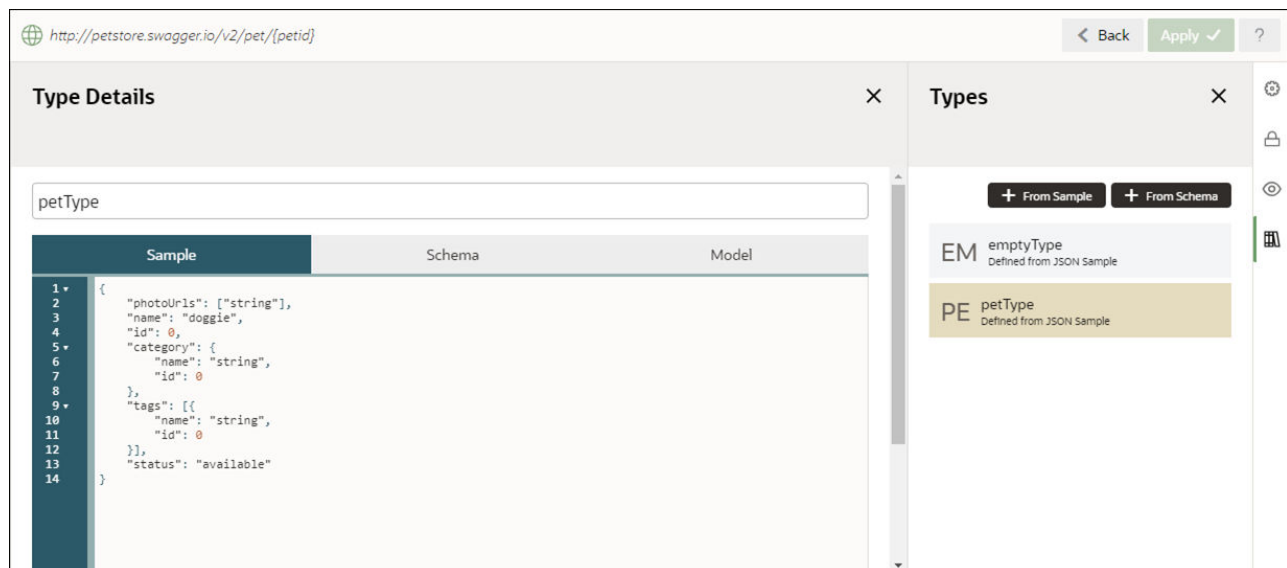
Ensure that the JSON output of the REST connector does not contain elements that start with a number or contain spaces. These elements may throw up a failed mapping error during runtime when converted to XML. Element names are case sensitive, and must start with a letter or underscore.

- d. When you select a business object in the **Body** field, you must also specify the message's media type in the **Media type** field. Process Automation supports *application/json*.
- e. In the **Parameters** table, configure one or more parameters for the operation. Click an empty row to add a line, and complete name, style, and description fields. Depending on the selected method, supported style options include template, header, body, and query. When you add, edit, or remove a parameter token in the operation's path (for example, {Id}), its corresponding template parameter is automatically created, edited, or deleted.
- f. Click **Apply** to save the operation. Apply is active only when all required information has been entered.

Define REST Connector Input and Output Types

Define the data types that the REST connector will accept as input and return as output. When the connector is used in a structured or dynamic process, these types are shown in the data association in Designer enabling you to map data from a process to this connector and to appropriate fields. You can create types from a schema or a JSON sample, then select types when configuring operations.

1. From a selected process application, click the **Connectors** tab to open the Connectors page, and open a connector.
2. Click **Types**  from the side menu.
The Types pane is displayed, showing all types defined for the connector.
3. Select **+From Sample** or **+From Schema** to add a type.



Secure the REST Connector

You can secure your REST connector by specifying a security policy and credential details.



- [Configure Basic Auth Security](#)
- [Configure OAuth Security with Client Credential](#)
- [Configure OAuth Security with JWT Assertion](#)
- [Configure Security with OCI Signature Policy](#)

Another way to secure your connector is to select a Process Automation Administrator defined global credential. See [Use Global Credentials](#).

Configure Basic Auth Security

Basic authentication security provides a simple authentication scheme built into the HTTP protocol. The client sends HTTP requests with an authorization header that contains the word Basic, followed by a space, and a base64-encoded string username:password. The application or endpoint requires that users and services provide security credentials for access.

Defining a basic security policy involves providing a username and password.

1. From a selected process application, click the **Connectors** tab and open a connector.
2. Click **Security**  from the side menu.
The Security pane is displayed for the connector.
3. Click **Edit** .
4. In the **Security Type** field, select **Basic Auth** from the drop-down list.
5. Under Basic Auth Security Info, enter the username and password that users must enter to call the REST service.
6. Click **Save**.



Configure OAuth Security with Client Credential

With the OAuth with Client Credential security policy, the client application directly obtains access on its own without the resource owner's intervention using its client ID and client secret.

The application or endpoint requires that users and services provide security credentials for access.

For an OAuth grant (client credential) flow, you have to know the client ID and client secret from the OAuth provider. For example, if you're using an Oracle IDCS-based OAuth client, determine this information using the IDCS admin console.

See the **Security, Authentication and Authorization** section in *REST API for Oracle Cloud Infrastructure Process Automation*.

1. From a selected process application, click the **Connectors** tab and open a connector.
2. Click **Security**  from the side menu.
The Security pane is displayed for the connector.
3. Click **Edit** .
4. In the **Security Type** field, select **OAuth** from the drop-down list.
5. In the OAuth Security Info fields, enter the following details:

Field	Information to Enter
Grant Type	Select Client Credential .
Client ID	Enter the OAuth client ID.
Client Secret	Enter the OAuth client secret.
Scope	Specify the scope you defined while creating the OAuth confidential application. Scope sets limits on an application's access to a user's account. For example, use the scope parameter to enable the access token to grant different levels of access to Oracle IDCS APIs.
OAuth Token URL	Either select Local Identity Domain or enter the token URL from the OAuth provider. For example: <code>https://idcs-aaaaaaaa.identity.aaaaaa.aaaa.com/oauth2/v1/token</code> . <ul style="list-style-type: none"> • Select Local Identity Domain to auto-populate the field with the current service instance's IDCS OAuth token URL. • You can also manually enter the IDCS OAuth token URL in the field. For example: <code>https://idcs-aaaaaaaa.identity.aaaaaa.aaaa.com/oauth2/v1/token</code>.

6. Save and test the security.
 - Click **Test** to validate your settings.
 - Click **Save** to save your settings and complete the connector's security.

Configure OAuth Security with JWT Assertion

Configure your connector to use OAuth 2.0 to authenticate with your API.

See the Security, Authentication and Authorization section in *REST API for Oracle Cloud Infrastructure Process Automation*.


For an OAuth assertion grant (jwt-bearer) flow, you'll need to:

- Generate a key pair
- Extract public and private keys from the key pair
- Obtain the public certificate's x5t thumbprint


- Create a trusted OAuth client in the OAuth provider

See [Quick Reference](#).

To configure OAuth with JWT Assertion:

1. From a selected process application, click the **Connectors** tab and open a connector.
2. Click **Security**  from the side menu.

The Security pane is displayed for the connector.

3. Click **Edit** .
4. In the **Security Type** field, select **OAuth** from the drop-down list.
5. In the OAuth Security Info fields, enter the following details:

Field	Description
Grant Type	Select JWT Assertion .
Client ID	Enter the OAuth client ID.
Private Key	Enter the base64-encoded private key
Scope	Specify the scope you defined while creating the OAuth confidential application. Scope sets limits on an application's access to a user's account. For example, use the scope parameter to enable the access token to grant different levels of access to Oracle IDCS APIs.
OAuth Token URL	Either select Local Identity Domain or enter the OAuth token URL from the OAuth provider. <ul style="list-style-type: none"> • Select Local Identity Domain to auto-populate the field with the current service instance's IDCS OAuth token URL. • You can also manually enter the IDCS OAuth token URL in the field. For example: <code>https://idcs-aaaaaaa.identity.aaaaa.aaa.com/oauth2/v1/token.</code>
X5t	Base64 encoded SHA-1 thumbprint of public certificate.

6. Test and save the security settings.
 - Click **Test** to validate your settings.
 - Click **Save** to save your settings and complete the connector's security.

Note that when the connector is used for a service task, it uses the process instance creator's identity; but when the connector is used for a form control, it uses the logged-in user's identity.

Configure Security with OCI Signature Policy

You can securely call Oracle Cloud Infrastructure (OCI) APIs using the OCI Signature security policy. This policy authenticates the service consumer (client), and ensures that communication between the OCI service and the service consumer happens in a secure manner.

In this policy, the client (who makes the http request) uses an API key to encrypt the required header fields and content. The encrypted data is then added to the `Authorization` header in the http request. Upon receiving such a request, the OCI service can verify that the request was made by an authentic user.



See [Request Signatures](#) in OCI documentation.

Prerequisites

Before you can configure a connector to use OCI Signature Policy, you must perform the following tasks:

1. Get the user OCID and tenancy OCID. See [Where to Get the Tenancy's OCID and User's OCID](#).
2. Create an API signing key. See [How to Generate an API Signing Key](#).
3. Get the fingerprint of the key. See [How to Get the Key's Fingerprint](#).
4. Optionally, if you've generated the key with a passphrase, you need to obtain the passphrase.

To configure OCI Signature Policy security for a connector:

1. From a selected process application, click the **Connectors** tab and open a connector.
2. Click **Security**  from the side menu.
The Security pane is displayed for the connector.
3. Click **Edit** .
4. In the **Security Type** field, select **OCI Signature Policy** from the drop-down list.
5. Under OCI Signature Security Info, enter the following details:

Field	Information to Enter
Tenancy OCID	Enter the tenancy OCID that you obtained. See Prerequisites .
User OCID	Enter the user OCID that you obtained.
Fingerprint	Enter the fingerprint that was generated while creating the key.
Private Key	Enter the private key that you obtained.
Passphrase	If you have generated the key with the passphrase, enter the passphrase obtained. Otherwise leave this field blank.

6. Click **Save**.

Use Global Credentials



You may want to use a global credential that is already created in Workspace by a Process Automation Administrator instead of creating application specific credentials in Designer. The advantage of global credential is that you can use the same credential in multiple process applications.

See [About Credentials](#).

Only Process Automation Administrators can create global credentials from the Workspace Administration page. A Process Automation Administrator can also edit or delete global credentials in Workspace. See [Manage Credentials in Workspace](#).

Note that any changes a Process Automation Administrator makes to a global credential will be applied to all connectors in which you use the global credential.

To use a global credential:

1. From a selected process application, click the **Connectors** tab and open a connector.
2. Click **Security**  from the side menu.
The Security pane is displayed for the connector.
3. Click **Edit** .
4. In the **Security Type** field, select **Global Credential** from the drop-down list.
5. Search for and select a global credential.
Only global connectors that use the same base URL display in search results.
6. Click **Save**.

Specify Connector Visibility in Process Editors

You can specify a connector's visibility settings including if it would appear in Activities palette of a structured process or under the Connector category of a dynamic process.


Note:

Connectors defined in your process application are always visible when editing form controls that support events. See [Add Dynamic Behavior to Forms](#).

In a structured or dynamic process, designers can implement a connector in one of these ways:

- By adding a **service activity** and selecting the connector in its implementation. Designers can configure a connector this way regardless of visibility settings.
- By selecting the connector directly from the Activities palette's **Integration** category of a structured process or from the Activities drop-down list's **Connectors** category of a dynamic process. Listing connectors in this way enables designers to quickly make use of connectors configured by others.

Follow these steps to configure a connector's visibility in process editors. Note that connectors are always visible for implementation in service tasks.

1. From a selected process application, click the **Connectors** tab and open a connector from the Connectors main page.
2. Click **Visibility**  from the side menu.
The Visibility pane is displayed.
3. In the **Show in Palette** field, specify whether the connector should be visible as an integration activity in the Activities palette of a structured process or a dynamic process's **Connector** category.
4. In the **What to show** field, specify the visibility level you want to show - connector, resource, or operation.
5. In the **How to show it** field, optionally specify a different label to display for the connector, and its resources and operations, under the **Integrations** category of the Activities palette in a structured process or under the **Connectors** category of the Activities drop-down list in a dynamic process.

Use Connectors in Process Applications

After a connector is configured, you can use it in your process applications by implementing it in processes or forms.

You can implement a connector in:

- **Service activities** in structured or dynamic processes
- **Integration activities** in structured processes
- **Forms**, when a field needs values from a REST call

Implement a Connector in a Structured Process

1. Position a connector on the canvas in one of two ways.
 - Drag a service activity to the canvas.
 - Select an available connector or its resource and operation from the **Integrations** category in Activities palette. Drag and drop the integration activity to the canvas.
2. Open the connector's properties and complete implementation fields. Select **Service**, **Resource**, and **Operation** entries.
3. Implement data association to and from the connector.

Implement a Connector in a Dynamic Process

1. Position a connector on the canvas in one of two ways.
 - Drag a service activity to the canvas.
 - Select an available connector, or its resource and operation from the **Connectors** category in the Activities drop-down menu.
2. Open the connector's properties and complete implementation fields. Select **Service**, **Resource**, and **Operation** entries.
3. Implement data association to and from the connector.

Implement a Connector in a Form


1. In the full form editor, select a control that supports connectors. You can use connectors in multiple ways in forms.
 - As a control data source.
 - As a local event connector.
 - As a global connector.
2. Configure the control's connector properties. Select **Service**, **Resource**, and **Operation** entries.
3. Implement data association to and from the connector.

Use Data Association to Configure Connector Input and Output

Whether you implement a connector in a process or a form, use data association to configure its input and output. For example, if the REST connector uses a GET operation to retrieve email content from another service, and a PUT operation to update the email with the response, use data association to configure the business object to store the values as they're changed by the service calls. See [Configure Data Association](#).

Delete a Connector

Before deleting a connector, ensure it's not being used in a process or form. You can't delete a REST service connector that is being called by a service task.

1. From a selected process application, click the **Connectors** tab.
Connectors configured for your process application are displayed.
2. Hover over the connector to delete, click **Options** , and choose **Delete**.
3. In the Delete connector pane, click **Delete**.

The connector is removed and no longer appears on the Connectors page or the process application's main page.

Work with Integrations

In Oracle Cloud Infrastructure Process Automation, you can use active integrations from Oracle Integration that are designed with REST triggers in your process applications. In this way you can leverage the rich array of adapters that Oracle Integration offers to communicate and interact with other services in the cloud.

A simple integration includes at the least the source or the trigger connection and the target or the invoke connection, and data mapping between these two connections. The source connection is the entry point to the integration from Process Automation. It is represented by a source system or endpoint which triggers the integration to execute.

For more information, see [Understand How Integrations Work](#) in *Using Integrations in Oracle Integration 3*.

Note that you can work with integrations in Process Automation only when one of the following prerequisites are fulfilled.

- You have a Process Automation instance that is enabled with an Oracle Integration 3 service instance. See *Enable Process Automation with Oracle Integration 3* in *Administering Oracle Cloud Infrastructure Process Automation*.
- You have registered Oracle Integration from the Registered Services page of Workspace. See [Register Services](#).



Note:

Ensure that the OAuth policy is configured on the REST triggers of the Oracle Integration.

If a Process Automation instance is *enabled* with an Oracle Integration 3 instance, you can navigate between the design-time interfaces of the two instances.

- To navigate to Oracle Integration 3 from Process Automation, open the navigation menu in Process Automation Designer and click the punch out URL for Oracle Integration. The Oracle Integration 3 home page opens in another browser tab.
- To navigate to Process Automation from Oracle Integration 3, open the navigation menu in the Oracle Integration home page, and click the punch out URL for Process Automation. The Process Automation Designer opens in another browser tab.

Topics:

- [Use Integrations](#)
- [View Integration Details](#)
- [About Integration Statuses](#)
- [Discover and Use Integrations Within Oracle Integration 3 Projects](#)

Use Integrations

To use an integration in your process application, you have to first create a connection to the integration with a connector, and then implement the integration in structured processes and forms using the connector.

To use integrations in process applications:

1. [Create a Connector with an Integration](#)
2. [Implement an Integration in Processes and Forms](#)

Create a Connector with an Integration

To connect to an integration in Oracle Integration, you must create a connector using the integration.

1. From a selected process application, click **Add** in the upper right corner of the page.
2. In the Add Component pane, expand **Connectors** and choose **Integration**.


Available active integrations designed with REST triggers from Oracle Integration are listed.

3. Choose an integration from the list or search for the integration and select it.
4. Click **Create**.

The connector configured with the integration is displayed in the Connectors page as well as the process application's main page. You can identify it as an integration connector by the

Integration icon .

You can also do the following:

- Click the integration connector to view details about the integration. See [View Integration Details](#).
- Click the **Properties** icon  from the side menu to open the Properties pane and quickly view the integration connector's properties - name, description, and identifier. Note that you cannot edit the properties.
- Specify if the connector, resources, and operations linked with the integration are visible from the Activities palette while designing structured processes. See [Specify Connector Visibility in Process Editors](#).

Implement an Integration in Processes and Forms

After creating a connector with the integration, you can use it in processes and forms to implement and use the integration in your process application.



Note:

You can only use integrations in structured processes.

1. Select and open a structured process in your process application.
2. In the Activities palette, expand **Integrations** and select the integration connector to use.
You can also select resources and operations if you've specified that they appear in the Activities palette from the **Visibility** settings of the integration connector.
3. Drag and drop the integration connector onto the process flow wherever you want to add it.
4. Configure the input and output flows of the integration with data association. See [Configure Data Association](#).

You can also implement the integration in a structured process by using the **Integration** activity under **Systems** and configuring its properties (service, resource, and operation) from the Properties pane.

You can implement the integration connector in forms like any other connector. See **Implement a Connector in a Form** in [Use Connectors in Process Applications](#).

View Integration Details

Click an integration on the Connectors page or on the process application's main page to view its name, version, identifier and so on from the integration details window that opens.

You can view the following details of an integration from the integration details window

- The name of the integration.
- The unique identifier for the integration.
- The status of the integration. See [About Integration Statuses](#).
- The description of the integration.
- The base URL for the integration.
- The **Versions** section: Under this section all available versions of the integration are listed. If there are other versions available to be chosen then a **Use** button is shown next to the integration versions that can be used.

About Integration Statuses

Some integrations may have badges to indicate their status.

Know more about integration status badges and what they mean in the following table:

Integration Status	Description
New	A new integration that is ready to be used in forms and processes.

Integration Status	Description
New Version Available	<p>There is a new version available for the current integration.</p> <p>Under the Versions section of the integration details window, the current and new versions of the integration are listed. If you want to use the new version of the integration, click the Use button. You'll be prompted to confirm. On confirmation, the integration gets updated to the new version.</p> <p>You'll get a warning if the new version that you are trying to use is incompatible and hence can cause validation errors. You can choose to cancel the update or update anyway and fix the errors later.</p>
Update Model	<p>The integration (current version) has been updated and activated in Oracle Integration.</p> <p>Click and open the integration details window. Under the Versions section, the currently used integration is listed with a Update button. Click Update and then confirm when prompted to update to the latest version.</p>
Fetch Failed	<p>Fetching the integration from Oracle Integration failed, as the Oracle Integration instance that was registered or associated with the Process Automation instance is no longer available.</p>
Missing Integration	<p>The integration has been deleted from the Oracle Integration instance.</p> <p>As a workaround, you can delete the integration and use an available active integration.</p>

Discover and Use Integrations Within Oracle Integration 3 Projects

For structured processes, you can use the Process Automation integration wizard to quickly connect to and invoke active integrations within Oracle Integration 3 projects without requiring to configure various settings like authentication and defining endpoints. Process Automation uses service registration credentials to discover and invoke the integrations.

To connect and invoke an integration in Oracle Integration 3 projects:

1. In Designer, open a process application and click **Add** in the upper corner. The **Add component** pane appears.
2. Expand **Connectors**, and click **Integrations**. The **Add component** pane appears.
3. Select **Project**.
4. From the **Select project** drop-down, select the required project. A list of active integrations in the project appear.
5. Select the required integration and click **Create**.

Embed a Start Form in an External Application

Sometimes users of an external service may want to invoke a process instance from within their application without navigating to Workspace. They can do so by embedding the start form in their application. This not only saves the effort of navigating to Process Automation from their application, but ensures seamless user experience as they're always in the context of their primary application.

For example, in case of an Employee Onboarding application in Oracle HCM Cloud, a new employee must fill a form with details and submit it. This is the [start form](#) that would initiate the onboarding process. By embedding the start form's URL in the Oracle HCM Cloud application, the start form can be accessed and opened from Oracle HCM Cloud itself without navigating to [Workspace user interface](#) in Process Automation. The URL of the start form can open in a new browser window or can be embedded in iFrame.

Process automation lets you load a Process Automation start form with the context of an existing application by passing data through the URL query parameters. Data in the URL query parameters can then be used in the On Load events to make API calls to external systems to get more information. This saves the user the effort of re-entering details required to finish tasks. Additionally, you can also pass the data in base64 encoded format.

You can only pass first level data in the URL query parameters. For example, `form.department=Cloud`.

For example, Oracle HCM Journeys lets you create an external URL based tasks and pass the context details of the journey, task and user performing the activity to the URL. This data can be populated in the Process Automation start form. Subsequently, the Process Automation start form can trigger an On Load event to query a HCM API to retrieve more information about the user, and then pre-populate fields such as name, department, and cost center.

Here is an example of how you can send data using URL query parameters:

```
http://<IP address and port>/process/workspace/pages.html?  
root=startForm&reloadOnSubmit=false&processDefinitionKey=ProcessApplicationSam  
ple-1.0-EmployeeOnboardingProcess&form.profile={"firstName":"first name of  
user","userName":"name of the user"}&form.team=TeamName&form.project=Name of  
Project&form.manager=Name of Manager
```

Here is an example of how you can send data in Base64 encoded format using URL query parameters:

```
http://<IP address and port>/process/workspace/pages.html?  
root=startForm&reloadOnSubmit=false&processDefinitionKey=ProcessApplicationSam  
ple-1.0-  
EmployeeOnboardingProcess&form.profile=SI6IkFzaHdpbmkRGhha2Vkiwi&form.team=V  
kaGFrYXIgUmF&form.project=U2FuIEZyYW5jaXNjbjbywgQ2&form.manager=EZyYW5jaXNjbjby
```

 **Note:**

In case of a [Process Automation instance paired with a Fusion based Oracle Cloud Application](#), user authentication is seamless. However for all other applications and services users would be prompted to authenticate before using the start form.

Base URL:

```
https://{OPA_service_instance}/process/workspace/pages.html?  
root=startForm&processDefinitionKey=#&reloadOnSubmit=#
```

URL Query Parameters

Parameter	Value Type	Description	Default Value	Required (in Query Parameter)
processDefinitionKey	string	The Process Definition Key parameter uniquely identifies a process belonging to an application. This key can be found in Process Interface definition which is available upon calling the Process Interfaces REST endpoint. <code>https://{OPA_service_instance}/process/api/v1/process-interfaces</code> See List Process Interfaces in <i>REST API for Oracle Cloud Infrastructure Process Automation</i> .	"	Yes
reloadOnSubmit	string	Reloads the form after submit action is performed.	false	Optional
form.	string	Populates the start form with the value in this variable.	Not applicable	optional

Embed a Task List in an External Application

Sometimes users of an external service may want to invoke a Process Automation task list from within their application without navigating to Process Automation Workspace. They can do so by embedding the task list in their application.

This not only saves the effort of navigating to Process Automation from their application, but ensures seamless user experience as they're always in the context of their primary application. These users can then view details of the tasks and take action on them.



Note:

In case of a [Process Automation instance paired with a Fusion based Oracle Cloud Application](#), user authentication is seamless. However for all other applications and services, users are prompted to authenticate before accessing the task list.

Base URL:

`https://{OPA_service_instance}/process/workspace/pages.html?root=taskList`

Process automation lets you load a Process Automation task list with the context of an existing application by passing data through the URL query parameters. The URL query parameters can:

- Control the actions that are available to users such as reassign and escalate the task.
- Control the details that are shown in the task details. For example, disable the comments.
- Filter the types of tasks that are shown to the user.

Here is an example of how you can send data using URL query parameters:

```
http://<IP address and port>/process/workspace/pages.html?
root=taskList&hideTabBar=true&selectedContext=teamTasks
```

URL Query Parameters of a Task List

The following table lists all the URL query parameters of a task list:

URL Query Parameter	Description	Value Type	Default value and sample formats
hideTabBar	Hides the tabs. Defaults to My Tasks tab when set to true.	Boolean	false
selectedContext	Specifies the type of assignment of the task. The valid ENUM are: <ul style="list-style-type: none"> myTasks teamTasks startedByMe managedByMe 	String	myTasks
hideToolBar	Hides the search bar and filters in the header of the task list page form.	Boolean	false
displayOptions	Configures the display options for a task list. The value must be a JSON object. The available parameters are: <ul style="list-style-type: none"> hideSystemActions Shows or hides the system actions. hideCustomActions Shows or hides the custom actions. enableToolBarEvents Enables the toolbar events. disablePagination Shows or hides the custom actions. The value type for all these parameters is a string, and the default value is false.	JSON object	{ } Sample format: Display Options for Task List (Value should be Proper JSON Object)
pageSize	Configures the number of tasks that Process Automation fetches for listing.	Number	10

URL Query Parameter	Description	Value Type	Default value and sample formats
taskDetailOptions	<p>Configures the task details options when the task list loads. The value must be a JSON object.</p> <ul style="list-style-type: none"> hideActions hideSave hideSystemActions hideSummary hideForm hideComments hideHistory hideAttachment hideMoreInfo <p>The value type for all these parameters is a Boolean, and the default value is <i>false</i>.</p>	JSON object	<p>{}</p> <p>Sample format:</p> <pre>{ "hideSystemActions": true, "hideComments": true }</pre>
systemActions	<p>Configures the system actions when the task list loads.</p>	String	<p>SUSPEND, ESCALATE, RENEW, REASSIGN, REQUEST_FOR_INFO, SUBMIT_INFO, WITHDRAW, ACQUIRE, PURGE, DELETE, RESUME, RELEASE, CLAIM</p>

Embed a Task Detail in an External Application

Sometimes users of an external service may want to invoke a Process Automation task details from within their application without navigating to Process Automation Workspace. They can do so by embedding the task details in their application.

This ensures seamless user experience as they're always in the context of their primary application and saves the effort of navigating to Process Automation from their application. These users can then view details of the task and take action on it.



Note:

In case of a [Process Automation instance paired with a Fusion based Oracle Cloud Application](#), user authentication is seamless. However for all other applications and services, users are prompted to authenticate before accessing the task details.

Base URL:

```
https://{OPA_service_instance}/process/workspace/pages.html?
root=taskDetail&taskNumber=####
```


Process automation lets you load a Process Automation task details with the context of an existing application by passing data through the URL query parameters. The URL query parameters can:

- Control the details of the task that are displayed.
- Show or hide the action buttons
- Show or hide the system action buttons

Here is an example of how you can send data using URL query parameters:

```
http://<IP address and port>/process/workspace/pages.html?
root=taskDetail&taskNumber=####&hideComments=true
```

In this example, the task comments do not appear to the user.

URL Query Parameters of Task Details

The following table lists all the URL query parameters of task details:

URL Query Parameter	Description	Value Type	Default value and sample formats
taskNumber	Displays the taskId of the task.	String	
hideActions	Hides all action buttons.	Boolean	false
hideSave	Hides the Save button.	Boolean	false
hideSystemActions	Hides the system action buttons.	Boolean	false
hideSummary	Hides the summary section.	Boolean	false
hideForm	Hides the task form.	Boolean	false
hideComments	Hides the comments section.	Boolean	false
hideHistory	Hides the history section.	Boolean	false
hideAttachment	Hides the attachments section.	Boolean	false
hideMoreInfo	Hides the More Information section.	Boolean	false

10

Explore Workspace

Workspace is the runtime environment of Oracle Cloud Infrastructure Process Automation. It can be accessed by all authenticated users.

You can access Workspace directly from Designer (the design-time environment). Click the user icon on the top-right of the Designer UI, then click **Workspace**. The Workspace UI opens in another tab of your browser.

The way you and others use Workspace depends entirely on how the process application and tasks are set up. But explore the following topics to learn about things you typically do.

Topics:

- [Start an Application](#)
- [Work on Tasks](#)
- [Track Processes](#)
- [Use Analytics](#)
- [Set to Out of Office](#)

If you're a Process Automation Administrator, learn about additional options in [Workspace Administration](#).

Start an Application

In Workspace, open the navigation menu, and then select **Start Requests**. From the list of available options on the Start Requests page, click the application you want to start.

Complete the start form, and click **Submit**. You get a confirmation that an instance has been created. This initiates the first task in the application.

Note the following:

- If the list of application is extensive, you can also use the **Search** field to search for the application you want to start.
- If you want to view default versions of applications, click **Default Version**.
- If you want to view all available versions of applications, click **All**.

Note:

If the application you want to start doesn't appear on the Start Requests page, you may not have the correct role. Contact the Process Automation Administrator who can assign the required role to you.

Work on Tasks

User tasks are listed on the Tasks page in Workspace. To access the Tasks page, open the Workspace navigation menu and then choose **Tasks**.

The task list displays the following information about tasks.

- **Title:** The title of the task.
- **Process Name:** The process to which the task belongs.
- **Reference Id:** The business identifier (if any) associated with the task.
- **Created Date:** The date and time on which the task got created.
- **Due Date:** The due date (if any) of the task.
- **Status:** The present state of the task in its lifecycle.
- **Actions:** Click the **Actions** icon to display a list of actions that you can perform on the task. See [Perform Task Actions](#).

You can filter the task list and view specific tasks in the task list. See [Find a Task](#).

To view details about a task and complete it, select and open the task. See [View and Complete Tasks](#).

Find a Task

There are various options available in the Tasks page using which you can find specific tasks.

Find a task on the Tasks page:

- Use the **Search** field to search for particular tasks. See [Search for a Task](#).
- Use filter options available to display specific tasks.
 - [Use the Task Tab Filters](#)
 - [Use the Task Lifecycle and Due Date Filters](#)
 - [Apply Advanced Search Filters](#)

Search for a Task

Use the **Search** field on the Tasks page to search for specific tasks.

You can do a simple search based on keywords such as task title, description, task Id, and process instance Id.

Click **Try business search** on top right of the Tasks page to enable advanced search with business data.

- [About Business Search](#)
- [Search Use Cases](#)

 **Note:**

Searches are not case sensitive. For example a search for `Task` can return results with matches such as `task`, `TASK`, `Task`, or `Task`.

About Business Search

Business search gives end users a rich search experience by providing them the ability to do complex searches that can include one or more fields, standard query operators, wildcards, fuzzy term search, and more.

Different types of query options are supported in business search. Some of the most used ones are listed below:

Simple query

A simple query uses a simple syntax build with terms, quote-enclosed phrases, and operators. It uses Lucene's [simple query parser](#). This type of query is more flexible and will attempt to interpret a search request even if it's not perfectly composed.

You can use the following operators in a simple query:

Operator	Description
+	Specifies AND operation.
""	Wraps multiple terms to specify them as exact phrases.
	Specifies OR operation.
-	Specifies NOT operation.
*	Specifies prefix query if used at the end of a term.

For example: `"expense report" +limit -reject`

The search result for the above query returns the exact phrase `expense report` AND the word `limit`, and it does NOT contain the word `reject`.

Advanced query

An advanced query uses complex syntax that can be build with operators, wildcards, as well as one or more fields. It uses Lucene's [full query string syntax](#). While this type of search is versatile, the query is strict and returns an error if the query string includes any invalid syntax.

The following table lists some examples of advanced query.

Query	Description
<code>status:open</code>	The field <code>status</code> contains the term <code>open</code> .
<code>title:expense AND total:(3000 TO 4000)</code>	The field <code>title</code> contains the term <code>expense</code> and the field's total value should be between 3000 and 4000.
<code>hotel_name:(hotel NOT motel) AND category:'Resort and Spa'</code>	The field <code>hotel_name</code> contains the term <code>hotel</code> and doesn't contain the term <code>motel</code> ; and the field <code>category</code> matches <code>Resort and Spa</code> .
<code>book.*:(quick OR brown)</code>	Any of the fields such as <code>book.title</code> , <code>book.content</code> or <code>book.date</code> contains <code>quick</code> or <code>brown</code> .

Auto-suggestion

Auto-suggestion enables real-time suggestion of fields that are configured with business data. As you type, a list of [business search enabled fields](#) will be displayed that you can use to query. For example, if you want to search by using a patient's name, then on entering the patient name, a list of possible fields that you may want to use is suggested. Selecting one of the fields will then prompt you for the value that you want to search in this field.

Boosting a term

Term boosting refers to ranking a match higher if it contains the boosted term, relative to others that don't contain the term. To boost a term, use the caret ^ symbol with a boost factor (a number) at the end of the term you're searching. Note that the default boost factor is 1.

For example: `make: nissan^10 OR city: "new york"`

In the above query `make: nissan` receives a boost factor of 10 as compared to `city: "new york"` with boost factor of 1. Hence in the search result, matches that contain `make` as `nissan` will get higher relevance and ranking.

Fuzzy search

Fuzzy search matches on terms that are similar, including misspelled words. To do a fuzzy search, use the tilde ~ symbol at the end of a single word. An optional parameter of value between 0 and 2 can specify the maximum number of edits allowed. If no value is specified, then the default value is 2.

For example to search for a term similar in spelling to "book" use the fuzzy search: `book~1`. The search will return terms like books and cook.

Key things to note in search results

- **Relevance:** Search results are sorted in accordance to their relevance. For example, in a free text search, if the same term matches in one task's title and in another task's summary, then the task where the term match is in the title is given higher preference and is ranked higher in the task results.
- **Synopsis:** A fragment text containing the matching search criteria is provided in the search results. Thus providing a summary to users as to why certain tasks are ranked higher in the search results. To view the synopsis, click the expand arrow next to a match.
- **Highlights:** Matching criteria are highlighted in the search results to draw users attention.

Search Use Cases

Explore some common search use cases.

- [Free text search](#)
- [Free text search with phrases](#)
- [Advanced search](#)

Free text search

You can do a free text search when you have some data but don't know the exact field names against which the search query should be based. So you query one or more terms, where any of the terms is considered a match.

For example, a hospital's front desk can search for a patient's first name `john` in the admission records. But there may be a number of patients with the first name John, so the search is refined to include the patient's last name `doe`. The search may still return multiple results as there may be more than one patient with the name John Doe. So the search can be refined further to include the last four digit of the patient's social security number (SSN).

The search query can be: `+(john doe) +(*1234)` which means that matches containing the terms `john`, `doe` and a number `1234` should come up in the search result.

Free text search with phrases

In the above use case of free text search, the search results may be anywhere. For example, (john doe) can be in a comment: *I got the confirmation from John that someone named Doe will drop the package.*

To eliminate this kind of anomalies and reduce noise in the search results you can do a free text search with phrases where the exact phrase is enclosed in double quotes. This ensures that the search matches the whole phrase and in that order in the search result.

For our hospital use case, the search query can be: `("john doe" +(*1234))` which means that only matches that contain the whole phrase `john doe` and in that order comes up in the search result.

Advanced search

In the above use case where you do a free text search with phrases, you may still end up with undesirable search results. For example, `john doe` may occur in some comments or in fields other than the targeted field. Or, there's a record where an orderid or telephone number ends with `1234`.

To eliminate such cases, you can refine the search to specific field names. In the search field, you can enter a query of `field: value`, where `field` is the `fieldName` and `value` is the value that `fieldName` should contain. Each `field: value` combination is called a clause. You can have as many clauses as you want with nested hierarchy in the search query.

For example: `+(field1: (value1 OR value1`)) + (field2: value2 OR field3: value3) - (field4: value4)`

where:

MUST field1 should have value1 or value1`

MUST

MUST field2 should have value2 OR field3 should have value3

NOT field4 to have value4 (This field with value shouldn't be present at all)

For our hospital use case, the search query can be: `name: "john doe" AND ssn: *1234` which should return the exact matching result where the name of the patient is John Doe whose SSN ends with the number 1234.

You can expand the above queries using number and date ranges.

Use the Task Tab Filters

You can use the task tab filters present on top of the Tasks page to filter the task list according to My tasks, Team tasks, Started by me, and Managed by me.

Filter	Description
My tasks	Displays tasks claimed by you or tasks that others have assigned to you.

Filter	Description
Team tasks	Displays tasks that are assigned to a number of users, including the group that the user belongs to. To work on a task that is displayed on the Team tasks tab, you need to first claim the task. On the task row, click Actions , and then click Claim . In the Claim task dialog, click Claim again. The task now gets displayed under the My tasks tab and you can work on it.
Started by me	Displays tasks from process instances you started.
Managed by me	Displays tasks from process instances you have started, and other tasks that you interacted in the past, for example, a task that you may have reassigned to someone.

Use the Task Lifecycle and Due Date Filters

You can use the task lifecycle and due date filter chips available under the **Search** field to further filter tasks under each task tab filter (My tasks, Team tasks, Started by me, and Managed by me). For example under **My tasks**, select **Assigned** to display tasks assigned to you.

Note that the task lifecycle and due date filter chips change in the context of the task filter tab that you choose. For example, if you select the **My tasks** tab, then the following task lifecycle and due date filter chips are displayed: **Assigned**, **High priority**, **Info requested**, **Overdue**, and **Due soon**. But if you select the **Team tasks** tab, then the following task lifecycle and due date filter chips are displayed: **All**, **High priority**, **Overdue**, and **Due soon**.

The highlighted filter chip shows the active filter applied on the task list.

Filter	Description
Assigned	Displays all tasks assigned to you. The assigned state indicates a task has been assigned to a user or role and it is ready to be acted upon.
High priority	Displays all tasks marked as high priority.
Info requested	Displays all tasks for which further information have been requested.
Overdue	Displays tasks for which due dates are over. A red Overdue tag is displayed under Due Date for such a task, along with the number of days that are overdue.
Due soon	Displays all tasks for which due date is soon.

Apply Advanced Search Filters

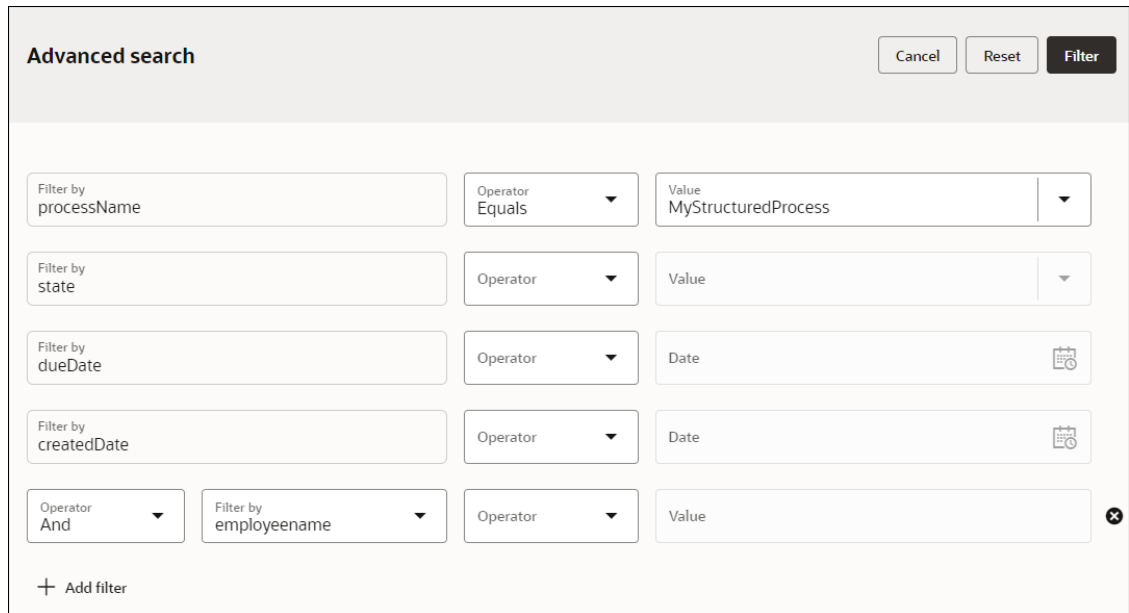
Choose **More filters** to open the Advanced search dialog and apply advanced filters to display specific tasks in the task list.

If you're on the default search page, specify the following in the Advanced search dialog:

1. In the **Process** field, select the process associated with the task from the options available in the drop-down list.
2. In the **Status** field, select the task's state in its lifecycle from the drop-down list.
3. In the **Due Date** field, specify and narrow down the search by specifying one of the options:
 - On
 - Before

- After
 - Between
4. Select a particular date and time in the **Select Date** field.
 5. You can also specify a particular date and time range in the **Created Since** and **Created Until** fields.
 6. Click **Filter**.

If you're on the *business search* page where search based on business data is enabled, then you can use various operators for each of the **Filter by** fields to do a more accurate search for those fields. Additionally, you can also filter the search using the [configured searchable fields](#) that you defined in design-time. You can add as many filters as you want using configured searchable fields by clicking the **Add filter** option.



The tasks that meet the criteria you defined in the Advanced search dialog are displayed in the task list.

The applied filters are also displayed as filter chips below the **Search** field.

Perform Task Actions

You can perform required actions on tasks that are assigned to you.

Click **Actions** on the task's row, and choose an action from the menu that displays to act on the task. When you take action on a task, it moves to the next step in the process instance.

Action	Description
SUBMIT	Submit a task. Click SUBMIT from the Actions menu, and in the resulting dialog, click SUBMIT . Optionally, review the task form and enter information in any required fields of the form by clicking the View Details link. You can also add your comments in the Comments field before you perform the submit action.

Action	Description
APPROVE	<p>Approve a task.</p> <p>Click APPROVE from the Actions menu, and in the resulting dialog, click APPROVE. Optionally, review the task form and enter information in any required fields of the form by clicking the View Details link. You can also add your comments in the Comments field before you perform the submit action.</p>
REJECT	<p>Reject a task.</p> <p>Click REJECT from the Actions menu, and in the resulting dialog, click REJECT. Optionally, review the task form and enter information in any required fields of the form by clicking the View Details link. You can also add your comments in the Comments field before you perform the submit action.</p>
Claim	<p>If a task is assigned to multiple users or a group, the task will appear as <i>Unclaimed</i> to all the users or group in the Team tasks page. Any of the users can claim the task to act upon it. After a user claims a task, other assignees won't be able to access the task as the task will disappear from the Team tasks page.</p> <p>The task will be available under the My tasks page of the user who claimed the task, and the status of the task changes to <i>Assigned</i>.</p>
Withdraw	<p>If you don't want a task, you can withdraw it.</p> <p>Once withdrawn, the task disappears from the task list. The next activity of the process instance becomes available to be acted upon.</p> <p>From the My tasks, Started by me, and Managed by me pages, you can do a bulk withdrawal of tasks.</p> <ul style="list-style-type: none">• Select the tasks that you want to withdraw.• Click Withdraw.
Escalate	<p>You can escalate a task to other users and/or roles.</p> <p>Click Escalate from the Actions menu. In the resulting Escalate Tasks dialog, select one or more user and/or role in the drop-down list available from the Select members field. Add a comment in the Comment field to let users know the reason for escalation. Click Escalate.</p> <p>The task will be escalated to the specified users and/or roles. If the task is escalated to a single user then it will appear as an <i>Assigned</i> task in the task list of the user, and the task's status will be displayed as <i>Escalated</i>. If the task is escalated to more than one user and/or role then it will be in <i>Unassigned</i> state, with the sub-state <i>Escalated</i>.</p> <p>Note that escalation can also be automatically triggered for a task if it hasn't been acted upon even after a certain time interval has elapsed. To do so, you have to define the escalation policy for the human task in design-time. See Configure Task Escalation or Expiration.</p>
Request Info	<p>You can request for more information on a task assigned to you.</p> <p>Click Request Info from the Actions menu, and in the resulting Request more information dialog, select a user from the User drop-down list from whom you want more information. Add a comment in the Comment field. Click Request Info.</p> <p>The task will be available to the user from whom you requested information. The status of the task changes to <i>Info Requested</i>.</p>
Submit Info	<p>You can provide the information requested for a task.</p> <p>You can act upon a task with the status <i>Info Requested</i> by selecting Submit Info from the Actions menu of the task. Provide the required information in the resulting dialog, and click Submit Info.</p> <p>The task will be available to the user who requested for the information. The status of the task changes to <i>Info Submitted</i>.</p>

Action	Description
Reassign	<p>You can reassign a task assigned to you to a different user or role. Select the task, and then click Reassign from the Actions menu.</p> <p>From the My tasks, Started by me, and Managed by me pages, you can do a bulk reassign of tasks.</p> <ul style="list-style-type: none"> • Select the tasks that you want to reassign. • Click Reassign . <p>In the resulting Reassign Tasks dialog, select a user or a role in the drop-down list available from the Select members field. Optionally, add a comment in the Comment field. Click Reassign. Note that the task/tasks disappear from your task list.</p> <p>The task/tasks will appear as <i>Assigned</i> task/tasks in the task list of the user to whom you had reassigned.</p>
Release	<p>Suppose you realize that you don't need to act upon an assigned task, you can release the task and make it available in the Team task page so that other users can act upon it as required.</p> <p>The task will be available in the Team task page. Its status changes to <i>Unclaimed</i>.</p>

Additionally, you can also select and open the task and perform actions from the task's details window. See [View and Complete Tasks](#).

View and Complete Tasks

Select and open a task to view detailed information about the task and also perform certain actions on it to complete the task.

In the task details window you can do the following:

- Depending on the type of task, you can see actions that you can perform on the task such as **Submit** (if it is a submit task) or **Approve** and **Reject** (if it is an approve task).
- Set the task's priority. The options available are **Low**, **Medium**, and **High**.
- Click **Save** to save the changes you made to the task.
- Click the **More Actions** button to choose an action that you can perform on the task.

Additionally, on the right side of the task details page, you can find a vertical bar containing the following icons to access other task details:

- Forms
- Comments
- Documents
- History
- More information



You can click each of these icons to view details and complete the task if the task is assigned to you.

Section	Description
Forms	View details and fill required information on the form attached to the task. If required, you can use the Expand Form/Collapse Form button. Click SUBMIT from the Actions menu, and in the resulting dialog, click SUBMIT . Optionally, review the task form and enter information in any required fields of the form by clicking the View Details link. You can also add your comments in the Comments field before you perform the submit action.
Comments	Enter a comment for the task. Click Post to post the comment so that others can view the comment. Note that by default the All Comments radio button is selected, and you can view all process level comments. If you want to view comments only for that particular task, select the Task Comments radio button.
Documents	Attach required documents to the task.
History	View the task's history. The task history also displays the comments provided by the users on custom actions and system actions so that you can understand the reason why the action was taken.
More information	View general information about the task such as its state, Task Id, Instance Id, the user to whom the task is assigned, the due date (if any) for the task, the date on which the task was created, if any business identifier is associated with the task, and so on.

Track Processes

Workspace provides the Tracking page where you can track structured and dynamic processes that are part of activated applications.

Click the Workspace navigation menu, and from the options click **Tracking** to open the Tracking page. All structured and dynamic processes that belong to applications activated by you are displayed in the process list. You can identify structured and dynamic processes by the icons displayed besides their names.

-  Structured Process
-  Dynamic Process

You can do a number of things on the Tracking page:

- Search and find a process. See [Find a Process](#).
- Apply filters to display specific processes on the process list. See [Filter Processes](#).
- Sort the process list. [Sort Processes](#).
- View the state of a process. [About Process States](#).
- Work on structured and dynamic processes. See [Work on Processes](#).

Find a Process

There are various options available in the Tracking page using which you can find specific process/processes.

Find a process on the Tracking page:

- Use the **Search** field to search for a specific process. You can do a simple search or switch to an advanced search mode by clicking **Try business search** on top right of the Tracking

page. Business search works in the same way in both Tasks and Tracking pages. The only difference is that in the task page it is used for searching specific tasks, while in the Tracking page it is used for searching specific processes. See [About Business Search](#) and [Search Use Cases](#).

- Use the filter options available to display specific processes. See [Filter Processes](#).

You can also sort the process list on the Tracking page by using the various sorting options. See [Sort Processes](#).

Filter Processes

You can filter processes in the Tracking page by using the different process status filters and also by applying advance filters.

- [Use the Process Status Filters](#)
- [Apply Advanced Filters](#)

Use the Process Status Filters

You can use the process status filter chips available below the **Search** field to filter and display processes according to their states.

To know more about the different states of structured and dynamic processes, see [About Process States](#).

Filter	Description
In progress	Displays all structured and dynamic processes that are in progress.
Completed	Displays all structured and dynamic processes that have been completed.
Terminated	Displays all structured and dynamic processes that are terminated.
Suspended	Displays all structured and dynamic processes that are suspended.
Faulted	Displays all structured and dynamic processes that fails to complete due to internal error of the system.

Apply Advanced Filters

Choose **More filters** to open the Advanced search dialog and apply advanced filters to display specific processes in the process list.

If you're on the default search page, specify the following in the Advanced search dialog:

1. In the **Search applications** field, specify the application to which the process belongs.
2. In the **Application version** drop-down field, select the version of the application.
3. In the **Select Process** drop-down field, select the process.
4. Click the field to select a user name from the suggested options.
5. In the **State** drop-down field, select a state.
6. In the **Period** drop-down field, select the time period:
 - **Past week**
 - **Past month**
 - **Past year**
 - **Custom**


7. Click Filter.

The process/processes that meet the criteria you defined in the Advanced search dialog is displayed in the search results. The applied filters are also displayed as filter chips below the **Search** field.

If you're on the *business search* page where search based on business data is enabled, then you can use various operators for each of the **Filter by** fields to do a more accurate search for those fields. Additionally, you can also filter the search using the [configured searchable fields](#) that you defined in design-time. You can add as many filters as you want using configured searchable fields by clicking the **Add filter** option.

Sort Processes

You can sort the process list on the Tracking page by using the different sorting options available.

Click the **Sort order**  icon to sort the task list in ascending or descending order. You can click the field next to the **Sort order** icon and select any of the sort by options available from the drop-down list.

Sort By	Description
Created On	Sort the processes based on their date of creation
Application Name	Sort the processes based on the name of their application.
Created By	Sort the processes based on their creator.
Process Title	Sort the processes based on their title.

About Process States

A process can go through different states in its lifecycle.

The table below lists and describes the different states of a process.

State	Description
In progress	A process that is still ongoing. A user can terminate or suspend a process that is in the <i>In progress</i> state.
Completed	A process that has been completed. All tasks associated with the process has been completed successfully, and no more further action is required.

State	Description
Terminated	The <i>Terminated</i> state indicates that the process was stopped abruptly without all tasks associated with the process being completed successfully. A process can be terminated manually by a user action or REST API. A dynamic process can also be terminated through conditions defined for stage or activity or manually. See Define Conditions for Stage or Activity and End Dynamic Processes and Activities .
Suspended	A process can be suspended manually by a user action or REST API. A user can either terminate or resume a process that is in the <i>Suspended</i> state. When resumed the suspended process transitions into the <i>Active</i> state. If a process was suspended from a faulted state, when resumed it transitions back to the <i>Faulted</i> state.
Faulted	A process that fails to complete due to internal error of the system.

Work on Processes

You can work on a process from the process instance details window.

In the Tracking page, click a process to open its process instance details window. You can do the following from the process instance details window:

- [View Process Details](#)
- [Work with Activities](#)
- [View Process Audit](#)
- [Post and View Comments](#)
- [Upload and View Documents](#)
- [View Process Data](#)
- [View Process Information](#)
- [End Dynamic Processes and Activities](#)

View Process Details

You can view details about your process in the process instance details window.

To view process details:

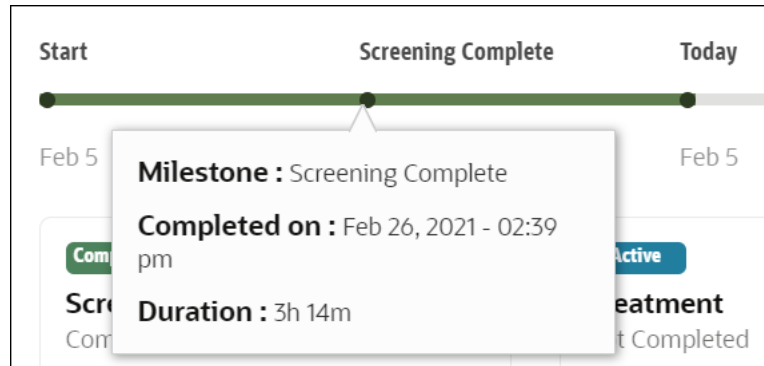
1. From the process list in the Tracking page, click the process whose details you want to view.
2. In the resulting process instance details window, the title bar displays the process name and the application name to which the process belongs. You can click the **Close** button to exit and return back to the Tracking page.
3. You can view the process state and the date of creation.
4. If required, you can change the state of the process. Click **Actions** on the top right to display the process state transition options. The options available in the list depends on the present state of the process and the process itself. For example, an *In Progress* structured process can have the options - *Terminate* and *Suspend*, whereas an *In Progress* dynamic process can have the options - *Terminate*, *Suspend*, and *Complete*.

See [About Process States](#).

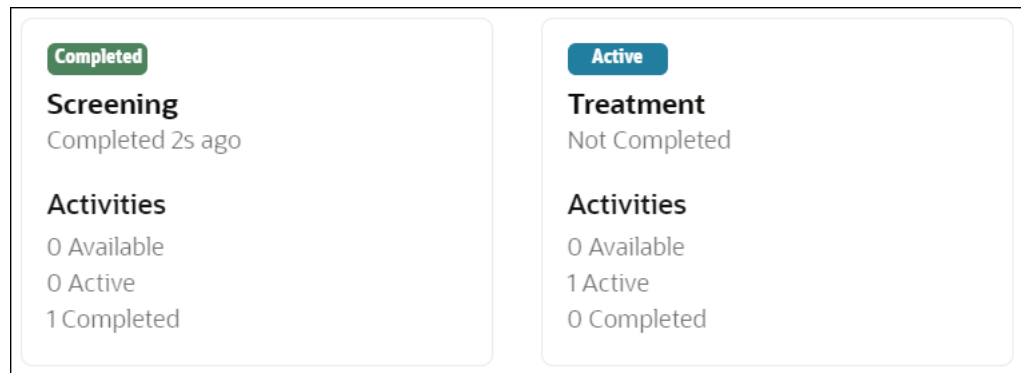
5. If there are milestones and stages defined for a dynamic process, expand **Milestones and Stages** to view details about them.

The progress bar at the top shows the status of the dynamic process instance.

- Click a milestone in the status bar to view details about the milestone.



- View details about stages. You can view the number of available, active, and completed stages.



Work with Activities

You can work on task activities associated with the process from the process instance details window.

To work on an activity:

1. For a structured process, expand **Actionable Activities**. The activities that you can take actions on are listed under it.
 - a. Click **Open** on the activity to display and open the form associated with the activity.
 - b. Edit and fill required information in the form.
 - c. If you want to save the form and submit it later, click **Save**.
 - d. To submit the form, click **Submit**. If it is an approval form, click **Approve** or **Reject**.

Note:

When you submit the form, any validation issues are displayed at the top. You can expand the banner to see the list of errors. Click an error to navigate to the corresponding control where the error occurs.

- e. Click **Cancel** to exit the form and return back to the process instance details window.
2. For a dynamic process, expand **Activities**. All open, running, and completed activities are listed under it.
 - a. Click the **Actions** icon (three dots) on a *completed* activity to open and view the form associated with the activity.
 - b. Click the **Actions** icon (three dots) on a *running* activity to open the form associated with the activity and perform certain actions. For example, you can terminate or suspend a service task activity. Click the **Actions** button on the form and select an option from the drop-down list.
 - c. Click the **Actions** icon (three dots) on an *available* human task activity to open the form associated with the activity. You can perform actions such as the following on the form.
 - i. View the form.
 - ii. Edit and fill required information in the form.
 - iii. If you want to save the form and submit it later, click **Save**.
 - iv. Submit or approve/reject the form

 **Note:**

When you submit the form, any validation issues are displayed at the top. You can expand the banner to see the list of errors. Click an error to navigate to the corresponding control where the error occurs.

- v. Click **Cancel** to exit the form and return back to the process instance details window.
- d. You can search for an activity by clicking the **Search** icon and entering the name of the activity in the search field.
- e. Use the **Filter by Stage** drop-down list to filter activities by a stage.
- f. Click **Filter/Sort** to choose which activities to display (Activated, Running, or Completed) and to sort the activities by stage or status.

Note that non-actionable activities such as service tasks and milestones are displayed under **Audit**. See [View Process Audit](#).

View Process Audit

You can view the audit details of the process instance from the process instance details window.


To view the audit details:

1. Expand **Audit**.

A list of actionable activities such as human tasks and non-actionable activities such as stages, milestones, and service tasks are displayed.

The human tasks listed also show the associated comments provided by the users who performed actions on the tasks.

2. To search for an activity, click the **Search** icon and enter the activity name in the search field . You can also sort the list of activities in ascending or descending order by activity name, completed date, or duration.

3. Click **Open**  next to an activity to open the activity details window.
4. Expand **History** to view the audit details of the activity.



Post and View Comments

You can post and view comments for the process instance under **Comments** in the process instance details window.

To post a comment:

1. Expand **Comments**.
2. Enter the comment in the field provided to write the comment. Note that it has a limitation of maximum 2000 characters.
3. Click **Post**.

The comment added is displayed below the comment field.

Upload and View Documents

You can upload documents and view uploaded documents related to your process in the process instance details window.

To upload a document:

1. Expand **Documents**.
2. In the **Drag and Drop** field, click **Add**. In the resulting pane, select a file from your local system.

Optionally, select a file in your local system, and then drag and drop it into the field.

The file uploads and is displayed below the **Drag and Drop** field. To delete the file, click **Delete** next to the file.

View Process Data

You can view data for a dynamic process under **Data** in the process instance details window.

Expand **Data**. All data objects configured for the process are listed under it. Click the data object to view its payload data.

For example, the image below shows the payload data for formArg.

formArg

```
{
  "name": "gdfgdsfgdfg",
  "dob": "2021-10-29",
  "gender": "option1",
  "email": "sdgfsdg@oracle.com",
  "reason": "hhh",
  "loanAmount": 23,
  "income": 23,
  "score": null,
  "eligible": true,
  "status": ""
}
```

View Process Information

You can view detail information about a process under **More Information** in the process instance details window.

Click **More Information** to see details such as the status of the process instance, the instance Id, the application it belongs to, and the date and time it was created.

▼ **More Information**

<p>Status</p> <p>In Progress</p> <p>Process</p> <p>Loan Submission</p>	<p>Instance Id</p> <p>4db9ead6-267d-11ec-a66a-8ee177b0346c </p> <p>Created Date</p> <p>Oct 6, 2021 1:43 PM</p>
--	--

End Dynamic Processes and Activities

Occasionally, you may need to end a dynamic process and all its associated activities and linked processes.

Note that only users with **Manage** permission can terminate a dynamic process and its activities. See [Users, Groups, External Applications and Permissions](#).

To end a dynamic process in runtime

1. From the process list in the Tracking page, click the dynamic process that you want to end.
2. In the resulting dynamic process instance details window, click **Actions** to display the available dynamic process state transition options.
3. Click **Terminate**.

Note that terminating the dynamic process will:

- Terminate the dynamic process instance
- Terminate the process execution
- Terminate the human task activities

4. In the Terminate dialog that displays:
 - In the **Comment** field, enter a reason for ending the dynamic process.
 - If you want to end child processes (structured process activities) linked to the dynamic process, ensure to select the **Terminate associated child processes** check box.

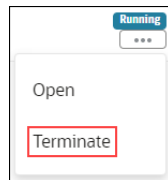


You can also end child processes linked to a dynamic process with REST APIs by setting the `terminateChildInstances` property flag to true. See [Perform an Action on Process Instance](#) in *REST API for Oracle Cloud Infrastructure Process Automation*.

5. Click **Terminate**.

To end a dynamic process activity in runtime

1. From the process list in the Tracking page, click the dynamic process whose activity you want to end.
2. In the resulting dynamic process instance details window, expand **Activities**.
3. Click **Actions** for the activity that you want to end, and then choose **Terminate**.



A dialog prompting you to confirm the termination appears.

4. In the Confirm dialog, click **OK**.
If you want to end child processes linked to the activity, ensure to select the **Terminate associated child processes** check box, before confirming the termination.

Alter the Flow of a Process

You can alter the flow of a running or suspended process instance by moving its execution from the current activity to another activity in the flow. Additionally if an activity in a process instance is failing due to errors in the value of the data objects and instance attributes, then you can modify them and retry running the current activity again.

Note the following:

- You can alter the flow of only *structured* processes.
- You must have **Manage** permission for a process application to alter the flow of a process instance.
- You cannot alter the flow of a terminated or completed process.

To alter the flow of a process instance:

1. From the Workspace navigation menu, click **Tracking** to open the Tracking page.
2. In the Tracking page, click the process instance for which you want to alter the flow.
3. In the process details window, click the **Actions** drop-down field and choose **Alter Flow**.

The screenshot shows the 'StructuredProcess1' details window. At the top, there are 'Save' and 'Close' buttons. Below the title, there is a section for 'Actionable Activities' which is currently empty. An 'Audit' section shows a timeline of events:

- Start event-StructuredProcess1**: Start activity, View input and output. Occurred at Aug 1, 2023 10:09 AM.
- User Task 1-StructuredProcess1**: Submitted a few seconds ago by idcsopausers@test.com. Occurred at Aug 1, 2023 10:09 AM.
- User Task 2-StructuredProcess1**: Assigned to idcsopausers@test.com, updated a few seconds ago. Occurred at Aug 1, 2023 10:10 AM.

 An 'Actions' dropdown menu is open on the right side, with 'Alter Flow' highlighted in red. Other options in the menu are 'Terminate' and 'Suspend'. At the bottom, there is a 'Comments' section with a right-pointing arrow.

The process instance is suspended and the Alter Flow window of the process instance opens.

4. In the **Reason** field, enter a suitable description for altering the flow of the process.
5. Move the process execution from a current activity to another activity in the process flow.

For example, you have a process instance with User Task 1, User Task 2, and User Task 3. The normal flow of the process will be that once User Task 1 completes, the process moves to User Task 2 and then to User Task 3. But for some specific reason you want to skip User Task 2, and want the process to move directly to User Task 3 from User Task 1. You can do so by using the Alter Flow action to move the process execution from the current activity (User Task 2) to the new activity (User Task 3) in the process flow.

- a. Note that by default the current activity of the process instance is displayed in the **Current activities** drop-down field.
- b. In the **New activities** drop-down field, choose the activity to which you want to move the process execution from the current activity. It is possible to move either forward or backward in the process flow depending on your requirement. So you can choose an activity that is either before or after the current activity in the process flow.

StructuredProcess1 Suspended

Alter Flow

Move this process instance to another activity in the flow. Use Change data to modify the associated data. Use Resume to restart the process instance with the changes.

Current activities: User Task 2

New activities: User Task 3

Reason: Skip user task 2

Buttons: Cancel, Change data, Resume

6. Modify the data attributes of an activity.

Sometimes a process instance may go into faulted state due to some wrong data attribute being passed into a data object for an activity. In this case, using the Alter Flow action in runtime you can rectify the data attribute error and restart the activity by resuming the process.

- a. Click **Change data**.
- b. In the resulting window, you can see the data objects associated with the process instance.
- c. Click the data object which has an error and update its attributes to rectify the error.

StructuredProcess1 Suspended

Alter Flow

Modify data associated with this process instance. Use Resume to restart the process instance with the changes.

Left panel: (x) taskOutc..., (x) myFor..., #creator

Right panel (JSON):

```

1 {
2   "name": "John Doe",
3   "address": "California",
4   "employeeID": ,
5   "email": "example@oracle.com"
6 }

```

Buttons: Cancel, Back, Resume

7. Click **Resume** to restart the process instance with the changes.

You get a confirmation that the Alter Flow action was successful.

You can also alter the flow of a process instance by using REST APIs. See [Alter Flow REST API Endpoints](#).

Use Analytics

In Process Automation, using **Analytics** you can get statistical data in the form of graphs, charts, and maps to monitor business processes. Starting from a summary view of analytics report for all your applications, you can drill down to get granular data for process instances and activities. This in turn helps you in understanding and getting insights into the performance of applications, processes, activities and also in identifying potential bottlenecks.

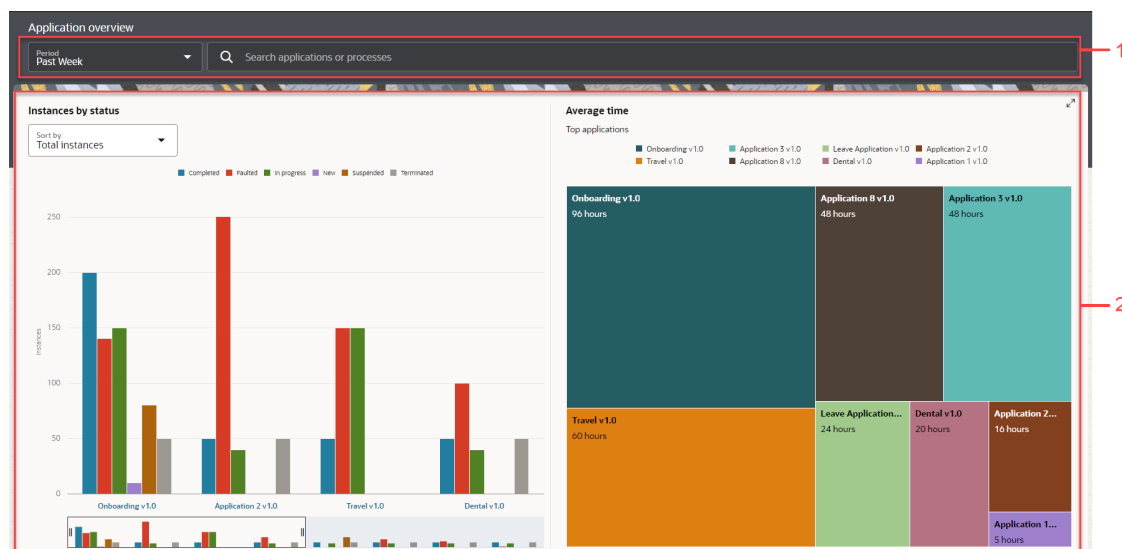
To view analytics data for a process application, you must have **Manage** permission for the process application. Typically, Process Owners and Process Automation Designers can view

analytics data for the process applications that they work on. So they can view analytics data for some but not all the process applications in a Process Automation instance.

If you want to view analytics data for all the process applications in a Process Automation instance, you've to be assigned the Process Application Administrator role. To know more about the Process Application Administrator role, see [About Process Application Roles](#).

In Workspace, open the navigation menu and click **Analytics** to open the Analytics main page.

The Analytics page is divided into two areas. On top of the page are the search and filter fields, and in the main area of the page is the visualization of data derived from process instances in the form of colored graphs, charts, and maps.



1. [Analytics: Search and Filter fields](#)
2. [Analytics: Visualization with Graphs, Charts, and Maps](#)

Analytics: Search and Filter fields

In the **Period** drop-down field, you can choose to filter process instances by duration. The available options are **Past Week**, **Past Month**, **Past Year** and **Custom**. Choose **Custom** if you want to view analytics for a particular duration by setting the start and end date.

Clicking the **Search** field will display all the applications and processes that are present. If you want to view analytics data for a particular application or process, select the application or the process from the drop-down list.

Analytics: Visualization with Graphs, Charts, and Maps

This area gives you the visualization of analytics data in process instances. Based on this data you can get insights into your applications and processes. For example, analyze the performance of your processes, determine areas that need improvement by identifying bottlenecks, analyze outliers within your processes, and so on.

Explore the following topics for using Analytics in Process Automation:

- [Analytics: Summary View](#)
- [View Analytics for an Application](#)
- [View Analytics for a Process](#)
- [View Analytics for Activities](#)

- [About Instance Status](#)
- [Analytics: Instance Tracking](#)

Tip: You can bookmark the URL of an Analytics page view, and use the bookmark to quickly open that view. Suppose you have bookmarked the Activity list page for a particular process while working on it. The next time you want to open the same view, you can use the bookmarked URL to quickly open the activity list instead of manually navigating to that level from the Analytics main page.

Analytics: Summary View

In the Analytics main page, you get a summary view of analytics data for all process applications present in the Process Automation instance.

The summary of application level analytics data in the form of bar graphs and charts is displayed in two sections:

- [Instances by Status](#)
- [Average Time](#)



Note:

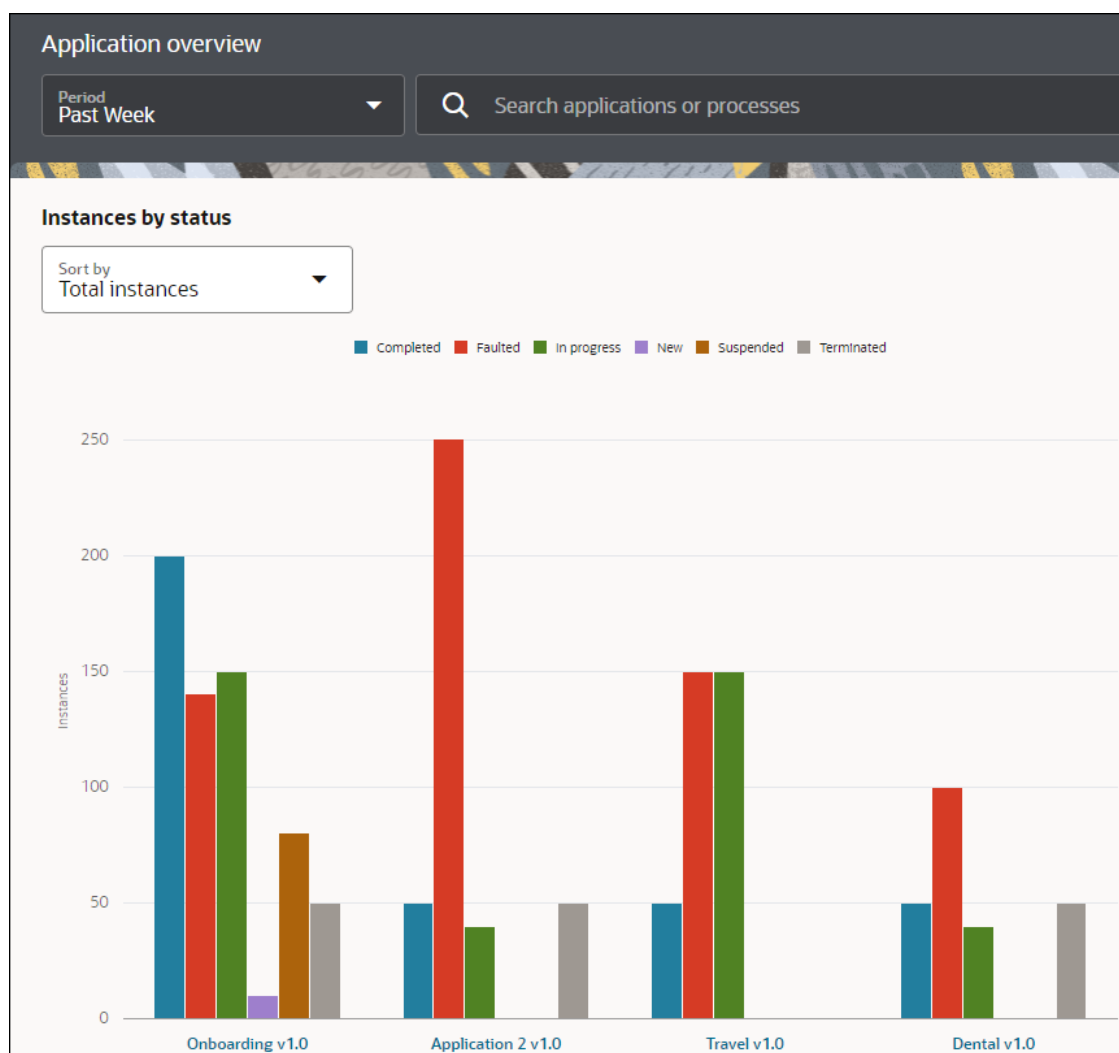
The terms *status* and *states* have been used interchangeably to describe instance status in the visualization area.

Instances by Status

The **Instances by Status** section is on the left of the visualization area and shows a bar graph representation of a summary of process applications. Each application lists total process instances by status. The X-axis lists the applications. The Y-axis represents the number of instances.

The colored bars represent the total number of instances by status: Completed, Faulted, In progress, New, Suspended, and Terminated. For more information on instance status, see [About Instance Status](#).

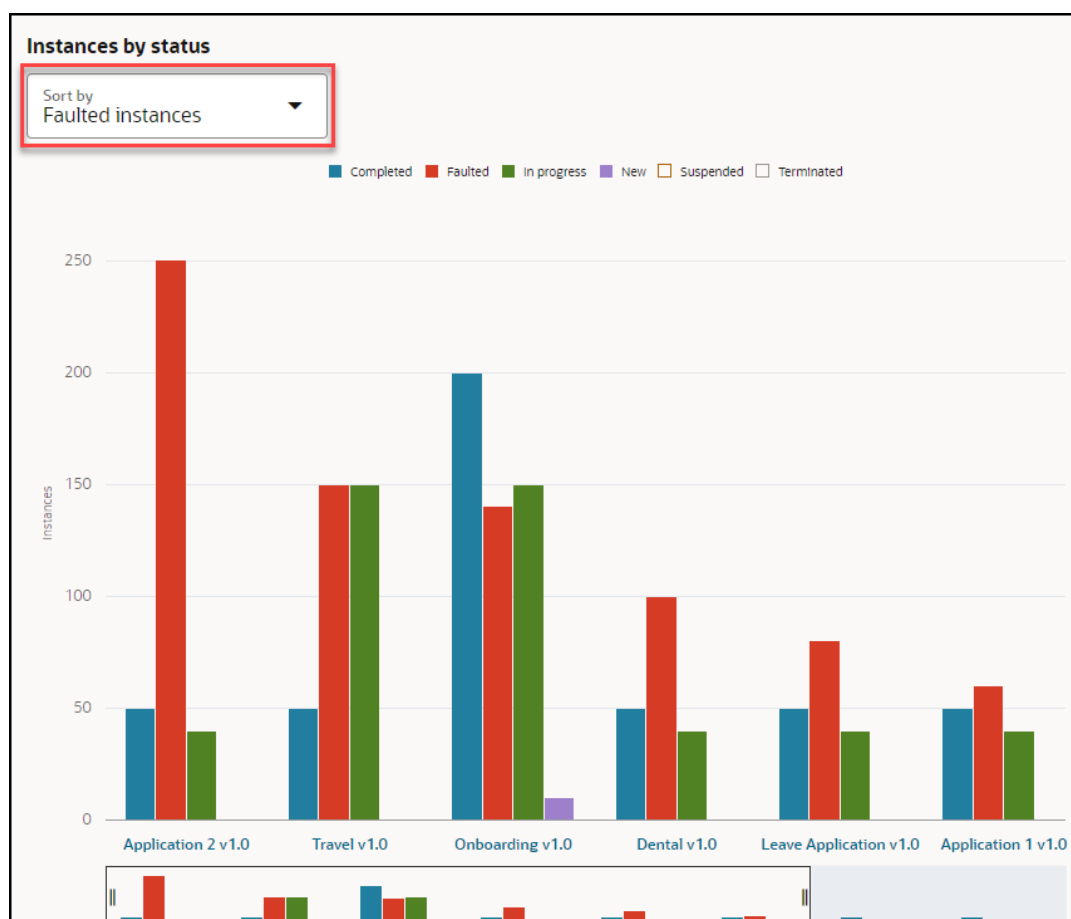
In the following example, the Onboarding application has 200 Completed, 140 Faulted, 150 In progress, 10 New, 80 Suspended, and 50 Terminated instances in the past week.



Note that if an application has multiple versions, all the different versions will be represented as separate applications in the X-axis of the bar graph.

Various options are available to further refine the view of the bar graph and get more details:

- Sort by:** The **Sort by** drop-down field allows you to sort application data represented in the bar graph in descending order based on the state that you select. For example, if you select **Faulted instances**, the bar graph changes with data represented in the descending order of the state (Faulted) selected. The application with the maximum number of instances in Faulted state will be represented first in the bar graph, the application with the second highest instances in Faulted state will be represented second, and so on.

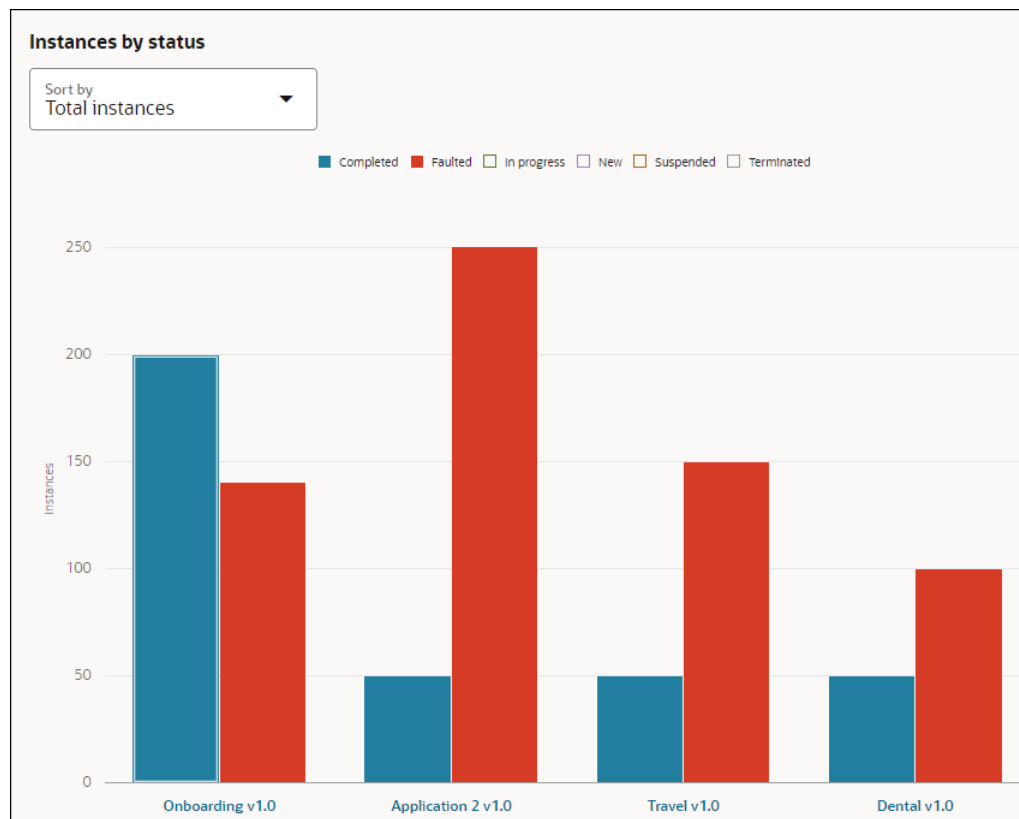


- **Interactive legends and bars:** When you hover over a legend on top or hover over a state bar in the bar graph, the other states dim, and the selected state gets highlighted in the bar graph.

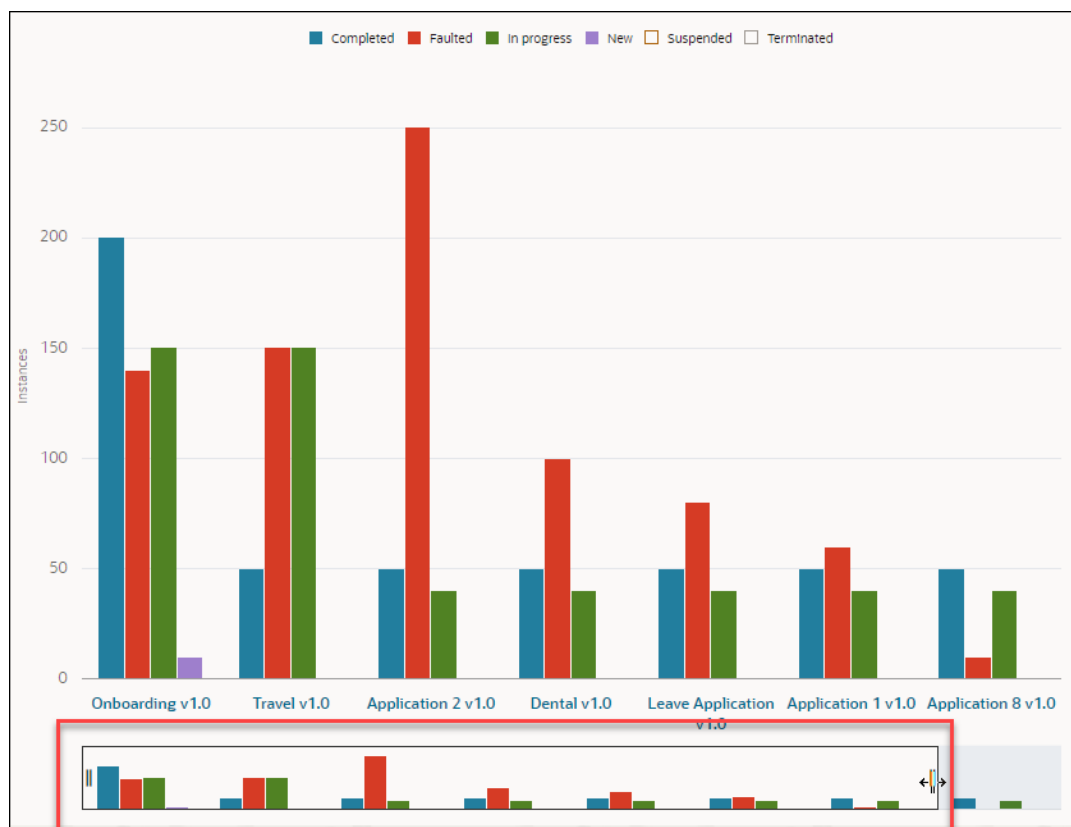
In the example below, hovering over the Faulted state bar in the Onboarding application, dims the other state bars in the applications. Also a tool tip on the selected bar is shown with the application name, the state, and the total count of instances in the Faulted state in the Onboarding application.



- Filter by State:** Suppose you want to view process instances that are in Completed and Faulted state only, click the other state legends on top to hide the states represented by them. The bar graph will only show the total number of Completed and Faulted process instances for the applications.

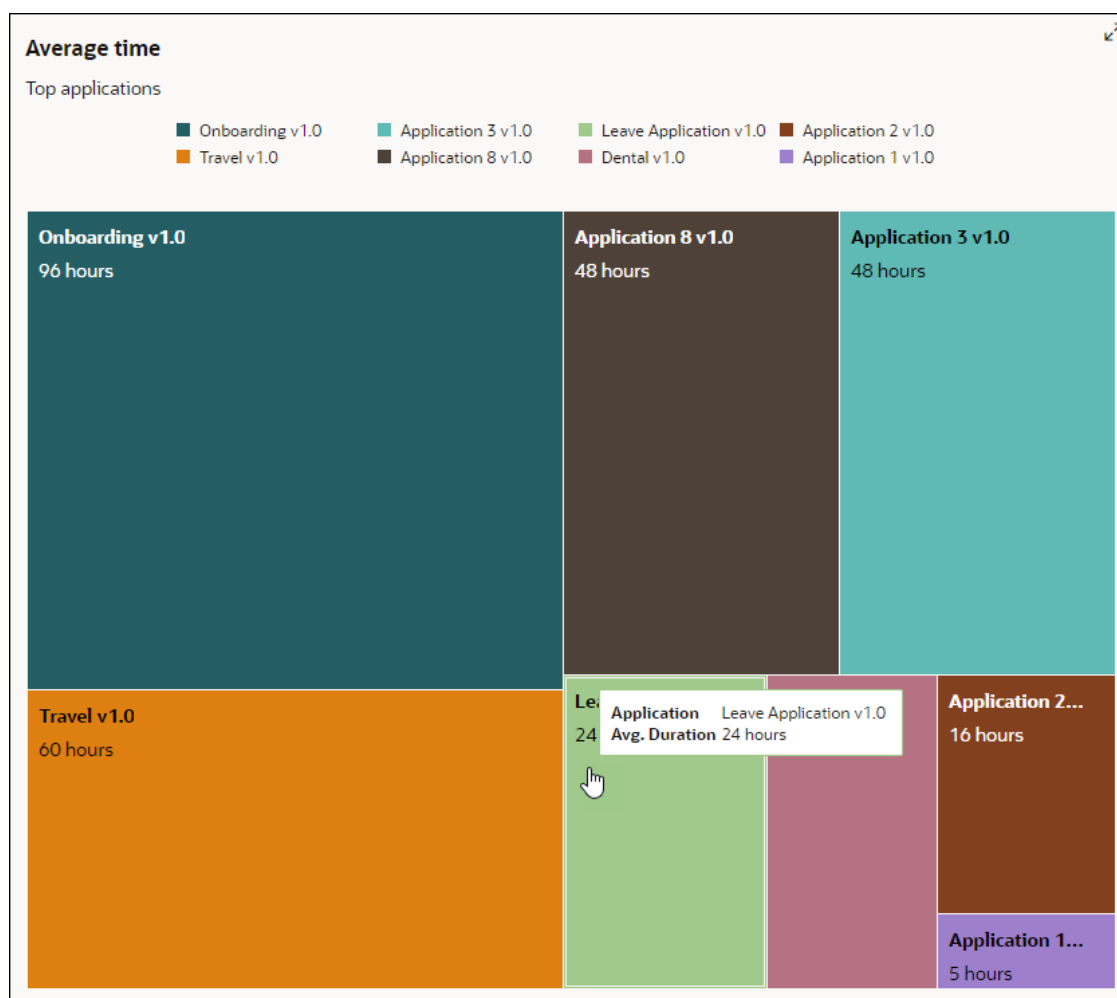


- Horizontal scroller:** By default, four applications are displayed on the X-axis of the bar graph. If you want to view more applications, use the horizontal scroller to scroll to your right and see all the applications present. Click the end of the scroller and expand it to see more applications in the bar graph.



Average Time

The **Average Time** section is present on the right of the visualization area and shows a Treemap view of the average time that the applications have taken to complete. Based on the average time taken by each application, the relative rectangular box for the application is created. The application that takes the maximum time appears in the biggest rectangle and so on. Hover over a rectangular box to get a tooltip with the application name and average duration that instances took to complete.



You can use the maximize icon on the top right to get a clearer view of the analytics visualization area.

View Analytics for an Application

You can view analytics report for a particular application to check how processes within that application are performing.

From the [Summary view](#) in the Analytics main page, to access and view analytics report for an application:

- Click the application name from the X-axis of the bar graph.
- Click an application node in the treemap view.
- Search and select the application from the search field.

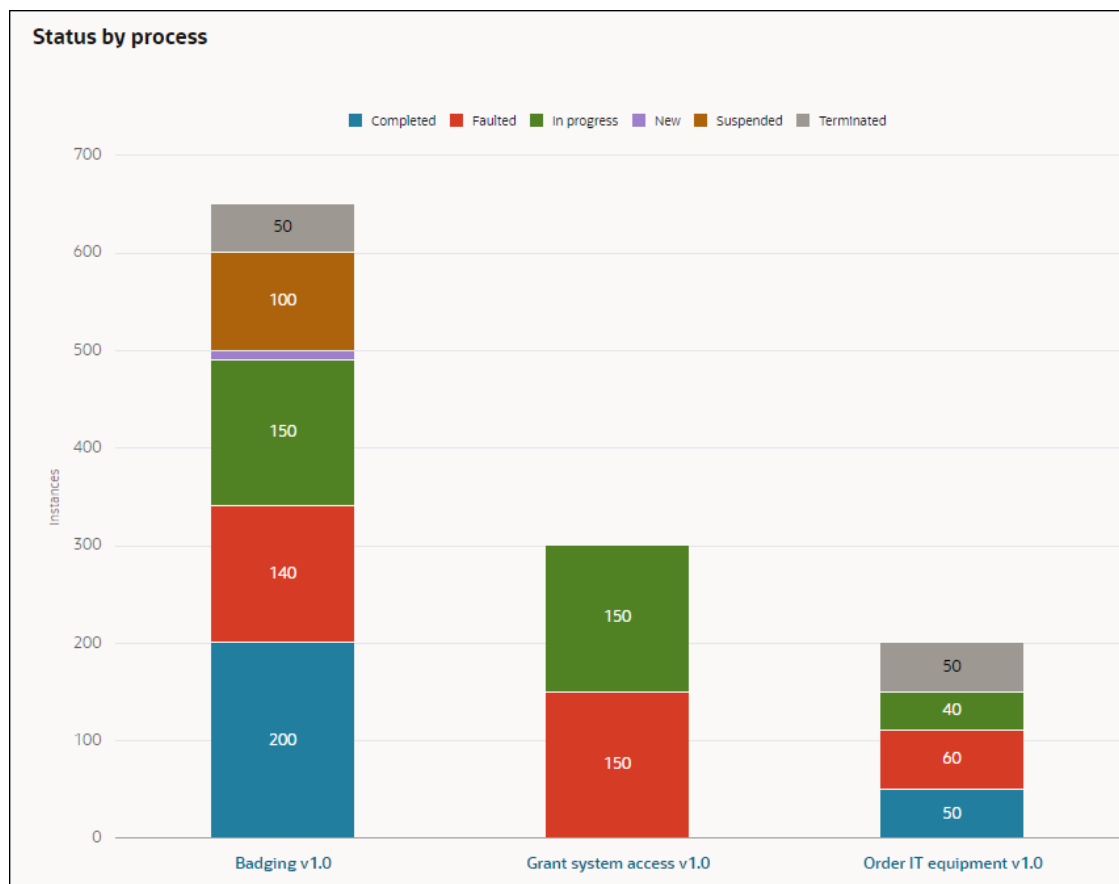
Tip: A quick way to switch to a different application is to use the filter chip displayed for the application below the search field.

Analytics report of all the processes in the application are displayed in the form of three views:

1. [Status by process](#)
2. [Total instances by process](#)
3. [Average time to complete](#)

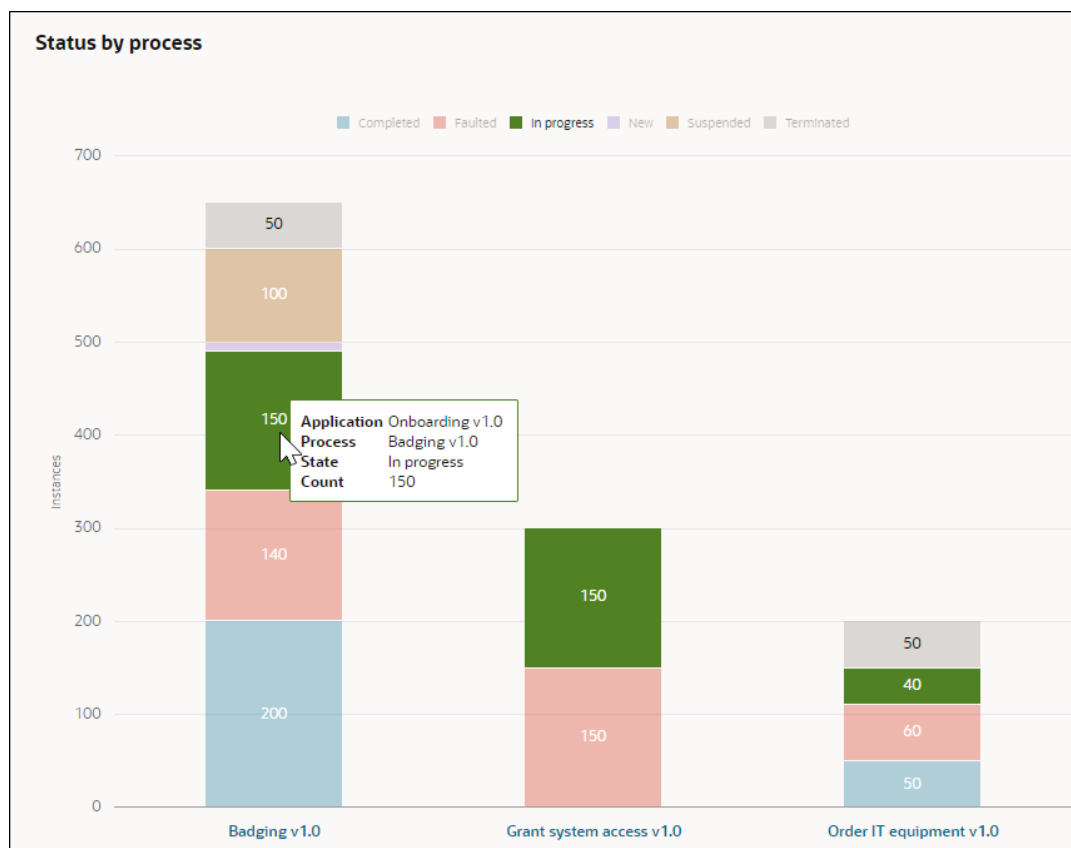
Status by process

The **Status by process** section is on the left where each process within the selected application is represented by a stacked bar chart. The stacked bar shows the cumulative count of a process in different instance states.

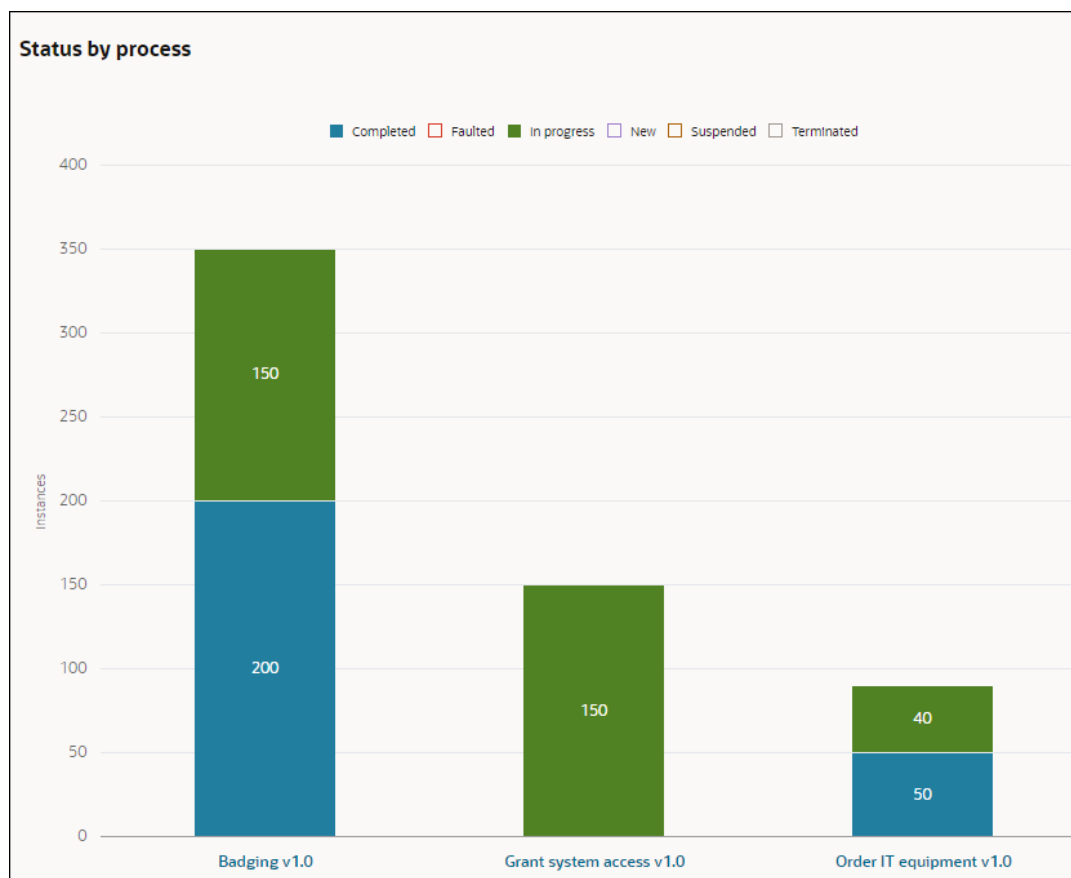


The following options are available to further refine the view of the stacked bar chart and get more details:

- **Interactive legends and stacked bars:** When you hover over a state legend or a state in the stacked bar chart, then the other states dim, and the selected state gets highlighted in the stacked bar chart.



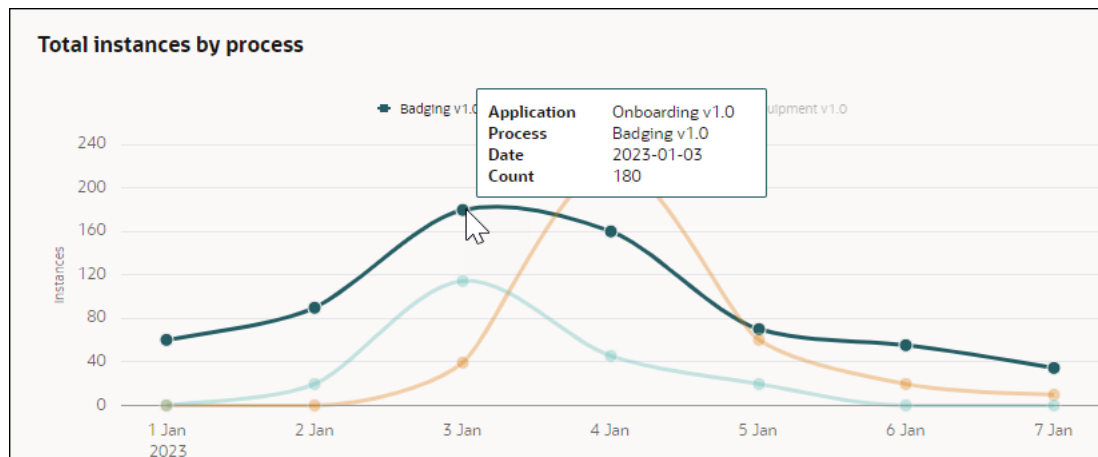
- Filter by State:** Suppose you want to view process instances that are in Completed and In progress state only, click the other state legends on top to hide the states represented by them. The stacked bar chart will only show the total number of In progress and Completed process instances for the processes.



Total instances by process

This is the first chart that is displayed on the right section. The report for the chart is based on the duration that the user has selected. For example, if you've selected **Past Week** in the **Period** field, then daily data for each day of the past week is shown. The X-axis of the chart will be represented by the days of the past week. The Y-axis represents instance counts. The different colored lines represent the processes present. Hovering over one of the plot points in a line shows the total number of instances of the process that were present on that given date (X-axis).

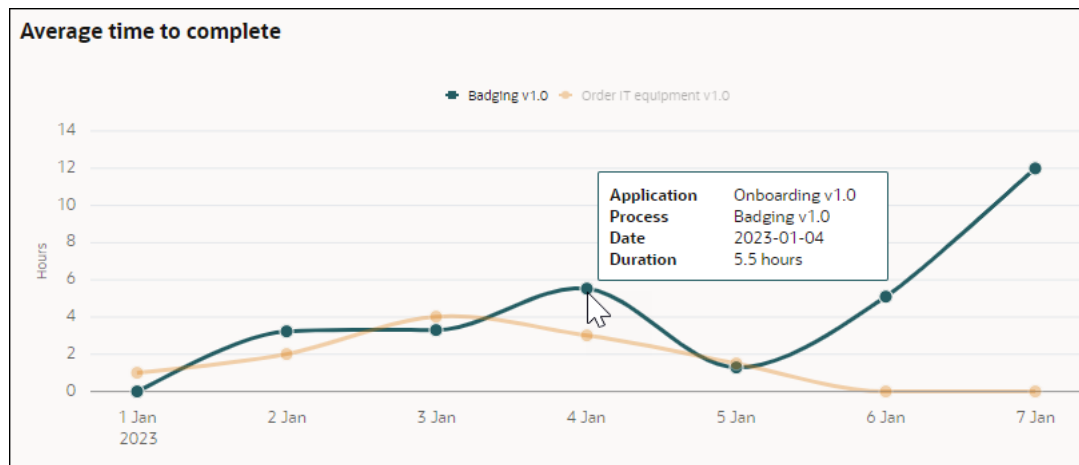
For example, hovering over the Badging process on Jan 3, shows that 180 process instances were present on that date.



Average time to complete

This is the second chart that is displayed on the right section. The report for the chart is based on the duration that the user has selected. For example, if you've selected **Past Week** in the **Period** field, then daily data for each day of the past week is shown. The X-axis of the charts is represented by the days of the past week. The Y-axis represents duration in hours. The different colored lines represent the processes present. Hovering over one of the plot points in a line shows the average time taken by the process to complete on that date (X-axis).

For example, hovering over the Badging process on Jan 4, shows that the average time taken by the process to complete on that date is 5.5 hours.



View Analytics for a Process

You can view analytics report for a process and get insights into the instances of the process over a period of time.

From the [application level visualization](#), to view analytics report for a process:

- Search and select the process from the search field.
- Click the process from the X-axis in the stacked bar graph.

Analytics reports for the instances in the process are displayed in the form of three views.

1. [Status over time](#)
2. [Instances by status](#)
3. [Average time to complete](#)

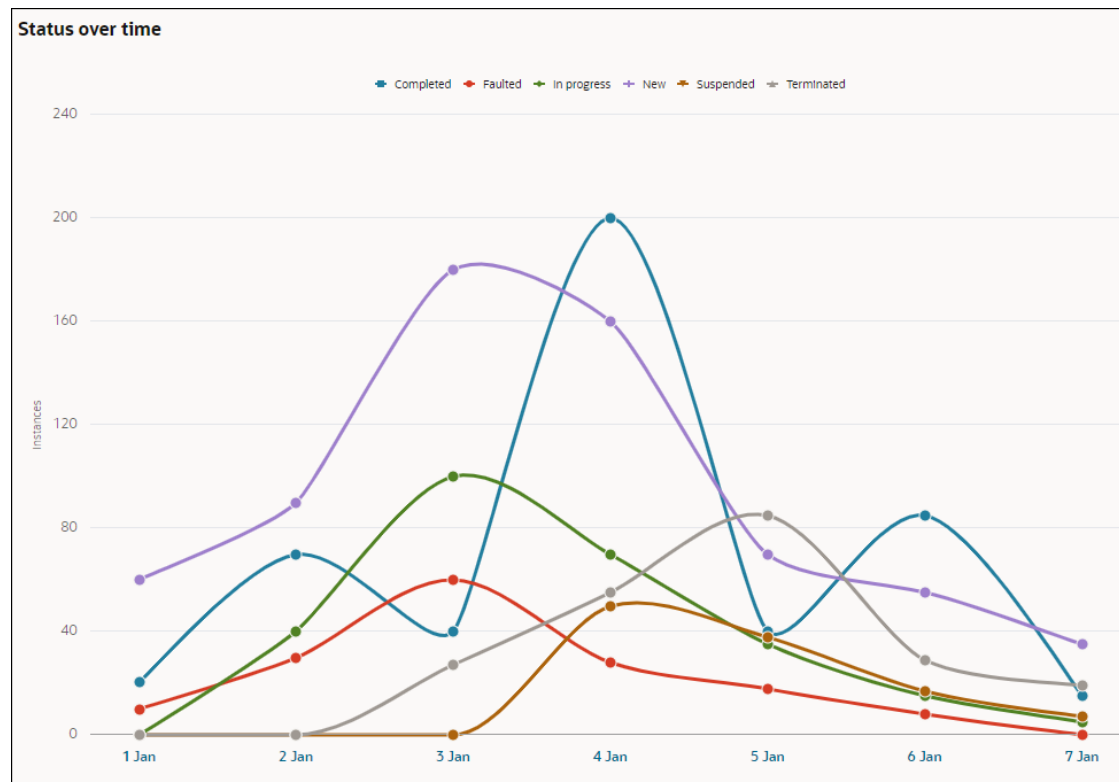
In addition, you can also drill down and [get analytics for a particular day](#).

Tip: A quick way to switch to a different process in the application is to use the filter chip displayed for the process below the search field.

Status over time

The **Status over time** section is present on the left where different states are represented by colored line graphs. The total instance counts for different states is shown over a duration that is selected in the **Period** field. The X-axis represents the time duration, and the Y-axis represents the total instance counts.

For example, if the period selected is past week, then you can view the total instance counts for different states on a given day in that week.

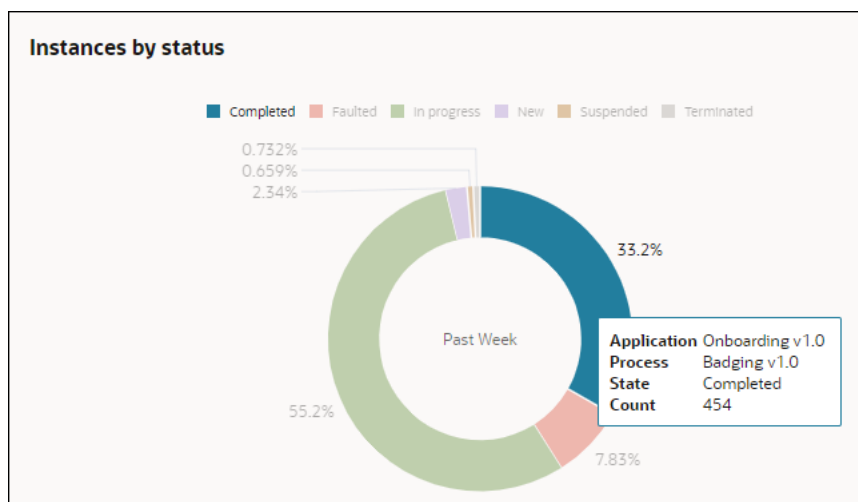


You can hover over a plotted point in a line graph to view the total instances in the selected state on that particular day.

Instances by status

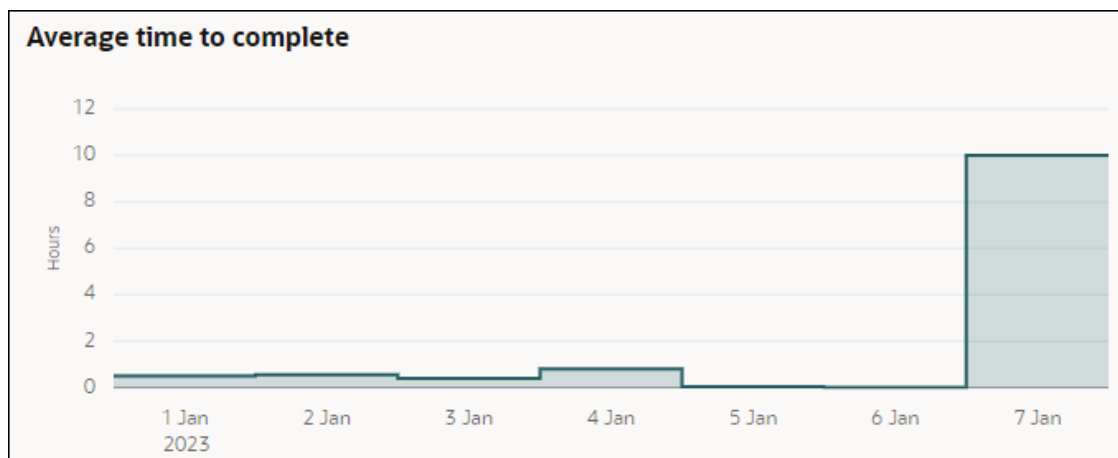
The **Instances by status** section is represented by a doughnut chart. By default, a summary view is shown in the doughnut chart based on the duration selected. The percentage value for the different instance states in the selected duration is shown on the rings of the doughnut chart.

You can hover over a state ring in the doughnut chart to view the total instance count for that state.



Average time to complete

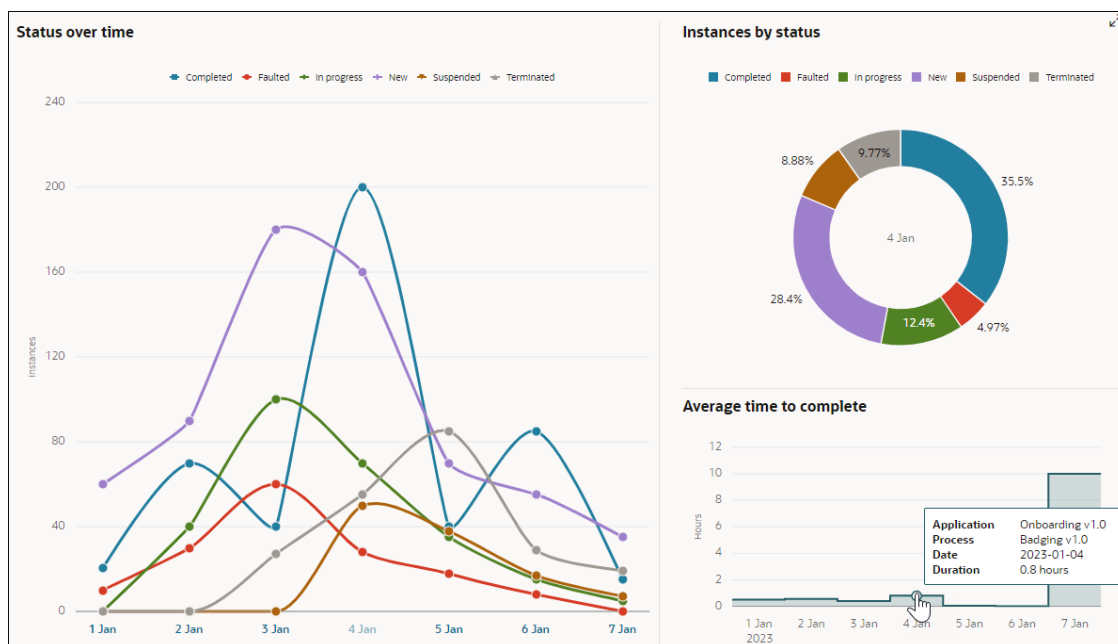
This chart shows the average time to complete in a line with area graph. Based on the duration selected, the average time taken to complete the process is presented in a line with area.



Get analytics for a particular day

To view analytics report for a day, from the [process level visualization](#) select the date in the X-axis of the **Status over time** chart. The visualization in the related **Instances by status** and **Average time to complete** charts on the right changes to show the reports for the selected day.

In the following example, on selecting Jan 4 in the X-axis of the **Status over time** chart, the **Instances by status** section's doughnut chart presents a visualization of the percentage of process instances in different states on Jan 4. The line with area chart of the **Average time to complete** section shows the average time that process instances took to complete on Jan 4.

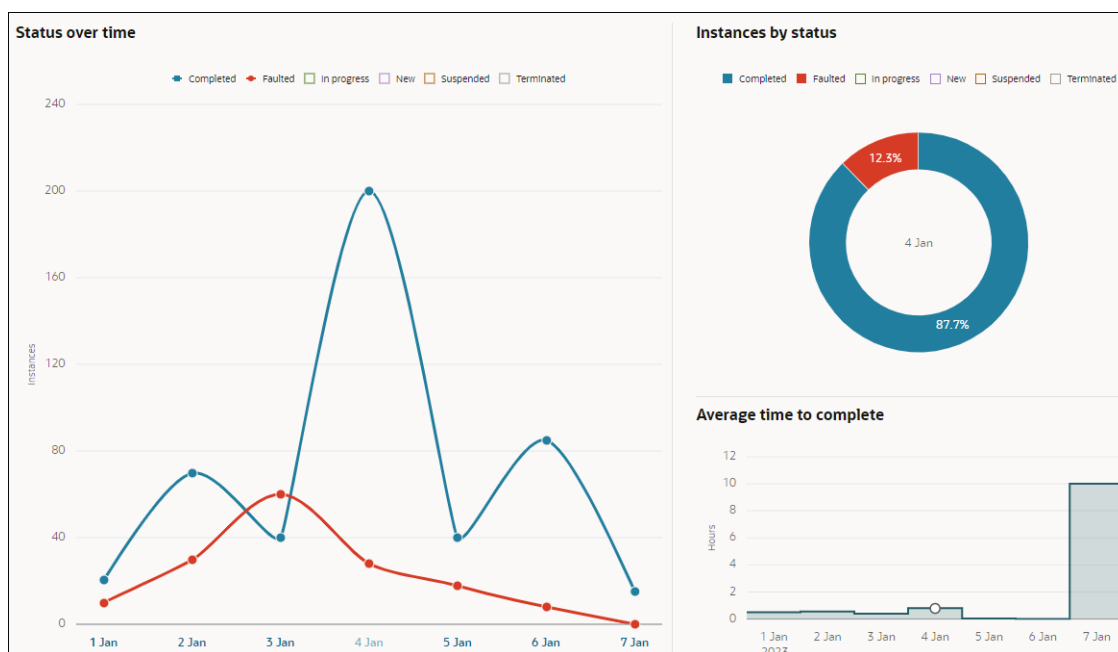


Note that if there are no Completed instances present in the process, then the **Average time to complete** section won't show any data.

As in the application and process level views, in the daily view too the state legends are interactive. Hover over a state legend on top or a state line graph of the **Status over time** section to highlight that state in the graph. The state also gets highlighted in the doughnut chart of the **Instances by status** section.

If you want to filter and view the instances in a particular state then disable the other states by clicking the state legends.

In the following example, the In progress, New, Suspended and Terminated states are disabled to get a visualization with only the Completed and Faulted states.



View Analytics for Activities

You can view analytics report for activities in a process and get insights into the performance of the activities.

To view activity level analytics:

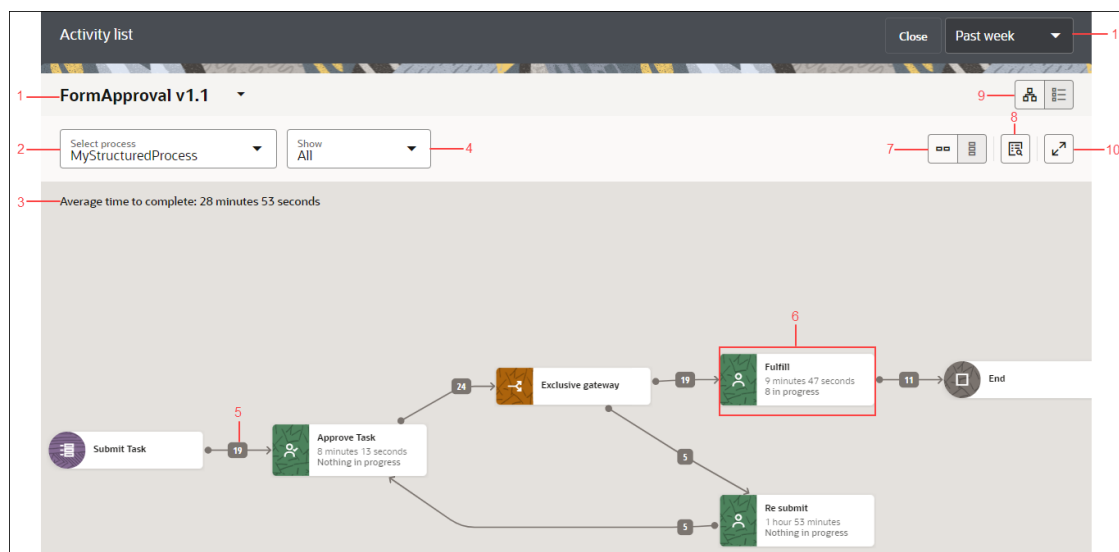
1. Navigate to the Activity list page of a process.
 - From the [application level visualization](#), click the **View process details** link above the Total instances by process section. By default, the activity list for the first process in the application opens in the [graphical view](#).
 - From the [process level visualization](#), click the **View process details** link above the Instances by status section. By default, the activity list for the selected process opens in the [graphical view](#).

The details of a process with its activities are available in the Activity list page in two types of visualizations: graphical view and list view. In the graphical and list views, you get high level analytics data for each activity in the process. For example, number of activity instances that were started, number of activity instances that are in progress, the time taken for certain activities to complete, and so on. See [Activity list - Graphical view](#) and [Activity list - List view](#).

2. Click an activity to view detail analytics report for the activity. See [View Analytics for a Human Task Activity](#) and [View Analytics for a Service Task Activity](#).

Activity list - Graphical view

The following is an example of graphical view that shows details of the activities in a structured process along with the activity flow.



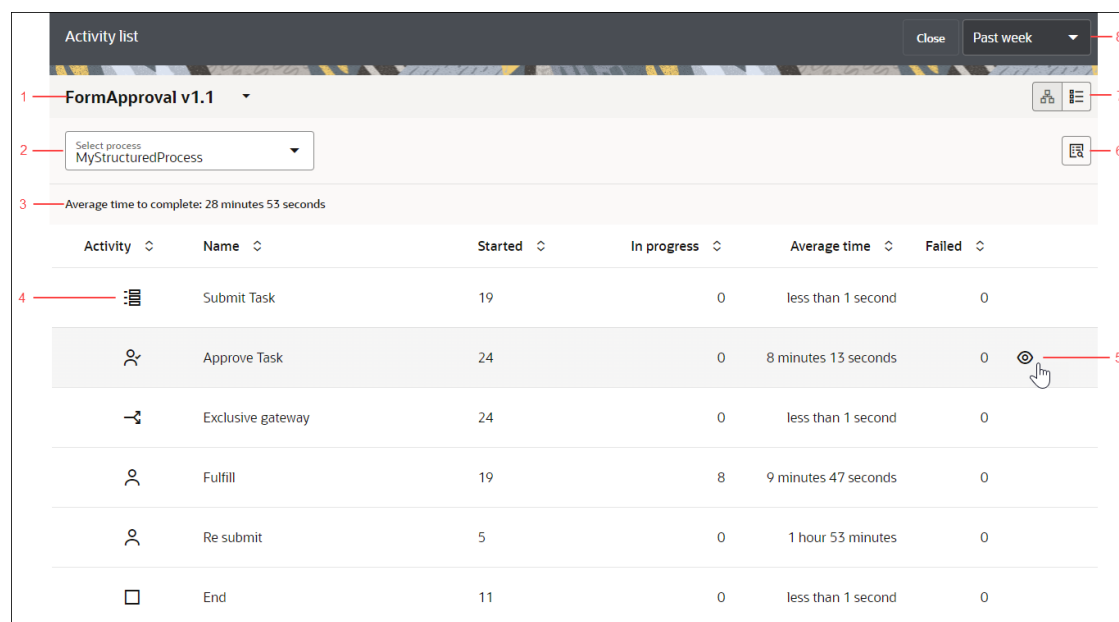
Note the following in the above example of graphical view:

1. The application to which the process (whose activity flow is depicted) belongs. If you want to switch to another application, click the drop-down field and choose a different application from the list.
2. The process whose activity flow is graphically represented. All processes available in the selected application version are listed when you click the drop-down field. If you want to switch to another process, choose it from the list.

3. The average time taken by the process to complete.
4. The **Show** drop-down field where you can select any of the available filters: **Errors**, **Time**, or **In progress**. By default, **All** is selected which means no filter is applied. When a filter is selected, only the filtered activities are highlighted in the activity flow, and all other activities are dimmed.
5. The number of instances that were started for a particular activity. For example, 19 instances of the form start event (Submit Task) were started.
6. Different activities are denoted by different nodes. Hovering over a node, depicts the activity that the node denotes. For example, in this case the node denotes a Submit human task activity. For some activities, you get additional information in the node, such as the time taken for the activity to complete and if the activity is in progress or not. An eye icon on an activity denotes that you can view detail analytics report for the activity. The eye icon is available for human task, service task, upload form and integration activities. See [View Analytics for a Human Task Activity](#) and [View Analytics for a Service Task Activity](#).
7. Use the **Horizontal layout** or **Vertical layout** icons to switch between the two layouts. In this example, **Horizontal layout** is selected.
8. Use the **View instances** icon to navigate to the Tracking page. See [Analytics: Instance Tracking](#).
9. Use the **Graphical view** or **List view** icons to switch between the two views. By default, graphical view is selected.
10. Use the **Maximize** icon to maximize the window.
11. The time period for which the analytics report is displayed. Click the drop-down field to choose another time period option from the list. The options are **Past week**, **Past month**, **Past year**, and **Custom**.

Activity list - List view

The following is an example of list view that shows details of the activities in a structured process.



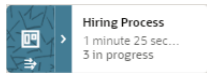
Activity	Name	Started	In progress	Average time	Failed
Submit Task	Submit Task	19	0	less than 1 second	0
Approve Task	Approve Task	24	0	8 minutes 13 seconds	0
Exclusive gateway	Exclusive gateway	24	0	less than 1 second	0
Fulfill	Fulfill	19	8	9 minutes 47 seconds	0
Re submit	Re submit	5	0	1 hour 53 minutes	0
End	End	11	0	less than 1 second	0

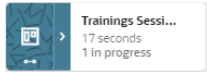
Note the following in the above example of list view:

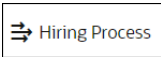
1. The application to which the process (whose activities are displayed) belongs. If you want to switch to another application, click the drop-down field and choose a different application from the list.
2. The process whose activities are displayed in the list view. All processes available in the selected application version are listed when you click the drop-down field. If you want to switch to another process, choose it from the list.
3. The average time taken by the process to complete.
4. Different icons are used to denote different types of activities. In this case, the form start event icon is highlighted. Hovering over an icon reveals the type of activity that the icon denotes.
5. An eye icon is displayed on selecting a human task, service task, upload form and integration activity. The eye icon indicates that you can view detail analytics report for the activity. See [View Analytics for a Human Task Activity](#) and [View Analytics for a Service Task Activity](#).
6. Use the **View Instances** icon to navigate to the Tracking page. See [Analytics: Instance Tracking](#).
7. Use the **Graphical view** or the **List view** icons to switch between the two views. In this example, the list view is selected.
8. The time period for which the analytics report is displayed. Click the drop-down field to choose another time period option from the list. The options are **Past week**, **Past month**, **Past year**, and **Custom**.


Some key points to note in the graphical and list views

- **Stages** in a dynamic process are by default shown in expanded view depicting all activities inside it.
- **Subprocesses** in a structured process are by default shown in collapsed view. Expand the subprocess to view the activities inside it and their flow.
- **Call** activity in a structured process is shown with a link to the called process. Click the link to navigate to the called process.
- **Multi-instance** activities are shown with respective icons to denote if they are parallel multi-instance or sequential multi-instance activities. Examples:

– Parallel multi-instance activity (graphical view): 

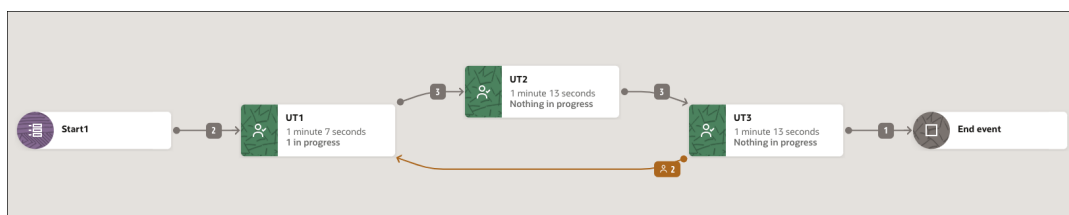
– Sequential multi-instance activity (graphical view): 

– Parallel multi-instance activity (list view): 

– Sequential multi-instance activity (list view): 

- If the process flow has been altered by manual intervention, for example certain activities were skipped, the **altered flow** of the process is depicted as a new line between the start and target activities. To indicate that this action was performed outside the normal flow of the process, a human icon is present in the transition line and the color of the line is different from the normal flow.

In the below example, the altered flow is depicted by the orange flow line and the human icon on the line indicates that the flow has been altered by manual intervention.



View Analytics for a Human Task Activity

Select and click a human task activity in the activity list to view detail analytics report for the human task activity.

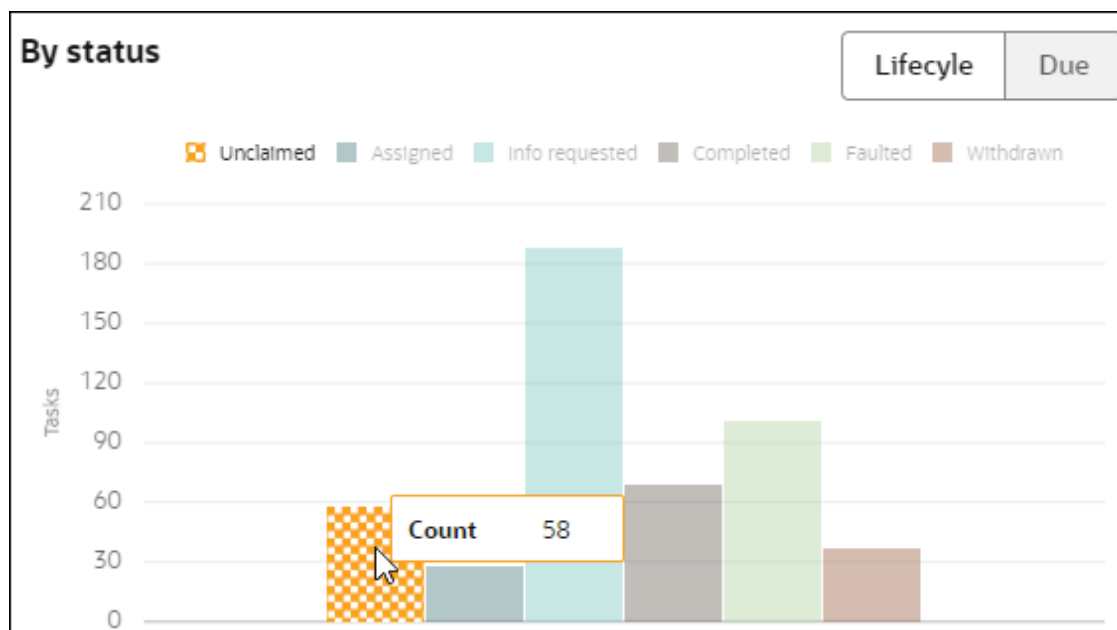
Analytics report for the human task activity is displayed in a drawer that opens on the right of the Activity list page. You can filter the analytics report either by the task's lifecycle or due date.

1. [By status](#)
2. [By assignee](#)
3. [Outcome](#)

By status

The **By status** section presents the different states of the human task instances represented by a colored bar chart. The colored bars on the X-axis represents the different states of the task. The Y-axis represents the instance counts of the task. The legends for the different states of the task are present on top. If you hover over a state legend or a state bar on the chart, the other states dim and the selected state gets highlighted in the bar chart.

Hover over a task state bar to know the total task instance count for that state. For example, hovering over the Unclaimed task state bar shows that 58 instances of the human task are unclaimed.

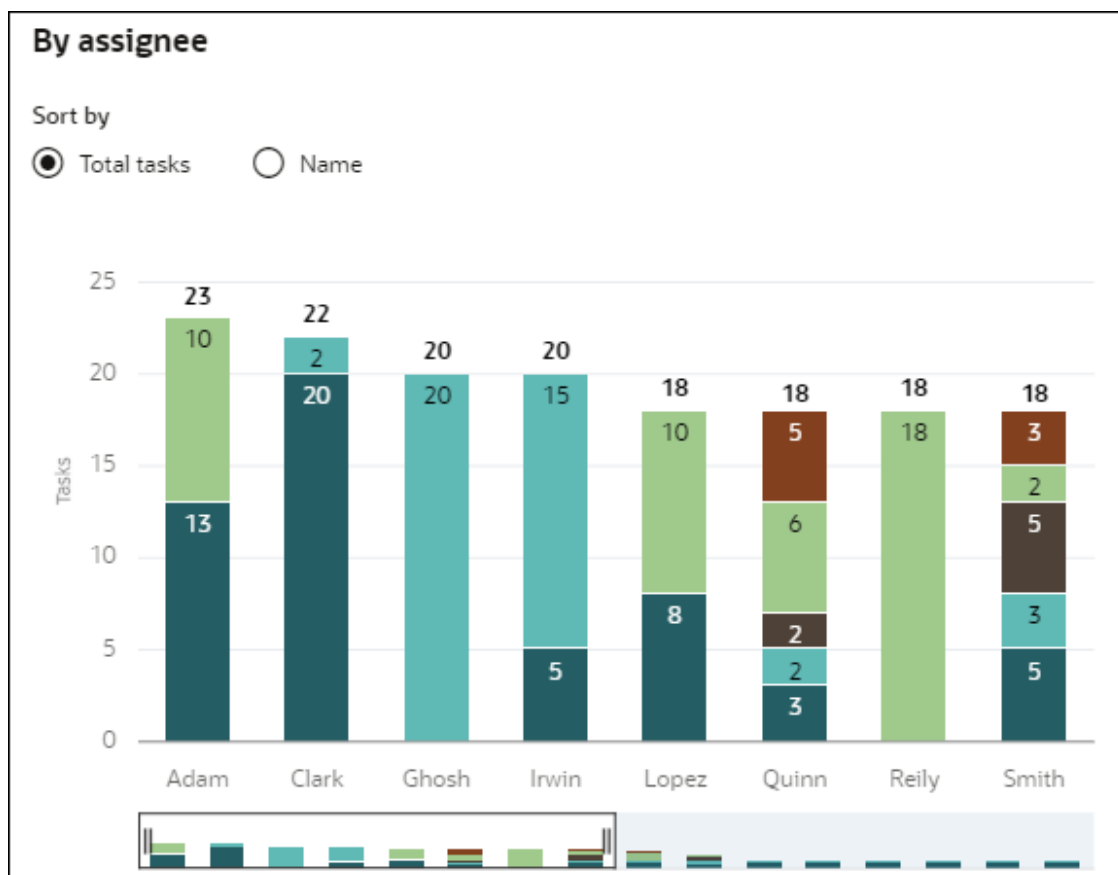


If you select **Due**, the bar chart view changes to display human task instances that are On track, Overdue or Expired.

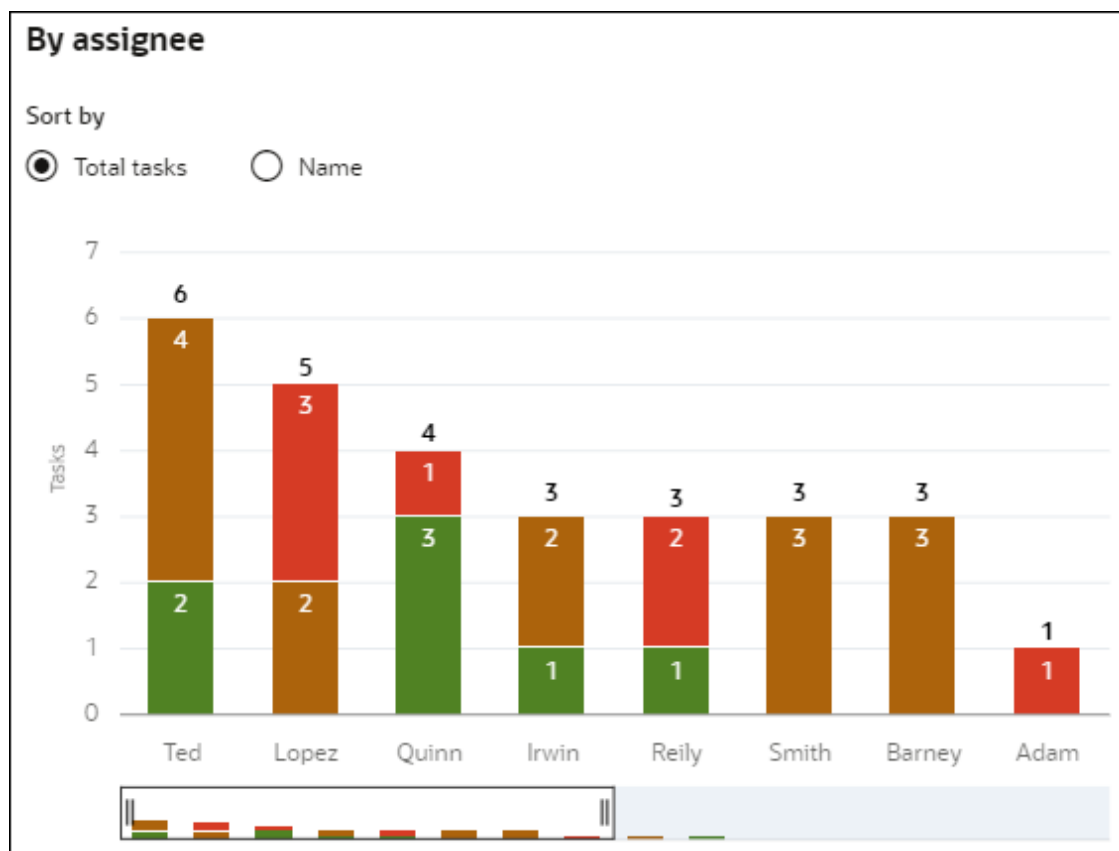


By assignee

The **By assignee** section is represented by a stacked bar graph visualization of task instances in their different states assigned to particular users. You can use the **Sort by** option to sort the stacked bars either by total task counts (**Total tasks**) or by name of the assignee (**Name**). If there are more than 8 users assigned to the task, you can use the horizontal scroller to display analytics for other assignees.

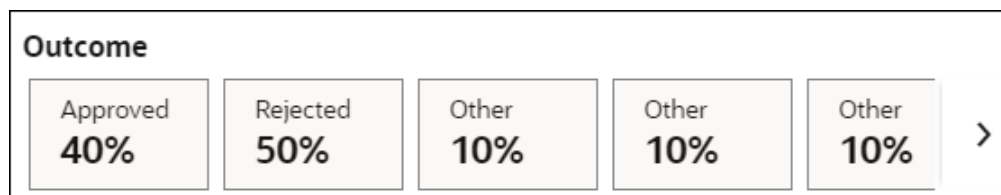


If you select **Due**, you get a stacked bar graph visualization of task instances that are On track, Overdue or Expired and assigned to different users. The X-axis represents users assigned the task, and the Y-axis represents the total task instance counts. You can sort the visualization data according to total tasks or name of the assignee.



Outcome

The **Outcome** section is displayed only for human task activities of type Approve. It shows the percentage of the human task instances that were approved or rejected. If there are any custom actions that were configured during design time for the activity, then the percentage of task instances for the custom action is also shown. If there are multiple custom actions configured for the human task activity, then you can use the right arrow to scroll to the right and view the percentage of instances for those custom actions.



View Analytics for a Service Task Activity

Select and click a service task activity in the activity list to view detail analytics report for the service task activity.

Analytics report for the service task activity is displayed in the form of two views in a drawer that opens on the right of the Activity list page.

1. [Total requests](#)

2. Error types

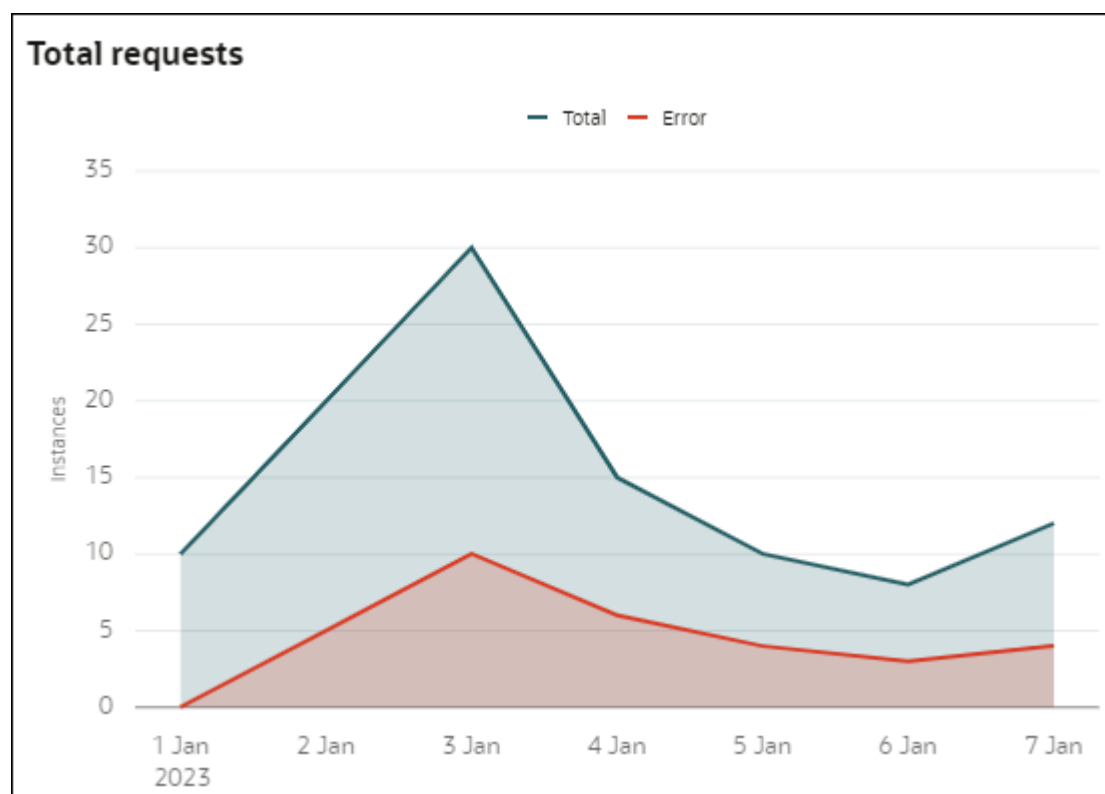


Note:

If a service task has any instance that was retried after it failed to run, a badge *Retried* will be shown for the service task activity under the Started column in the Activity list.

Total requests

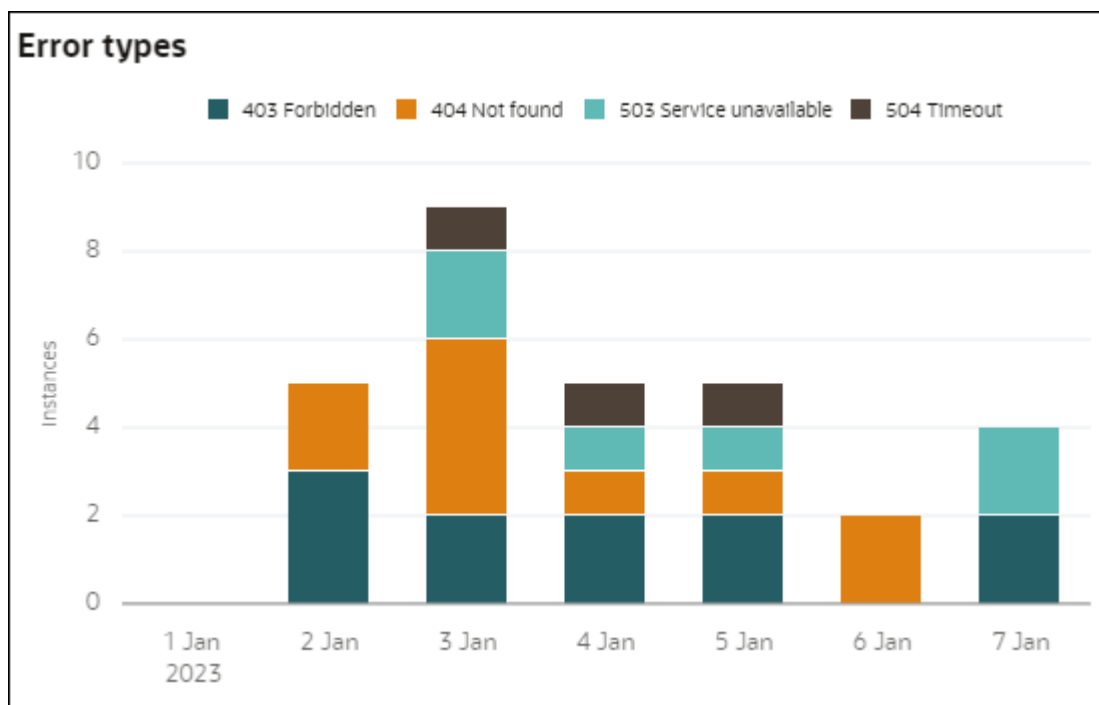
The **Total requests** section is a line chart with area that shows the total number of instances and the number of instances that has errors. The X-axis represents the time duration, and the Y-axis represents the total instance counts.



Error types

The **Error types** section is shown as a stacked bar chart. The stacked bar categorizes the errors that occurred in a particular duration into different error types and shows the cumulative count of errors. The X-axis represents the time duration and the Y-axis represents the total error instances.

For example, if on Jan 3 there were 9 error instances in total, then the stacked bar gives a visualization of those errors according to the different error types - 2 of those errors were 403 Forbidden, 4 errors were 404 Not found, 2 errors were 503 Service unavailable, and 1 error was 504 Timeout.




You can also choose to hide a particular error type by clicking the corresponding error legend.

Note that if a service task fails due to REST API error, instead of the visualization charts an error message is shown along with the OPC (Open Platform Communications) request ID using which you can track the error.

About Instance Status

The following table lists some instance status or states with their descriptions.

Status	Description
New	The number of instances started for a particular process on a given date. This metric shows process <i>demand</i> .

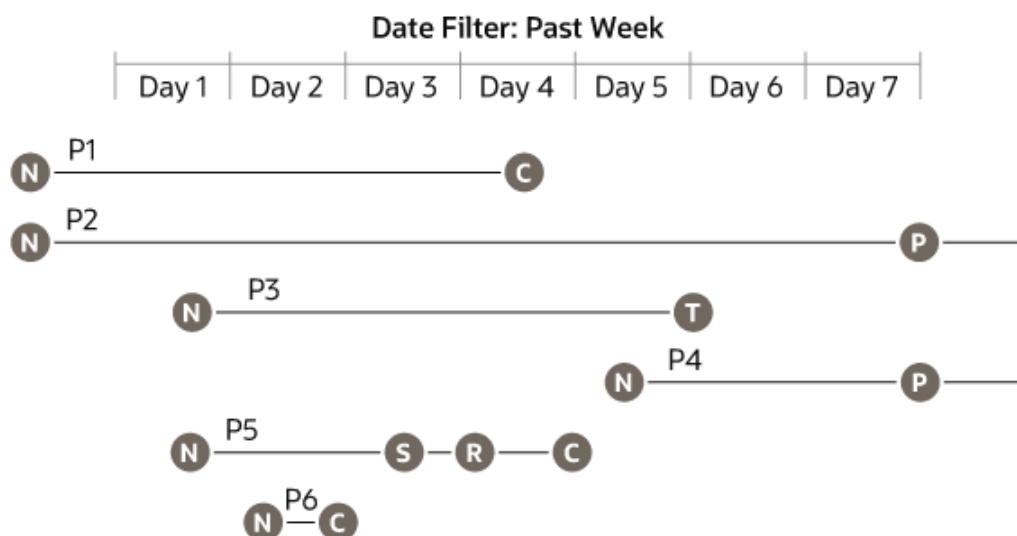
Status	Description
Terminal	<p>This status indicates the final state of a process instance.</p> <p>For the purposes of analytics, we consider all other process states apart from New and In-progress to be terminal.</p> <p>The following process states are considered to be in Terminal state in analytics.</p> <ul style="list-style-type: none"> • Completed • Faulted • Suspended • Terminated
	<div style="border: 1px solid #0070C0; padding: 10px; background-color: #E6F2FF;"> <p> Note:</p> <p>Usually a Terminal state will not change to other states such as Completed or Faulted. However, in some cases the terminal state can change to another terminal state or to the in progress state. For example, a <i>Suspended</i> process can be resumed and <i>Completed</i>.</p> </div>
In progress	<p>This state indicates that the process instance is still running or is active. It hasn't completed yet.</p> <p>This metric indicates <i>workload</i>.</p>

Calculating New, Terminal, and In Progress instances in Daily view

Status	Daily View Calculation
New	The calculation for this state in a daily view is based on whether the process start date is equal to the date viewed. In that case, the count for this state will be incremented by 1.
Terminal	<p>In a daily view, a process instance is shown as terminal if a terminal event for the process is received on that day. For example, if a process has completed or faulted today, in that case the count for the terminal state will increment by 1.</p> <p>If however a terminal event is superseded by an in-progress event (for example, a process is recovered) then this action would remove the terminal count (-1) and the process would show as in-progress.</p>
In progress	In a daily breakdown we show an in-progress state if we have not received a terminal event for that process on that day. For example Process A is created on Jan 12, and is active until a terminal event is received on Jan 15. Then for that process instance an in progress count would show on the 12th, 13th, and 14th Jan.

Calculating New, Terminal, and In Progress instances in Summary view

The following example illustrates the calculation of New, Terminal (Terminated, Completed, Suspended), and In progress instances in summary view for a duration of Past Week.




Summary	
New (N)	4 (P3, P4, P5, P6)
In Progress (P)	2 (P2, P4)
Terminated (T)	1 (P3)
Completed (C)	3 (P1, P5, P6)
Suspended (S)	0 (P5 was recovered)

Analytics: Instance Tracking

You can navigate to the instance Tracking page from any level of Analytics view. By navigating to the instance's Tracking page and viewing its audit, users can investigate and resolve any errors or bottlenecks identified in the analytics report.

Navigate to the [Tracking page](#) that opens on a new browser tab from Analytics by:

- Clicking the **View instances** link that is available in the analytics report window of an application, process, or activity.
- Selecting the status bar in an application or process level visualization.
- Selecting the **View instances**  icon on top right of the process details graphical or list view in the Activity list page.

The filters that are applied in the Analytics page are propagated to the Tracking page when you navigate to the instance Tracking page from the Analytics report view. So you can always contextually navigate to the Tracking page from an instance's analytics report view.

Suppose you're in the analytics view of an application for a selected duration. When you navigate to the Tracking page from the analytics view of that application by clicking the **View instances** link, then the same filters (application and duration) will be applied to the Tracking page too. If a user navigates to the Tracking page by clicking any of the process instance status bars in the Analytics page, then along with the application and duration period filters, the process name and the instance status filters also are propagated and applied to the Tracking page.

Similarly, [filters that are applied in the Tracking page](#) are also propagated to the Analytics page when the user navigates to the Analytics user interface from the Tracking page.

Set to Out of Office

To avoid delays in completing tasks, Workspace users can configure the out of office settings so that Process Automation can reassign tasks.

Process Automation lets Workspace users specify multiple periods of their absence so that their directly assigned tasks can be reassigned to other users or roles. Once you create a leave of absence, you can edit the details or cancel the leave.

Process Automation administrators can also configure the out of office settings for users in scenarios where the user is unable to update or configure the settings due to unplanned or extended leaves. Once the Process Automation administrator or the user configures the out of office settings, Process Automation displays the out of office banner on the My Tasks page showing the details of the out of office configuration.

When a task is reassigned to a role, the status of the task changes to unclaimed state and anyone with the assigned role can work on the task. The new task assignee can view the comment provided by the original user in the out of office configuration settings.



Note:

- The out of office feature does not work on role based task assignments.
- You cannot create multiple out of office periods that have overlapping dates. The start and end date of each out of office period must be exclusive to the dates in any other out of office period.
- Process Automation does not allow you to create an out of office period if you are assigning it to a user who has also created an out of office leave for the same period.

Click the user icon on the top-right of the Workspace UI, then click **Out of office**.

Create Out of Office Record

Users can specify multiple periods of their absence and specify an assignee or a role who can work on the assigned tasks.

1. Click the user icon on the top-right of the Workspace UI, then click **Out of office**.
The Out of office page appears.
2. In the **Start date** field, specify the starting date of the leave.
3. In the **End date** field, specify the ending date of the leave.
4. From the **Reassign to** drop-down field specify whether you want to assign the tasks to a user or a role.
5. In the **Assignee** field, enter the name of the user whom you to assign the tasks.
6. In the **Comment** field, enter the information that you want the assigned user to read.
7. Click **Submit** icon.

11

Workspace Administration

Process Automation Administrators (assigned the ServiceAdministrator IDCS application role) have access to the Administration pages in Workspace to manage and work on various administrative tasks.

On the Workspace navigation pane, expand **Administration** and choose an option under it to manage roles, notifications, credentials, and much more.

Topics:

- [Manage Roles in Workspace](#)
- [Manage Notifications](#)
- [Manage Credentials in Workspace](#)
- [Enable Break Glass](#)
- [Specify Data Management Settings](#)
- [Register Services](#)

Manage Roles in Workspace

As a Process Automation Administrator, you can manage global and application roles from Workspace. Application and global roles that are created in design-time by Process Automation Designers can be viewed, edited or deleted; and global roles can be created for production purposes. In addition, you can also assign a user to the Process Application Administrator role. Any change that you do is effective immediately.

- [View or Edit Roles After Activation](#)
- [Add a Global Role in Workspace](#)
- [Assign the Process Application Administrator Role](#)
- [Delete Roles in Workspace](#)

View or Edit Roles After Activation


You can view application and global roles that were created and activated in design time, and also modify them for production purposes from the Workspace Roles page.


1. From the Workspace navigation menu, click and expand **Administration** and then choose **Roles**.

A list of roles configured for the Process Automation instance is displayed.

2. Find the role you want to view or edit.
 - a. Select the **Global** or **Application** tab. Global roles are listed under the **Global** tab and application roles are listed under the **Application** tab.
 - b. Click the **Search** icon and enter the first few letters of a role's name (case sensitive) in the **Search** field.

3. Select the role to open and view its users/groups, permissions, and the process application/applications that the role is used.


You can also click **Actions**  and choose **Edit** to open and view the role's details.

To delete a role, click **Actions**  and choose **Delete**. Note that the *Process Application Administrator* global role is a system generated role and can't be deleted.

4. You can do the following in the role's details window:
 - a. Add users or groups to the role.
 - i. In the **Search by** field, select **Users**, **Groups**, or **External applications** from the drop-down list.
 - ii. In the search field, search and select the user, group or external application that you want to add to the role. A tick mark next to the user, group, or external application indicates that it has been selected, and the newly added user, group or external application gets listed under **Add members to this role**.

 **Note:**

Changing a global role's users, groups, or external application changes them for every application in which the global role is used.

- b. Update permissions for the role.
 - i. Expand **Permissions**. The Permissions section shows the application (or applications for global roles) in which the role is used.
 - ii. Click  in the **Action** column and choose a permission from the list to change the permission level.

This changes the user or group's permission level for the application in which the role is used.
- c. Additionally for global roles, you can assign permissions to applications.
 - i. In the **Search** field, search for the application to which you want to assign the permission or select it from the list of applications that display.
 - ii. By default, the **Read** permission is assigned. You can update the permission level if required.

 **Note:**

Permissions assigned to a global role can be deleted. However permission assigned (during design-time) to an application role cannot be deleted.

5. Click **Save**.

Add a Global Role in Workspace

You can create global roles in Workspace and assign users/groups or external applications, and permissions.

1. From the Workspace navigation pane, click and expand **Administration**, and then choose **Roles**.

A list of roles configured for the Process Automation instance is displayed.

2. Click **Create global role** in the upper right corner. The window for creating a global role appears.
3. Enter a name for the global role in the **Role name** field, optionally add a suitable description for the role, and click **Save**.
The new role is now listed on the **Global** tab.
4. Add users or groups to the role.
 - a. In the **Search by** field, select **Users**, **Groups**, or **External applications** from the drop-down list.
 - b. In the search field, search and select the user, group, or external application that you want to add to the role. A tick mark next to the user or group indicates that it has been selected, and the newly added user, group, or external application gets listed under **Add members to this role**.
5. Assign application/applications to the role.
 - a. Expand **Permissions**.
 - b. In the **Search** field, search for the application to which you want to add permission.
As a global role can be linked and used in many process applications, you can assign multiple applications to the role.
6. By default, Read permission is set for an application. If required, you can change the permission level to Inspect, Use or Manage for the application/applications that you added for the role.
7. Click **Save**.

Assign the Process Application Administrator Role

When you assign a user or group to the Process Application Administrator role, it allows the user or group to manage all resources in all process applications present in the Process Automation instance. Use caution in assigning this role.

Before you assign users or groups to the Process Application Administrator role, note the following:

- The Process Application Administrator role is created by default with a Process Automation instance.
- The Process Application Administrator role is a system generated role and *cannot* be deleted.
- The Process Application Administrator role has full permissions on all process applications, versions, and resources present in the Process Automation instance.

Note:

You need to be a Process Automation Administrator (assigned the ServiceAdministrator IDCS application role) to assign users and groups to the Process Application Administrator role.



1. From the Workspace navigation pane, click and expand **Administration** and choose **Roles**.

A list of roles configured for the Process Automation instance is displayed.

2. On the **Global** tab, search for `Process App`, and select **Process Application Administrator** to open it.
The role details window appears.
3. Add a user or group to the role.
 - a. In the **Search by** field, select **Users** or **Groups** from the drop-down list.
 - b. In the **Search for users and groups** field, search and select the user or group that you want to add to the role. A tick mark next to the user or group indicates that it has been selected, and the newly added user or group gets listed under **Users and Groups**.
4. Expand **Permissions**. Notice the single entry. It shows the Manage permission assigned to all process applications and versions (an asterisk indicates all selected) present in the Process Automation instance. You cannot change the permission level or delete the assigned permission.
5. Click **Save**.

Delete Roles in Workspace

You can delete an application or global role in Workspace. Before deleting a role, ensure that it's not being used in an active application.

1. Open the Workspace navigation pane, click and expand **Administration**, and then choose **Roles**.
A list of roles configured for the Process Automation instance is displayed.
2. Find the role you want to delete. You can choose a role on the **Global** tab or the **Application** tab.
3. See where the role is being used.
 - a. Select the role to open its details window. You can also click **Actions**  and choose **Edit** to open and view the role's details.
 - b. Expand **Permissions**, and view the applications that use the role.
 - c. Close the window by clicking **Cancel**.
4. Delete the role by selecting the role, then clicking **Actions**  and then selecting **Delete**.
The role is deleted and it no longer displays in the Roles page.

Manage Notifications

Configure Oracle Cloud Infrastructure Process Automation to use emails for notifying task assignees when events such as assignment and reassignment occur. You can also view the status of the notifications that are sent and resend or delete them if issues occur.

Note:

- To manage notifications, you must have the Service Administrator IAM role.
- To use your own from address like *no-reply@abc-demo.com*, configure Sender Policy Framework (SPF) and DomainKeys Identified Mail (DKIM) on the sender domain *abc-demo.com*. To configure DKIM, contact Oracle Support to create a service request. Configuring SPF and DKIM helps validate emails, avoid spoofing, and reduce fraud attacks. For more details on configuring DKIM and SPF, see [Configure Email Authentication Settings for SPF and DKIM](#). Additionally, you can also see [Troubleshoot Oracle Cloud Infrastructure Notification Email Configuration to Ensure Proper Delivery](#).
- Incorrect configuration of SPF and DKIM on the sender's domain server (DNS) can result in soft or hard bounces depending on the severity of the issue. This may lead to the email addresses being added to the suppression list.

Email Delivery in Oracle Cloud Infrastructure (OCI) automatically adds email addresses with bounce codes showing permanent failures, or user complaints to the suppression list to protect the sender reputation. Email Delivery doesn't send any messages to these recipients in the future. To remove email addresses from the suppression list, see [Managing Suppression List](#) and [Deleting an Email Address from Suppression List](#) in Oracle Cloud Infrastructure (OCI).

Before enabling email notifications in runtime, make sure you customize them, including their content, template, attachments, and subject lines in design time. See [Customize Notification Emails](#). After enabling email notifications, you can view the notification logs and resend emails to all or some of the original recipients.

1. Enable email notifications.
 - a. Open the navigation menu in Workspace, expand **Administration** and choose **Notification**.

The Notification page is displayed.

- b. Click **Approved Sender**.

An approved sender is a resource that enables Oracle Cloud Infrastructure Process Automation to send email with the registered From email address.

- c. On the **Register Approved Sender** pane, enter the email address in the **From Recipient** field to set it as the default sender for all email notifications.

Note that you cannot use certain reserved domains such as `@oracle.com`, `@gmail.com`, `@hotmail.com`, and `@yahoo.com` as approved senders. To learn more, see [Managing Approved Senders](#) in Oracle Cloud Infrastructure documentation.

 **Note:**

If the **From** address is not configured on the Notifications page, the **From** address defaults to the approved sender address of `noreply@processautomation.<region>.oraclecloud.com`. For example, if your region is Ashburn, the address is `noreply@processautomation.us-ashburn-1.oraclecloud.com`.

d. Click **Register**.

The default sender email address is registered with Oracle Cloud Infrastructure in your customer tenancy.

2. Test email notifications.

a. Click **Test Notifications**.

b. On the **Test Notifications** pane:

- Enter the email address of the person to whom the notification email has to be sent in the **Recipient** field.
- Enter the email's subject in the **Subject** field.
- Enter the email's content in the **Content** field.

c. Click **Send**.

d. Verify that the email was sent as expected.

3. View notification status and troubleshoot as needed.

You can view the status of notifications; if they have been successfully sent, if sent failed and you need to retry sending them, and also if any error occurred for the notifications.

Email notifications may fail due to lack of network availability, wrong email address, or temporarily unavailable email servers. By default, a resend is automatically attempted for all failed notifications after an interval of fifteen minutes.

To manually retry or delete a notification, select the notification and click **Actions**.

- **Retry:** Click **Retry** to retry sending the notification. Confirm when prompted to confirm the retry.
- **Delete:** Click **Delete** to delete the notification. Confirm when prompted to confirm deletion.

Manage Credentials in Workspace

As a Process Automation Administrator, you can edit application credentials and manage global credentials in Workspace.

- [About Credentials](#)
- [Edit Application Credentials](#)
- [Manage Global Credentials](#)

About Credentials

Credentials are used by connectors to authenticate and securely call other services.

You can use either *application* credentials or *global* credentials to secure your connector.

- **Application credentials** are specific to a particular process application and can be used only in connectors configured for that particular process application. They can't be shared with other process applications. These credentials are usually created by Process Automation Designers while creating and securing connectors for a process application in design-time. See [Secure the REST Connector](#).
- **Global credentials** can be used in connectors across many process applications. These credentials are created by Process Automation Administrators in Workspace.
 - [Add Global Basic Credential](#)
 - [Add Global OAuth Credential \(Client Credential\)](#)
 - [Add Global OAuth Credential \(JWT Assertion\)](#)
 - [Add Global OCI Signature Policy Credential](#)

It is *recommended* that you use global credentials. Once created these credentials can be used in connectors across many process applications in design-time. See [Use Global Credentials](#). Also, when you update a global credential, the update is applied to all the connectors (across process applications) where the global credential is used.

As a Process Automation Administrator, you can edit application credentials, and manage global credentials from Workspace.

- [Edit Application Credentials](#)
- [Manage Global Credentials](#)

**Note:**

Credentials are stored centrally. They are separate from a process application's metadata and aren't exported or imported with a process application.

Edit Application Credentials

From Workspace Administration, you can edit application credentials that were created in Designer.

1. From the Workspace navigation menu, click and expand **Administration** and then click **Credentials**.
2. In the Credentials page, click the **Application** tab.
3. All the application specific credentials created for the Process Automation instance are listed.
4. Search and select an application credential that you want to edit.
5. Click the **Actions** menu (three horizontal dots), and then click **Edit**.
6. In the Update credential dialog, you can edit the fields as required.
 - Note that some fields such as **Folder Name**, **Credential Name**, and **Grant Type** (in case of OAuth credentials) can't be updated. Such fields are greyed out.
 - After you edit an OAuth credential of either Client Credential or JWT Assertion grant type, you can test the updates made by clicking **Validate**.

For more information about the fields in the Update credential dialog, see:

- [Configure Basic Auth Security](#)
- [Configure OAuth Security with Client Credential](#)
- [Configure OAuth Security with JWT Assertion](#)
- [Configure Security with OCI Signature Policy](#)

Manage Global Credentials

As a Process Automation Administrator, from Workspace Administration you can create, edit or delete global credentials.

- [Add Global Basic Credential](#)
- [Add Global OAuth Credential \(Client Credential\)](#)
- [Add Global OAuth Credential \(JWT Assertion\)](#)
- [Add Global OCI Signature Policy Credential](#)
- [Edit or Delete Global Credentials](#)

Add Global Basic Credential

As a Process Automation Administrator, you can define a global basic credential in Workspace. In Designer, Process Automation Designers can find and use this credential while configuring connectors to interact with other applications and services.

1. From the Workspace navigation pane, click and expand **Administration**, and then click **Credentials**.

The Credentials page appears. If you're not on the Global credentials page, click the **Global** tab.

2. On the top right of the Global Credentials page, click **Create global credentials**.
3. From the drop-down list that displays, select **Basic Credentials**.

The Add a new Basic credential dialog displays.

4. In the Add a new Basic credential dialog, enter the following details:

Field	Information to Enter
Folder Name	Specify the folder in which to store the credential. This single level folder structure enables you to organize credentials by grouping them in folders. <ul style="list-style-type: none"> • To save the credential in the default Common folder, leave Common selected. • To save the credential in a new folder, click the Create new link and enter a new folder name. You can't change a global credential's folder later.
Credential Name	Enter a name to identify the credential. You can't change a global credential's name later.
Target URL	Enter the base URL that the connector will access in connector calls.
Username	Enter the username that users must enter to call the REST service.
Password	Enter the password that users must enter to call the REST service.
Confirm Password	Re-enter the password.
Description	Optionally, specify how the credential will be used.

5. Click **Submit**.

The global credential is listed under the **Global** tab. Process Automation Designers can now find and select the global credential when creating connectors that use the same base URL. See [Use Global Credentials](#).

Add Global OAuth Credential (Client Credential)

As a Process Automation Administrator, you can define a global OAuth credential of Client Credential grant type in Workspace. In Designer, Process Automation Designers can find and use this credential while configuring connectors to interact with other applications and services.

1. From the Workspace navigation pane, click and expand **Administration**, and then click **Credentials**.

The Credentials page appears. If you're not on the Global credentials page, click the **Global** tab.

2. On the top right of the Global Credentials page, click **Create global credentials**.
3. From the drop-down list that displays, select **OAuth Credentials**.

The Add a new OAuth credential dialog displays.

4. In the Add a new OAuth credential dialog, enter the following details:

To get the information you need to complete the fields, see the Security, Authentication and Authorization section in *REST API for Oracle Cloud Infrastructure Process Automation*.

Field	Description
Grant Type	Select Client Credentials from the drop-down menu.
Folder Name	Specify the folder in which to store the credential. This single level folder structure enables you to organize credentials by grouping them in folders. <ul style="list-style-type: none"> • To save the credential in the default Common folder, leave Common selected. • To save the credential in a new folder, click the Create new link and enter a new folder name. You can't change a global credential's folder later.
Credential Name	Enter a name to identify the credential. You can't change a global credential's name later.
Target URL	Enter the base URL the connector will access in connector calls.
Client Id	Enter the OAuth client ID that you obtained while configuring the confidential application in Oracle Identity Cloud Service (IDCS).
Client Secret	Enter the OAuth client secret that you obtained while configuring the confidential application in Oracle Identity Cloud Service (IDCS).
Scope	Specify the scope you defined while creating the confidential application in IDCS. Scope sets limits on an application's access to a user's account. For example, use the scope parameter to enable the access token to grant different levels of access to Oracle IDCS APIs.
OAuth Token URL	Either select Local Identity Domain or enter the token URL from the OAuth provider. For example: <code>https://idcs-aaaaaaaa.identity.aaaaaa.aaaa.com/oauth2/v1/token</code> .

For other fields, see [Configure OAuth Security with Client Credential](#).

5. Click **Submit**.

The global credential is listed under the **Global** tab. Process Automation Designers can now find and select the global credential when creating connectors that use the same base URL as described in [Use Global Credentials](#).

Add Global OAuth Credential (JWT Assertion)

As a Process Automation Administrator, you can define a global OAuth credential of JWT assertion grant type in Workspace. In Designer, Process Automation Designers can find and use this credential while configuring connectors to interact with other applications and services.

1. From the Workspace navigation pane, click and expand **Administration**, and then click **Credentials**.

The Credentials page appears. If you're not on the Global credentials page, click the **Global** tab.

2. On the top right of the Global Credentials page, click **Create global credentials**.
3. From the drop-down list that displays, select **OAuth Credentials**.

The Add a new OAuth credential dialog displays.

4. In the Add a new OAuth credential dialog, enter the following details:

To get the information you need to complete the fields, see:

- Security, Authentication and Authorization section in *REST API for Oracle Cloud Infrastructure Process Automation*.
- [Quick Reference](#).

Field	Information to Enter
Grant Type	Select JWT Assertion from the drop-down menu.
Folder Name	Specify the folder in which to store the credential. This single level folder structure enables you to organize credentials by grouping them in folders. <ul style="list-style-type: none"> • To save the credential in the default Common folder, leave Common selected. • To save the credential in a new folder, click the Create new link and enter a new folder name. You can't change a global credential's folder later.
Credential Name	Enter a name to identify the credential. You can't change a global credential's name later.
Target URL	Enter the base URL that the connector will access in connector calls.
Username	This is auto-populated with the username with which you logged into your Process Automation instance. This user name is needed only for validation for the JWT assertion credential. It's not part of the JWT credential itself and isn't stored in the Credential service.
Client Id	Enter the OAuth client ID that you obtained.
Private Key	Enter the base-64 encoded private key.
Scope	Specify the scope you defined while creating the confidential application. Scope sets limits on an application's access to a user's account. For example, use the scope parameter to enable the access token to grant different levels of access to Oracle IDCS APIs.

Field	Information to Enter
OAuth Token URL	Either select Local Identity Domain or enter the token URL from the OAuth provider. For example: <code>https://idcs-aaaaaaa.identity.aaaaa.aaa.com/oauth2/v1/token</code> .
x5t	Specify the base64 encoded SHA-1 thumbprint of public certificate.
Description	Optionally, specify how the authentication will be used.

5. Click **Submit**.

The global credential is listed under the **Global** tab. Process Automation Designers can now find and select the global credential when creating connectors that use the same base URL, as described in [Use Global Credentials](#).

Add Global OCI Signature Policy Credential

As a Process Automation Administrator, you can define a global OCI Signature Policy credential in Workspace. Process Automation Designers can find and use this credential in Designer while configuring connectors to interact with other services and applications.

1. From the Workspace navigation pane, click and expand **Administration**, and then click **Credentials**.

The Credentials page appears. If you're not on the Global credentials page, click the **Global** tab.

2. On the top right of the Global Credentials page, click **Create global credentials**.
3. From the drop-down list that displays, select **OCI Signature Policy Credentials**.

The Add a new OCI Signature Policy credential dialog appears.

4. In the Add a new OCI Signature Policy credential dialog, enter the following details:

To get the information you need to complete some of the fields, see the **Prerequisites** section in [Configure Security with OCI Signature Policy](#).

Field	Information to Enter
Folder Name	Specify the folder in which to store the credential. This single level folder structure enables you to organize credentials by grouping them in folders. <ul style="list-style-type: none"> To save the credential in the default Common folder, leave Common selected. To save the credential in a new folder, click the Create new link and enter a new folder name. <p>You can't change a global credential's folder later.</p>
Credential Name	Enter a name for the credential. You can't change a global credential's name later.
Target URL	Enter the base URL that the connector will access in connector calls.
Tenancy OCID	Enter the tenancy OCID. See Prerequisites .
User OCID	Enter the user OCID.
Fingerprint	Enter the finger print that was generated while creating the key.
Private Key	Enter the private key.
Passphrase	Optionally, if you generated the key with the passphrase, enter the passphrase.
Description	Optionally, specify how the authentication will be used.

5. Click **Submit**.

The global credential is listed under the **Global** tab. Process Automation Designers can now find and select the global credential when creating connectors that use the same base URL as described in [Use Global Credentials](#).



Note:

A global OCI Security Policy credential cannot be re-used across different Oracle Cloud Infrastructure (OCI) services.

Edit or Delete Global Credentials

As a Process Automation Administrator, you can edit or delete global credentials in Workspace.

- Credentials are stored centrally. Editing or deleting a credential in Workspace changes or deletes the credential wherever it is used in Designer.
- Before deleting a credential, ensure that it's not being used in an activated application.

To edit or delete a global credential:

1. From the Workspace navigation pane, click and expand **Administration**, and then click **Credentials**.

The Credentials page appears. If you're not on the Global credentials page, click the **Global** tab. You can see all the global credentials created under the **Global** tab.

2. Find and select the credential you want to edit or delete.
3. Click the **Actions** menu (three horizontal dots), and then choose **Edit** or **Delete**.
4. If editing a global credential, edit the required fields in the Update credentials dialog, and click **Submit**.
 - Note that some fields such as **Folder Name**, **Credential Name**, and **Grant Type** (in case of OAuth credentials) can't be updated. Such fields are greyed out.
 - After you edit an OAuth credential of either Client Credential or JWT Assertion grant type, you can test the updates made by clicking **Validate**.

For more information about the fields in the Update credential dialog, see:

- [Add Global Basic Credential](#)
 - [Add Global OAuth Credential \(Client Credential\)](#)
 - [Add Global OAuth Credential \(JWT Assertion\)](#)
 - [Add Global OCI Signature Policy Credential](#)
5. If deleting a global credential, ensure it's not being used and confirm the deletion.

Manage Out of Office Records

Process Automation administrators can create and update out of office records on behalf of Workspace users.

When Workspace users cannot create out of office records due to unplanned leaves or update the records due to extended leaves, Process Automation administrators can create and manage out of office records so that Process Automation reassigns the tasks.

When a task is reassigned to a role, the status of the task changes to unclaimed state and anyone with the assigned role can work on the task. The new task assignee can view the comment provided by the original user in the out of office configuration settings.

 **Note:**

- The out of office feature does not work on role based task assignments.
- You cannot create multiple out of office periods that have overlapping dates. The start and end date of each out of office period must be exclusive to the dates in any other out of office period.
- Process Automation does not allow you to create an out of office period if you are assigning it to a user who has also created an out of office leave for the same period.

Configure Out of Office Records for Users

Process Automation administrators can create and update multiple out of office records for Workspace users.

1. From the Workspace navigation menu, click and expand **Administration** and then choose **Out of office**.

The Out of office page appears.

2. Select the user for whom you want to create or update out of office records.
3. In the **Start date** field, specify the starting date of the leave.
4. In the **End date** field, specify the ending date of the leave.
5. From the **Reassign to** drop-down field specify whether you want to assign the tasks to a user or a role.
6. In the **Assignee** field, enter the name of the user whom you to assign the tasks.
7. In the **Comment** field, enter the information that you want the assigned user to read.
8. Click **Submit** icon.

Enable Break Glass

Oracle Break Glass for Oracle Cloud Infrastructure Process Automation enables you to securely restrict Oracle's access to your cloud environment.

The Break Glass for Oracle Oracle Cloud Infrastructure Process Automation feature is enabled if you have a Oracle Cloud Infrastructure Process Automation instance that is paired with a Fusion-based Oracle Cloud Applications subscription that includes Break Glass.

When you use Break Glass, Oracle Support representatives can access your cloud environment only after relevant approvals and authorization to troubleshoot any issues that may arise in your cloud environment.

Break Glass has these primary features:

- **Temporary access approval**, in which Oracle personnel can only access instance data through a strict customer approval process. Typically, such a process would only be initiated to help resolve a customer service request.

Such access is time limited. Any temporary access credentials are automatically reset after the agreed upon time.

Such access is logged and detailed reports are available.

- **The option to provide your own Transparent Data Encryption (TDE) master encryption key.**

By default, your data in the Oracle Cloud environment is encrypted at rest using TDE.

With Break Glass, you can upload your own TDE master encryption key and manage its lifecycle. If you provide your own key, your data will also be protected and audited using Data Vault. You can also periodically update the keys.

Temporary Access Approval

If you submit a service request (SR) and Oracle Support determines that it needs access to some of your data for debugging purposes, you can agree to give them temporary access to your service instance data. Here's the general flow of the process:

1. You submit an SR.
2. If Oracle Support determines that they need access to any of your data for debugging purposes, they will contact your administrator via email for approval to conduct a Break Glass session. (The email has a link to the Temporary Access Approval page of your Oracle Cloud Infrastructure Process Automation, where your administrator can click **Approve** or **Reject**.)
3. If your administrator approves the request, a temporary password is generated to enable Oracle Support to start a Break Glass session, in which they can access the required data.
4. Once Oracle support completes its work in the Break Glass session, they terminate the session. If they don't explicitly terminate the session, it expires automatically within the timeframe that you have agreed upon.

Provide Your Own Key

By default, Oracle provides and manages the TDE keys for encrypting the data in your Oracle Cloud Infrastructure Process Automation instance.

If your instance has Break Glass enabled, you can also replace the Oracle-provided private key with your own, which also enables you to rotate the keys as you require.



Note:

When you first switch to using your own key, you need to allow some time for your instance to be out of service. You should also back up any key artifacts in your instance.

- [Create and Import Your TDE Master Key](#)
- [Update the Key](#)

Create and Import Your TDE Master Key

To provide your own key, follow these steps:

1. In Workspace, open the navigation menu, and expand **Administration**. Click **Break Glass**.
2. On the Provide Your Own Key page, click **+ Provide Your Own Key**.
3. In the Add Key dialog, click **Public Key**. Copy and note the key value that is displayed.
4. Use OpenSSL to generate and encrypt your key:
 - a. Create a new directory for the key and assign it to an environment variable:

```
mkdir -p dir_of_key
```

```
export KEYPATH =dir_of_key
```

```
touch $KEYPATH/wrappingkey
```

Paste the public key value that you obtained in step 3 in the wrapping key.

- b. Make sure the directory is restricted:

```
chmod go-rwx $KEYPATH
```

- c. Generate the TDE master key:

```
openssl rand 32 > $KEYPATH/clearkey
```

- d. Encrypt your generated TDE master key with the Oracle public wrapping key that you downloaded in step 3:

```
openssl pkeyutl -encrypt -in $KEYPATH/clearkey -inkey $KEYPATH/  
wrappingkey -pubin -pkeyopt rsa_padding_mode:oaep -pkeyopt  
rsa_oaep_md:sha256 > $KEYPATH/wrappedkey
```

- e. Encode wrappedkey in Base64:

```
base64 $KEYPATH/wrappedkey > $KEYPATH/wrappedkey1
```

- f. Remove new line characters from wrappedkey1:

```
perl -p -i -e 's/\R//g;' $KEYPATH/wrappedkey1
```

- g. Rename the file back to wrappedkey:

```
mv $KEYPATH/wrappedkey1 $KEYPATH/wrappedkey
```

5. In the **External Key Data Source** field, upload the encrypted TDE master key (`wrappedkey`, as in the above example).
6. In the **Email Address** field, enter the email address of the person to notify when the reconfiguration of the Oracle Cloud Infrastructure Process Automation instance has finished and the instance is ready to be used again.
7. Click **Submit** and then **Confirm**.

Update the Key

If you have previously provided your own TDE key for your Oracle Cloud Infrastructure Process Automation instance, you can update that key.

1. In Workspace, open the navigation menu, and expand **Administration**. Click **Break Glass**.
2. On the Provide Your Own Key Page, click **+ Update the key**.
3. Click **Public Key** to download the Oracle public wrapping key that you will need to encrypt your own transparent data encryption (TDE) master key.
4. Use OpenSSL to generate and encrypt your key:
 - a. Create a new directory for the key and assign it to an environment variable:

```
mkdir -p dir_of_key
```

```
export KEYPATH dir_of_key
```

- b. Make sure the directory is restricted:

```
chmod go-rwx $KEYPATH
```

- c. Generate the TDE master key:

```
openssl rand 32 > $KEYPATH/clearkey
```

- d. Encrypt your generated TDE master key with the Oracle public wrapping key that you downloaded in step 3:

```
openssl pkeyutl -encrypt -in $KEYPATH/clearkey -inkey $KEYPATH/  
wrappingkey -pubin -pkeyopt rsa_padding_mode:oaep -pkeyopt  
rsa_oaep_md:sha256 > $KEYPATH/wrappedkey
```

5. In the **External Key Data Source** field, upload the encrypted TDE master key.
6. Click **Submit** and then **Confirm**.

 **Note:**

Once you create or update your key, you have to wait 16 days or more before you can update it again.

Specify Data Management Settings

As a Process Automation Administrator, you can set the number of days that data is retained for process instances that are in completed or closed state before it is purged.

1. In Workspace, open the navigation menu and expand **Administration**. Click **Data Management Settings**.

The Data Management Settings page displays.

2. In the **Purge retention days** field under Purge data settings, enter the number of days that data should be retained for completed process instances.
 - By default the number of retention days for data of completed process instances is 180.
 - The number of retention days that you specify can be any number that is equal to or greater than 1.
 - Data for completed process instances will be automatically purged after the retention days specified in the Purge data settings.
3. Click **Save**.
4. If you want to reset the purge retention days, then click **Reset** and enter a new number.

Register Services

You can register a new service and view already registered services from the Registered Services page in Workspace. When you register a service, Process Automation (client) can access the OAuth protected REST API endpoints of the service (resource) and start communicating with the service.

You have to be a Process Automation Administrator to access the Registered Services page in Workspace.

If there are services which are already registered with your Process Automation instance or associated with your Process Automation instance from the Oracle Cloud Infrastructure (OCI) console, then those services are listed on the Registered Services page. You can view details about a service such as the name, description, and date on which the service was created.

Note the following:

- Services that are registered from the Registered Services page in Workspace can be edited or deleted. Select the service, click the **Options** icon and then click **Edit** to edit its settings. You can update the name, description, and client credentials of the service. Click the **Options** icon and click **Delete** to delete the registered service.
- Services that are associated with your Process Automation instance from the OCI console can only be viewed but can't be edited or deleted from the Registered Services page in Workspace. Associated services are listed as *attached* services.

Before registering a service, you need to complete certain prerequisite tasks. After you've completed the prerequisites, you can register the service from Workspace.

- [Prerequisites for Registering a Service](#)
- [Register a New Service](#)

Prerequisites for Registering a Service

To register a service from Workspace and to access and call the REST API endpoints of the service, you have to complete certain prerequisite tasks.

Log in as an administrator to identity domain or Oracle Identity Cloud Service (IDCS) admin console in Oracle Cloud Infrastructure (OCI) and complete the following tasks.

1. [Create and configure an application](#)
2. [Assign the application to the ServiceDeveloper role](#)
3. [Get the client ID, client secret, and scope to register the service](#)

Create and configure an application

You have to create an application and configure OAuth settings including grants and scope for authentication and access management, and then activate the application.

The following steps are for creating and configuring an application from the identity domain.

1. Navigate to the Applications page.
2. On the Applications page, click **Add application**.
3. In the Add application dialog, select **Confidential Application**, and click **Launch workflow**.
4. Enter the application details such as name (for example, `OPAAuth`) and description, and click **Next**.
5. In the Configure OAuth page, select **Configure this application as a client now**.
6. Under **Allowed grant types**, select **Client credentials**. You can also select the **Resource Owner** and **JWT Assertions** grant types.
7. Under **Client IP Address**, select **Anywhere**.
8. Now, add the scopes of the service that you want to register to this application.

For example, if you want to register Oracle Integration, add the scopes specified in the Oracle Integration IDCS application instance to the application you are creating.

Adding scopes for the service you want to register enables the client application (that you're creating) to obtain access tokens against the scopes specified, and thus call and access the service instance's APIs.

- a. Under Resources, click **Add scope**.

In the resulting Add scope window, IDCS applications of provisioned services are displayed.

- b. Search for the IDCS application instance of the service you want to register.

For example, if you want to register Oracle Integration then search for the IDCS application of the Oracle Integration service instance.

- c. Click the expand arrow next to the application instance to display the scopes configured for it.

- d. Select the required scopes.

- e. Click **Add**.

Note that if you want to select all the scopes configured for the application, select the application and click **Add**.

9. Click **Next**.
10. Click **Finish**.
11. Activate the application. Click **Activate** and when prompted click **Activate application**.

The activated application is displayed on the Applications page.

Assign the application to the ServiceDeveloper role

In the IDCS application of the service that you want to register, assign the application that you created to the ServiceDeveloper role.

1. Navigate to the Oracle Cloud Services page.

2. Search for the IDCS application of the service instance that you want to register. For example, if you want to register Oracle Integration, search for the IDCS application of the Oracle Integration service instance.
3. Select the application to open the application details page.
4. Under **Resources**, click **Application roles**.
5. Under the **Application roles** section, select the **ServiceDeveloper** role and click the expand arrow.
6. Next to **Assigned applications**, click the **Manage** link.
7. In the Manage application assignment window, click the **Show available applications** link.
8. Search for the application (OPAAuth) that you created.
9. Select the application and click **Assign**.
10. Click **Close**.

Notice that when you expand the **ServiceDeveloper** role under the **Application roles** section, you'll see a number (1) besides **Assigned applications**, indicating that an application is assigned to the role.

Get the client ID, client secret, and scope to register the service

Now that you've created, configured, and activated an application, and assigned the ServiceDeveloper role to the application, you can obtain the client ID, client secret, and scope for registering the service from Workspace.

1. Navigate to the Applications page.
2. On the Applications page, search for the application (OPAAuth) that you created.
3. Select and click the application to open its details page.
4. Under **OAuth configuration**, scroll to the **General Information** section.
5. Copy and note the client ID.
6. Click the **Show secret** link and then copy and note the client secret.
7. Scroll to the **Token issuance policy** section, and copy the scope under **Resources**.

You're all set to register the service now from Workspace. See [Register a New Service](#).

Register a New Service

After you've completed the prerequisite tasks for the new service in the Oracle Cloud Infrastructure (OCI) console, you can register the service from Workspace.

1. In Workspace, open the navigation pane and expand **Administration**. Click **Registered Services**.

The Registered Services page displays.

2. Click the **Register new service** drop-down.

From the **Integration Version** drop-down, select **Oracle Integration 3** or **Oracle Integration 2**. Available cloud services that you can register with your Process Automation service are displayed.

3. Choose the service. For example, Oracle Integration.
4. Enter the following information to register the service.

Field	Information to Enter
Name	Enter a name for the service.
Description	Enter the description of the service.
Service URL	Enter the service URL.
Service OCID	Enter the OCID value that uniquely identifies the service instance. Obtain the service OCID from the service instance details page in the OCI console.
OAuth Token URL	Enter the OAuth token URL. For example: <code>https://<idcs_URL>oauth2/v1/token</code>
Client Id	Enter the client ID that you obtained. See Prerequisites for Registering a Service .
Client Secret	Enter the client secret that you obtained.
Scope	Enter the scope that you obtained.

5. Click **Register**.

The service now gets displayed as a registered service on the Registered Services page.

Once the service is registered, you can start using the service with Process Automation. For example, if you have registered an Oracle Integration service instance, then you can view integrations that are configured in Oracle Integration and start using REST based activated integrations into your process applications. See [Work with Integrations](#).

A

Quick Reference

The steps and commands below provide quick reference on how to get the required information for creating a trusted OAuth client and working with it. The information below can also be found through Google searches.

Generate a Key Pair Using OpenSSL

Use the command below to generate a public and private key ready for use:

#note: we can also provide a validity period e.g -days 365

```
openssl req -newkey rsa:2048 -nodes -keyout private_key.pem -x509 -out myPublicCertificate.crt
```

Generate a Key Pair Using Java Keytool

1. Generate the key store.

#note: we can also provide a validity period e.g -days 365

```
keytool -genkey -alias keyalias-keyalg RSA -keystore mykeystore.jks -storepass password -keysize 2048
```

you can optionally provide the keystore type (JKS or PKCS12 format) - (for example -storetype pkcs12)

2. Extract the public certificate and private key from the key store. (For all operations, provide the Import password when prompted.)

a. Extract the public key certificate from the JKS.

```
keytool -export -alias keyalias -keystore mykeystore.jks -rfc -file public.cert
```

b. To export the private key from JKS, convert the JKS to PKCS12 format.

```
keytool -v -importkeystore -srckeystore mykeystore.jks -srcaalias keyalias -destkeystore myp12file.p12 -deststoretype PKCS12
```

```
#extract private keyopenssl pkcs12 -in myp12file.p12 -nodes -nocerts -out private_key.pem
```

c. Extract the private key from the PKCS12 Keystore.

```
openssl pkcs12 -in myp12file.p12 -nodes -nocerts -out private_key.pem
```

d. Extract the public certificate from the PKCS12 Keystore.

```
openssl pkcs12 -in myp12file.p12 -nokeys -out myCertificate.crt
```

Obtain an x5t thumbprint of a public certificate

x5t is a base64 encoded SHA-1 thumbprint of a public certificate. It can be obtained from a .cert file using an openssl command. It is used to identify specific certificates.

```
echo $(openssl x509 -in mycertificate.crt -fingerprint -noout) | sed 's/SHA1  
Fingerprint=//g' | sed 's://g' | xxd -r -ps | base64
```

Configure a Trusted OAuth Client in Oracle Identity Cloud Service

Below is a summary of steps to define a trusted OAuth client in Oracle Identity Cloud Service. See the Oracle Identity Cloud Service documentation.

1. Sign in to Oracle Identity Cloud Service using administrator credentials.
2. Click **Add**.
3. Select **Confidential Application**.
4. Enter an application name and click **Next**.
5. Select the option **configure this application as client now**.
6. Enable **JWT Assertion** and other required grant types.
7. Select the **Trusted Client** checkbox and import a public certificate (refer to earlier section for details on how to obtain a public certificate).
8. Click **Add Scope** and add the necessary scopes to your OAuth client.
9. Click **Next**. Click **Next** again and click **Finish**.
10. Activate the application.