



SAP Build Process Automation

Generated on: 2024-02-26 21:19:51 GMT+0000

SAP Build Process Automation | Cloud

PUBLIC

Original content: https://help.sap.com/docs/PROCESS_AUTOMATION/a331c4ef0a9d48a89c779fd449c022e7?locale=en-US&state=PRODUCTION&version=Cloud

Warning

This document has been generated from the SAP Help Portal and is an incomplete version of the official SAP product documentation. The information included in custom documentation may not reflect the arrangement of topics in the SAP Help Portal, and may be missing important aspects and/or correlations to other topics. For this reason, it is not for productive use.

For more information, please visit the <https://help.sap.com/docs/disclaimer>.

What Is SAP Build Process Automation?

SAP Build Process Automation is a citizen developer solution to adapt, improve, and innovate business processes with no-code workflow management and robotic process automation capabilities.

SAP Build Process Automation enables business users and technologists to become citizen developers. With powerful yet intuitive low-code and no-code capabilities, the solution supports you in driving automation by tapping into the expertise of citizen developers.

Features

The solution offers the following key features:

- Build or adapt processes with an intuitive graphical interface.
- Create forms-based workflows using drag-and-drop functionality.
- Develop and manage decision logic in tabular, spreadsheet-like decision tables.
- Automate repetitive tasks within existing process flows using robotic process automation.
- Create intelligent actions and recommendations using machine learning capabilities.
- Work efficiently from a unified launchpad and task center.
- Hand over projects to professional developers, who can embed actions and advanced workflows into projects initiated by citizen developers.
- Support real-time, event-driven transparency into comprehensive processes and process instances with process visibility dashboards.

In addition, SAP Build Process Automation offers prebuilt content and features – such as bots, process steps, business rules, and workflow components.

Related Information

[Supported Languages](#)

[Quotas, Restrictions, and Limits](#)

[Known Limitations](#)

Use SAP Build Process Automation

SAP Build Process Automation is an SAP BTP service that allows you to create, run, automate, and monitor your business processes on one interface using low-code/no-code capabilities.

You can subscribe to SAP Build Process Automation using either the standard (paid) or free plan, with active subscriptions added to your SAP BTP subaccount.

Boosters

Follow our interactive guided boosters to **build** applications or use different platform services and features.

SAP Build Process X 🔍 All ▼

Extension Suite - Digital Process Automation (2)

Set up account for SAP Build Process Automation (Free)

Automated setup for SAP Build Process Automation in your account.

[Start](#)

Set up account for SAP Build Process Automation

Automated setup for SAP Build Process Automation in your account.

[Start](#)

For more information about subscribing to SAP Build Process Automation and accessing the service, see [Initial Setup](#).

And for more information about business processes and the role of citizen developers, take our SAP Learning journey:

[Utilize Low-Code/No-Code Applications and Automations for Citizen Developers](#).

Already subscribed to SAP Build Process Automation?

Once subscribed to SAP Build Process Automation, you can create, manage, and deploy projects using the Lobby.

For more information, see [Create and Manage Projects](#).

Create and Manage Projects

After subscribing to SAP Build Process Automation, you create and manage projects in the lobby.

This lobby includes projects you've created, those shared with you, or that you've imported from external sources or the store.

The screenshot shows the SAP Build interface. On the left is a sidebar with navigation links: Lobby, Connectors (Actions, Automation SDK), Store, Monitoring, and Control Tower. The main area has a purple header "Welcome to SAP Build" with the sub-header "Create apps, automate processes, and build business sites using productivity or no-code tools." Below this is a "Quick Start" section with three cards: "Access our SAP Build Learning Journeys" (Learning), "Create a Change and Innovation Approval Process" (Template), and "Create an Invoice Approval Process" (Template). The main content area shows "All Projects (2)" with a table:

Name	Versions	Type	Last Accessed	Members	Options
Supplier Approval and Creation in SAP S/4HANA Supplier Approval in SAP S/4HANA		Process Automation	Oct 26, 3:02 pm	Me	...
Job Offers Approvals Automates approval process of job offers pending in SAP SuccessFactors system.		Process Automation	Oct 19, 12:22 pm	Me	...

i Note

User principles are case insensitive. For example, “user@mail.com” is treated as the same entity as “USER@mAIL.cOM”. Therefore, sharing a Project/Package with them is transparent.

Project Types

With an SAP Build Process Automation subscription, you can create and manage the following project types:

- **Business Process Project** - Create, deploy, automate, and run digital business process by configuring process artifacts. Examples of business process projects include: Investment requests, invoice approvals, and sales order approvals.
- **Actions Project** - Embed external actions and capabilities into your business process projects by uploading an open API specification file in JSON format. Actions projects allow external systems and solutions to communicate with SAP Build Process Automation.

Create a Business Process Project

When creating a business process project, choose **Create** **Build an Automated Process** **Business Proces** to start from scratch. This creates an empty project, allowing you to define your own content.

The screenshot shows the SAP Build interface with a search bar and a prominent blue "Create" button highlighted with a red box. Below the search bar, there are two sections: "Last Accessed" (Nov 11, 4:54 pm) and "Members" (Everyone). A bookmark icon and three dots are also present in this section.

Alternatively, you can take advantage of the **Quick Start** project options. This creates the project container and prompts you to add the relevant steps for the project.

For more information about business process projects and skills, see [Business Process Projects](#)

This is custom documentation. For more information, please visit the [SAP Help Portal](#)

Create Actions Project

To create an actions project, choose **Connectors > Actions > Create > Choose an API Source** and then enter the necessary details.

For more information about Actions project, see [Create an Action Project](#).

Manage Existing Projects

For existing projects, choose **More Options** to see further project management abilities. These abilities depend on the permissions you hold and can include releasing, publishing, renaming, sharing, and deleting your projects.

Last Accessed	Members	
Nov 8, 1:13 pm	2 members	... <div style="background-color: #fff; border: 1px solid #ccc; padding: 5px; margin-top: 5px;"> Release Publish to Library Export Rename Project Manage Members Delete </div>
Oct 27, 5:41 pm	Everyone	

For more information about your project management options, see [Manage Existing Projects](#).

Business Process Projects

In SAP Build Process Automation, you create a business process project using a combination of one or more process focused skills or building blocks. These skills are known as **artifacts**.

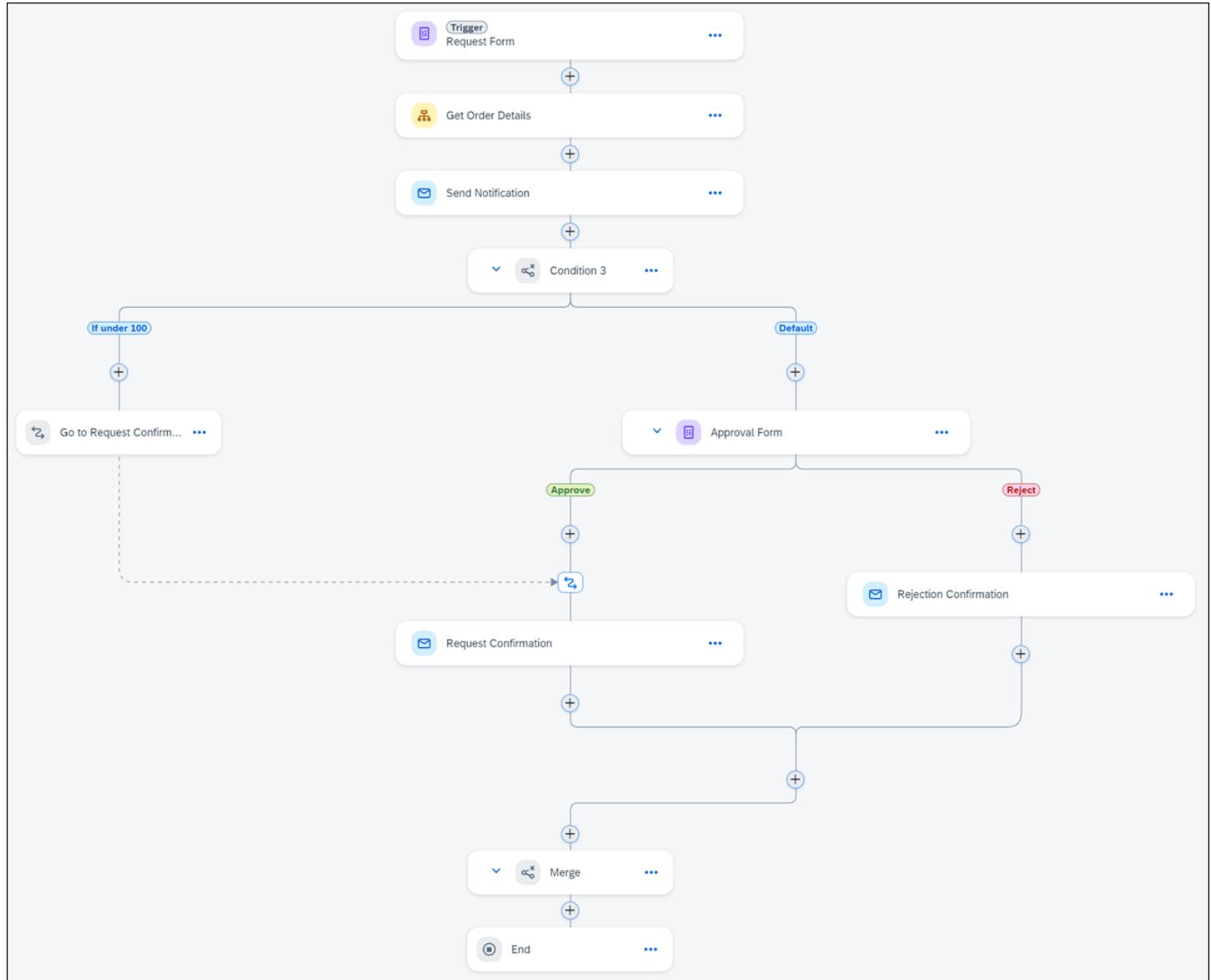
Artifacts help you to configure common steps or tasks needed for a process to run successfully. And the number and order of the artifacts you use depend on the process you're creating. As such, there are no mandatory artifacts needed in a business process project.

To help explain the artifacts available in SAP Build Process Automation, we'll divide them into two categories: Process and Automation

Process Artifacts

These include: Process editor, forms, approval forms, automations, decisions, and visibility scenarios

In this example, process artifacts have been used to create an investment request process:

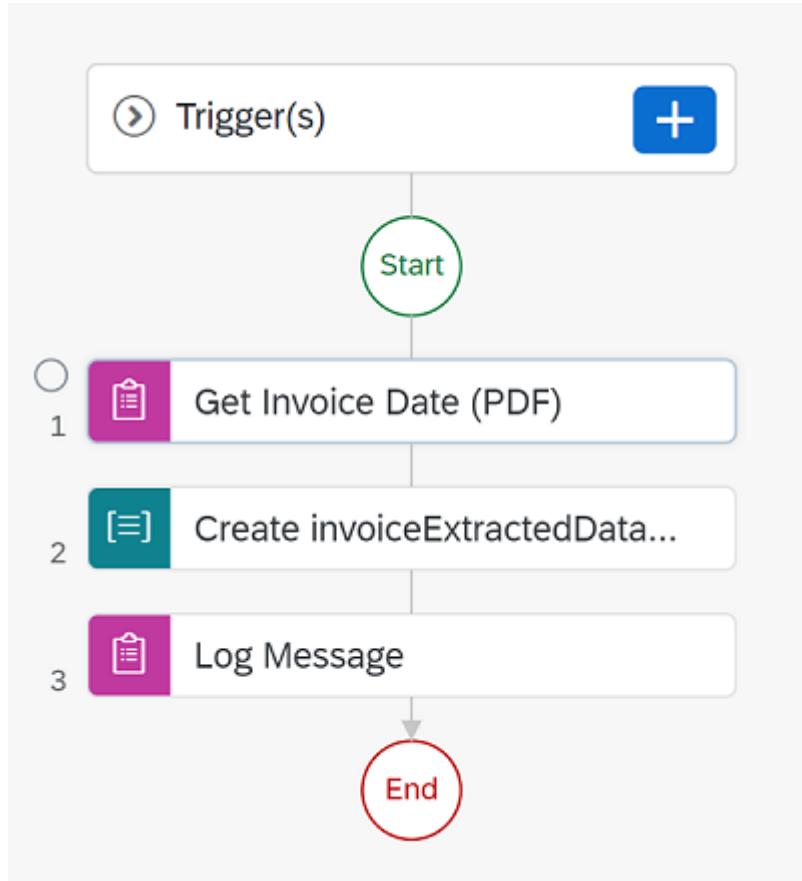


For an overview of available process artifacts, see [Process Artifacts](#).

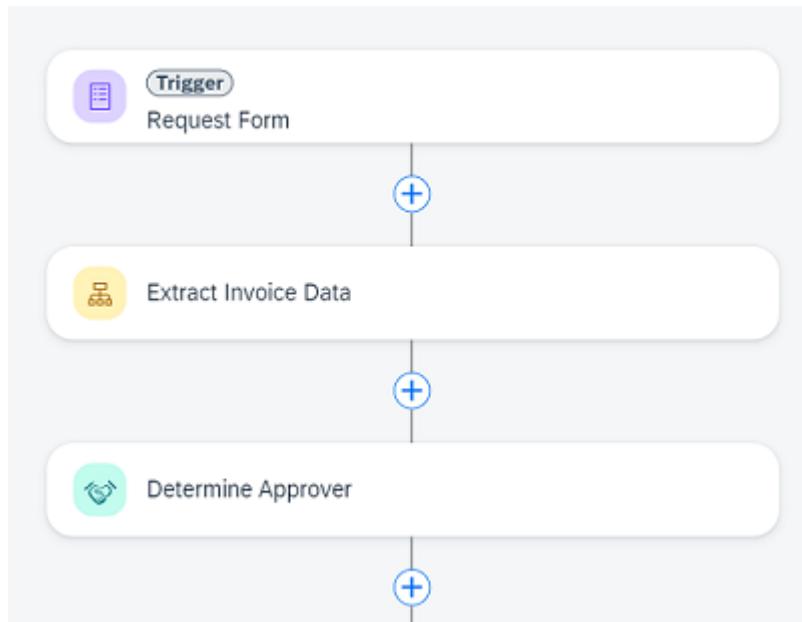
Automation Artifacts

These include: Alerts, applications, project launchers, user tasks, data types, files, and document templates

In this example, automation artifacts have been created to extract data from a PDF:



With this automation then added to a process:



For an overview of available automation artifacts, see [Automation Artifacts](#).

Process Artifacts

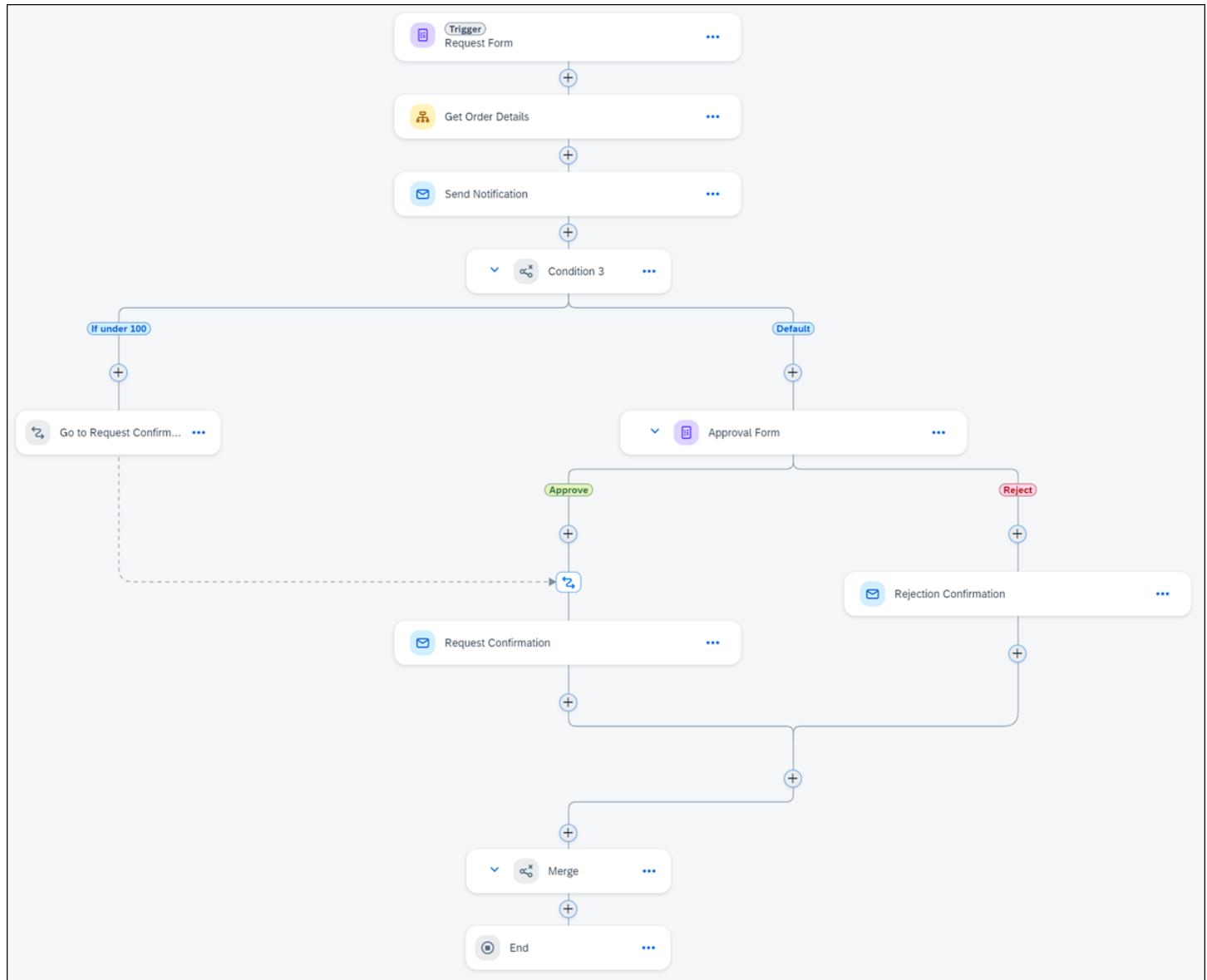
Process artifacts allow you to both create a multistage process and to configure how that process runs and is monitored.

Process Editor

This is a visual canvas on which you map out your business process from start to finish. Other process artifacts are then added to this canvas, with process controls and connectors used to decide how information flows when the process is running.

This is custom documentation. For more information, please visit the [SAP Help Portal](#)

For example, the process editor was used to create this basic investment request process:



This canvas view is the ‘behind the scenes’ of the process and isn’t shown to a person participating in that process.

For more information about using the process editor, see: [Create a Business Process](#)

Forms

Interactive forms are created by adding text elements and input fields to a blank page. Forms can be both starting triggers for a live process or used as optional steps later in the same process.

In the following example, a request form has been created:

The screenshot shows the SAP Build interface with a 'PR Creation Request' process step selected. On the left, there's a sidebar with various icons and a list of process steps. The main area displays a 'Purchase Requisition Creation Request' form. It includes a 'Requestor Information' section with fields for Name and Email, and a note stating: 'Please fill in the details to request the approval and creation of a purchase requisition for IT equipment.'

Published forms are often how someone submitting a request interacts with the process. Here they access a URL or a SAP Build Work Zone tile, enter their request details, then click 'Submit'.

For more information about creating and adding forms to your process, see [Create a Form](#)

Approval Forms

Based on the information submitted in the request form, approval forms share the request with those responsible for reviewing them. The assigned approvers for the process then decide whether to approve, query, or reject the request. Submitted requests appear as tasks in the approver's inbox, helping them to manage requests in one system.

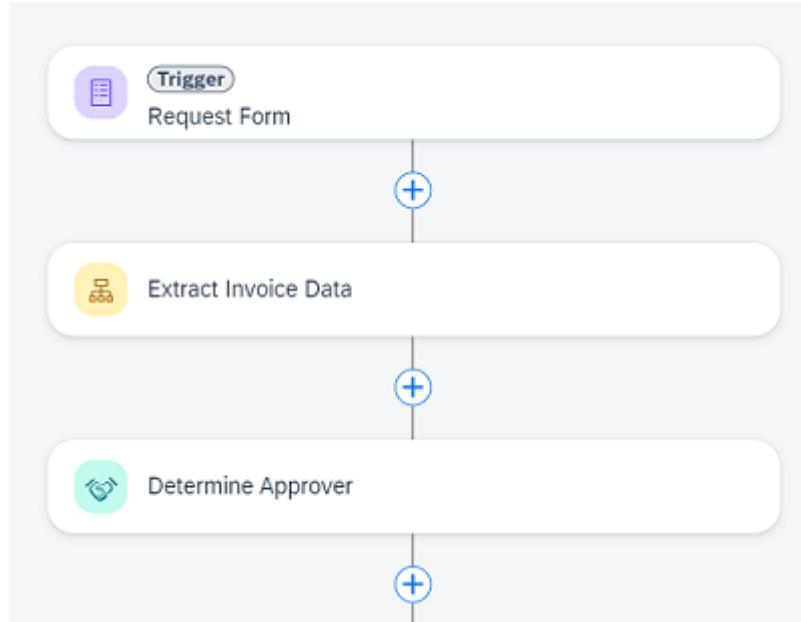
The screenshot shows the SAP Build interface with a 'PR Approval Form' task in the 'All Tasks' list. The task details show a 'Purchase Requisition Approval' request from 'tom.' with a priority of 'High'. The main panel displays the 'Purchase Requisition - Approval' form with fields for Requestor Name (Tom), Material (IPHONE-12), Quantity (1), Requestor Comments (Can you please approve this request?), and Manager Comments (an empty text input field).

For more information about creating and configuring approval forms, see [Create an Approval Form](#)

Automations

The use of intelligent bots to automate manual, repetitive tasks such as retrieving data from a spreadsheet or submitting information to a database. Automations can be both included in a business process, and configured using the process editor, or used as stand-alone projects. That's why you'll find them mentioned here, too.

In the following example, an automation that extracts invoice data has been added to a process:

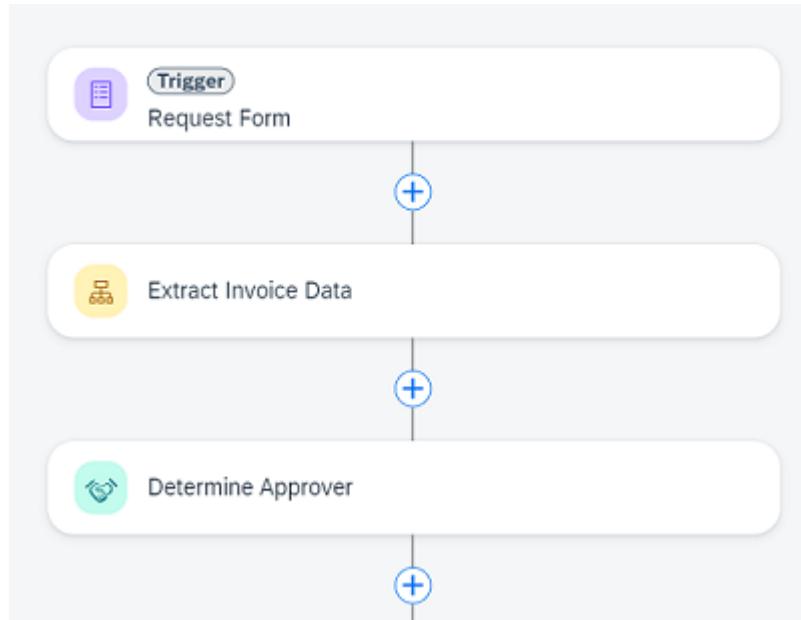


For more information about creating automations, see [Create and Design Automations](#)

Decisions

Effective processes respond and react to the information available. The way they behave in SAP Process Automation, and the direction the process moves, is based on adding and configuring process decisions. For example, if a request comes in from someone in Location A, then root that process to Location A's purchasing team.

In this example process, a decision determines who should approve the submitted request:



Using decisions, one process can effectively handle many scenarios at once.

For more information about creating and configuring process decisions, see [Create a Decision](#)

Visibility Scenarios

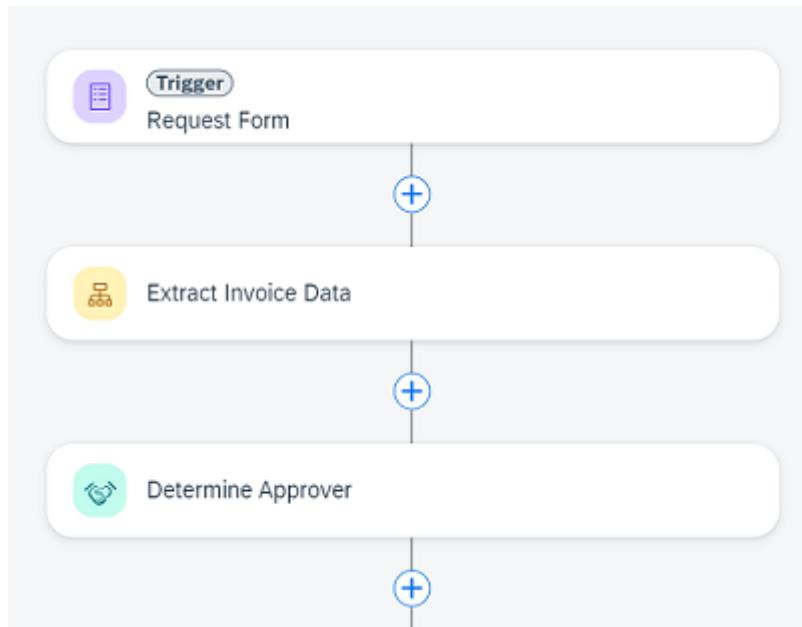
While not added to the process diagram, a visibility scenario allows you to configure and then monitor live versions of the process. By selecting the processes and choosing the type of information displayed, you have greater insights into the efficiency of your process (and react where necessary).

For more information about configuring visibility scenarios for your process, see [Configure a Visibility Scenario](#)

Automation Artifacts

Automation artifacts can be used to create and run an automation. These artifacts are created in the project overview area and can then be added to and configured in an existing automation. As such, they can't be used independently in your business process.

In this example, an artifact has been created to extract information from an invoice:

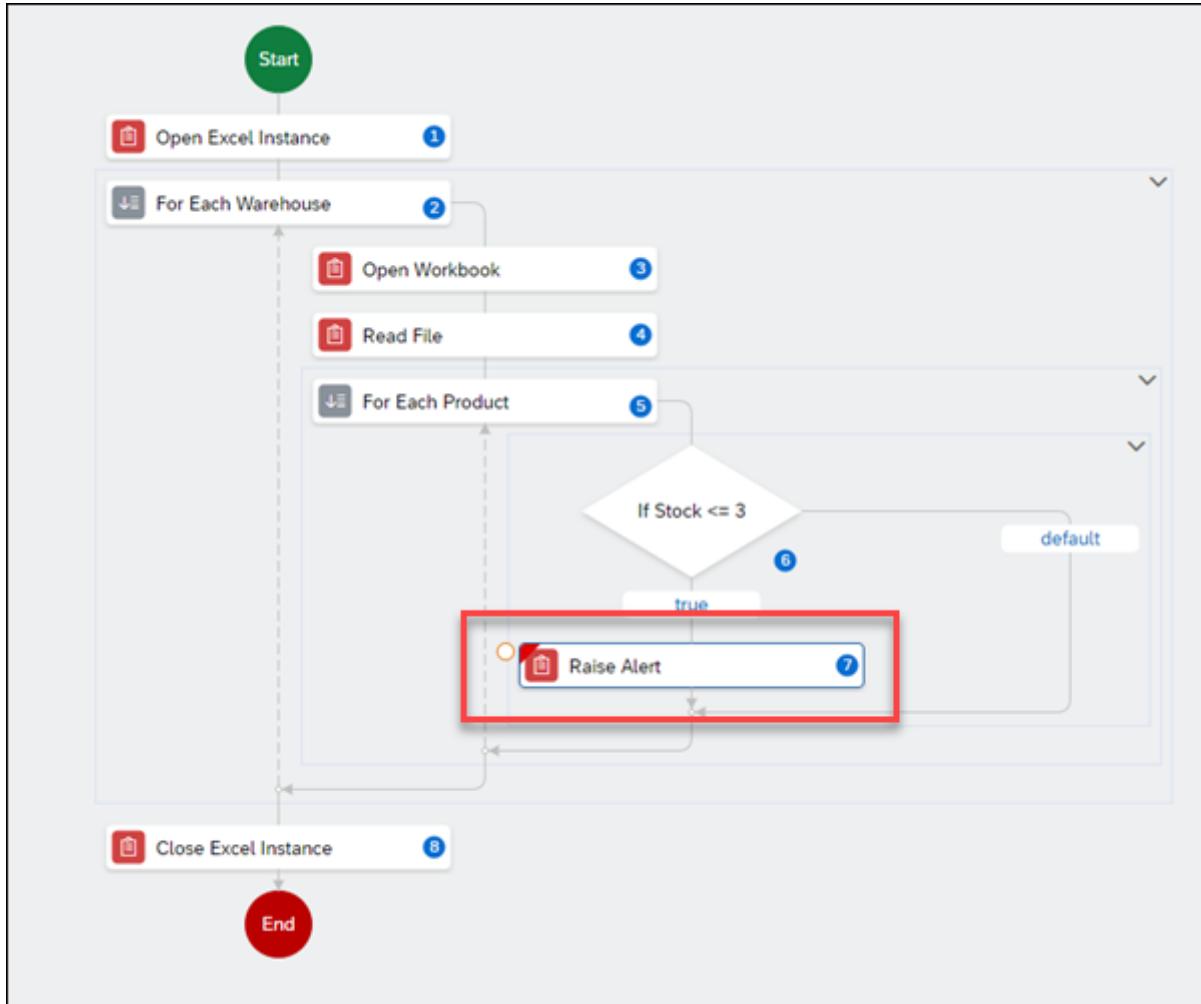


As you can see, an automation can be both a stand-alone project and included in a wider process. In both situations, there are several automation focused artifacts available in SAP Build Process Automation that help to build more complex, intelligent bots.

Alerts

These identify when specified business events occur while an automation is running. For example, when an item is out of stock or a limit has been reached. Alerts are then enhanced using alert handlers. These trigger an email to be sent to the people or system who need to know this information.

In this example, an alert has been created to notify those responsible when stock level has fallen under 3 units:



For more information about adding alerts to your automation, see [Create an Alert](#)

Applications

This artifact allows you to identify applications to control during a live automation, such as a website or software. Known as capturing and declaring applications, the automation can be configured to recognize a screen and interact with the elements displayed.

The screenshot shows the SAP Application Development interface with the 'Select Screen' feature open. On the left, a sidebar lists various open screens, including 'Candidates.xlsx - Excel'. The main area displays an Excel spreadsheet titled 'Candidates.xlsx - Excel' with data for four individuals: Paul Smith, Alice Norton, and John Barker. The 'Screen Details' panel on the right provides options for capturing the application, with 'Capture Application' selected. It also includes fields for 'Application Name' (set to 'Candidates.xlsx - Excel'), 'Application Identifier' (set to 'candidatesxlsxExcel'), and 'Technology' (set to 'Excel sheet listing candidates'). A note in the panel states: 'There might be a specific Excel activity for your use case. If so, you do not need to capture it.'

Project Launcher

Attended automations (those run under human supervision) can be started from a 'systray' agent or system tray agent on your device. By designing a project launcher, you configure which automations can be started (or launched) from that agent. These can either be run manually or automatically based on meeting certain factors.

In this example, the project launcher has been configured to run three automations from the systray:

The screenshot shows the SAP Fiori interface for a 'Project Launcher'. The top navigation bar includes 'Overview' and the current page title, '*Project Launcher'. Below the header, there are two main sections: 'Launch manually from the agent' and 'Launch automatically by events'.

Launch manually from the agent:

Automation name	Agent label
<input type="checkbox"/> Extract Data	Extract Data

Launch automatically by events:

Automation name	Agent label	Element	Event	Action
<input type="checkbox"/> Open Excel			0	<button>Add event</button>
<input type="checkbox"/> Populate Form			0	<button>Add event</button>

For more information about creating a project launcher, see [Create a Project Launcher](#)

User Task

User tasks take the form of messages received in a person's inbox while a process is running. When creating a user task, you define in a message a specific action that needs to be done by the recipient. For example – to enter personal information, attach a file, or choose from a list of options.

For more information about creating user tasks for your automations, see [Create a User Task](#)

Data Type

Most automations and business processes require data to be inputted, stored, and shared to successfully run. This data always depends on the specific process or those running it, however. As such, there are different data types for different circumstances. By manually creating data types, you can configure and use the data needed for your specific process.

In this example, three data types have been created to be used in an automation:

The screenshot shows the SAP Build Process Automation interface with the 'Investment Details' data type selected. On the left, there's a table with columns: Name, Type, Sample, List, and Required. Three fields are listed: 'InvestmentType' (String), 'Country' (String), and 'BusinessUnit' (String). Each row has a 'New Child' button. On the right, there's a panel titled 'Data Type Details' with sections for General Information, Identifier, Description, and checkboxes for 'Data type is active' and 'Strict'.

Name	Type	Sample	List	Required	
InvestmentType	String		No	No	<button>New Child</button>
Country	String		No	No	<button>New Child</button>
BusinessUnit	String		No	No	<button>New Child</button>

Data Type Details

General Information

Name: *

Identifier:

Description:

Data type is active

Strict

For more information about creating and maintaining data types, see [Create a Data Type](#)

Files

Files can be both created and edited in SAP Build Process Automation (text, YAML, JSON, and XML) or uploaded and stored within a project. These files can then be integrated and used within an automation, too.

Document Template

Information can be extracted from a document while an automation is running. For the automation to do this, a document template can be created or uploaded to the project. This template is then used to identify the data available and extract it when directed to do so. For example – retrieving customer information from a frequently used invoice layout.

For more information about working with document templates, see: [Document Processing and Information Extraction](#)

Manage Existing Artifacts

In SAP Build Process Automation, you create a business process project using a combination of one or more process focused skills or building blocks. These skills are known as **artifacts**. Once created, you can manage your artifacts from your business process project overview page.

All Artifacts			Search...		Create	Import
Name	Description	Type	Last edited	Last edited	Created On	
alert	No value	Alert	21 days ago	tom.b...	November 18, 2022	...
autom...	No value	Automation	18 days ago	tom.b...	November 18, 2022	
Employ...	No value	Data Type	18 days ago	tom.b...	November 18, 2022	
form	No value	Form	21 days ago	tom.b...	November 18, 2022	
Process	No value	Process	4 days ago	tom.b...	November 18, 2022	...

Options

When managing your artifacts, you've the following options:

Artifact	Create	Read	Update	Delete	List	Rename	Activate / Deactivate	Copy / Paste / Duplicate	Import Inside
Action Project	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes
Alert	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	N/A
Application	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Partly (Excel files only)
Data Type	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	N/A
Decision	Yes	Yes	Yes	Yes	Yes	Yes	No	No	Yes
Document Template	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes
File	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes
Form	Yes	Yes	Yes	Yes	Yes	Yes	No	Yes	N/A
Approval Form	Yes	Yes	Yes	Yes	Yes	Yes	No	Yes	N/A
Process	Yes	Yes	Yes	Yes	Yes	Yes	No	No	N/A
Project Launcher	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Only Copy / Paste	N/A
User Task	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	N/A
Visibility Scenario	Yes	Yes	Yes	Yes	Yes	Yes	No	No	N/A

Using Artifacts Within Other Artifacts

Many artifacts can be used within existing artifacts, such as forms being used within a process.

i Note

If you modify or delete an artifact used within other artifacts, this can affect the validity of these artifacts.

The following table summarizes the available options:

	Action Project	Alert	Application	Automation	Data Type	Decision	Document Template	File	Form	Approval Form	P
Automation	Yes	Yes	Yes	Yes	Yes	N/A	Yes	Yes	N/A	N/A	N
Data Type	N/A	N/A	N/A	N/A	Yes	N/A	Opposite	N/A	N/A	N/A	N
Decision	N/A	N/A	N/A	N/A	Yes	N/A	N/A	N/A	N/A	N/A	N
Document Template	N/A	N/A	N/A	N/A	Yes	N/A	N/A	N/A	N/A	N/A	N
Form	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N
Approval Form	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N
Process	Yes	Yes (via an automation)	Yes (via an automation)	Yes	Yes	Yes	Yes	No	Yes (via an automation)	Yes	Y
Project Launcher	N/A	N/A	Yes	Yes	N/A	N/A	N/A	N/A	N/A	N/A	N
User Task	N/A	N/A	N/A	N/A	Yes	N/A	N/A	N/A	N/A	N/A	N
Visibility Scenario	N/A	N/A	N/A	N/A	N/A	Yes	N/A	N/A	Yes	Yes	N

Create a Business Process

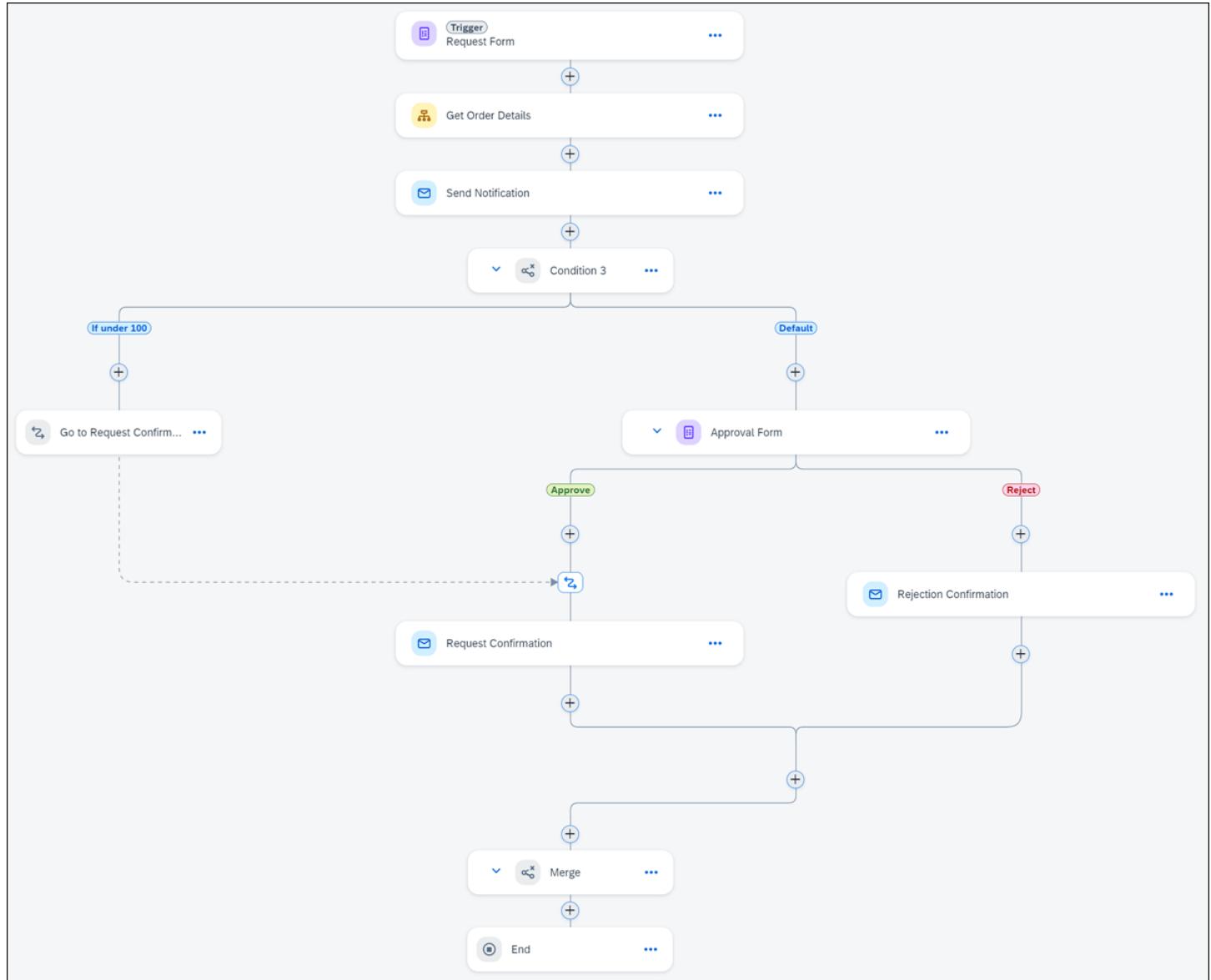
With SAP Build Process Automation, you can visually create a business process version using a combination of artifacts (such as forms and decisions) and process controls (such as branches, conditions, and mail notifications).

Context

A business process is started by defining a trigger, an event that indicates to your SAP Build Process Automation tenant to start a process instance.

Process triggers can be a form, such as a request form, an API call, where an external system starts the process, or an event.

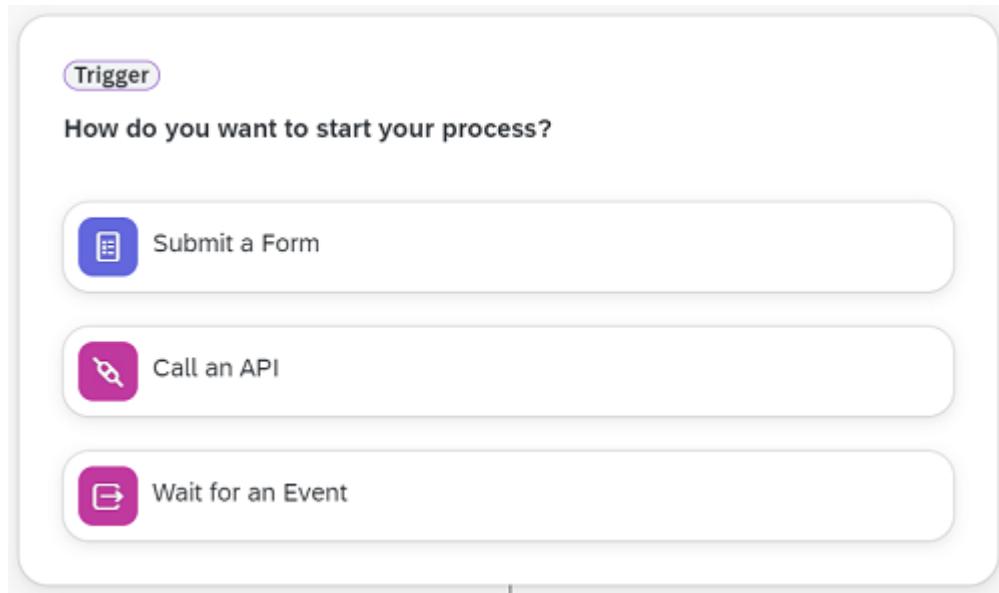
In this example, the process is triggered when a request form is submitted:



The process then continues with a combination of [Process Artifacts](#) (such as forms, approval forms, and decisions) and optional process controls.

Procedure

1. To create your project and process in the **Lobby**, choose **Create > Build an Automated Process > Business Process**.
 2. Enter a name and description for your project, and choose **Create**.
 3. Enter a name, identifier, and optionally a description for your process.
 4. Choose **Create** again.
- The process editor loads.
5. Configure your process trigger, choosing between:
 - o API trigger - [Configure an API Trigger to Start a Process](#)
 - o Form trigger - [Configure Settings for Forms and Approval Forms](#)
 - o Event trigger - [Create Event Triggers](#)

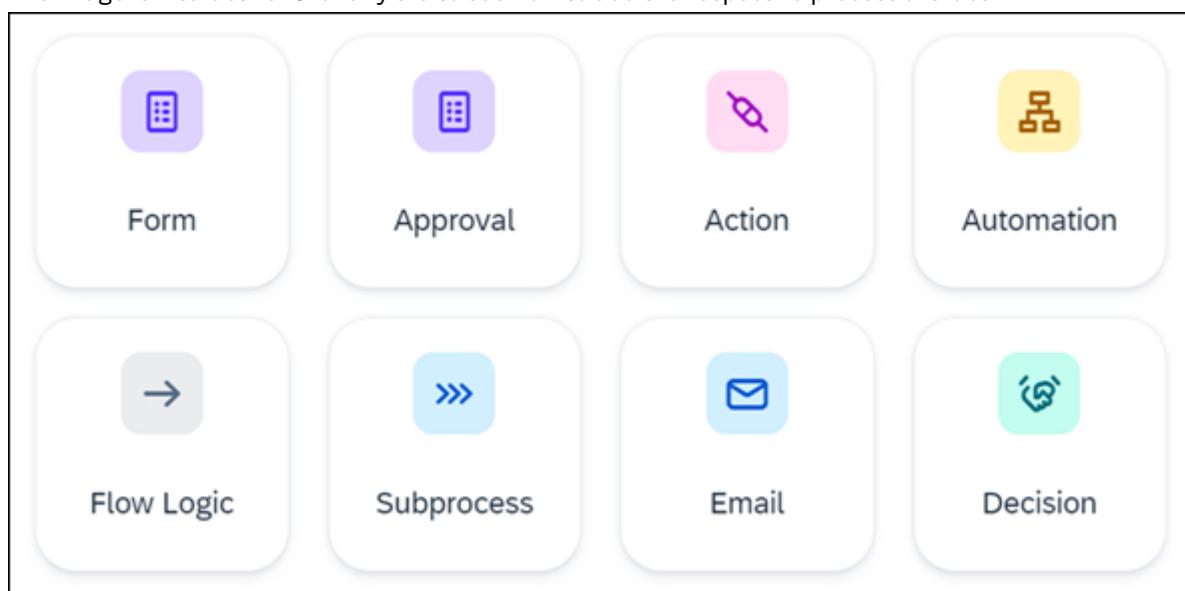


6. Click **+** and select your required process artifacts or process controls. Sometimes, you get to select again whether to create a new artifact or reuse an existing one. Repeat this step as necessary to continue building your process.

i Note

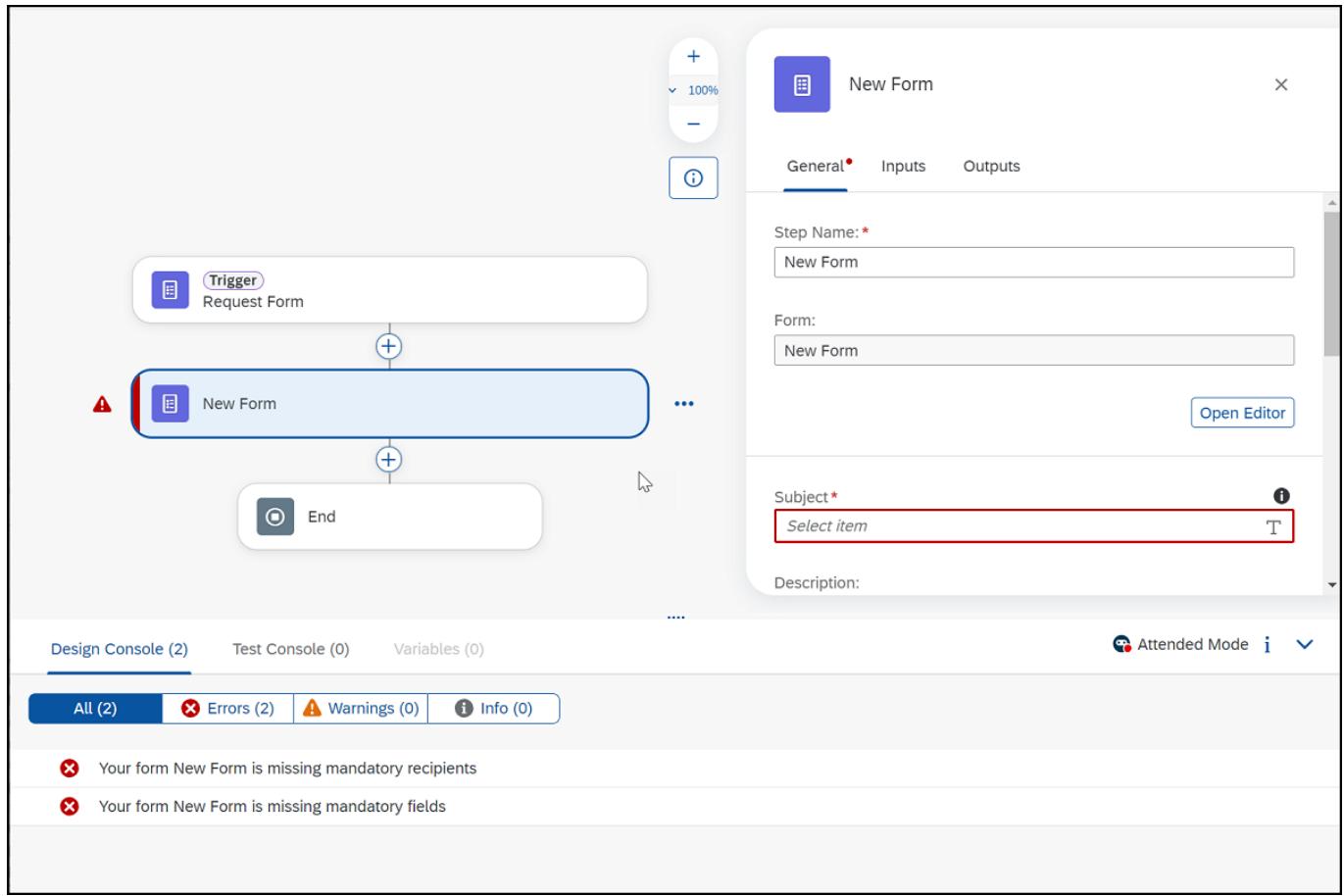
You can change the order of your process steps using drag and drop. The selected step is automatically connected to the process.

The image is interactive. Click any tile to see how to add the respective process artifact.



7. Review and correct all process errors, indicated by a red icon on the impacted artifact.

In this example, the form artifact is missing a subject. The design console displays the error details.



8. Once the process is completed and all errors are corrected, save your process.

The project is ready to be released and deployed.

Related Information

[Release a Project](#)

[Deploy a Project](#)

Configure Process Variables

The outputs of any step that you add to your process can be consumed as inputs for the subsequent steps in a process. In addition, you can create custom variables on a project level that aren't bound to a particular step.

Context

If you add actions or subprocesses to your process, their data types become available as step input to any later step in the process. For data types that contain deep structures, such as business partner information, this saves you much repetitive work.

Using custom variables, you can make any information available at the global level at any stage in the process. For more information on the use cases, see [Use Cases for Custom Variables](#).

Procedure

1. For the input variables, add an action to your process. See [Add Actions to a Process](#).
2. Click anywhere on the canvas and open the side panel.
3. On the **Process Details** side panel, choose **Variables** > **Process Inputs**.

4. To configure process input variables, next to **Process Inputs**, choose **Configure**.

a. Choose **Add Input**, and enter the following data:

- **Name and Identifier:** For example, `businessPartner`.
- **Type:** Select any data type from the list:
 - The basic data types
 - All data types of the current project
 - All date types of any dependent project. These dependencies are, for example, created when adding actions or subprocesses to the project.

The screenshot shows the 'Configure Process Inputs' dialog. At the top, there is a message: 'Some inputs might be bound to other processes and deleting them can cause errors.' Below this is a table titled 'Inputs' with columns: Name *, Identifier *, Type *, Required, and List. In the 'Name' field, 'emaillist' is entered. In the 'Identifier' field, 'emaillist' is also entered. In the 'Type' column, a dropdown menu is open, showing 'String' selected. A checkbox labeled 'Required' is checked. A checkbox labeled 'List' is checked. To the right of the table is a vertical scroll bar. Below the table is a dropdown menu with several options: 'Basic' (selected), 'String', 'Number', 'Boolean', 'Date', 'DateTime', 'Current Project' (with 'Table List Object' and 'WF data type' listed), 'petstoreactionproject' (with 'ApiResponse', 'Category', 'get_findPetsByStatus_200_output_schema', 'get_findPetsByTags_200_output_schema', 'get_getOrderById_200_output_schema', and 'get_getPetById_200_output_schema' listed), and 'petstoremodelproject' (partially visible). At the bottom right of the dialog are 'Apply' and 'Cancel' buttons.

- To create a list, select **List**.
- To make this variable a required entry, select **Required**.

5. Next to **Process Outputs**, choose **Configure** and add outputs in the same way as you added inputs.

6. To configure custom variables, choose **Configure** next to **Custom Variables**. Then, choose **Add Variable**, and enter the following data:

- **Name and Identifier:** For example, `businessPartner`.
- **Type:** Select any basic data type from the list.
- To create a list, check **List**.

7. Choose **Apply**.

8. Save your changes.

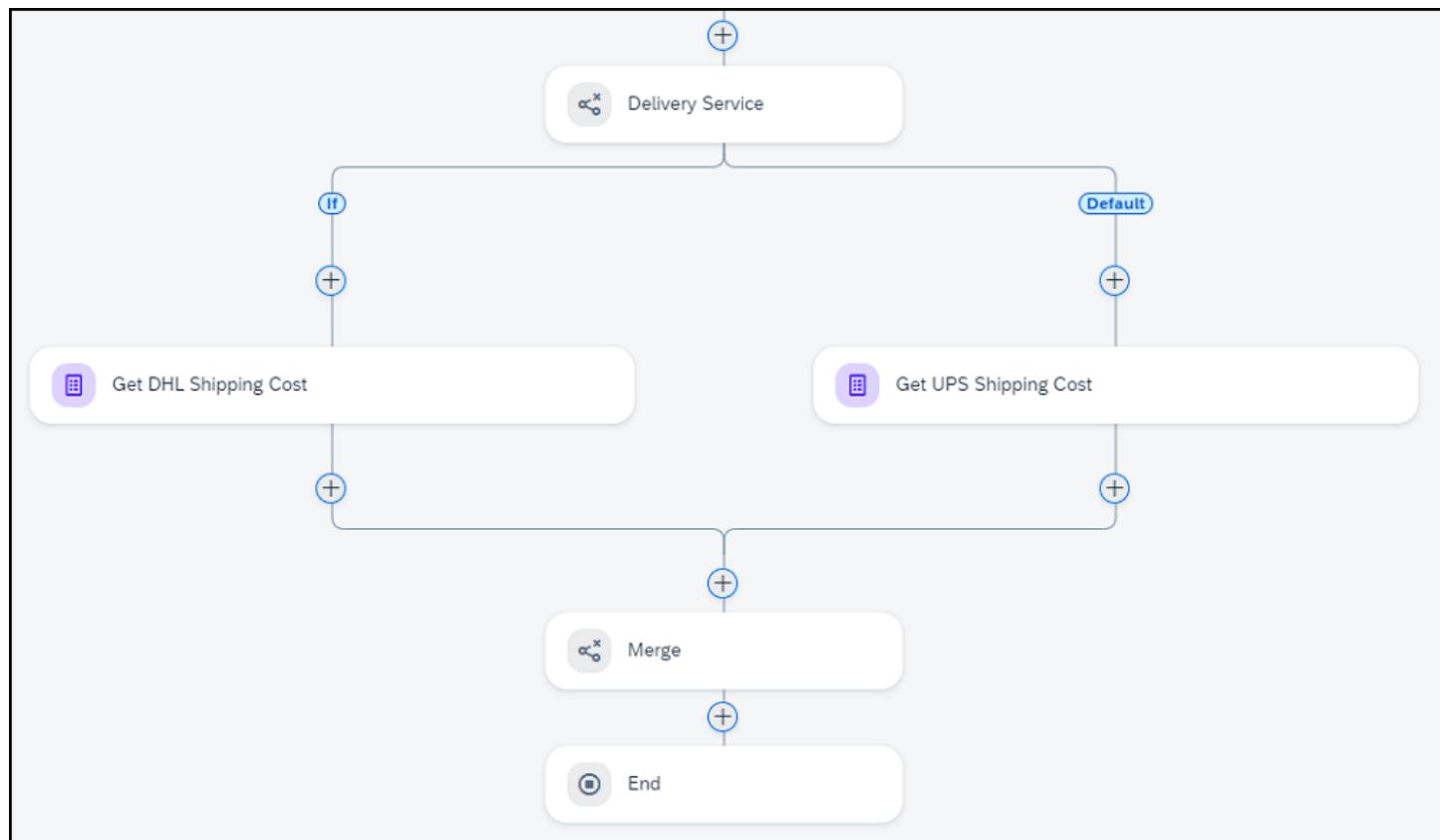
Use Cases for Custom Variables

Using custom variables, you can make any information available at the global level at any stage in the process.

Using custom variables, you can make any information available at the global level at any stage in the process. Custom variables have the following use cases:

Independency of Branch Outcome

Make the step that follows a condition branching independent of the outcome of the conditional branches. Let's look at an example: We have a condition where the branches have steps for two shipping companies. The **Final Approval** form needs to cover both possible outcomes to map the **Shipping Cost** field. So, we map an independent custom variable instead of the specific process output from the steps in the branches.



The following shows the order, in which the custom variable is inserted into the process.

1. We define the **shipping cost global** custom variable for the process.

Process Details

- General Variables Visibility
- > Process Inputs
- > Process Outputs Configure
- > Custom Variables Configure

T bike model global *
shipping cost global *
bike price global *

2. In the two steps in the branches, we use the new custom variable on the **Outputs** tab.

The screenshot shows a process flow with a 'Trigger' step followed by a 'Condition' step. Two branches emerge from the condition: one labeled 'DHL' leading to a 'Get DHL Shipping Cost' step, and another labeled 'Default' leading to a 'Get UPS Shipping Cost' step. Both branches converge at a 'Final HR Approval' step before ending. To the right, the 'Get DHL Shipping Cost' form is open on the 'Outputs' tab, where the 'Shipping Cost' output is mapped to the custom variable '# Shipping Cost'.

3. In the **Final Approval** form, we can now map the **Shipping Cost** field to the custom variable as well, and make it independent of the condition branches.

The screenshot shows the same process flow as the previous one, but the 'Get DHL Shipping Cost' step is highlighted. To its right, the 'Final HR Approval' form is open on the 'Outputs' tab, where the 'Shipping Cost' output is mapped to the custom variable '# Shipping Cost'.

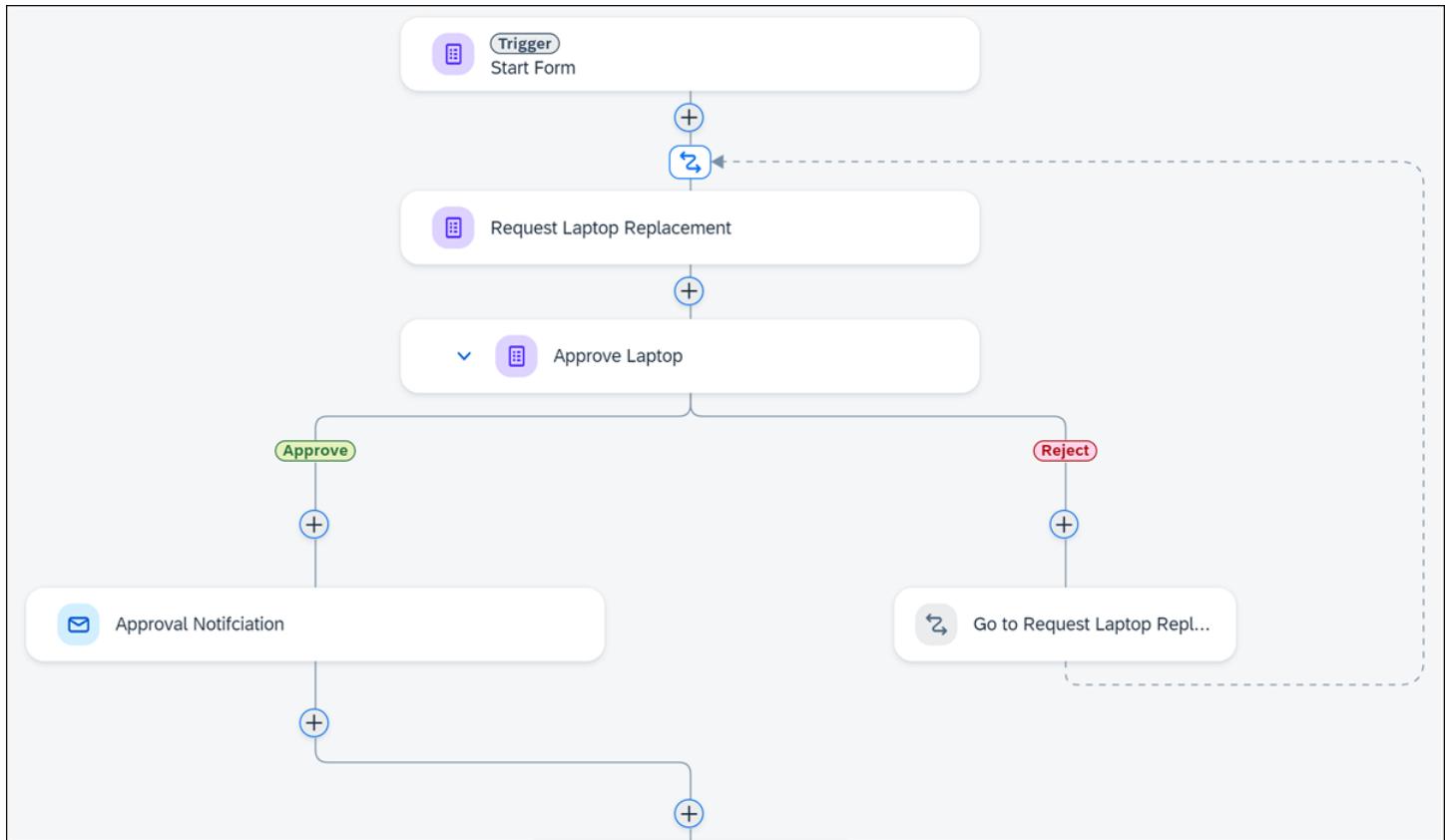
4. At runtime, the final approval form receives an input for the shipping cost, no matter which branch is used.

Rework Request

Send a request back for rework while preserving the already-entered values and the comments added by the parties involved.

Let's assume that a step needs rework before the approver accepts it. To send the process back to the previous step that needs rework, you can define a custom variable that is already available before the approval step outcome is available. In this way, the This is custom documentation. For more information, please visit the [SAP Help Portal](#)

previous step can anticipate the result of the approval step and use the correct variable.



One Field, Multiple Processes

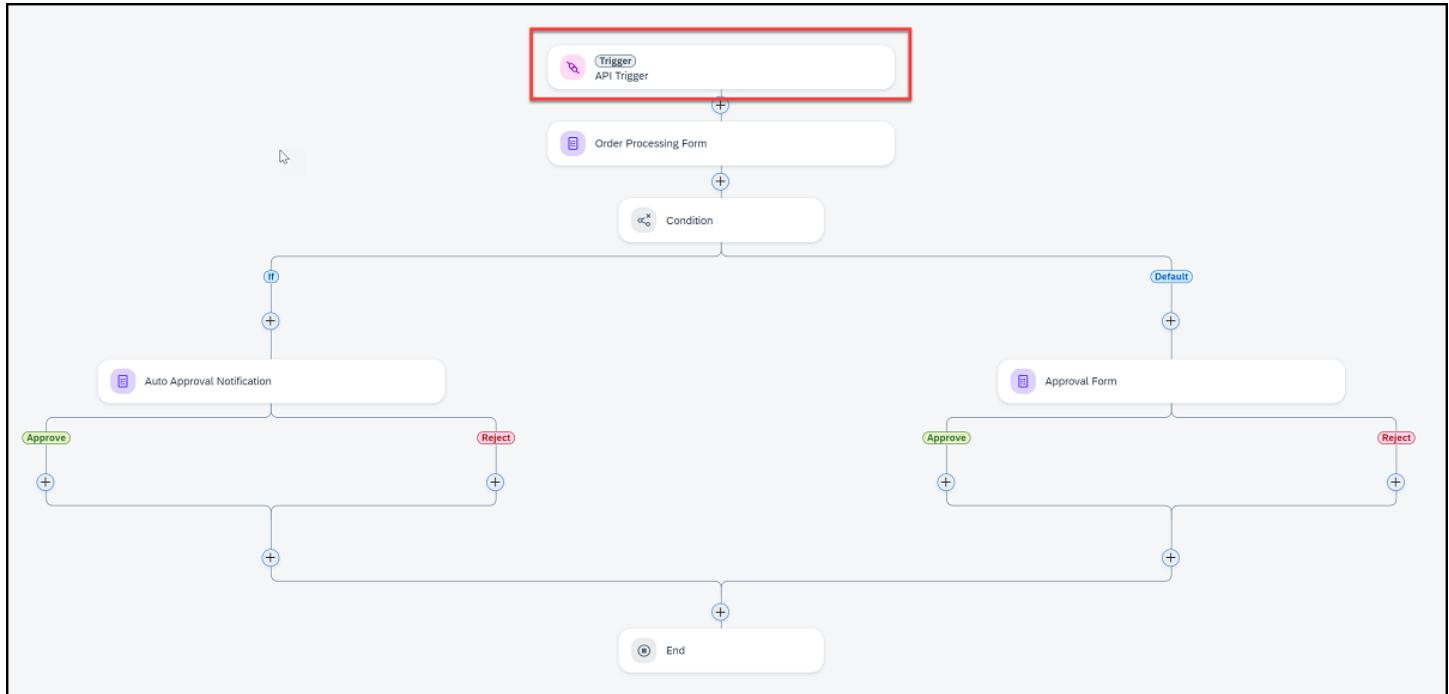
Multiple process steps edit the same field. This field is immediately available and doesn't need to be mapped and carried forward.

Configure an API Trigger to Start a Process

You can start an instance of your process using an API call, with the inputs for the call configured in the process editor. These inputs can then be used as input fields in your process, for example, in a form or approval form.

Context

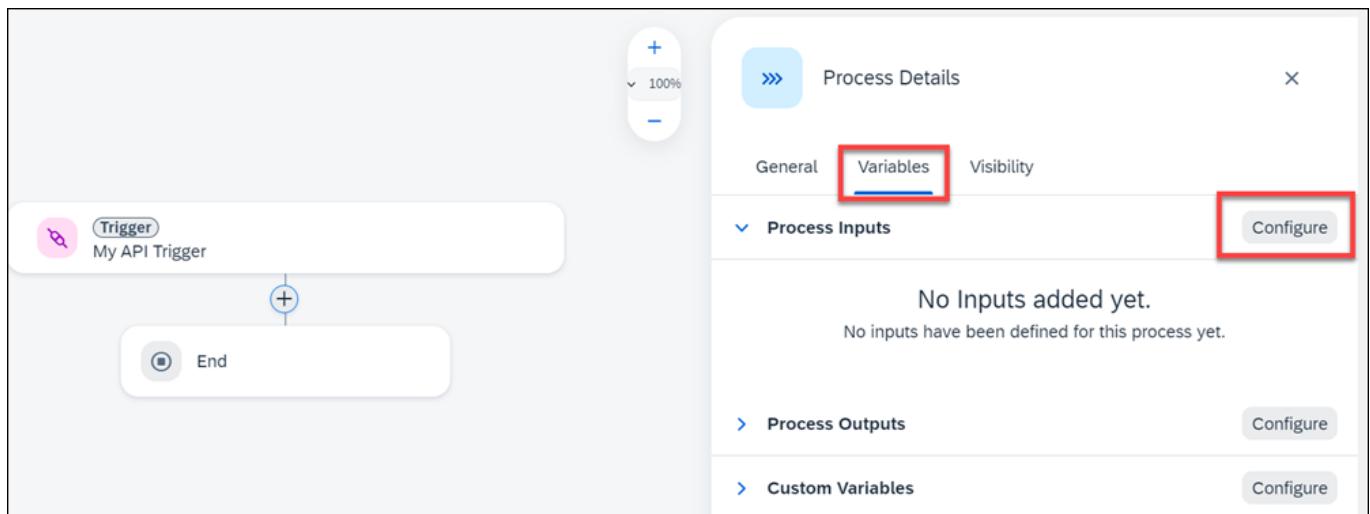
In this example, an API call is used to trigger a sales order process:



You can display such API triggers in the same overview as the automation triggers. For deployed processes, go to [Monitor](#) > [Manage](#) > [Triggers](#) from the lobby.

Procedure

1. On the **Trigger** tile in the process editor, choose **Add a Trigger**. Then, select **Call an API**.
 2. Enter a name for your trigger and a description. The identifier is generated based on the name.
 3. Click anywhere on the canvas to display the **Process Details** side panel.
 4. Choose  **Variables**  **Process Inputs**  **Configure** .



5. Click **Add Input** and then enter a **Name**, **Identifier**, and **String**, and specify if this is a **Required Input**.
Repeat this step to add all necessary inputs.

Configure Process Inputs

Some inputs might be bound to other processes and deleting them can cause errors.

Inputs

Name *	Identifier *	Type *	Required	List
No Inputs added.				

Add Input

Apply Cancel

6. Choose **Apply**.

Configure Process Inputs

Some inputs might be bound to other processes and deleting them can cause errors.

Inputs

Name *	Identifier *	Type *	Required	List
Cost Center	costCenter	String	<input checked="" type="checkbox"/>	<input type="checkbox"/> Delete
External Sourcing	externalSourcing	String	<input checked="" type="checkbox"/>	<input type="checkbox"/> Delete
Investment Value	investmentValue	String	<input checked="" type="checkbox"/>	<input type="checkbox"/> Delete

Apply Cancel

The inputs are displayed in the process settings side panel.

Trigger My API Trigger

+

100%

-

End

Process Details

General Variables Visibility

Process Inputs

- Cost Center *
- External Sourcing *
- Investment Values *

Configure

7. Once a form has been added to your process, you can then map these inputs to the form fields:

Process Content

Search

Process Inputs

- Cost Center
- External Sourcing
- Investment Values

Process Metadata

Trigger My API Trigger

+

My First Form

+

End

My First Form

General Inputs Outputs

Cost Center Select item T

External Sourcing Select item T

Investment Value Select item T

8. Save your changes.

9. With the API call set as your process trigger, you can now test the call using Postman. See [SAP Community - How to Test Your API Call Trigger Using Postman](#).

i Note

We support the ISO 8601 format for date and time:

YYYY-MM-DD (2023-03-16)

hh:mm:ss (15:33:16)

Create Event Triggers

Prerequisites

- Your SAP S/4HANA Cloud system is available and connected with SAP Build Process Automation, and is able to receive all required notification events. For more information, see [Enable the Consumption of SAP S/4HANA Cloud Events](#).
- You have set up an Event Mesh service instance in your SAP BTP cockpit subaccount. For more information, see [Setting Up SAP Event Mesh in BTP Cockpit](#).
- You have created your queue and your queue subscriptions. For more information, see [Manage Queues](#).
- You have created a webhook subscription to subscribe to an event queue. For more information, see [Manage Queue Subscriptions](#).
 - You have created a service key. For more information, see [Create a Service Key for the SAP Build Process Automation Instance](#).
 - You have fulfilled the **Webhook URL** by using the URL from the instance of the **SAP Build Process Automation plan standard**  **Service Key**  **api** endpoints and adding `internal/be/v1/events`. As a result, your webhook URL should be as follows:
`https://spa-api-gateway-bpi-eu-prod.cfapps.sap.hana.ondemand.com/internal/be/v1/events`.
 - You have inserted the **Client ID**, the **Client Secret**, and the **Token URL** by adding `oauth/token` to the latter. You can copy and paste these credentials from the service key.

Context

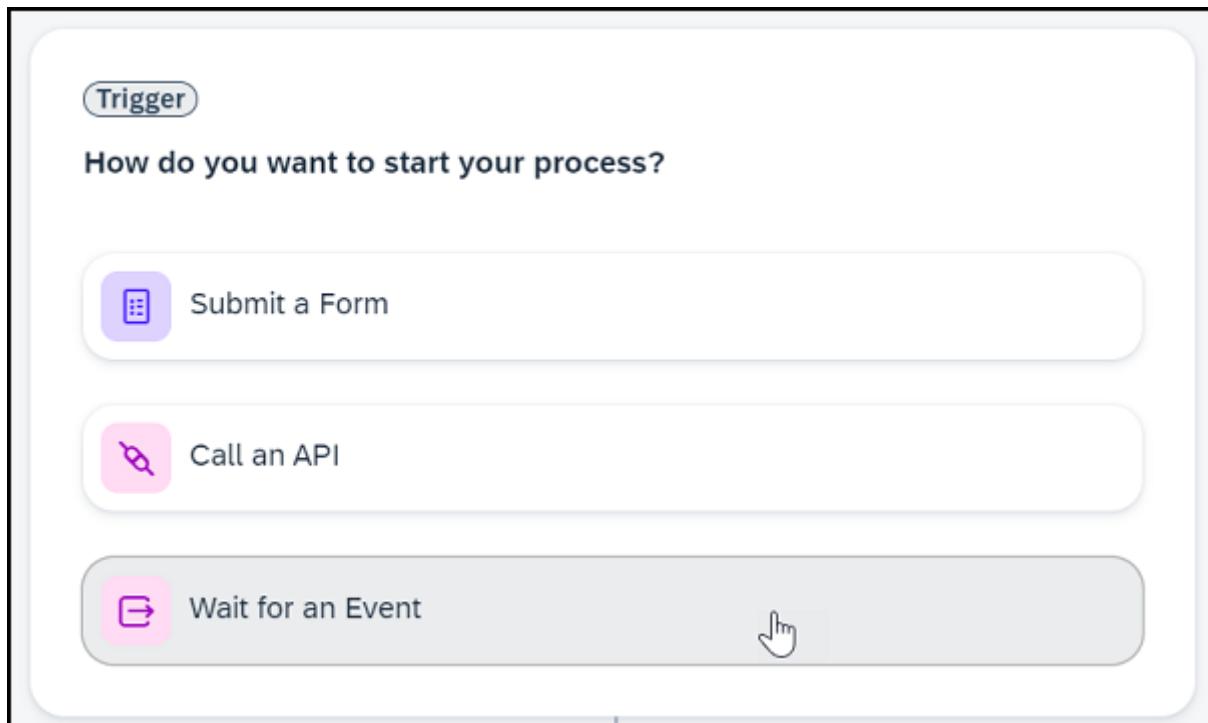
The event triggers onboard and listen to back-end events emitted from an external source system, and reacts to such events by triggering artifacts such as processes and automations.

All public events published on SAP Business Accelerator Hub for [SAP S/4HANA Cloud](#) and [SAP S/4HANA](#) are supported.

You already have created a project and can now add a process as an artifact that has to be started by an event trigger.

Procedure

1. On the **Trigger** tile in the process editor, choose **Add a Trigger**. Then, select **Wait for an Event**.



2. Enter the following data:

Field	Description
General	
Name	<p>Enter the name of your event trigger.</p> <p>i Note The event trigger name cannot be longer than 64 characters.</p>
Executes	Set by default to the current process name and is read-only.
Add Description	<p>Enter the description of your event trigger.</p> <p>i Note The description cannot be longer than 1024 characters.</p>
Event Configuration	
Event Object	Select an event object from the predefined list with consolidated event groups. You can use the fuzzy search to match your choice easier.
Event	Select an event that's available for the event object you have chosen at the previous step.

Create Event Trigger

General

Name: *	Executes: *
<input type="text" value="Enter a name"/>	<input style="background-color: #e0f2e0; border: 1px solid #c0e0c0; padding: 2px 10px; width: 150px; height: 20px; border-radius: 5px; vertical-align: middle;" type="text" value="» MyProcess"/> ▼
Description:	
<input style="width: 100%; height: 40px; border: 1px solid #c0e0c0; border-radius: 5px; padding: 5px; vertical-align: middle;" type="text" value="Enter a description"/>	

Event Configuration

Event Object *	Event *
<input style="width: 250px; height: 25px; border: 1px solid #c0e0c0; border-radius: 5px; padding: 2px; vertical-align: middle;" type="text" value="Select Event Object"/> ▼	<input style="width: 250px; height: 25px; border: 1px solid #c0e0c0; border-radius: 5px; padding: 2px; vertical-align: middle;" type="text" value="Select Event"/> ▼

Create **Cancel**

3. Choose **Create**.

A success message for creating the event trigger appears.

Related Information

[Manage Event Triggers](#)

Manage Event Triggers

You can edit, enable or disable, and delete event triggers using the different options of SAP Build Process Automation.

Related Information

[Create Event Triggers](#)

Manage Event Triggers from the Project Overview

You can edit, enable or disable, and delete event triggers using the option of project **Overview**.

Procedure

1. To manage your event trigger, choose **Overview > Triggers > ... (More)**.

Artifacts (27) **Triggers (1)** Dependencies (4)

Triggers

Name	Executing
User Starts Work	Trigger Start

...

Edit
Deactivate
Delete

2. Select one of the following options:

- **Edit:** To change the event trigger's **Name** or **Description**
- **Deactivate:** To not deploy the trigger
As a result, the event trigger becomes grayed out.
- **Activate:** To redeploy the event trigger
- **Delete:** To delete the event trigger artifact

3. Save your changes.

Manage Event Triggers from the Process Editor

You can edit, enable or disable, and delete event triggers using the option of process editor.

Procedure

1. To manage your event trigger, go to **Overview** and choose... (**More**).

2. Choose one of the following actions:

- **Edit:** To change the event trigger's **Name** or **Description**



- **Delete:** To delete the **Event** trigger artifact

3. Save your changes.

Manage Event Triggers from the Triggers View

You can edit, enable or disable, and run event triggers using the option of the **Triggers** view.

Procedure

- In SAP Build, choose **Control Tower**, then choose the **Environments** tile.
- Select your environment and choose the **Unattended Triggers** tab.
- Search for your trigger using the Type, Time range validity, Project, and Attributes filters.

Trigger	Executing	Project & Version	Next Execution	Last Updated	Attributes
Test trigger	Auto 2	Second... V1....		Over a month ago	

- Choose ... (**More Options**) for the selected event trigger and select one of these options.

- To pause the trigger action, choose **Disable**.

As a result, the event trigger becomes grayed out.

- To execute the event trigger processes, choose **Enable**.
- To change the event trigger **Name** or **Description**, choose **Edit**.
- To execute the event trigger in order to initiate event delivery with the user-provided data immediately, choose **Run Now**.

- If you choose **Run Now**, the **Run My Event Trigger** pop-up window appears. The **Event** and **Executes** fields are predefined with the event trigger data.
- In the **Test with following data** field, a structured message header must be provided, including the event data with the respective inputs. Ensure the following properties are correctly configured, as in the example below:

Sample Code

```
{
  "type": "sap.s4.beh.salesorder.v1.SalesOrder.Created.v1",
  "specversion": "1.0",
  "source": "internal",
  "id": "0794ef45-7741-1eea-b7be-ce30f48e9a1d",
  "time": "2023-12-04T06:21:52Z",
  "datacontenttype": "application/json",
  "data": {
```

```

    "SalesOrder": "3016330",

    "EventRaisedDateTime": "2023-08-08T10:16:27.919Z",

    "SalesOrderType": "TA",

    "SalesOrganization": "1010",

    "DistributionChannel": "10",

    "OrganizationDivision": "00",

    "SoldToParty": "1000360"

}

}

```

- Ensure the correctness of the message header properties:

Property	Description
"type"	Represents the type of the event associated with the trigger. Update it using the same structure but with the corresponding event object and event.
"specversion"	The default value is "1.0" and must not be altered.
"source"	The default value is "internal" and must not be altered.
"id"	Random identifier with a maximum length of 64 characters. It must be unique. If not unique, then event will not appear as a separate record in Event monitoring UI.
"time"	Date-time in the ISO 8601 format.
"datacontenttype"	Specifies the type of content. The default value is "application/json" and must not be altered.
"data"	Represents the parameters of events utilized in the process definition, including corresponding input values. <p>i Note</p> <p>You should not provide any properties that are not defined in the process definition.</p> <p>In case of such, the validation will be activated, and then you will not be able to execute the trigger.</p>

- If the event trigger is disabled, the **Run Now** option is grayed out.

5. Save your changes.

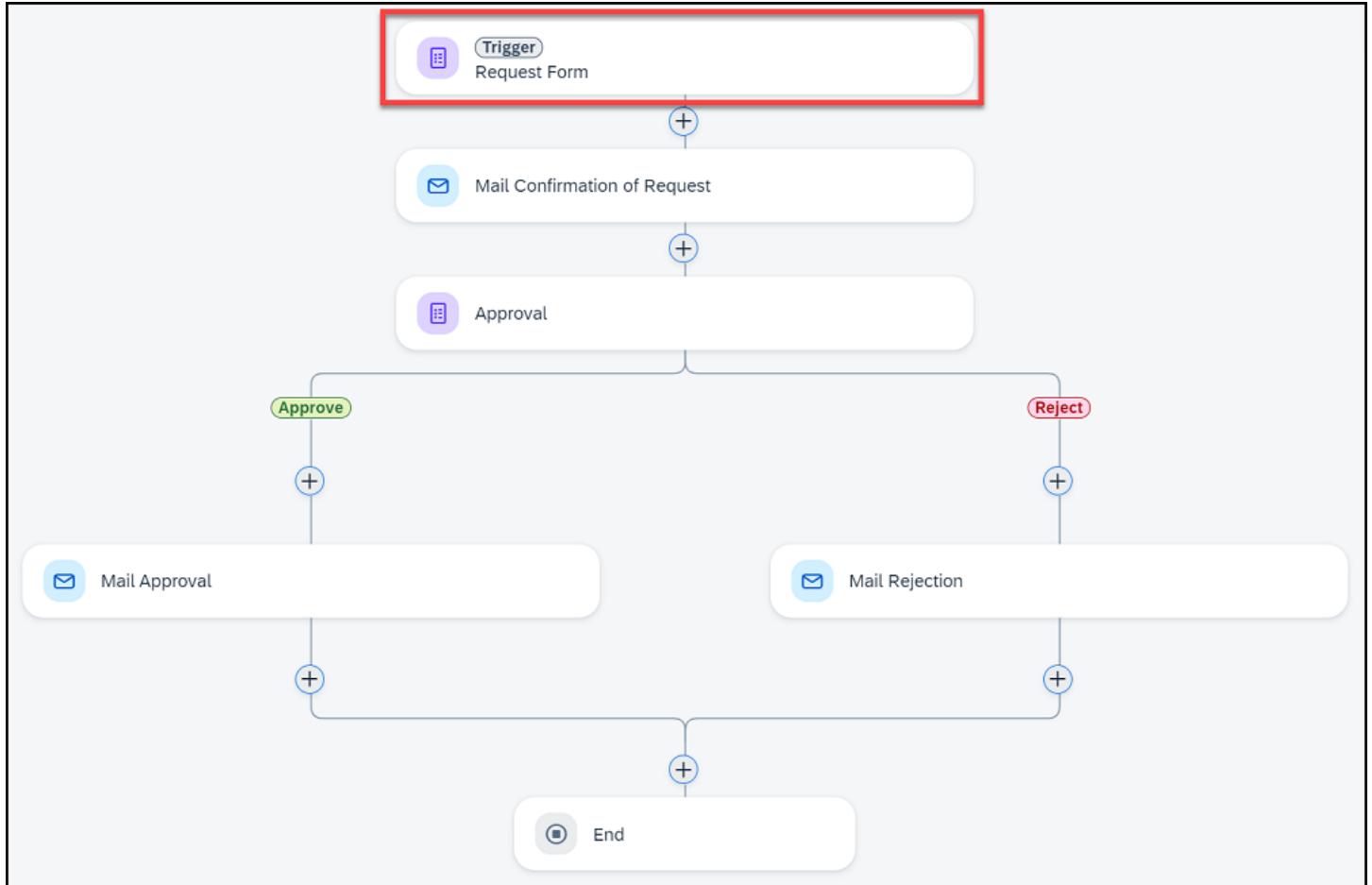
To monitor the business events, acquired from connected systems, see [Acquired Events](#).

Create a Form

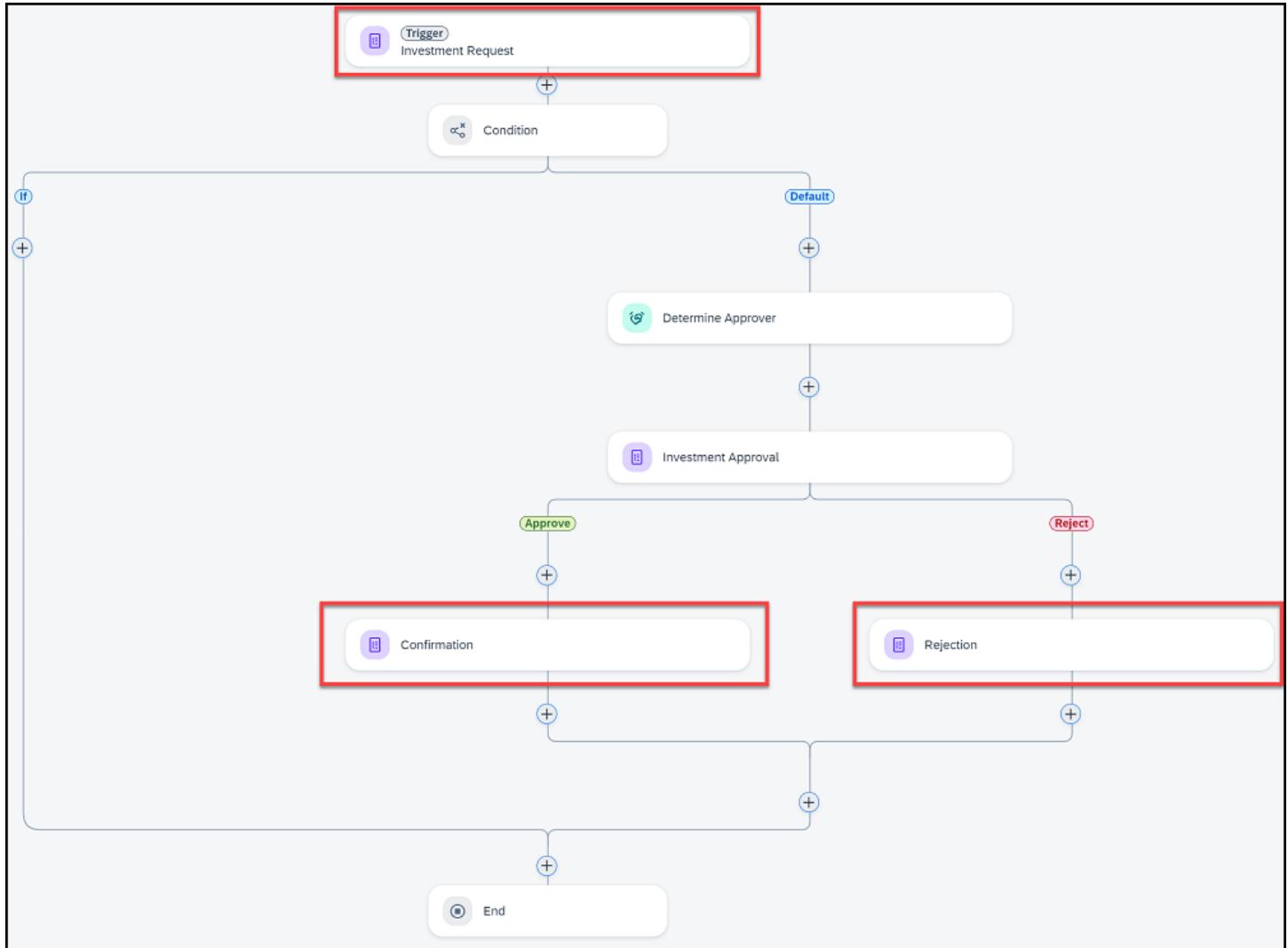
You can create interactive forms in SAP Build Process Automation, allowing you to capture and share information during a running process. Forms can then be a start trigger for a process and added as additional steps in the same process.

Context

In the following simple request process, a request submission form starts the process running. The requestor can access the published request form using a direct link or using a configured file in their SAP BTP.

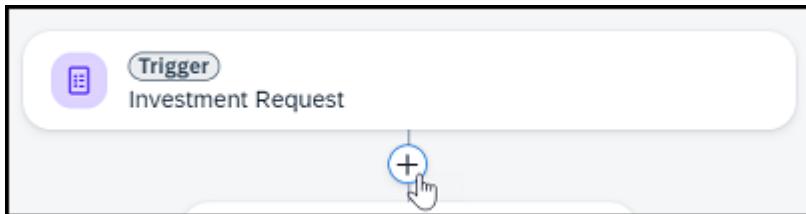


And in the following investment approval request, a request submission form also starts the process running. The requestor can access the published request form using a direct link or using a configured file in their SAP BTP. Then additional confirmation and rejection forms are included in the process, which will be displayed to the recipients as a task in their inbox.



Procedure

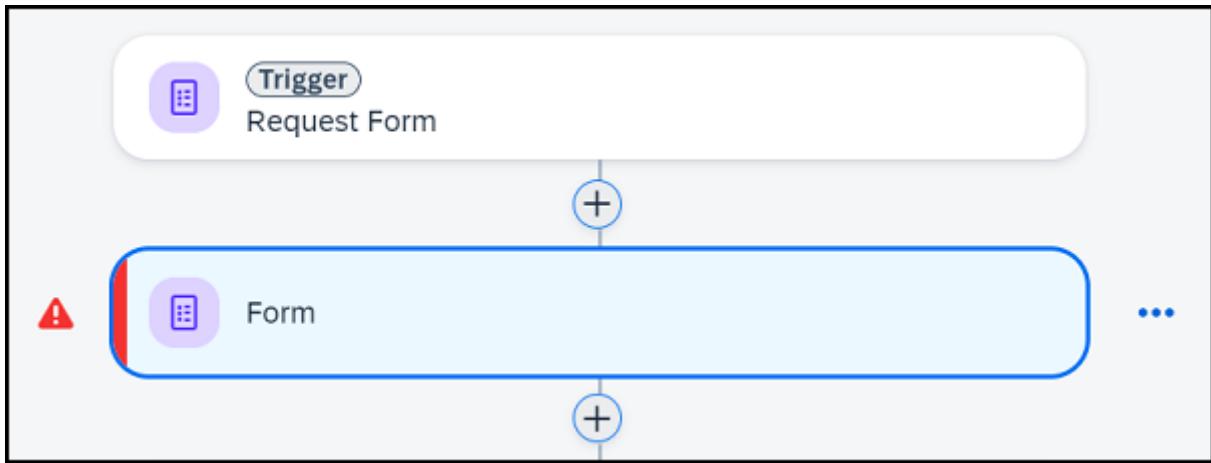
1. To create another form in the business process editor, choose **Form > Blank Form**.



2. Enter a **Name**, **Identifier** (used as an internal reference), and an optional **Description**.

3. Choose **Create**.

The form is created and added to the process editor canvas.



4. Double click and open the **Trigger** or **Form** element on the process editor canvas.
5. Drag and drop **Layout** and **Input** fields to the form as required. For an overview of form fields, see [Form Input Fields](#).

The screenshot shows the SAP Build Process Automation interface. The top navigation bar includes 'SAP Build Process Automation', 'Request Project' (which is the active tab), and 'Editable'. Below the navigation is a toolbar with icons for Overview, Request Process, Request Form, Release, and Save. The main content area is titled 'Request Form' and contains the following fields:

- Name ***: Text input field with placeholder 'Enter Text'.
- Office Location ***: Text input field with placeholder 'Enter Text'.
- Item Required ***: Text input field with placeholder 'Enter Text'.
- Enter the Reason**: Text input field with placeholder 'Enter Text'.
- Order Amount**: Text input field with placeholder 'Enter a Value' and a '#' symbol.
- Delivery Date**: Text input field with placeholder 'Select a Date'.

A red box on the left side of the screen highlights the sidebar where various input field types are listed, such as H1, H2, AA, T, E, etc.

6. To add a table, drag and drop the **Table** element to the form and enter a name.
- a. To add a column, choose **+** (Plus). For an overview of column types, see [Form Tables](#).

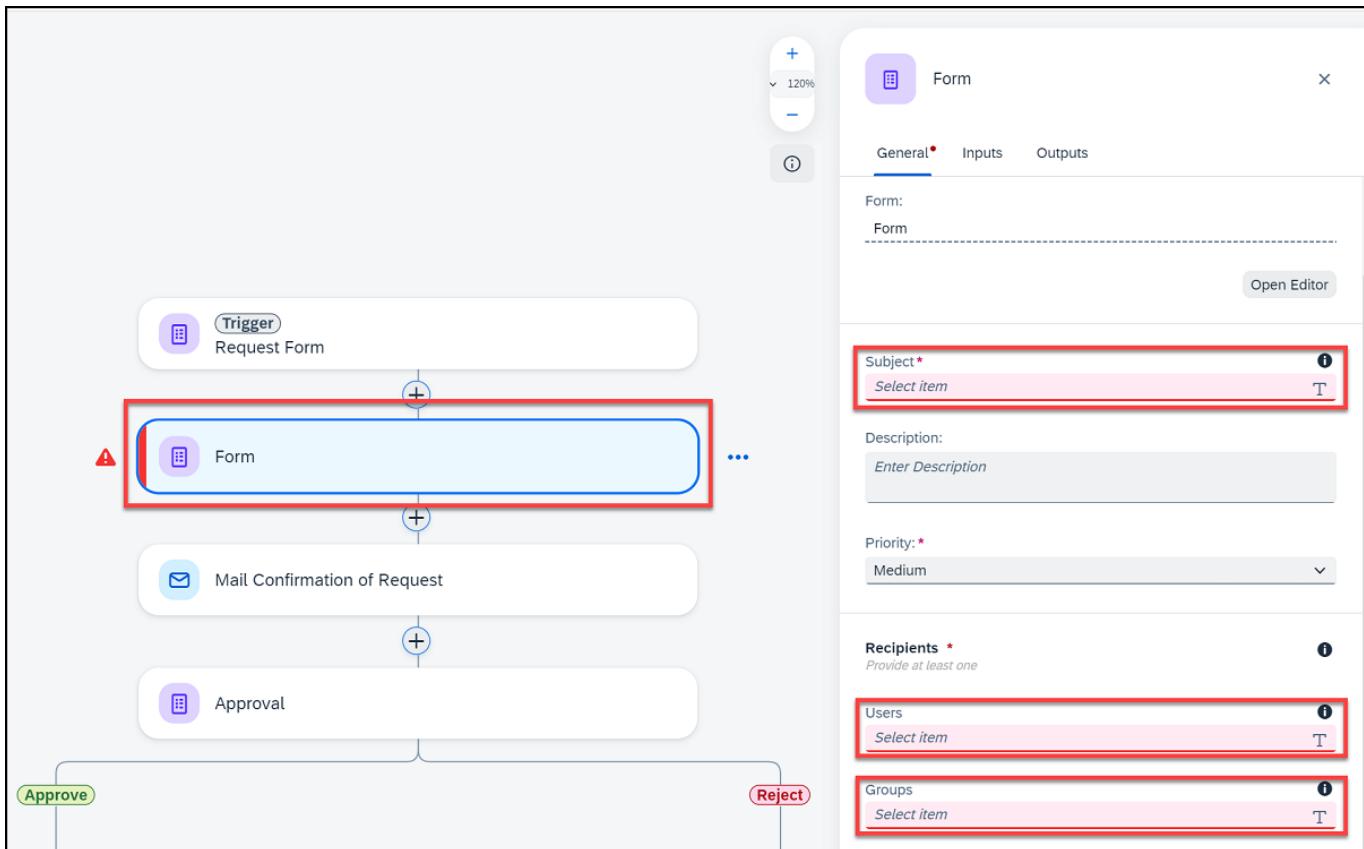
The screenshot shows the configuration of a table named 'Line Item Details'. The table has four columns with headers: 'Description *', 'Quantity *', 'Amount', and 'Expected Delivery Date'. Each column contains a text input field with placeholder text ('Enter Text', 'Enter a Value', 'Enter a Value', 'Select a Date') and a corresponding unit indicator ('T', '#', '#', 'Select'). A red box highlights the '+' button in the bottom right corner of the table header, which is used to add new columns.

7. **Optional:** Customize the default **Submit** outcome button and use **Next**, **Send**, or a **Custom** text. Choose the cogwheel icon, and select the required text from the **Button Title** dropdown.

If you select **Custom**, enter your own text. This text is then only available in the language in which you entered it. The predefined labels are translated. If several labels are translated using the same term in the target language, then the application only shows one label.

8. Save your changes.

9. Return to the process editor canvas and click on the form title, opening the form settings side panel.



10. Configure the **Form Settings** as required. See [Configure Settings for Forms and Approval Forms](#).

11. Save your changes.

Form Input Fields

You can add form input fields to both a standard form and an approval form, allowing you to design and manage the information displayed to the process participant.

Available Input Fields

In the form editor, you can directly edit the form name. You can drag and drop the following input fields:

Field	Available Values or Options
Headline 1	Enter text. You can move the element up and down.
Headline 2	
Paragraph	Enter text. You can move the element up and down.
Text (maximum 256 characters)	<ul style="list-style-type: none"> Minimum character limit Maximum character limit Read Only Required Description

Field	Available Values or Options
Text Area (maximum 1700 characters)	<ul style="list-style-type: none"> • Minimum character limit • Maximum character limit • Read Only • Required • Description
Dropdown	<p>Use one of the following options to define the dropdown entries:</p> <ul style="list-style-type: none"> • Use data from a data set by selecting an action from the library. When a data source is value help enabled, then the dropdown also shows the search icon and a popup for the search entry. See Add and Use Data Sources in Form Input Fields. ◦ Required ◦ Description • Enter text to define the options manually. ◦ Required ◦ Read Only ◦ Multiple Selection ◦ Description
Choice	<ul style="list-style-type: none"> • Enter text for the selection option entries • Vertical Alignment • Read Only • Required • Description
Checkbox	<ul style="list-style-type: none"> • Read Only • Required • Description
Number	<ul style="list-style-type: none"> • Fixed decimal places • Minimum value • Maximum value • Read Only • Required • Description
Date	<ul style="list-style-type: none"> • Based on the user's location • Allow selection of: All Dates, Only Past Dates, and Only Future Dates • Read Only • Required • Description

Field	Available Values or Options
File Upload / Attachment Requires an active SAP Document Management Service subscription. See Configure SAP Document Management Service for Process Attachments .	<ul style="list-style-type: none"> • Maximum number of files • Required • Read Only • Description
Tables	See Form Tables
Link	<p>Enter a link address of type HTTPS, HTTP, or mailto.</p> <p>Enter a link text to display instead of the URL, a label, and a description that appears in a question mark tool tip.</p> <p>Your input is the default link content. If you use a form that contains a link, the link content can be overwritten by other process content because the link field is available for mapping.</p>

Input Field Settings

You can directly edit input field texts. To move the field up or down or to delete it, use the **Menu** icon.



The **Configuration** section, let's you define more settings. Depending on the field type, you can, for example, mark the field as:

- **Required** - The participant must input data here to progress the active process.
- **Read Only** - The participant can see information entered earlier in the process, but can't edit it.
- **Multiple Selection** - The participant can see select more than one entry.

The **Input Validation** section, let's you define whether and which validation is used for text fields and text areas. When you select an option, the regular expression for that selection is displayed. You can copy it to create a more sophisticated custom expression that you enter as a custom validation. You can directly test the validation you defined using the **Test Text** field. For examples on regular expressions, see, for example, the [Mozilla cheat sheet](#).

The following validation options are available:

- **Letters** - The participant can enter letters only.
- **Numbers** - The participant can enter numbers only.
- **Letters and Numbers** - The participant can enter any character.
- **Alphanumeric** - The participant can use latin letters (a-z, A-Z) and arabic numbers.
- **Custom Validation** - You can enter a regular expression that exactly defines what the participant can enter. For example, you can require three letters, one number....

Add and Use Data Sources in Form Input Fields

You can add data sources to your form input fields, allowing process participants to select from information managed in external systems. This removes the need to manually add fields and information when creating a form.

Prerequisites

Before you can add and use data sources in your forms, make sure that the data sources have been configured in your library.

See [Configure Data Sources for Form Input Fields](#).

i Note

- Only JSON files are supported and the file size is limited to 5 MB.
- Open API specification files with versions 2.x.x and 3.x.x of JSON type are supported.
- Must include a **GET** request with an array.
- Must not have mandatory input parameters defined.
- To make an action available as a data source, define it's output by setting the **Main Output Array** tag.
- To use the search feature for a data source, the action must have the \$search parameter defined.

Context

Adding data sources for form input fields allows process participants to select from information shared from external systems. These fields can then be mapped across process steps, ensuring that process data is consistent.

As an example, a dropdown field in the following form is configured to allow process participants to select from customer data shared from SAP S/4HANA. In this case, they can select the customer's country, ID, address, and the company name.

New Dropdown ⋮ X

▽ Data to display

From data set **Manual definition**

Data Source *

Get entities from Custom...

Destination Variable *

Data_S4HANA i

Available Data *

ID X CompanyName X S...

ID	CompanyName	S...
<input type="checkbox"/>	Fax string	
<input type="checkbox"/>	City string	
<input type="checkbox"/>	Phone string	
<input type="checkbox"/>	Region string	
<input checked="" type="checkbox"/>	Address string	
<input checked="" type="checkbox"/>	Country string	
<input checked="" type="checkbox"/>	CustomerID string	
<input type="checkbox"/>	PostalCode string	
<input checked="" type="checkbox"/>	CompanyName string	

i Note

You can only use dropdown fields.

The list shows the labels of the output values that are defined for the action. These labels can be changed in the action editor.

Procedure

1. In the form editor, drag and drop a **Dropdown** input field to the canvas.
2. In the side pane under **Data to display**, choose **From data set**.

New Dropdown : X

▼ Data to display

From data set Manual definition

Option 1

Option 2

Configuration

Required

Read Only

Multiple Selection

3. Select a library under **Data Source**.

New Dropdown : X

▼ Data to display

From data set Manual definition

Data Source *

X

Configuration

Required

Add Description

4. Once you've found the **Action** to use, choose **Add**.

Browse library

Supplier

Action Type	Projects
GET X	X

🔗 Action

Get entities from Suppliers

Project: Northwind actions for Forms Value... Add

5. Select or create a **Destination Variable**. Destination variables are used to connect the action to an external system.
6. Select the **Available Data** that you want to display to the process participant in this field.

New Dropdown ⋮ X

⋮ X

▼ Data to display

From data set Manual definition

Data Source *

Get entities from Suppliers X Delete

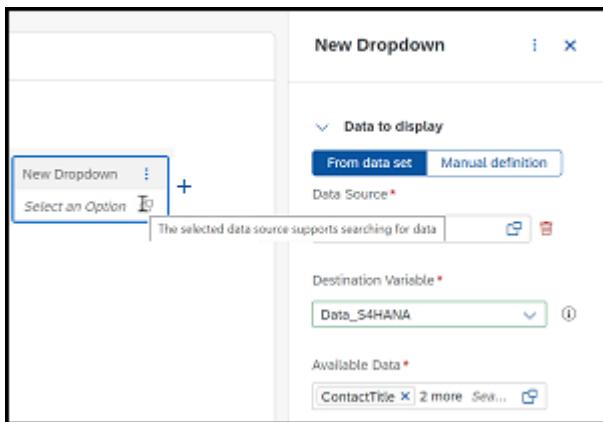
Destination Variable *

Data_S4HANA ⓘ

Available Data *

CompanyName X 2 more S... X

If the the \$search parameter is defined, the icon in the dropdown changes:



7. Save your changes.

Form Tables

As with form input fields, you can add tables to both a standard form and an approval form, allowing you to design and manage the information displayed to the process participant.

You define the columns of the table, and the process participant adds content with as many rows as needed.

If you create a form based on a form with a table, then you can map those fields as input of the new form using [Bind List](#).

i Note

You must bind the table to a data type list of type **Object** and the **Required** constraint, even for single-column tables of type text with string input. Lists with other data types, for example, strings, numbers, or booleans don't work.

The table in the new form is set to read only by default but you can change that setting.

The following general rules apply:

- The number of columns isn't limited.
- The number of rows can't be predefined.
- The **Read Only** setting applies to the entire table and can't be used for individual columns.

The following columns elements are available:

Column	Available Values or Options
--------	-----------------------------

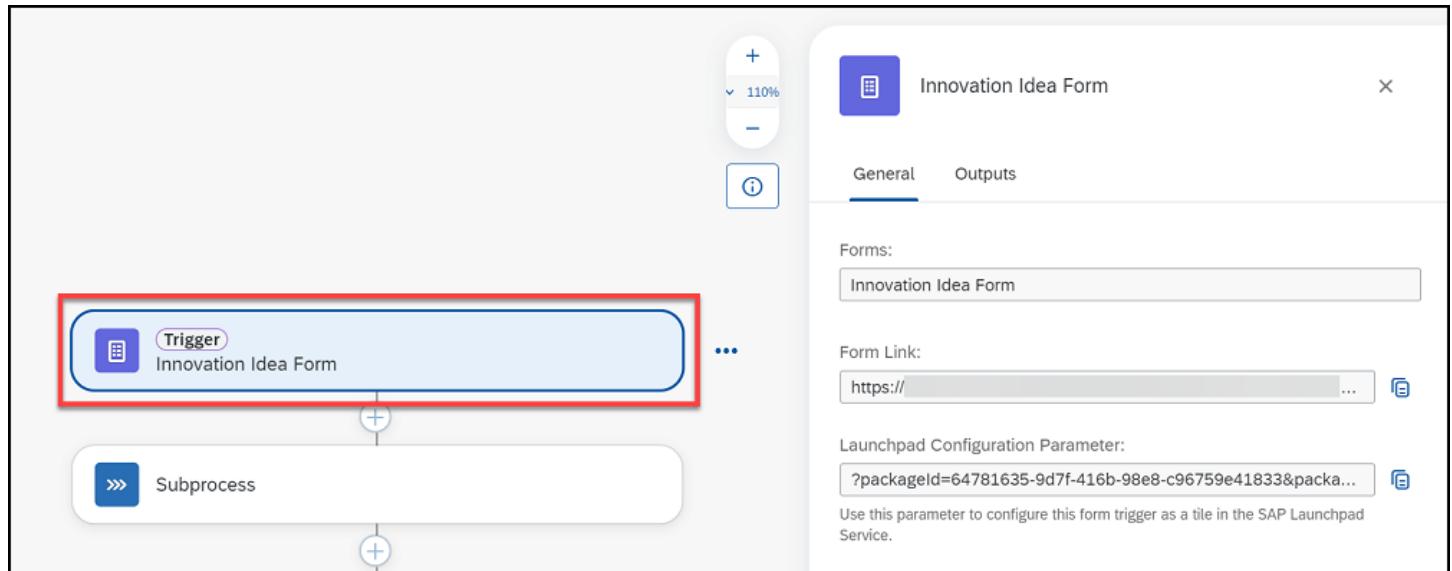
Column	Available Values or Options
Text (maximum 256 characters)	<ul style="list-style-type: none"> • Minimum character limit • Maximum character limit • Required • Description
Checkbox	<ul style="list-style-type: none"> • Required • Description
Number	<ul style="list-style-type: none"> • Fixed decimal places • Minimum value • Maximum value • Required • Description
Date	<ul style="list-style-type: none"> • Based on the user's location • Allow selection of: All Dates, Only Past Dates, and Only Future Dates • Required • Description
Dropdown	<p>Use one of the following options to define the dropdown entries:</p> <ul style="list-style-type: none"> • Use data from a data set by selecting an action from the library. When a data source is value help-enabled, then the dropdown also shows the search icon and a popup for the search entry. See Add and Use Data Sources in Form Input Fields. • Enter text to define the options manually. <ul style="list-style-type: none"> ◦ Required ◦ Description <p>i Note</p> <p>If you use a dropdown in the table, you can no longer set the whole table to read only.</p>

Configure Settings for Forms and Approval Forms

You can configure the settings for your forms using the process editor, allowing you to define how the form is displayed to the process participant.

Settings for Start Trigger Forms

You can configure your process start trigger settings by clicking on the **Trigger** tile.



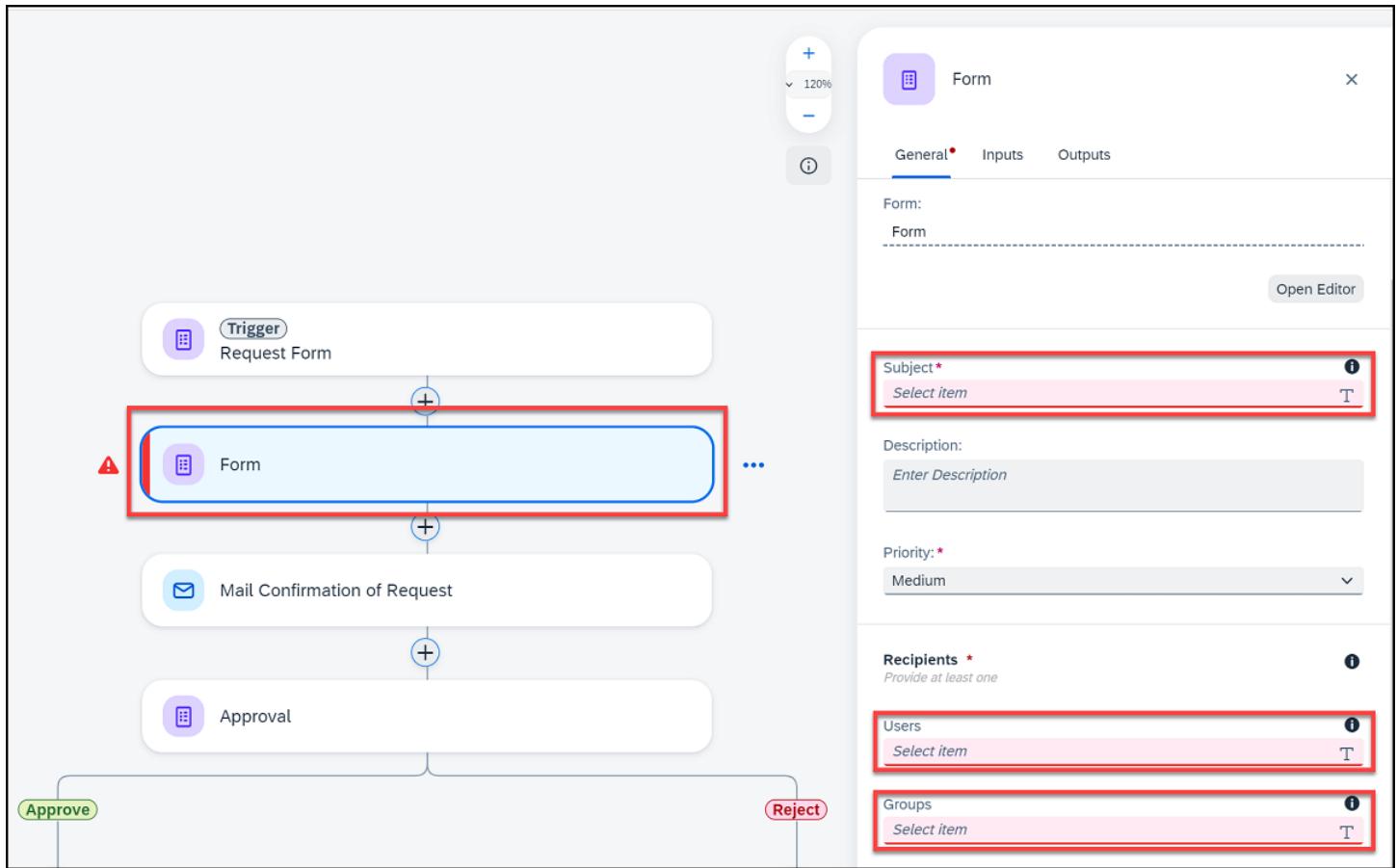
You have the following configuration options:

Field	Description
Forms	Select the form you want to use as the trigger for your process. Only forms available in the same business process project can be selected.
Form Link	Available for deployed processes only, this provides a direct link to the published form. The process participant can access this link, enter the relevant information, and then click submit.
Launchpad Configuration Parameter	Available for deployed processes only, this provides the parameters needed to create a tile in SAP Build Work Zone. Clicking the tile opens the form and from there the process participant can enter the relevant information and then click submit.
Outputs	This section provides an overview of the process metadata included in the trigger form. This is a read-only section.

Additional Settings for Forms or Approval Forms

You can configure additional form or approval form settings for forms visible on the process editor canvas only, rather than all forms in the project. These form settings allow you to configure how the form information is displayed as a task in the inbox and who that task is sent to.

To configure the form settings, click on the form element on the process editor:



You have the following configuration options:

Field	Description
Step Name	This is the name under which the form displays in the process canvas.
<Name of the form>	This is the name you entered for the form when creating it.
Subject	This is the subject of the task received in the inbox, allowing the participant to identify the item. In the following example, this is displayed as: Purchase Requisition - Approval .
Description	This provides the description displayed within the received task, giving the participant more information about the process. In the following example, this is displayed as: Please approve or reject following request .
Priority	This defines the priority of the task received in the inbox, giving an indication of when a response is needed. In the following example, this is displayed as High .
Recipients	You can assign process metadata and context fields, for example, lists of users or the user who started the process. Alternatively you can enter specific email addresses of users or groups who should receive the task. For more detailed guidelines, see Guidelines for Specifying Recipient Users .
Allow Forwarding	Select to allow the recipient to forward the task to another user.

Field	Description
Due Date	<p>You can set a due date for the task received in the inbox, giving a deadline for the response from the time that the task arrives. Due dates can be set to minutes, hours, days, months, and yearly durations.</p> <p>Choose one of the following due date options:</p> <ul style="list-style-type: none"> • No Due Date • Static Duration <p>Enter an integer between 1 and 999, and select the unit (minutes, hours, days, months, and years).</p> <ul style="list-style-type: none"> • Dynamic Duration <p>Select a number input between 0 and 2147483647 from the process content.</p> <ul style="list-style-type: none"> • Reference Date <p>Select a date input (UTC + 0) from the process content.</p>
Inputs	You can map the process metadata fields here, ensuring that the correct information persists throughout the process.
Outputs	This section provides an overview of the process metadata included in the form. This is a read-only section.

Example of a Task

The following is an example of a task received to the inbox, with the fields taken from the form settings configured using the process editor.

The screenshot shows the SAP Fiori task inbox interface. On the left, there is a sidebar titled "All Tasks (1)" with a search bar containing "tom." and a refresh button. A single task card is visible, titled "Purchase Requisition Approval" with a priority level of "High". The main content area is titled "PR Approval Form" and contains the following details for the task:

Purchase Requisition - Approval

Please approve or reject following request.

Requestor Name: Tom

Material: IPHONE-12

Quantity: 1

Requestor Comments: Can you please approve this request?

Manager Comments:

Guidelines for Specifying Recipient Users

When you specify recipient users for a user task, consider the following:

i Note

Carefully consider the impact that the changes described here might have on your overall scenario. Changing certain settings after productive use has started can have a negative impact on scenarios that are incompatible with the change. If

applicable, use mechanisms that restrict the impact to the specific scenario.

- Evaluate whether you can use "recipient groups" instead of "recipient users" because there are limits as to how many recipient users may be specified. If you can, you must configure the assignment of users to groups in the identity management-related function of the platform or the identity management back-end systems. This has, for example, the benefit that the assignment of a task to a certain user can be removed using these central identity management functions instead of in the workflow definition and related locations. This usually improves compliance with company and legal requirements. For example, removing an assignment typically becomes effective as soon as authentication tokens expire.
- If you cannot avoid specifying user names using constants or expressions, make sure that you apply the necessary lifecycle actions on the respective events to achieve compliance. For example, use the administrative REST APIs of SAP Build Process Automation, to remove recipient users when they should no longer be assigned to a task because they left their department or the company. Also, ensure that user interfaces that allow configuration of user IDs apply appropriate validation on the task recipients.
- Ensure that the case and spelling of the user ID matches the respective fields of the authentication tokens exactly. It is important that lower or upper case is also considered, because SAP Build Process Automation matches them as is. SAP Build Process Automation must also consider case sensitivity for user names that look like email addresses. There is no metadata that indicates whether user names are actual email addresses or whether case sensitivity is irrelevant. For this purpose, check your identity management system and the related configuration of SAP Business Technology Platform.
- Check the **User ID Source** and its related settings. Consider using "E-Mail" as the configured value, because this might improve consistency of user names in a scenario.
- Ensure that the identifiers, as validated against the user database, are provided. Do not rely, for example, on user names as entered directly by the user.
- Evaluate whether you can disable the creation of "shadow users". In certain constellations, this prevents users from logging in with user names that do not correspond to the canonical identifier, but use a different case.
- Evaluate whether you can configure that user inputs are automatically converted to the expected case. If you can, see the documentation of the SAP Cloud Identity Services feature **Apply Function to Subject Name Identifier** in [Convert Subject Name Identifier to Uppercase or Lowercase](#) or the respective configuration of your custom identity management system.

Configure a Start Trigger Form with SAP Build Work Zone

Using this trigger, you can start a business process.

Prerequisites

You have set up SAP Build Work Zone. See [Configure SAP Build Work Zone for SAP Build Process Automation](#) and particularly [Configure SAP Build Work Zone Content](#).

Procedure

- Start the Site Manager:
 - In your SAP BTP Cockpit subaccount, choose **Services** **Instances and Subscriptions**.
 - Search for your SAP Build Work Zone application and choose the **Go to Application** icon next to the app name.
- Open **Content Manager** **My Content**.
- Create a local copy of the **Process Trigger** app by selecting the **Process Trigger** item and choosing **Create a Local Copy**.

4. To configure the local copy, choose **Edit**

- Enter a title.
- On the **Navigation** tab, enter a parameter named **uri**.

To do so:

- In the process editor of SAP Build Process Automation, select the trigger and on the Trigger Settings panel, copy the configuration parameter URL.
 - Paste the configuration parameter URL into the **Default Value** field.
- On the **Translation** tab, edit any translated texts as needed.
 - Save your changes.

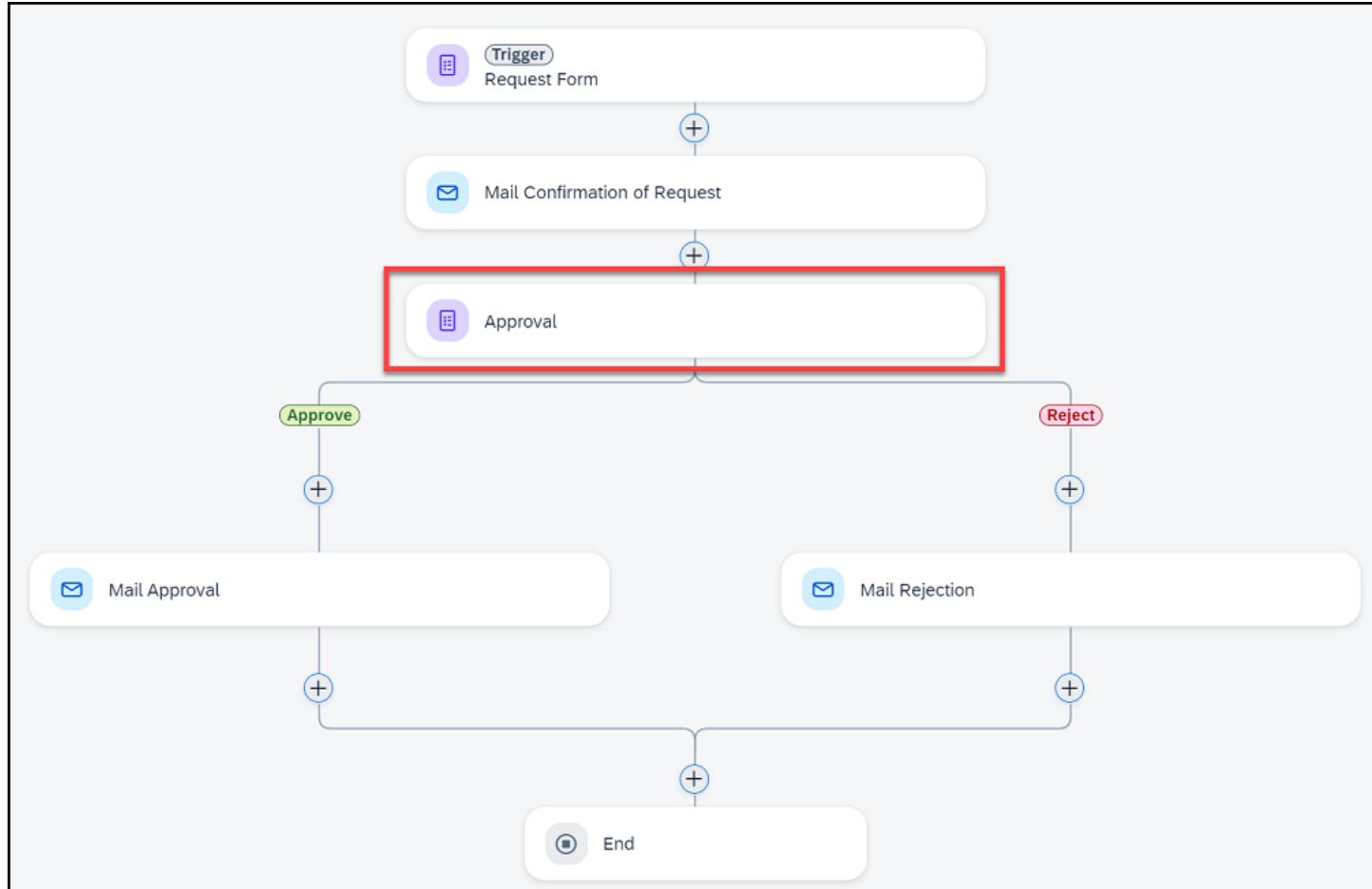
5. Assign the app to your users using standard means of SAP Build Work Zone.

Create an Approval Form

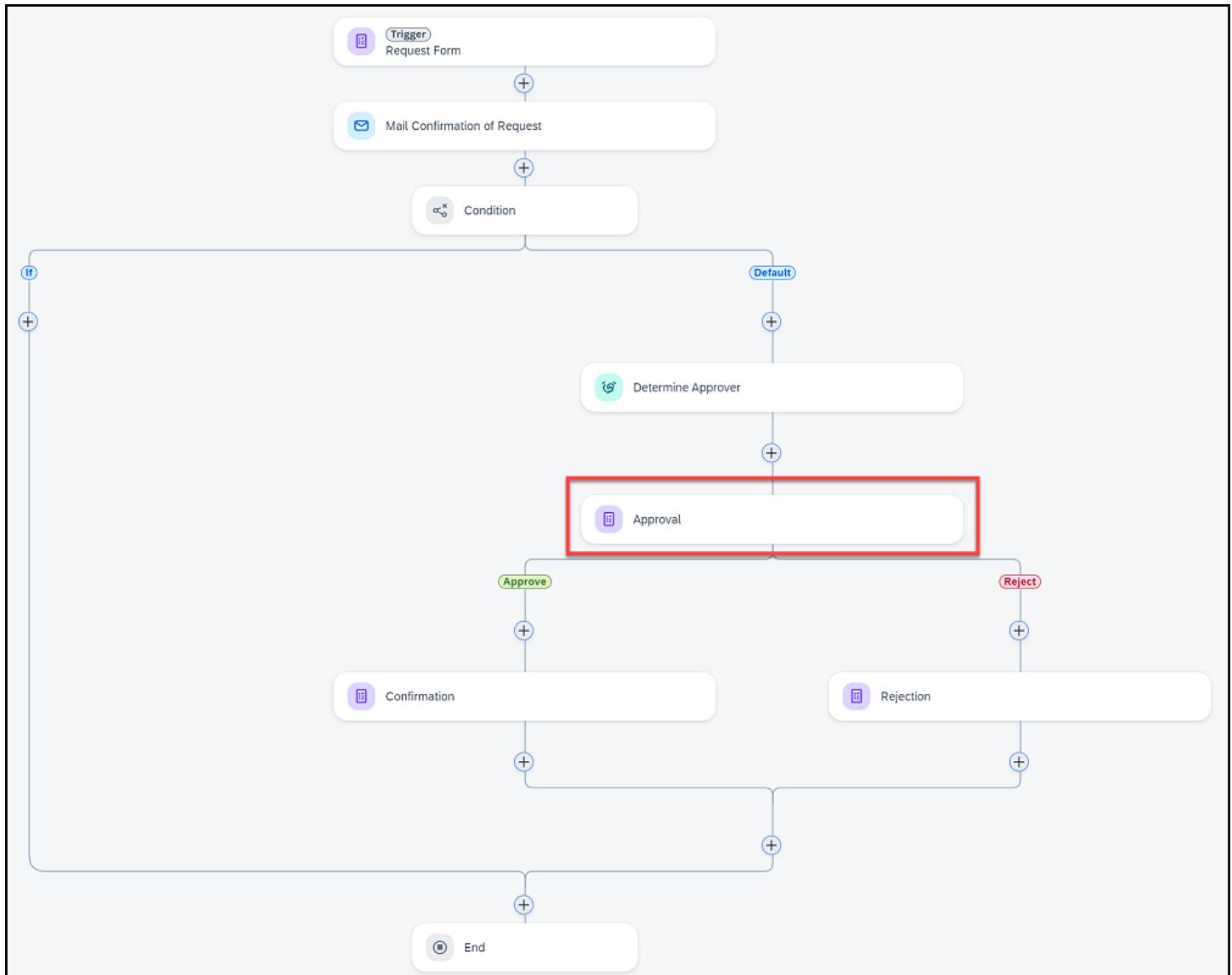
Approvals are an important part of business processes, whether performed manually or automatically approved based on process conditions. With SAP Build Process Automation, you can manage approvals by creating and adding an approval form to a business process.

Context

In the following simple request process, an approval is needed for all submitted requests:



And in the following investment approval request, an approval is only needed when an investment value is higher than a specified amount:



When a process is running, the approver receives a task in their inbox containing the request information. They can then choose to accept, query, or reject the request. Their decision then determines how the process proceeds.

The following is an example of a task received to the inbox, with the fields taken from the form settings configured using the process editor.

All Tasks (1)	
tom.	X
REFRESH	
Purchase Requisition Approval	
High	

PR Approval Form

Purchase Requisition - Approval

Please approve or reject following request.

Requestor Name
Tom

Material
IPHONE-12

Quantity
1

Requestor Comments
Can you please approve this request?

Manager Comments

Procedure

1. From the process editor canvas, click **+** and select **Approval - New Approval Form**.
2. Enter a **Name**, **Identifier**, and optionally a **Description**.
3. Choose **Create**.

The approval form element is now added to your process editor canvas.

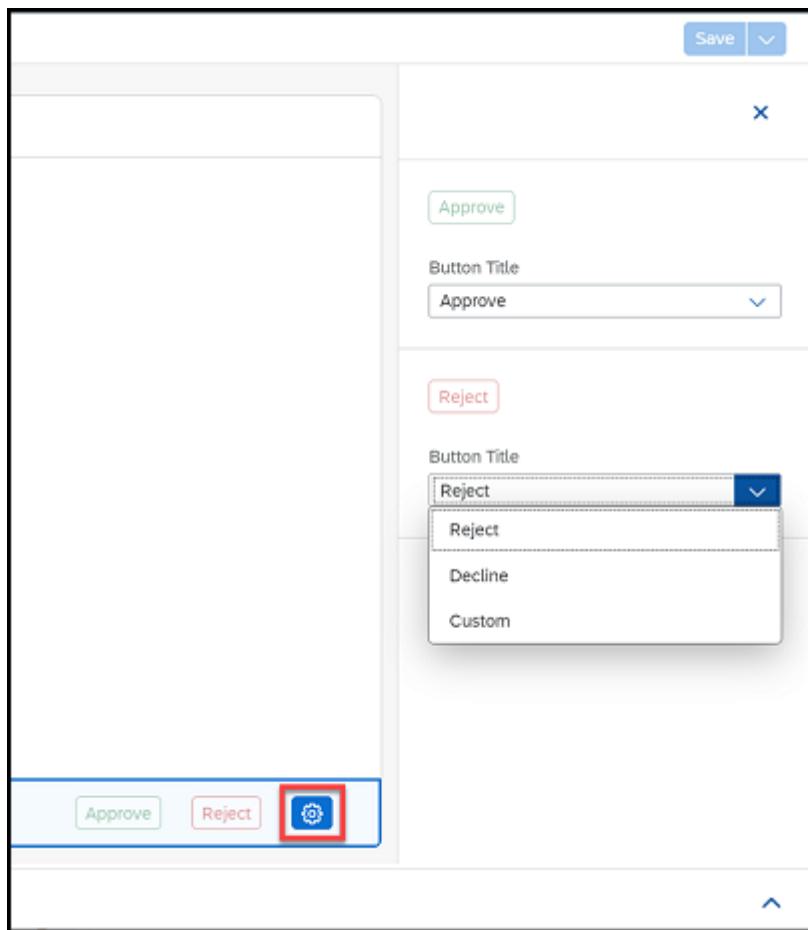
4. Open the form editor by double-clicking the approval form element on the canvas.

5. Configure your approval form as required using **Layout** and **Input Fields**.

For more information about input fields, see [Form Input Fields](#).

6. **Optional:** Customize the default outcome buttons. You can adapt the default **Approve** button to **Accept** or a **Custom** text, and the default **Reject** button to **Decline** or a **Custom** text. Choose the cogwheel icon, and select the required text from the **Button Title** dropdown.

If you select **Custom**, enter your own text. This text is then only available in the language in which you entered it. The predefined labels are translated. If several labels are translated using the same term in the target language, then the application only shows one label.



7. Save your changes.

The approval form has now been saved to your process.

8. Navigate back to the process editor.

9. Click on the approval form element on the canvas, opening the approval settings menu.

10. Configure your approval settings as needed.

For more information about configuring approval settings, see [Configure Settings for Forms and Approval Forms](#).

11. Save your changes.

The configured approval form is added to your business process.

Import an SAPUI5 Form

Import an SAPUI5 form and use it in your process in SAP Build Process Automation. This allows you to create SAPUI5 apps and use them as task UIs in your process.

Prerequisites

- You've created an SAPUI5 application. For more information about creating an application, see [Creating a Custom Task UI](#).
- You've added the namespace "sap.bpa.task" to the manifest.json for your application and you've defined the inputs, outputs, and outcomes for the application. For more information, see [Technical Information for Adapting the SAPUI5 Application](#).
- You've adapted the Component.js file and the App.view.xml file. For more information, see [Technical Information for Adapting the SAPUI5 Application](#).
- You've deployed your application. For more information, see [Build and Deploy](#) in the documentation for SAP Business Application Studio.

Context

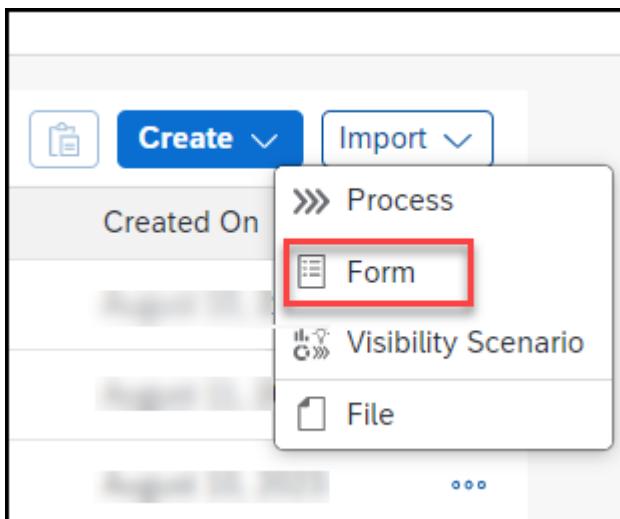
When you import a SAPUI5 form into SAP Build Process Automation, you can use this SAPUI5 app in the process editor in the same way as you use any other form. You can map the inputs and outputs of your form to the rest of the process and you can also use the outputs of the previous task in your form.

i Note

You cannot use an SAPUI5 form as a start form.

Procedure

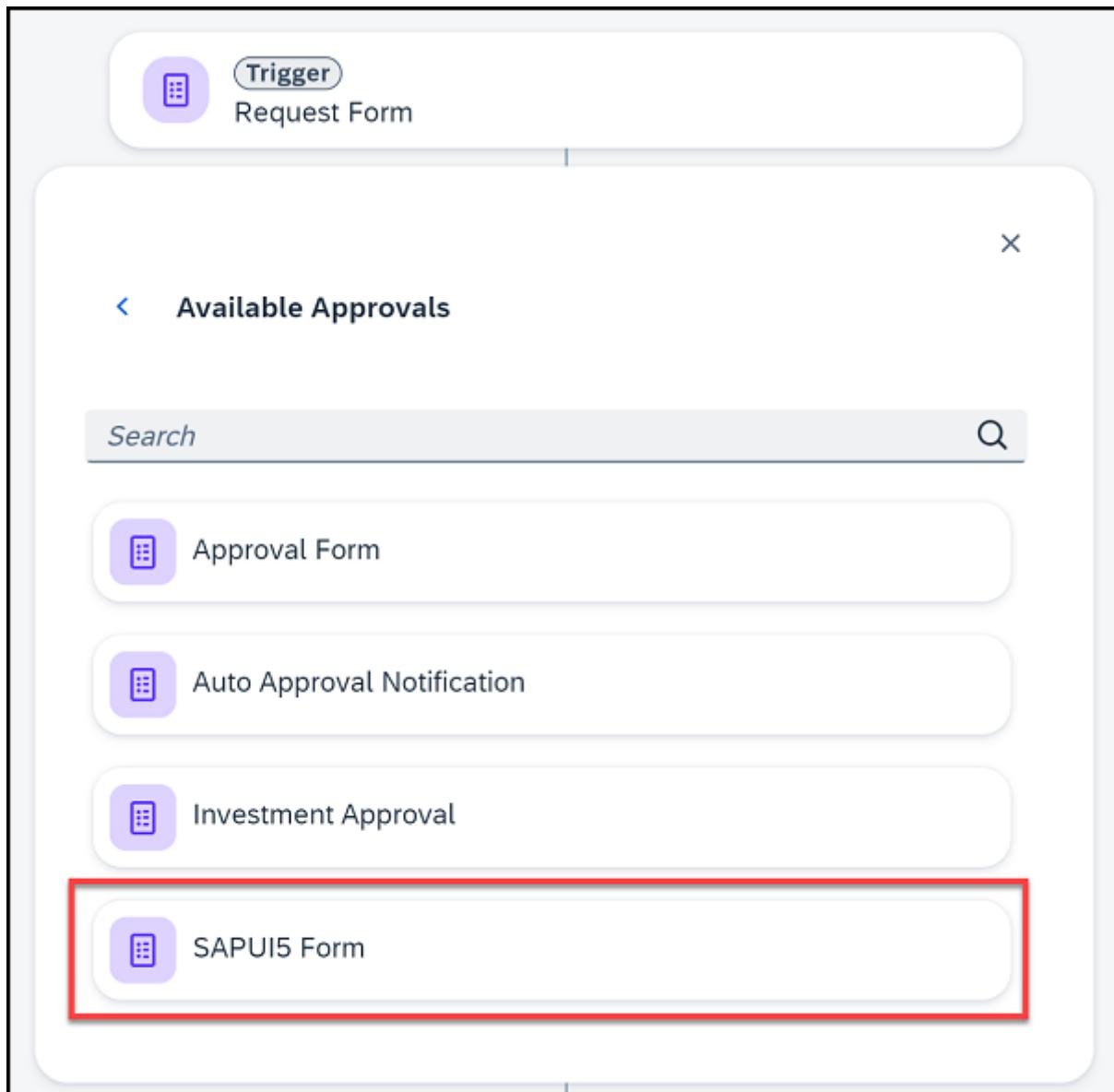
1. In the Overview tab of your project, choose **Import > Form**.



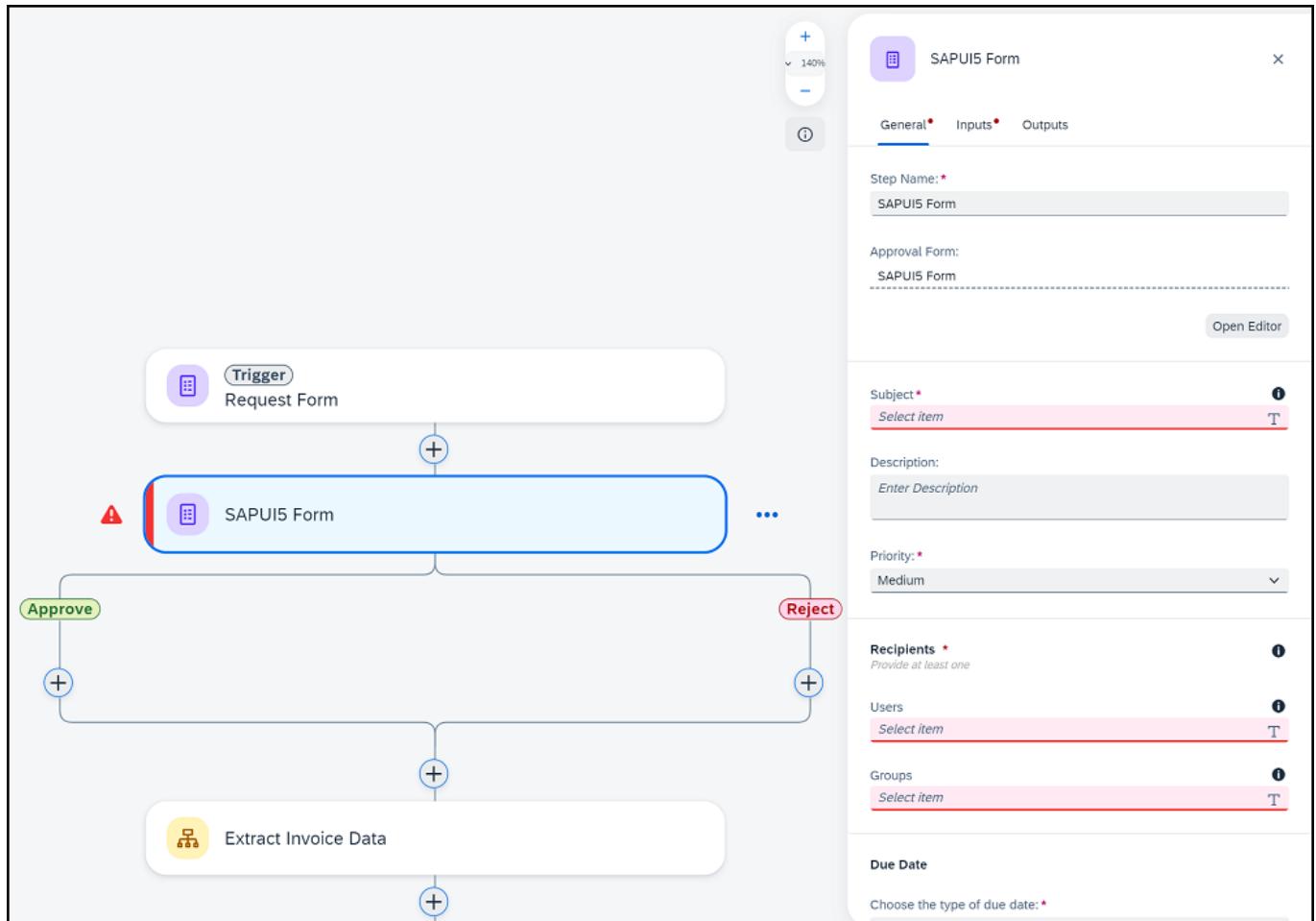
2. In the **Import Form** screen, enter the **Application ID** and the **Application Version** from the manifest.json.

The form is imported and the manifest URL is displayed. You can now add the form to your process.

3. From the process editor canvas, choose **+**, then **Form** or **Approval** (depending on the category of the form) and choose your SAPUI5 form.



4. Choose the form element on the canvas to open the form settings.



5. Configure your form settings as required. For more information, see [Configure Settings for Forms and Approval Forms](#).

6. Save your changes.

The configured SAPUI5 form is added to your business process.

Technical Information for Adapting the SAPUI5 Application

To use SAPUI5 applications as forms in SAP Build Process Automation, you need to add the "sap.bpa.task" namespace to the manifest.json file. You also need to adapt the Component.js file and the App.view.xml file in accordance with the manifest.json. The examples shown here are all compatible with each other.

Configuring the manifest.json File

Add the namespace "sap.bpa.task" to the manifest.json for your application. The following is an example:

❖ Example

```
{
  ...
  "sap.bpa.task": {
    "_version": "1.0.0",
    "outcomes": [
      {
        "id": "approve",
        "label": "Allow"
      }
    ]
  }
}
```

```

},
{
  "id": "reject",
  "label": "Deny"
}
],
"inputs": {
  "$schema": "http://json-schema.org/draft-07/schema",
  "title": "input",
  "type": "object",
  "required": ["newText"],
  "properties": {
    "newText": {
      "type": "string",
      "title": "Textfield",
      "description": "Description for Textfield"
    }
  }
},
"outputs": {
  "$schema": "http://json-schema.org/draft-07/schema",
  "title": "output",
  "type": "object",
  "required": ["newText"],
  "properties": {
    "newText": {
      "type": "string",
      "title": "Textfield",
      "description": "Description for Textfield"
    }
  }
},
"category": "approval"
}
}

```

The namespace must have the following properties:

- The `_version` must be the namespace version.
- The `category` can be either `standard` or `approval`, depending on the type of form you want to use in SAP Build Process Automation. The outcomes depend on the category you choose.
- The outcomes can be an array of objects that have an `id` and a `label`. The `id` can be `submit`, `approve` or `reject`. You can enter any text for the `label`.
 - If the `category` is `standard`, the outcome must be `[{id:"submit", label: "AnyLabel"}]`
 - If the `category` is `approval`, the outcome must be `[{id:"approve", label:"SomeLabel"}, {id:"reject", label: "SomeLabel2"}]`.

i Note

The outcomes you define in the `manifest.json` determine the actions that the application has to create for [My Inbox](#).

- The inputs must be a draft-7 JSON schema and define the context object you receive with the [Workflow API for Cloud Foundry](#). For more information, see the [Inputs and Outputs supported by SAP Build Process Automation Forms](#) section below.
- The outputs must be a draft-7 JSON schema and define the context object you have to send with the [Workflow API for Cloud Foundry](#). For more information, see the [Inputs and Outputs supported by SAP Build Process Automation Forms](#) section below.

Configuring the Component.js File

Once you've created your SAPUI5 application, you need to adapt the Component.js file.

Replace the definition of the Component.js file, which is the second parameter of UIComponent.extend, using the following example:

❖ Example

```
{
  metadata: {
    manifest: "json",
  },
  /**
   * The component is initialized by UI5 automatically during the startup of the app and can
   * @public
   * @override
   */
  init: function () {
    // call the base component's init function
    UIComponent.prototype.init.apply(this, arguments);

    // enable routing
    this.getRouter().initialize();

    // set the device model
    this.setModel(models.createDeviceModel(), "device");

    this.setTaskModels();
    const rejectOutcomeId = "reject";
    this.getInboxAPI().addAction(
      {
        action: rejectOutcomeId,
        label: "Deny",
        type: "reject",
      },
      function () {
        this.completeTask(false, rejectOutcomeId);
      },
      this
    );
    const approveOutcomeId = "approve";
    this.getInboxAPI().addAction(
      {
        action: approveOutcomeId,
        label: "Approve",
        type: "approve",
      },
      function () {
        this.completeTask(true, approveOutcomeId);
      },
      this
    );
  }
}
```

```

        label: "Allow",
        type: "accept",
    },
    function () {
        this.completeTask(true, approveOutcomeId);
    },
    this
);
},
setTaskModels: async function () {
    // set the task model
    var startupParameters = this.getComponentData().startupParameters;
    this.setModel(startupParameters.taskModel, "task");

    // set the task context model
    var taskContextModel = new sap.ui.model.json.JSONModel(
        this._getTaskInstancesBaseURL() + "/context"
    );
    this.setModel(taskContextModel, "context");

    // parse Date objects and set in own model
    await taskContextModel.loadData(this._getTaskInstancesBaseURL() + "/context");
},
_getTaskInstancesBaseURL: function () {
    return (
        this._getWorkflowRuntimeBaseURL() +
        "/task-instances/" +
        this.getTaskInstanceID()
    );
},
_getWorkflowRuntimeBaseURL: function () {
    var ui5CloudService = this.getManifestEntry("/sap.cloud/service").replaceAll(".", "")
    var ui5ApplicationName = this.getManifestEntry("/sap.app/id").replaceAll(".", "");
    var appPath = `${ui5CloudService}.${ui5ApplicationName}`;
    return `/ ${appPath}/api/public/workflow/rest/v1`;
},
getTaskInstanceID: function () {
    return this.getModel("task").getData().InstanceID;
},
getInboxAPI: function () {
    var startupParameters = this.getComponentData().startupParameters;
    return startupParameters.inboxAPI;
},
completeTask: function (approvalStatus, outcomeId) {
    this.getModel("context").setProperty("/approved", approvalStatus);
    this._patchTaskInstance(outcomeId);
},
_patchTaskInstance: function (outcomeId) {

```

```

const context = this.getModel("context").getData();
var data = {
  status: "COMPLETED",
  context: {...context, newText: context.newText || ''},
  decision: outcomeID
};

jQuery.ajax({
  url: `${this._getTaskInstancesBaseURL()}`,
  method: "PATCH",
  contentType: "application/json",
  async: true,
  data: JSON.stringify(data),
  headers: {
    "X-CSRF-Token": this._fetchToken(),
  },
}).done(() => {
  this._refreshTaskList();
})
},
}

_fetchToken: function () {
  var fetchedToken;

  jQuery.ajax({
    url: this._getWorkflowRuntimeBaseURL() + "/xsrf-token",
    method: "GET",
    async: false,
    headers: {
      "X-CSRF-Token": "Fetch",
    },
    success(result, xhr, data) {
      fetchedToken = data.getResponseHeader("X-CSRF-Token");
    },
  });
  return fetchedToken;
},
}

_refreshTaskList: function () {
  this.getInboxAPI().updateTask("NA", this.getTaskInstanceID());
},
}

```

Configuring the App.view.xml File

If you want to use a text input, replace the App element in your app.view.xml file as follows:

❖ Example

```

<App id="app">
  <Input id ="input" value = "{context>/newText}"></Input>
</App>

```

Inputs and Outputs supported by SAP Build Process Automation Forms

The following schema shows all property definitions a SAP Build Process Automation form can output or handle as input.

↳ Sample Code

```
{
  "$schema": "http://json-schema.org/draft-07/schema",
  "title": "<\"input\" or \"output\">",
  "type": "object",
  "required": [],
  "properties": {
    "newText": {
      "title": "New Text",
      "description": "",
      "type": "string",
    },
    "newTextArea": {
      "title": "New Text Area",
      "description": "",
      "type": "string",
    },
    "newNumber": {
      "title": "New Number",
      "description": "",
      "type": "number"
    },
    "newDate": {
      "title": "New Date",
      "description": "",
      "type": "string",
      "format": "date"
    },
    "newChoice": {
      "title": "New Choice",
      "description": "",
      "type": "array",
      "uniqueItems": true,
      "items": {
        "type": "string",
        "enum": ["Option 1", "Option 2"]
      }
    },
    "newChoice2": {
      "title": "New Choice",
      "description": "",
      "type": "string",
      "enum": ["Option 1", "Option 2"]
    },
    "newDropdown": {
      "title": "New Dropdown",
      "description": ""
    }
  }
}
```

```
"type": "string",
"enum": ["Option 1", "Option 2"]
},
"newCheckbox": {
    "title": "New Checkbox",
    "description": "",
    "type": "boolean"
},
"newTable": {
    "title": "New Table",
    "description": "",
    "type": "array",
    "items": {
        "$ref": "#/definitions/newTableItems"
    }
}
},
"definitions": {
    "newTableItems": {
        "type": "object",
        "required": [],
        "properties": {
            "newText": {
                "title": "New Text",
                "description": "",
                "type": "string"
            }
        }
    }
}
}
```

Update the xs-app.json File

In the `xs-app.json` in your UI module folder (the same folder that also contains the webapp), you must replace the following code snippet:

☰ Sample Code

```
{  
  "source": "^/bpmworkflowruntime/(.*)$",
  "target": "/public/workflow/rest/$1",
  "service": "com.sap.spa.processautomation",
  "endpoint": "api",
  "authenticationType": "xsuaa"
},
```

Use it with the following code snippet instead. Make sure that it is the first entry of the routes array.

```
{  
  "source": "^/api/(.*)$",  
  "target": "$1",  
  "service": "com.sap.spa.processautomation",
```

```
  "endpoint": "api",
  "csrfProtection": true,
  "authenticationType": "xsuaa"
},
```

Troubleshooting

When you build your application, if you receive an error message stating that your UI5 CLI installation is outdated, update the `@ui5/cli` version in the package `.json` file of the UI module. You can use versions from version 3 onwards, for example `"^3.4.0."`.

Add Flow Controls

You can add and configure controls to your business process.

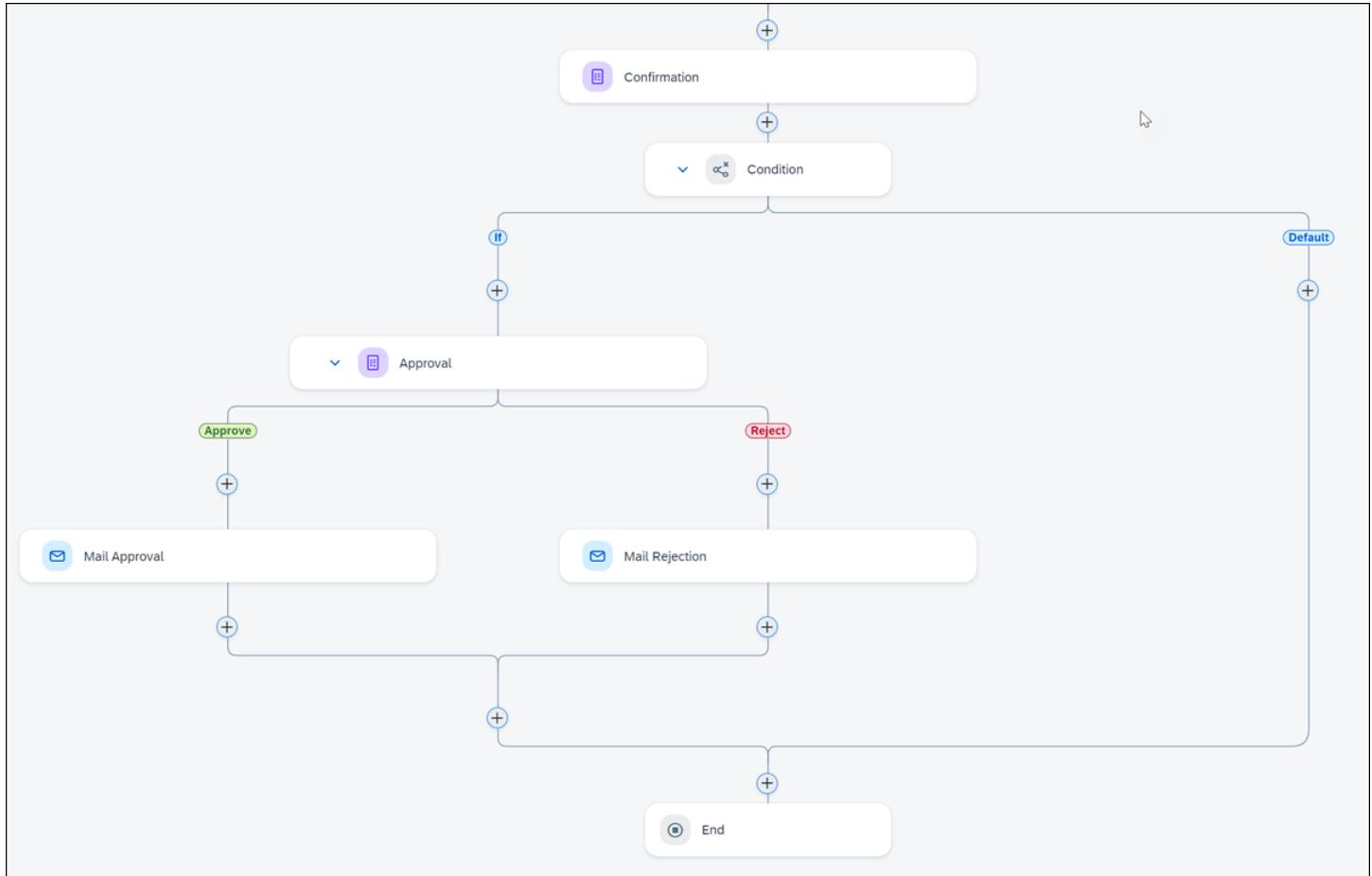
- [Add and Configure Process Conditions](#)
- [Add a Branch to a Process](#)

Add and Configure Process Conditions

You can add and configure conditions to a business process, enabling you to route a running process based on applying IF or DEFAULT rules to the process context.

Context

In the following example, this process condition checks if the value of a submitted request is over 1000. If yes, then the request is sent for approval. If no, then the request is considered as a default request, automatically approved, and a confirmation mail is sent.



For this example, the condition is configured like this:

Edit Branch Condition

Satisfies:

All Any of the following:

is greater than

Add Add Group

Summary:

Apply Cancel

You can also create more detailed conditions by using the **All** or **Any** feature.

In the following example, the process must include **Any** of the following three values for the process to be sent for approval:

- Value of the request is greater than 1000.
- Office location is Potsdam.
- The item requested isn't a laptop.

Edit Branch Condition

Satisfies:

All **Any** of the following:

» Order Amount ×	is greater than	1000	X
» Office Location ×	is equal to	Potsdam	X
» Item Required ×	is not equal to	Laptop	X

Add Add Group

Summary:

if value of » Order Amount is greater than 1000
or value of » Office Location is equal to Potsdam
or value of » Item Required is not equal to Laptop

Apply Cancel

You can also apply both the **All** and **Any** rules to groups of rules, such as in the following example:

In this example, the process must include **any** of the first group conditions AND **all** of the second group of conditions for the request to be approved.

Edit Branch Condition

Satisfies:

All **Any** of the following:

» Order Amount ×	is greater than	1000	X
» Office Location ×	is equal to	Potsdam	X
» Item Required ×	is not equal to	Laptop	X

All Any Remove Group

» Name ×	is equal to	Maya	X
» Delivery Date ×	is earlier than	» Delivery Date ×	X

Add

Summary:

if value of » Order Amount is greater than 1000
or value of » Office Location is equal to Potsdam
or value of » Item Required is not equal to Laptop
or value of » Name is equal to Maya
and value of » Delivery Date is less than » Delivery Date

Apply Cancel

Procedure

This is custom documentation. For more information, please visit the [SAP Help Portal](#)

- From the process editor canvas, choose **►+ (Plus) ► Flow Login ► Condition**.

The condition is added to the process and the side panel is displayed.

- Optional:** Enter a **Condition Name** and a **Branch Name** to customize how the condition is displayed in the process.
- Choose **Open Condition Editor**.
- Configure a minimum of one condition to be satisfied.

For example, the office location must be Potsdam:

Edit Branch Condition

Satisfies:

All Any of the following:

Office Location is equal to Potsdam

Clear All X

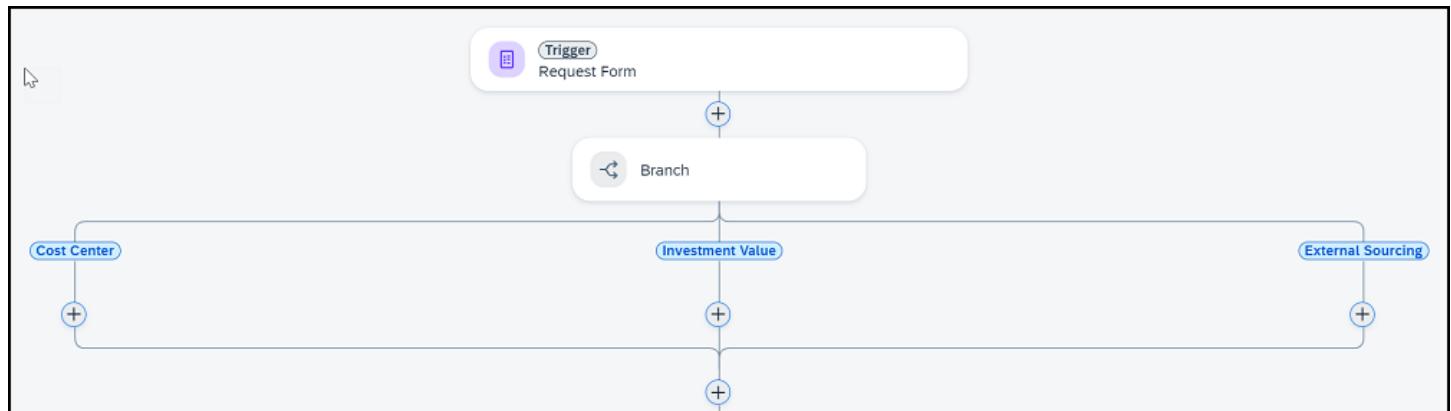
- Choose **Apply**.
- Save your changes.

Add a Branch to a Process

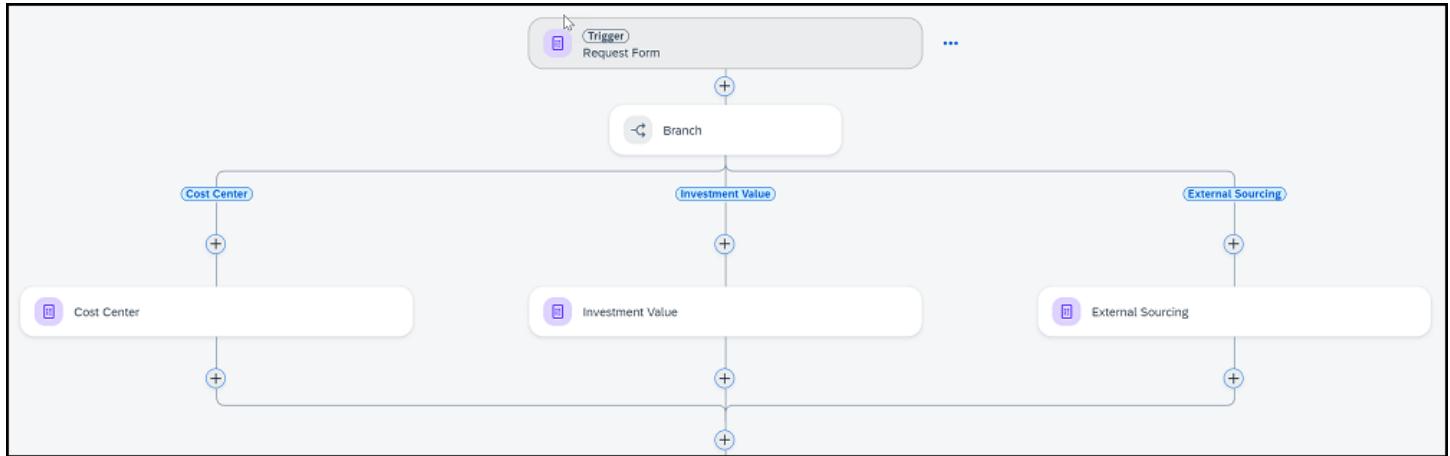
You can add a maximum of 10 parallel branches to your business process, enabling you to run multiple process tasks at the same time. These tasks run concurrently, with the process progressing once all tasks are complete.

Context

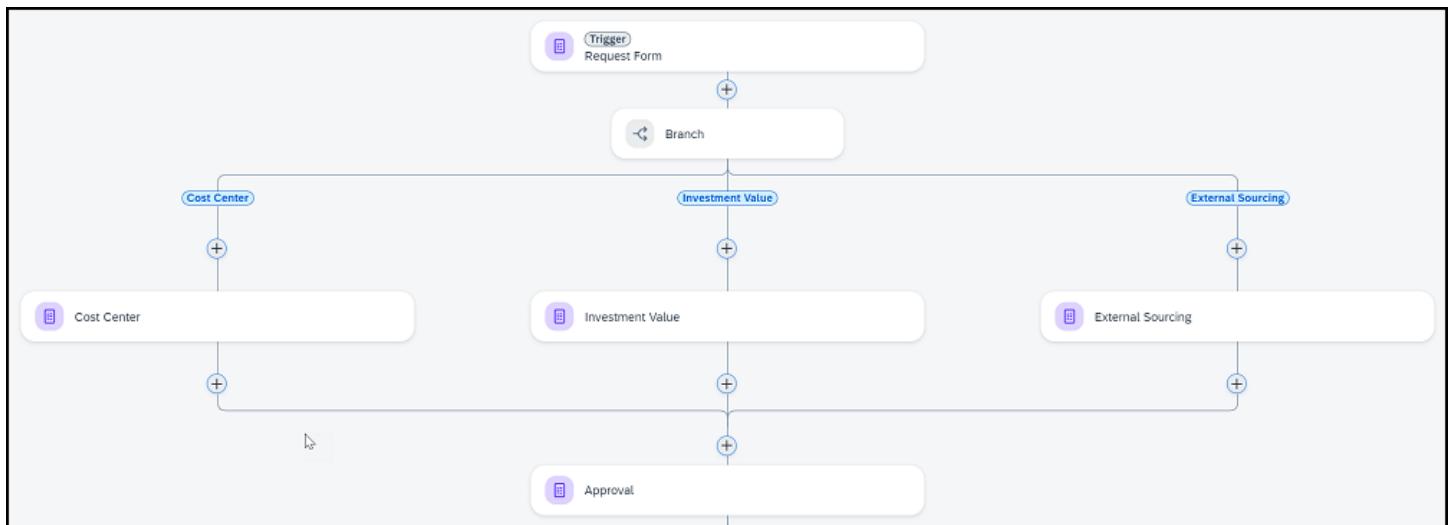
In the following example, three branches have been added to an investment request process and labeled accordingly:



These branches are then enhanced by adding process steps, in this case three additional forms:



An approval form is then added to the process. This approval is only triggered once all three branches have successfully processed.



You can monitor the status of branch tasks in a running process using the [Execution Log](#) in the [Monitoring](#) area.

In this example the process has reached the branch:

EXECUTION LOG

Do

Task "4711 IT costs" available

Aug 5, 2022, 2:08:32 PM

Instance ID: [5339c8fe-14b7-11ed-8f7c-eeee0a9bd7bc](#)

Recipients:

Initiator:

Do

Task "Yes" available

Aug 5, 2022, 2:08:31 PM

Instance ID: [5291a219-14b7-11ed-8f7c-eeee0a9bd7bc](#)

Recipients:

Initiator:

Do

Task "500.000 EUR" available

Aug 5, 2022, 2:08:31 PM

Instance ID: [527fc7c3-14b7-11ed-8f7c-eeee0a9bd7bc](#)

Recipients:

Initiator:

Do

"Branch" reached

Aug 5, 2022, 2:08:31 PM

Do

completed the task "5e6d3522-14b3-11ed-8f7c-eeee0a9bd7bc"

Aug 5, 2022, 2:08:31 PM

Do

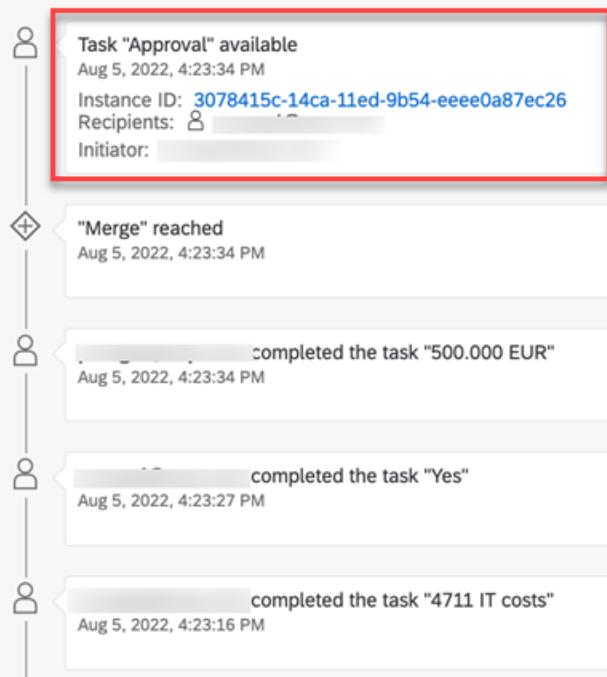
Task "5e6d3522-14b3-11ed-8f7c-eeee0a9bd7bc" available

Aug 5, 2022, 1:40:13 PM

Instance ID: [5ea3fc9e-14b3-11ed-8f7c-eeee0a9bd7bc](#)

And in this example all branch tasks have successfully processed and the approval has triggered:

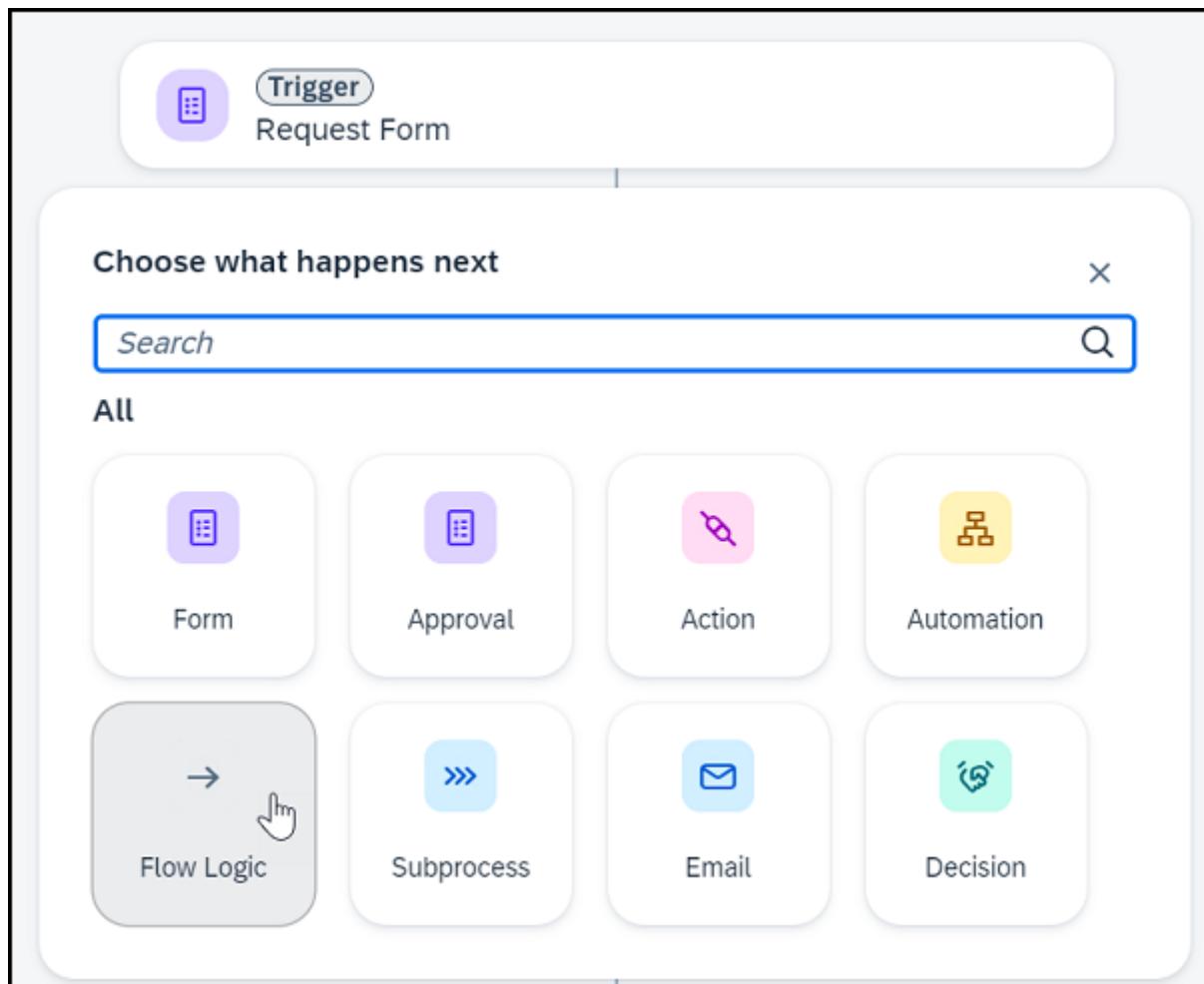
EXECUTION LOG



Procedure

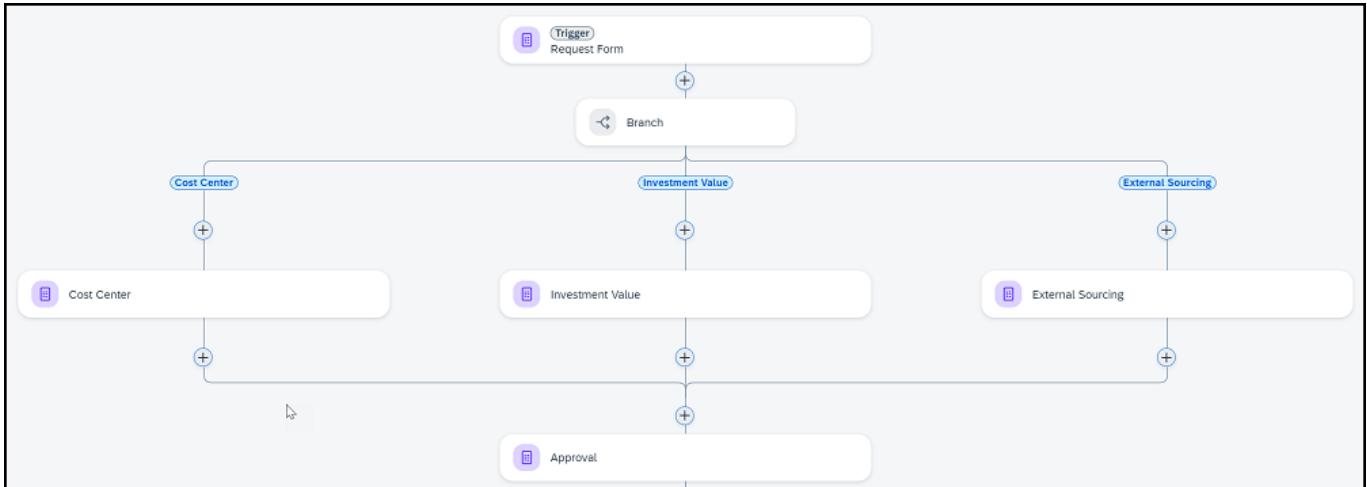
- From the process editor, click **+** and select **Controls - Branch**.

From the process editor, choose **Flow Logic**.

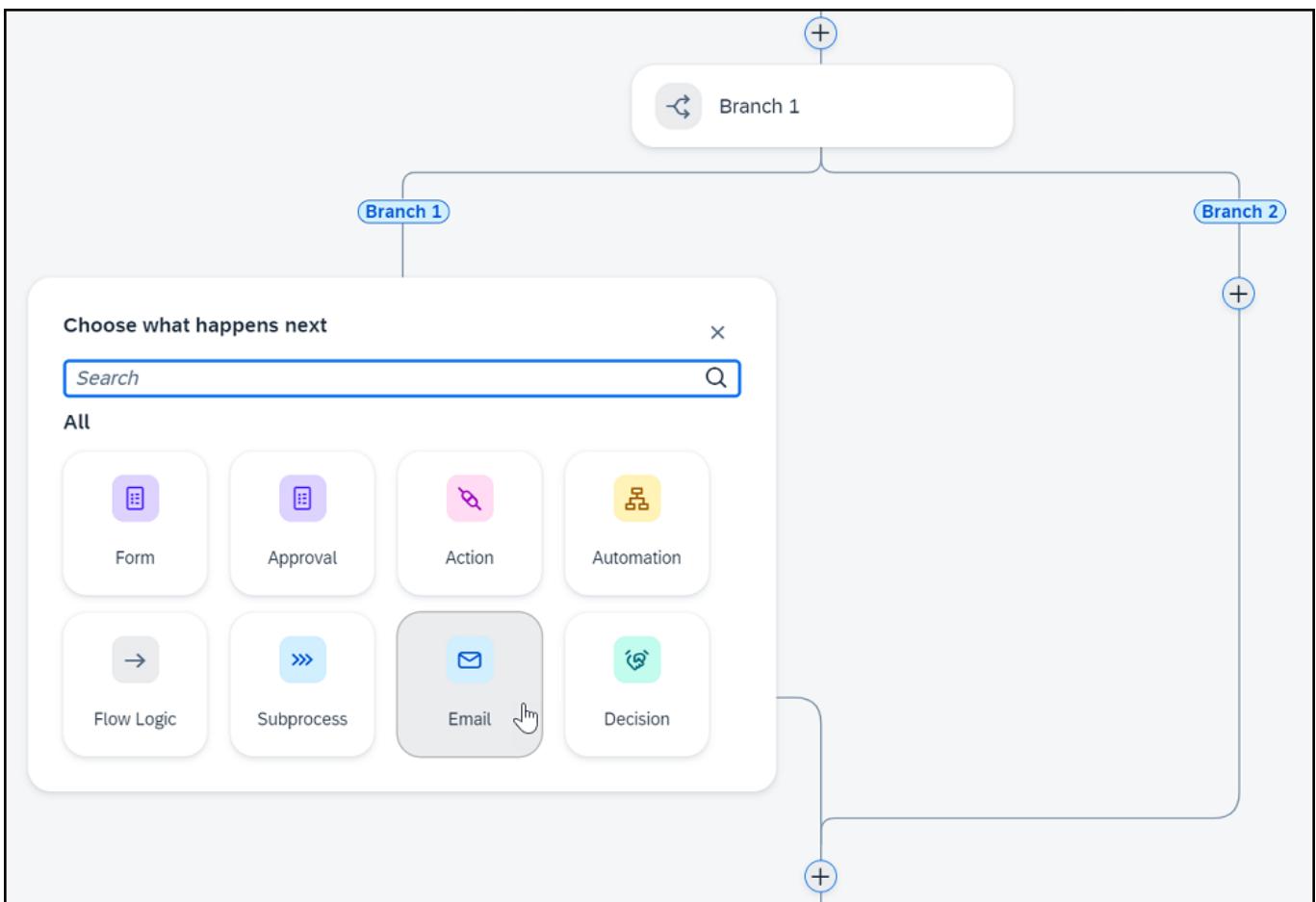


Two branches are added to the process and the branch settings are displayed.

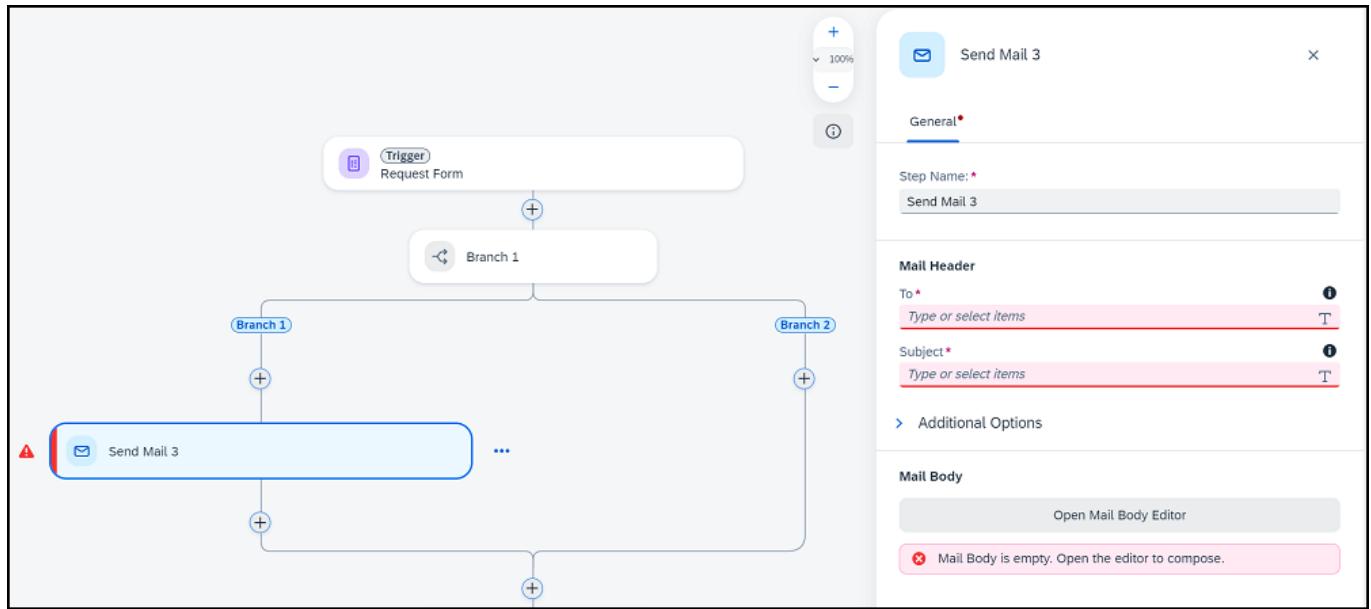
2. Configure the branch accordingly, adding an optional **Branch Name** and additional branches using **Add branch**.



3. Add process steps to each branch using **+** and selecting the required step. In this example, a mail notification is added to the branch.



4. Configure the process step, ensuring that all errors are corrected. In this example, the mail notification is missing the **Mail Header** information.



- Once all branches are configured, save your changes.

The branches have been added to the process and will be processed concurrently when the process is running.

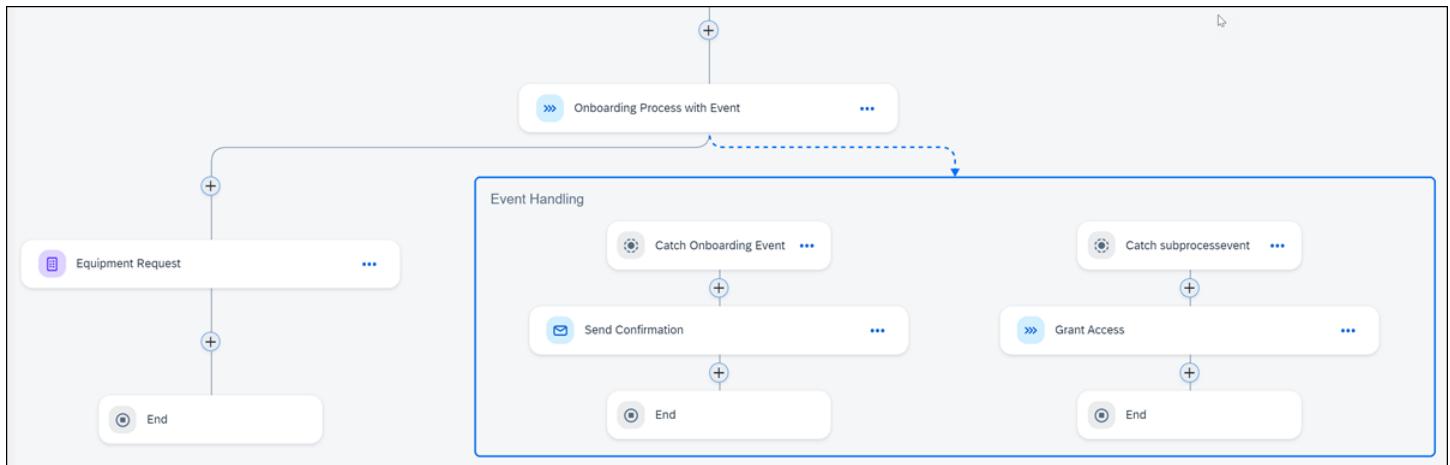
Raise an Escalation Event in Process Editor

You can raise and configure escalation events from anywhere in a business process.

Context

For an escalation event, you only define its name without any action attached. If the process reaches that event, it throws an escalation to its parent process. If you configure event handling for this event in the parent process, then the escalation is caught and you can send, for example, a notification.

In the following sample, we have a parent process with a subprocess that catches two raised events.



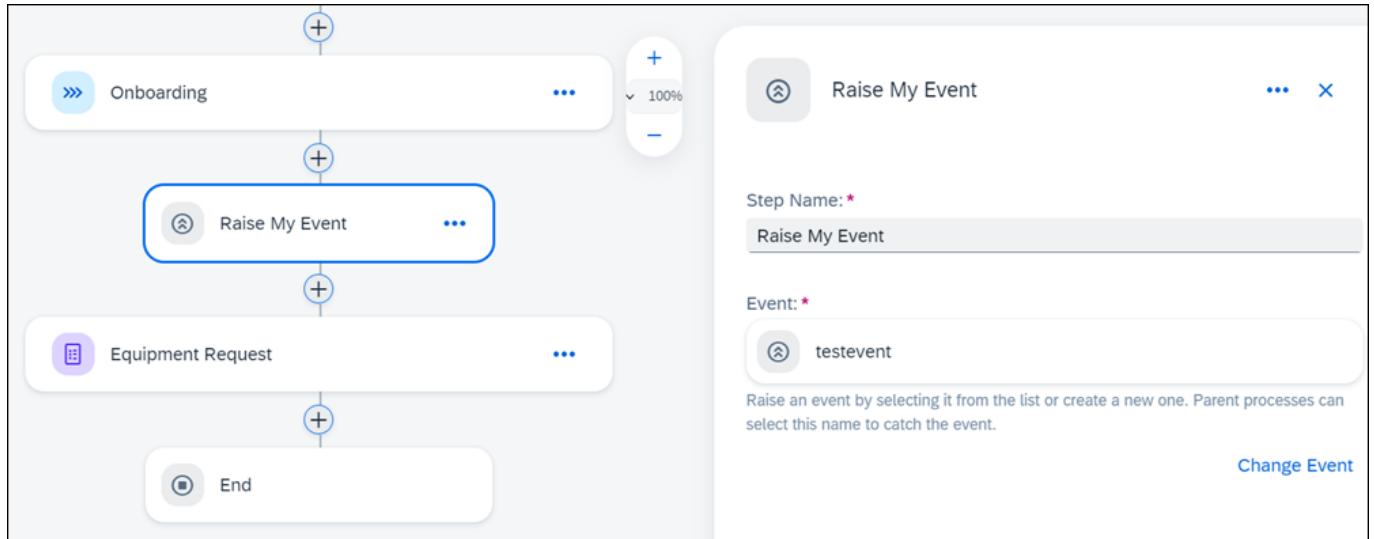
Procedure

- From the process editor canvas, choose **+ (Plus) > Flow Controls and Events > Raise Event**.
- On the **Raise Event** tile, choose **Create Event**.
- Enter a name for the event and choose **Create**.

The event is added to the process.

4. **Optional:** On the side panel, you can change the event by choosing **Change Event**.

You can then select an event from the list that opens or create another event from there.



5. Save your changes.

Handle Events Using the Parent Process

You can catch events from a business process or workflow.

Context

The parent process can catch an event. Not catching an event is not considered an error.

The following types of events can be raised by a subprocess or a workflow:

- Escalation event: This event type can be caught by its parent process.
- Timer events: This event type can be caught by its parent process or a user task, for example, an approval.

Handle Escalation Events

Procedure

1. From the step menu of the process or workflow, choose **Options**, and then choose **Handle Event**.

If your process does not contain a raised event yet, choose **Open Subprocess** to add an event to it.

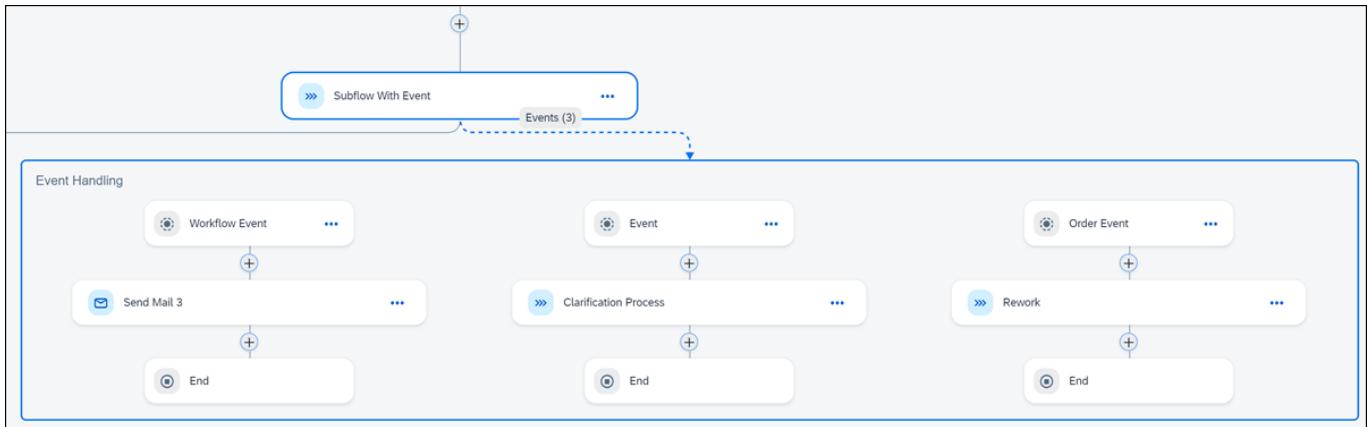
2. On the **Event Handling** tile, choose **Catch Event**.

3. In the catch flow that is created, choose **+** (plus) to add the step to catch the event.

You can catch multiple events, and they are then displayed next to each other. The number of handled events is displayed if you click the process or workflow. If you then click the number, the **Event Handling** tile opens.

4. Configure the event on the side panel.

5. Save your changes.



Handle Timer Events

Procedure

1. From the step menu of the process or workflow, choose **Options**, and then choose **Handle Event**.

If your process does not yet contain a raised , choose **Open Subprocess** to add an event to it.

2. On the **Event Handling** tile, choose **Timer Event**.

3. In the catch flow that is created, choose **+** (plus) to add the step to catch the event.

You can catch multiple events and they are then displayed next to each other. The number of handled events is displayed if you click the process or workflow. If you then click the number, the **Event Handling** tile opens.

4. Configure the event on the side panel.

Select whether the timer is based on the task creation date or a reference date:

- **Task Creation:** Set the toggle to **On Task Creation** to start the event handling directly or to **After Task Creation** to start the event handling only after the completion of the wait duration that you define.
- **Reference Date:** Select a date from the process context list, for example, the sales order date.

5. Save your changes.

Handle a Timer Event with a User Task

You can catch timer events from a user task, for example, an approval or a form.

Procedure

1. From the step menu of the user task, choose **Options**, and then choose **Handle Event**.

2. On the **Event Handling** tile, choose **Timer Event**.

3. In the catch flow that is created, choose **+** (plus) to add the step to catch the event.

You can catch multiple events that are then displayed next to each other. The number of handled events is displayed if you click the process or workflow. If you then click the number, the **Event Handling** tile opens.

4. Configure the event on the side panel.

Select whether the timer is based on the task creation date, the due date, or a reference date:

- **Task Creation:** Set the toggle to **On Task Creation** to start the event handling directly or to **After Task Creation** to start the event handling only after the completion of the wait duration that you define.

- **Due Date:** Start the event handling when the due date of the user task arrives.

If you select this option for a user task that has no due date defined, you get a notification telling you to update the user task first before you can use this option.

- **Reference Date:** Select a date from the process context list, for example, the sales order date. The event handling is started on that date.

5. Save your changes.

Add Actions to a Process

You can embed external skills and capabilities into your SAP Build Process Automation projects using actions. Action projects can either be created in the lobby, allowing you to upload an Open API specification file, or imported from the store to your library.

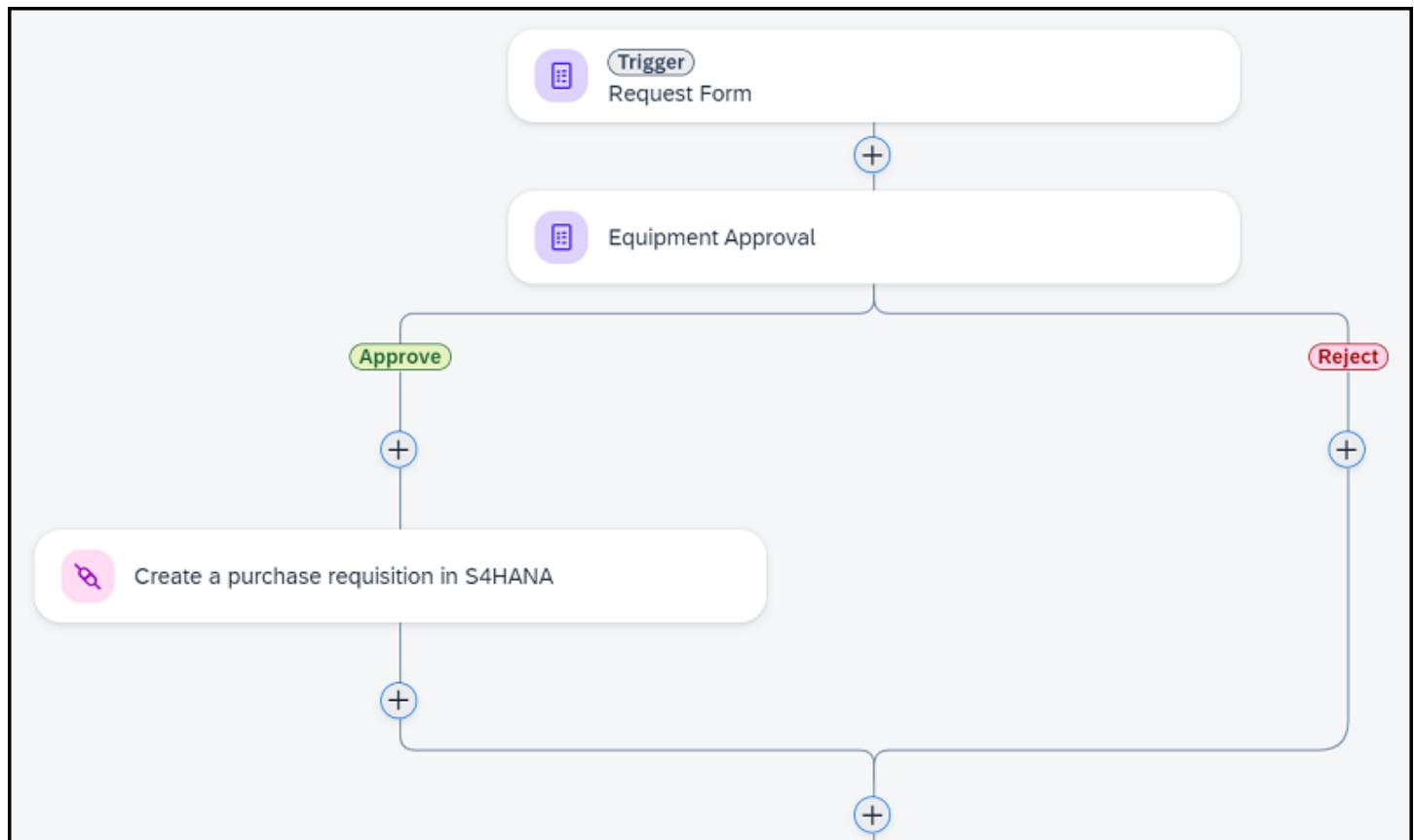
Prerequisites

Before adding an action to your business process, the corresponding action project must be available in your lobby. There are two ways to add an action project to your lobby:

- [Create an Action Project](#)
- [Use the Store](#)

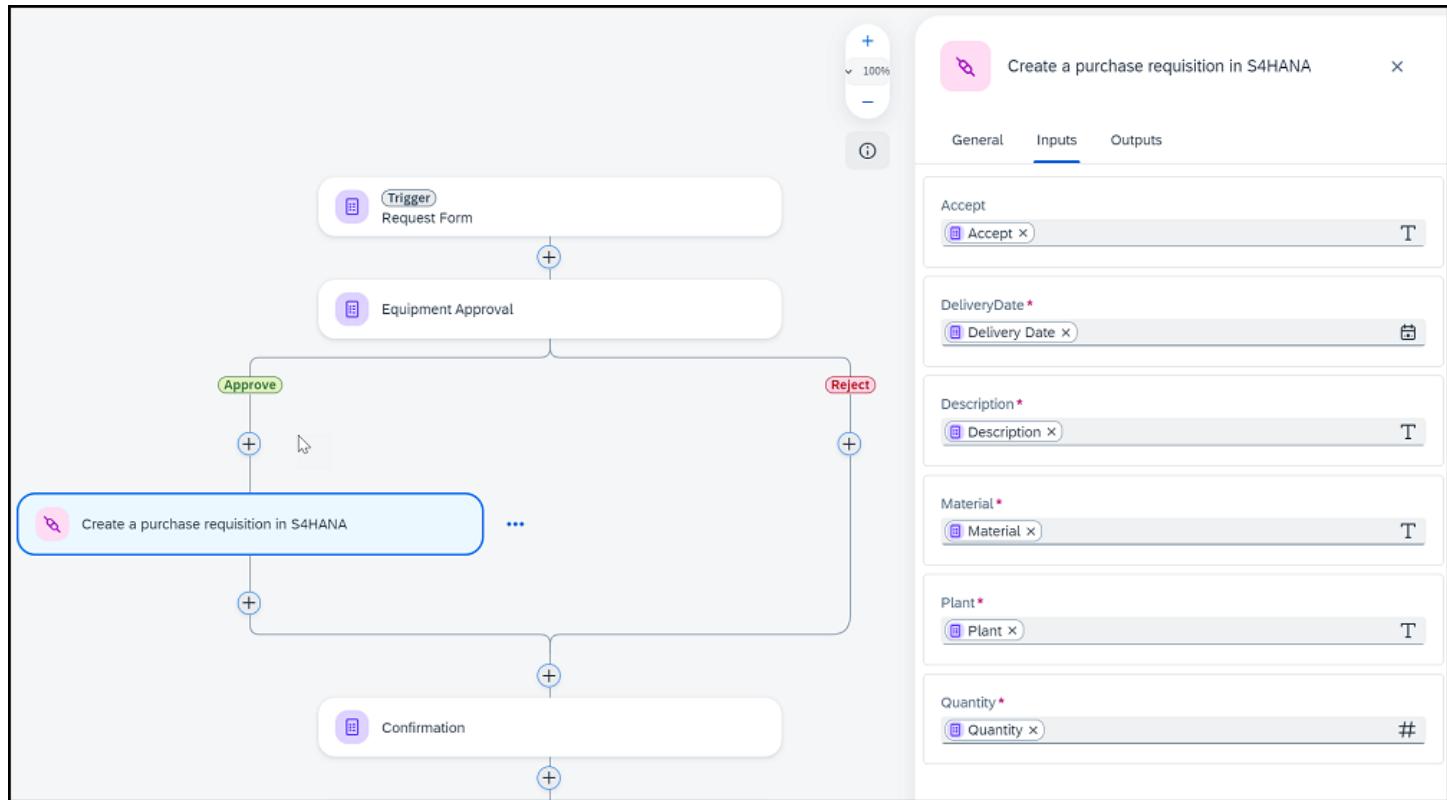
Context

In the following simple request process, a request submission form starts the process running. If the request is approved, an action is used to create a purchase requisition in an external system:



The graphic is explained in the accompanying text.

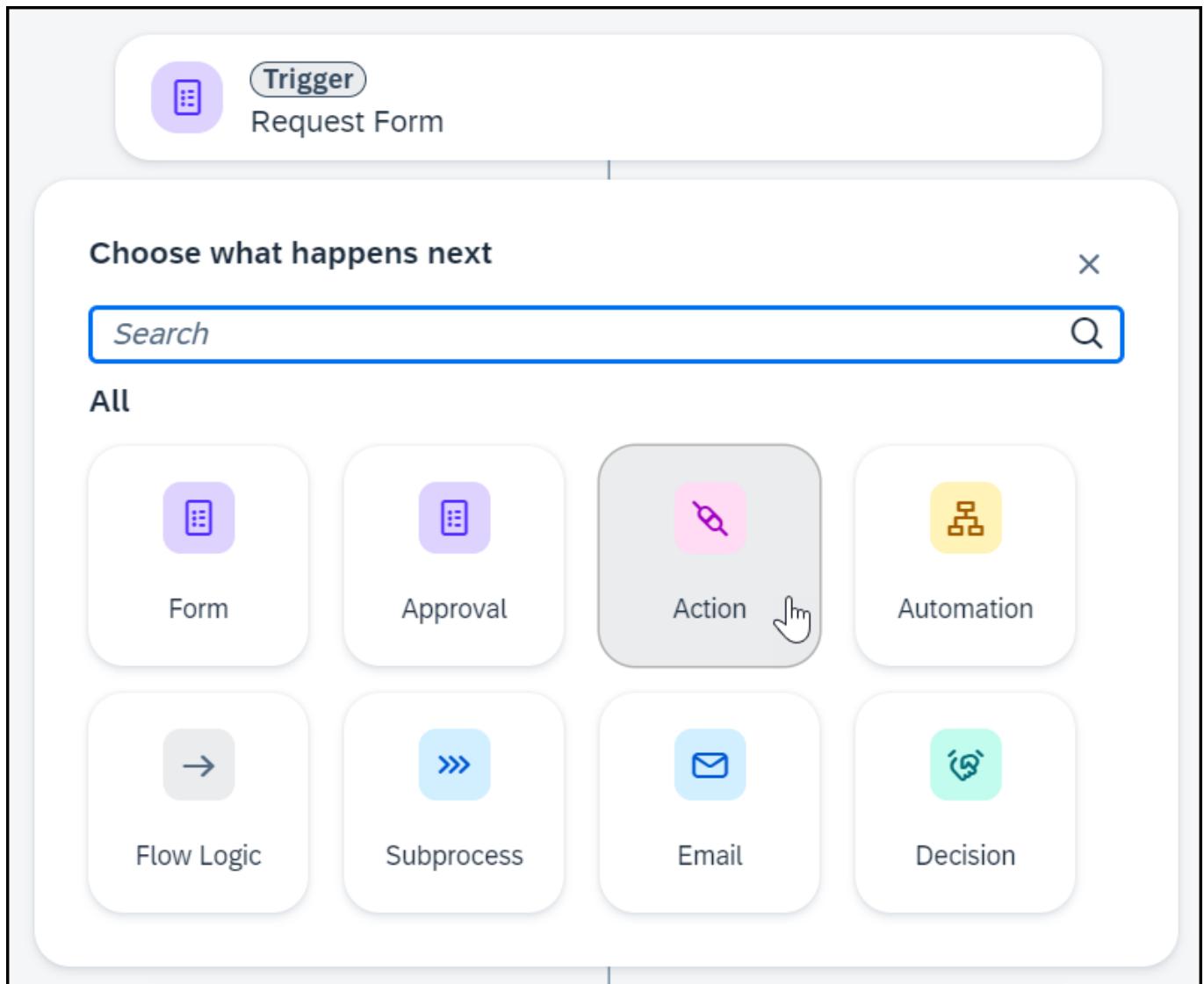
And the inputs of the purchase requisition are mapped from the fields used in the form:



Procedure

- From the process editor canvas, choose **+** and select **Actions - Browse Library**.

From the process editor, choose **►+ > Flow Logic > Parallel Branch**.



2. Search for the action you want to add to your process, and choose **Add**.

Browse library

Search Sort by Artifact Name: Ascending

Action Type	Projects	Line Of Business	Products
Action Add a new pet to the store	Action Add a new pet to the store Add a new pet to the store	Action Add new entity to AllowedStatusesSet	
Project: Julians Petstore Add	Project: PetStoreActionProject Add	Project: MM - POC - ZOMR - Status... Add	
Action Add new entity to AllowedStatusesSet	Action Add new entity to AllowedStatusesSet	Action Add new entity to BusinessUserBusinessRoleAssign...	
Project: MM - POC - ZOMR - Status... Add	Project: MM - POC - ZOMR - Status... Add	Project: faizk Add	
Action Add new entity to BusinessUserSubscriptionAssignm...	Action Add new entity to CustomerOrderItemCollection	Action Add new entity to JobRequisition	

Close

The action is added to your process editor canvas.

3. Click the **Action** on the canvas, opening the configuration side panel.

The screenshot shows a process editor canvas with a workflow step 'Create a purchase requisition in S4HANA'. This step is highlighted with a red warning icon. To the right, a configuration side panel is open for this step. The side panel has three tabs: 'General', 'Inputs', and 'Outputs'. The 'General' tab shows the step name 'Create a purchase requisition in S4HANA'. The 'Inputs' tab has a field 'Destination variable:' which is highlighted with a red border, indicating it is a required configuration item. The 'Outputs' tab is currently empty.

4. Configure your action as needed. Outstanding configuration items are highlighted in red and displayed in the design console.

In this example, the action requires a destination to be configured and for inputs to be mapped.

i Note

When configuring your action, you may need to add and configure other process or automation artifacts before your project can be released and deployed.

5. Save your changes.

Your action has been added to your process. It can now be consumed when the process is running.

Run Step on Behalf Of

With this feature, you can allow business users who participate in the business process to perform an action on external systems or to execute a subprocess or workflow in the same system.

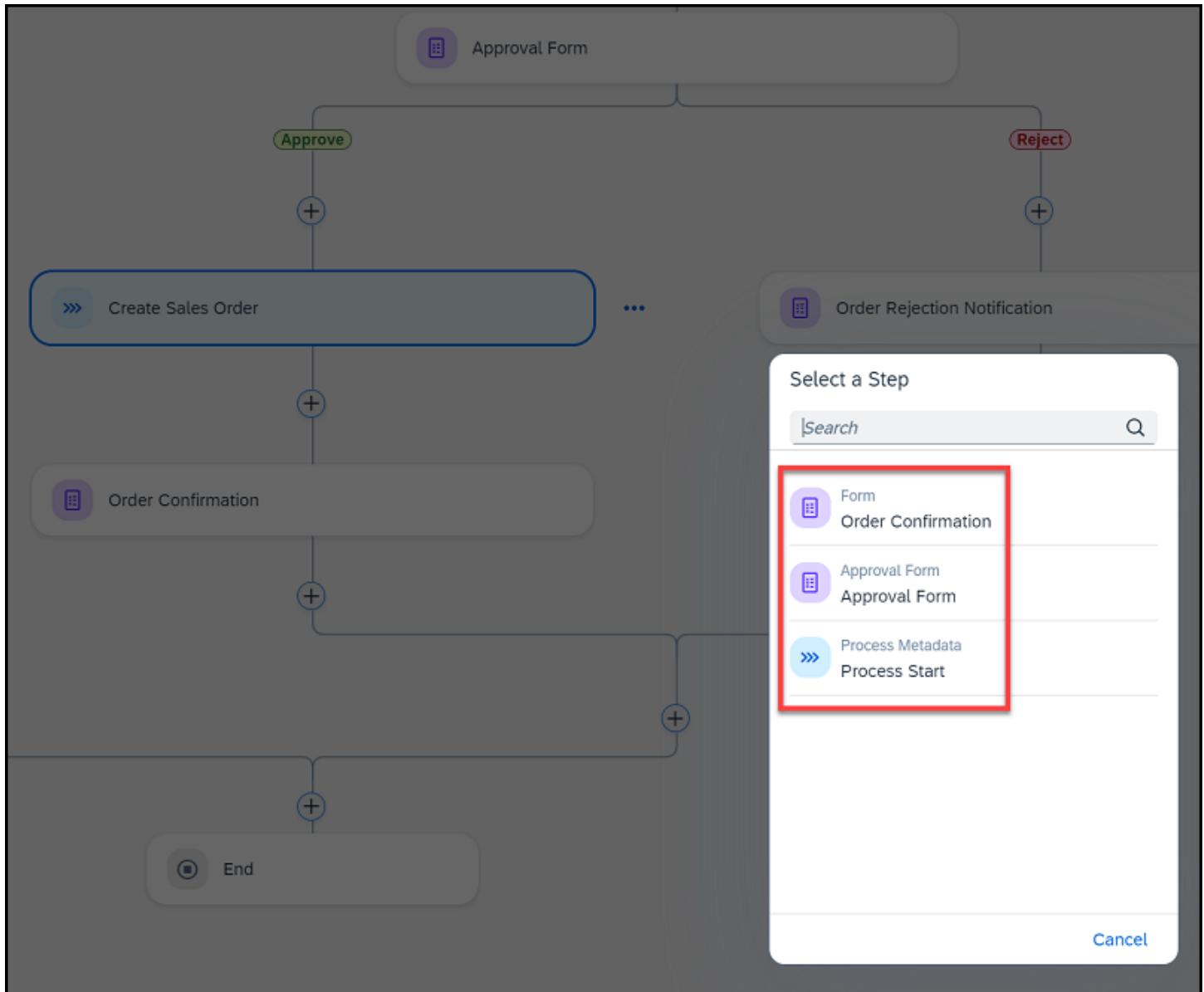
This feature also provides clear information on who triggered the step. This information can help your IT department and is valuable during the audit process.

Prerequisites

- To enable principal propagation, the identity of cloud users gets forwarded to a remote system or service in the Cloud Foundry environment. Principal propagation is possible between two cloud systems or between a cloud and an on-premise system. Set up the connections to the required system using the SAP Destination service. See [Configure SAP Build Process Automation Destinations](#).
- It's important that the step that you are mapping includes the details of a human user who has access to the required system. If the user doesn't have access or if the user is a technical user, the process might fail at runtime.
- The steps eligible to be run on behalf of another user are actions or subprocesses.

Context

You can select the process participant, on whose behalf the step runs, from the process steps. That is, you either map the user who started the process or a user who executed any previous step of the process. That is, you select a step and doing so, indirectly reuse its user in the subprocess. Then, the action or subprocess is executed on behalf of that process participant.



In a procurement process, for example, an employee requests a laptop and gets the manager's approval. As a result, a purchase requisition is created on behalf of the manager who approved the request. Therefore, the purchase requisition is run on behalf of the manager.

Sometimes determining the user of this previous step is tricky:

- We don't recommend using the user of a step inside a condition branch. This is because the outcome of the condition might come from another branch that doesn't include the step you are referring to, in which case, the user that you set isn't available and the process fails.
- By default, selecting a user in **Run step on behalf of** is optional. However, in a sequence of steps like a purchase requisition, it becomes mandatory because the subprocess has an action with **Run step on behalf of** configured to process start.

Example

You create a process A that includes an action for which you set **Run step on behalf of** to the user who started the process.

You create a second process B in which you include process A as a subprocess. Now, this subprocess requires an entry in **Run step on behalf of**.

- It's also possible, that you set up your process well, but then you delete the step that contained the user that you referred to in the action or subprocess. An error message prompts you to enter a user for the action or subprocess.

again.

Related Information

[User Propagation from the Cloud Foundry Environment to SAP S/4HANA Cloud](#)

[User Propagation from the Cloud Foundry Environment to SAP SuccessFactors](#)

[User Propagation between Cloud Foundry Applications](#)

Add a Subprocess to a Process

You can add subprocesses to your main process in the process editor, allowing you to modularize your business process and reuse subprocesses..

Prerequisites

The subprocess must be in the same or a dependent project of your main process in the process editor.

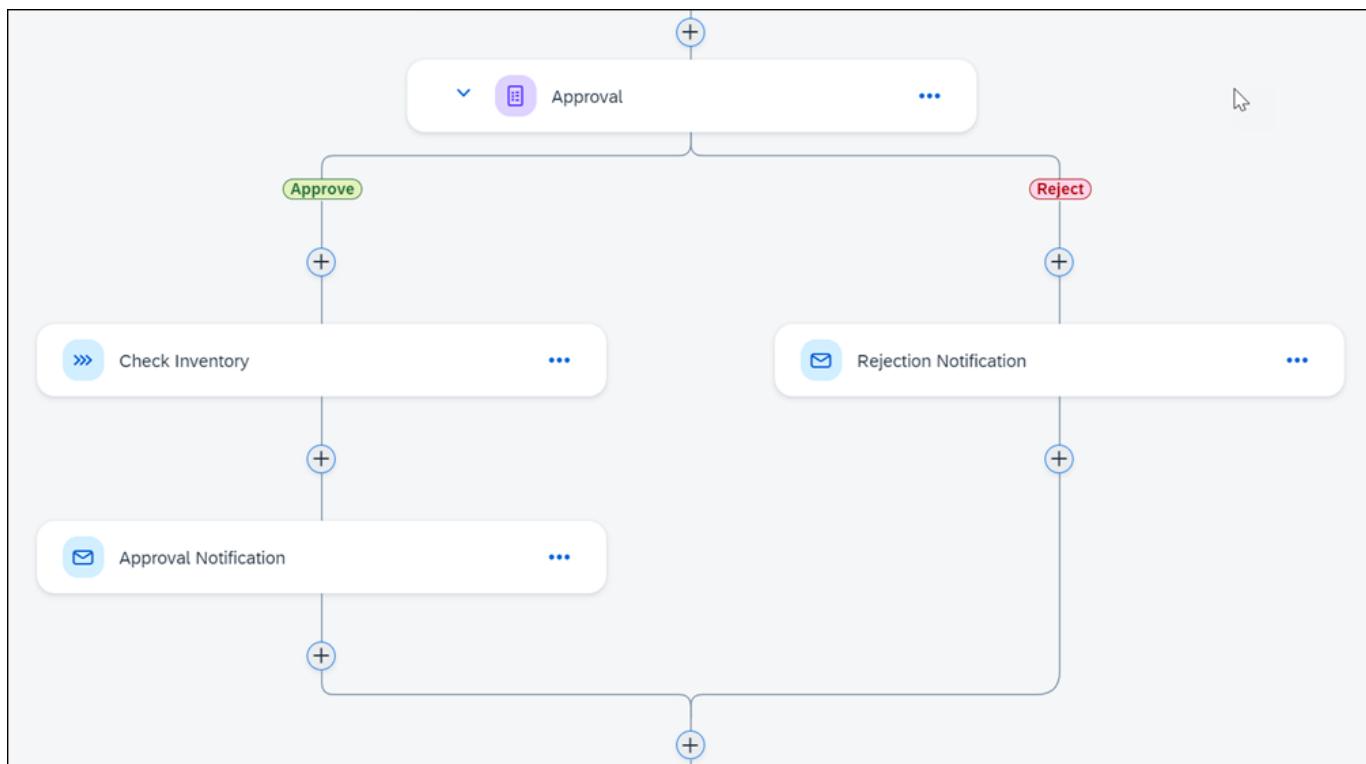
Context

Subprocesses can help you organize your processes. Technically they are simply processes but you can use them to create reuse sequences. Then, you can add them to multiple processes and increase your efficiency and consistency. There are the following types of subprocesses:

- A process that is included, and later run, as part of a main business process.

Eligible subprocesses need to be available in the same business process project. All processes contained within the same project can be a main process or used as a subprocess, depending on how you use them.

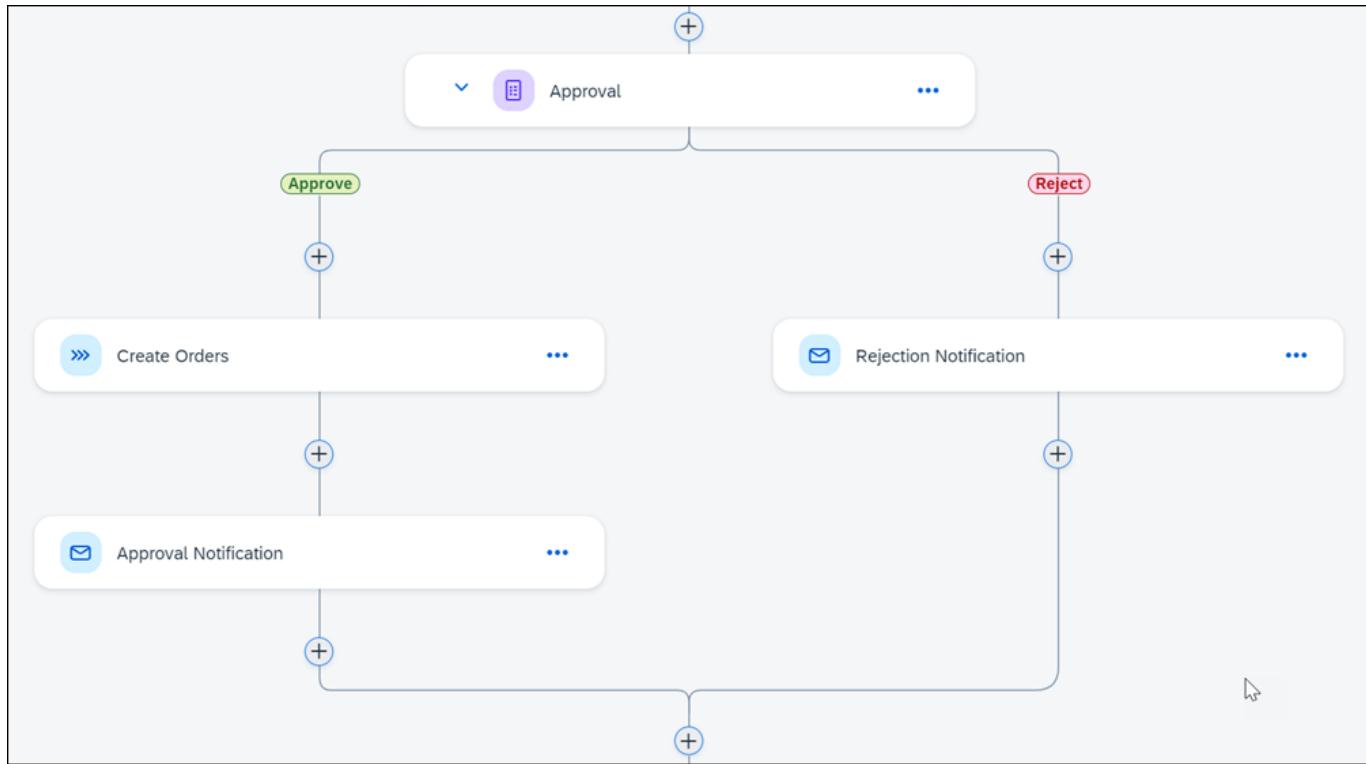
As an example, a subprocess that checks for inventory levels is included as part of the following purchase order request process:



The subprocess must run successfully before the approval mail notification is sent to the requestor.

- A workflow that was developed in SAP Business Application Studio or was added as part of live process projects from the store.

As an example of a workflow started from a process, this request process includes a workflow to create orders for approved requests:



In this case, the workflow runs only for approved requests. This workflow must run successfully before the overall process progresses, which means the approval mail is only sent once the order has been created.

Procedure

- From the process editor canvas, choose **+ (plus) > Subprocess**
- Select one of the following options:
 - Use a listed existing process. If a listed process is grayed out, it doesn't fulfill all criteria.
 - Create a new subprocess by choosing **Blank Subprocess**. This is simply another process that you create from scratch.
 - Use an existing workflow by choosing **Call a Workflow**.
- Configure any necessary process or workflow **Inputs** and **Outputs**.
- Save your changes.

The process or workflow is added as a subprocess and must run successfully before the main process continues.

- If you inserted a blank process, a new workflow, or a workflow from the library, choose the respective link in the image below and design that subprocess.

Related Information

[Create a Business Process](#)

[Call a Workflow](#)

[Modeling a Workflow](#)

[Create a Workflow Module](#)

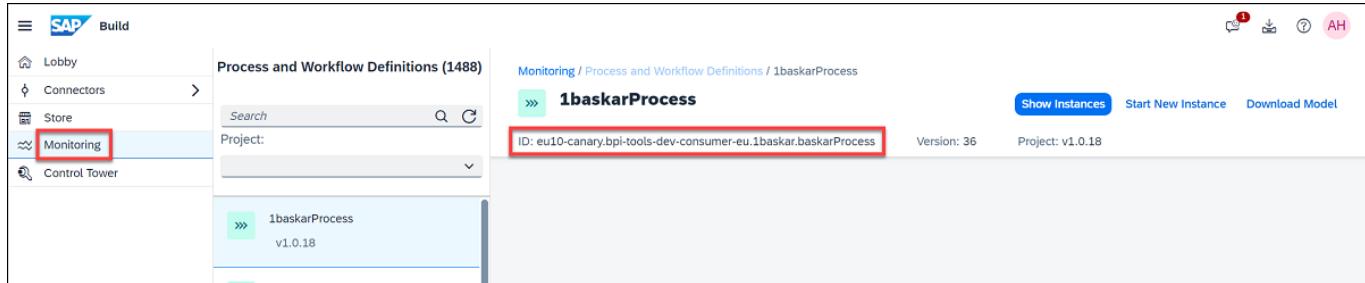
[Introducing Subprocesses as Referenced Subflows in SAP Build Process Automation](#)

Determine the Workflow Definition ID

Find the workflow definition ID of live process projects.

Procedure

1. In SAP Build, choose **Monitoring**.
2. Under **Manage**, choose **Processes and Workflows**.
3. Select the relevant workflow.



Call a Workflow

You want to add a prebuilt workflow as a subprocess.

Prerequisites

From the process editor canvas, you choose **Call a Workflow** to insert an existing workflow as a subprocess.

Procedure

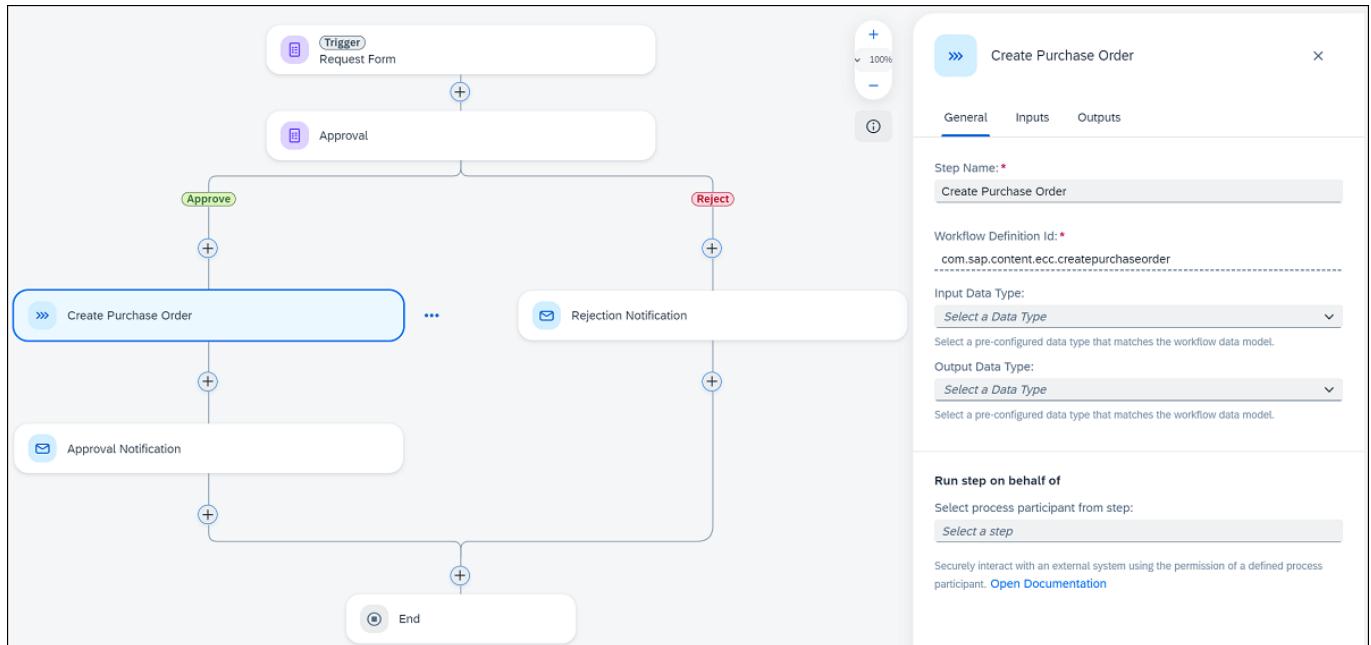
1. On the **Call Workflow** screen, select a workflow and choose **Add**.

On the **General** tab, the workflow definition ID is filled in automatically and is read only. The workflow name is also prefilled but can be changed.

2. Select an existing **Input Data Type** or click **+ New Data Type** to create one.

If creating a new data type, you must then configure this via your **Project Overview** area. See [Create a Data Type](#).

3. Select an existing **Output Data Type** or click **+ New Data Type** to create one.



4. Save your changes.

Add Mail Notifications to a Process

You can add mail notifications to your business process using the process editor, allowing you to send preconfigured emails to recipients while a process is running.

Prerequisites

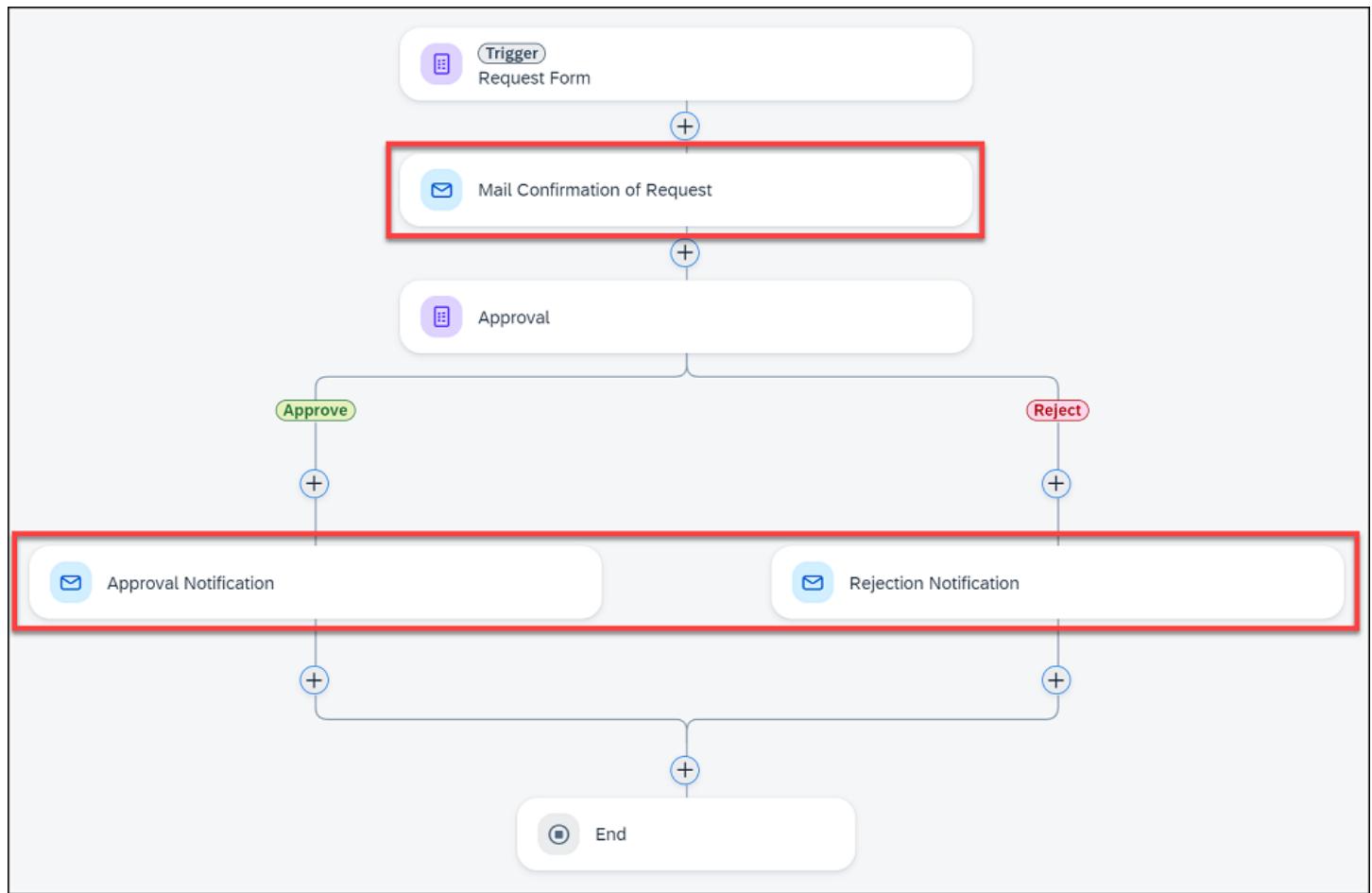
Before adding mail notifications to a process, you must first configure your SMTP mail destinations for SAP Build Process Automation. See [Configure an SMTP Mail Destination](#).

This configuration includes assigning the 'From' address for sent mail, for example: user@example.com. This inbox must be able to receive replies.

Context

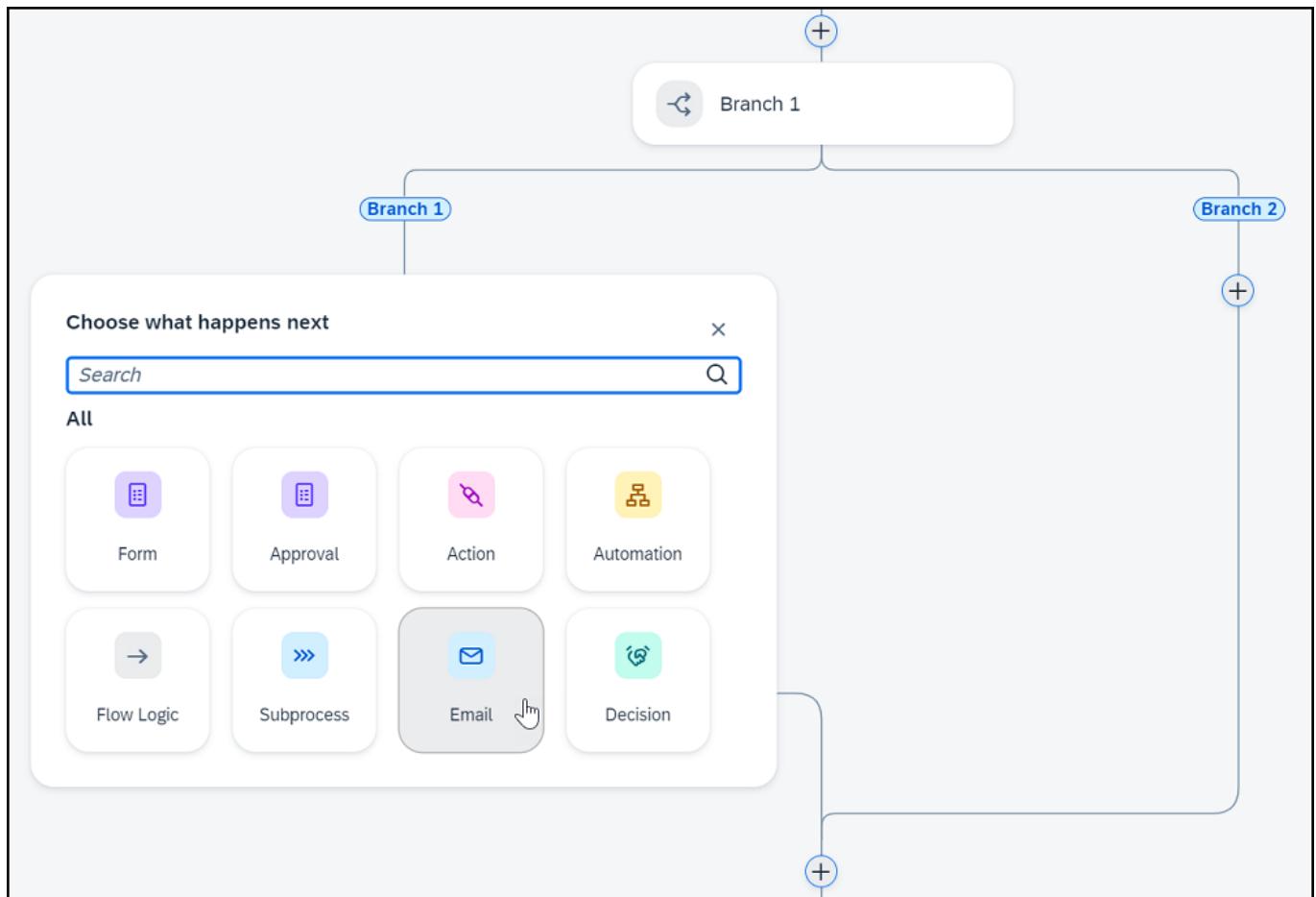
As an example, there are three mail notifications in this simple request process:

- Confirmation that the request was received
- Confirmation that the request was approved
- Confirmation that the request was rejected

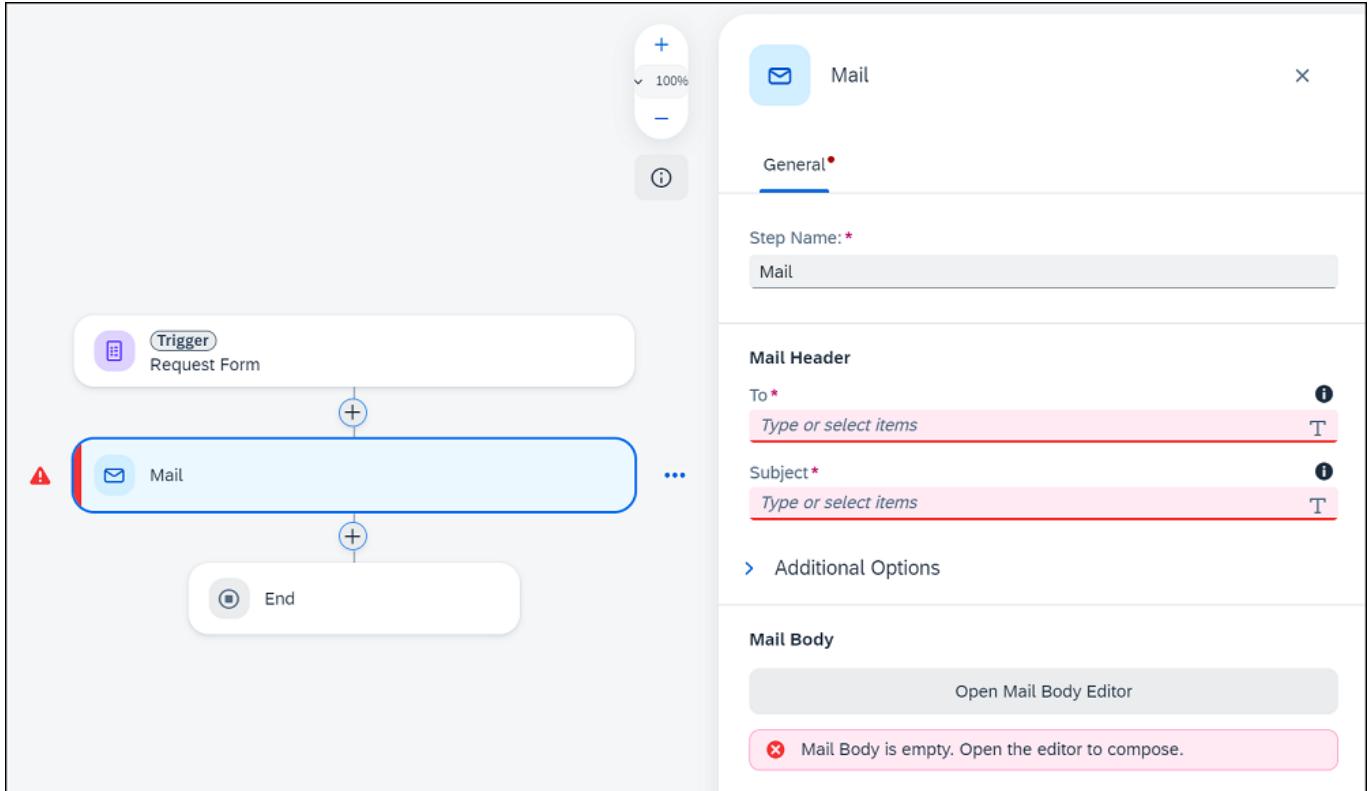


Procedure

1. Either choose **+** or right-click on the process editor canvas and select **Mail**.



The mail notification is added to the process and the settings are displayed in the side panel:



2. Configure the **Mail Header** fields.

Field	Description
To	The recipient of the mail notifications. Either add specific mail addresses or use information taken from process metadata or context. For more detailed guidelines when adding recipients, see Guidelines for Specifying Recipient Users
Subject	The subject of the mail itself. Either add specific text here or use information taken from process metadata or context.
CC	The copied recipient of the mail notifications. Either add specific mail addresses or use information taken from process metadata or context. For more detailed guidelines when adding recipients, see Guidelines for Specifying Recipient Users
BCC	The blind copied recipient of the mail notifications. Either add specific mail addresses or use information taken from process metadata or context. For more detailed guidelines when adding recipients, see Guidelines for Specifying Recipient Users

3. Choose **Open Mail Body Editor** and configure the mail body. The mail body can include the following:

- Text
- Process context information (such as the Form fields in the example)
- Process metadata (such as the 'Process Started By' information)

Edit Mail Body

Value Binding

Search

Form (Trigger)

- T Office Location
- T Name
- T Item Required
- # Value
- Deadline

Process Metadata

- T Process Started By
- T Process Instance Id

Hello

Your request has been received:

Your request reference number is:

The deadline for a decision is:

Thanks

4. Choose **Apply**.

5. Review and fix any missing mandatory fields that are still marked in red.

Results

The mail notification is added to the process, with mails sent to recipients when a process is running.

Add Wait for Duration to a Process

You can add a wait to a process, controlling how long the live process waits (or pauses) before continuing.

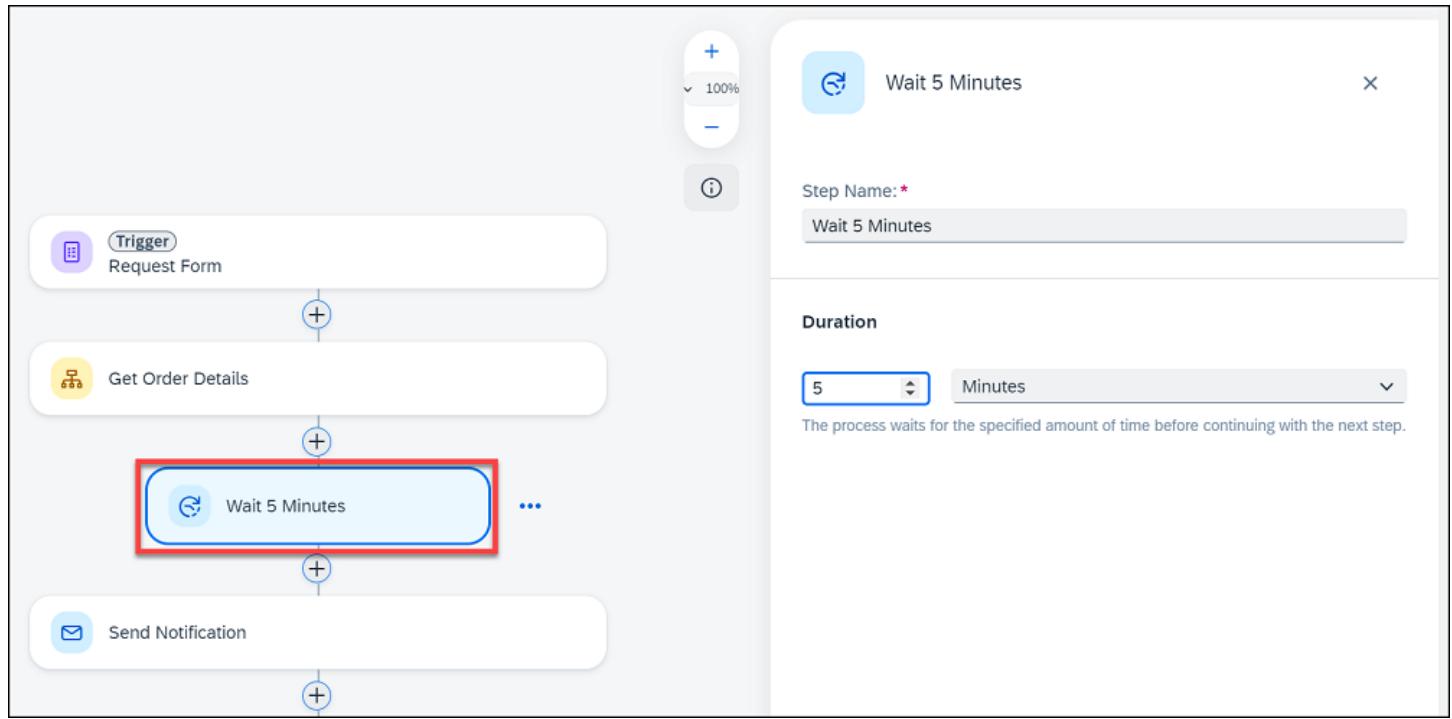
Context

Waits can be added and configured for the following intervals:

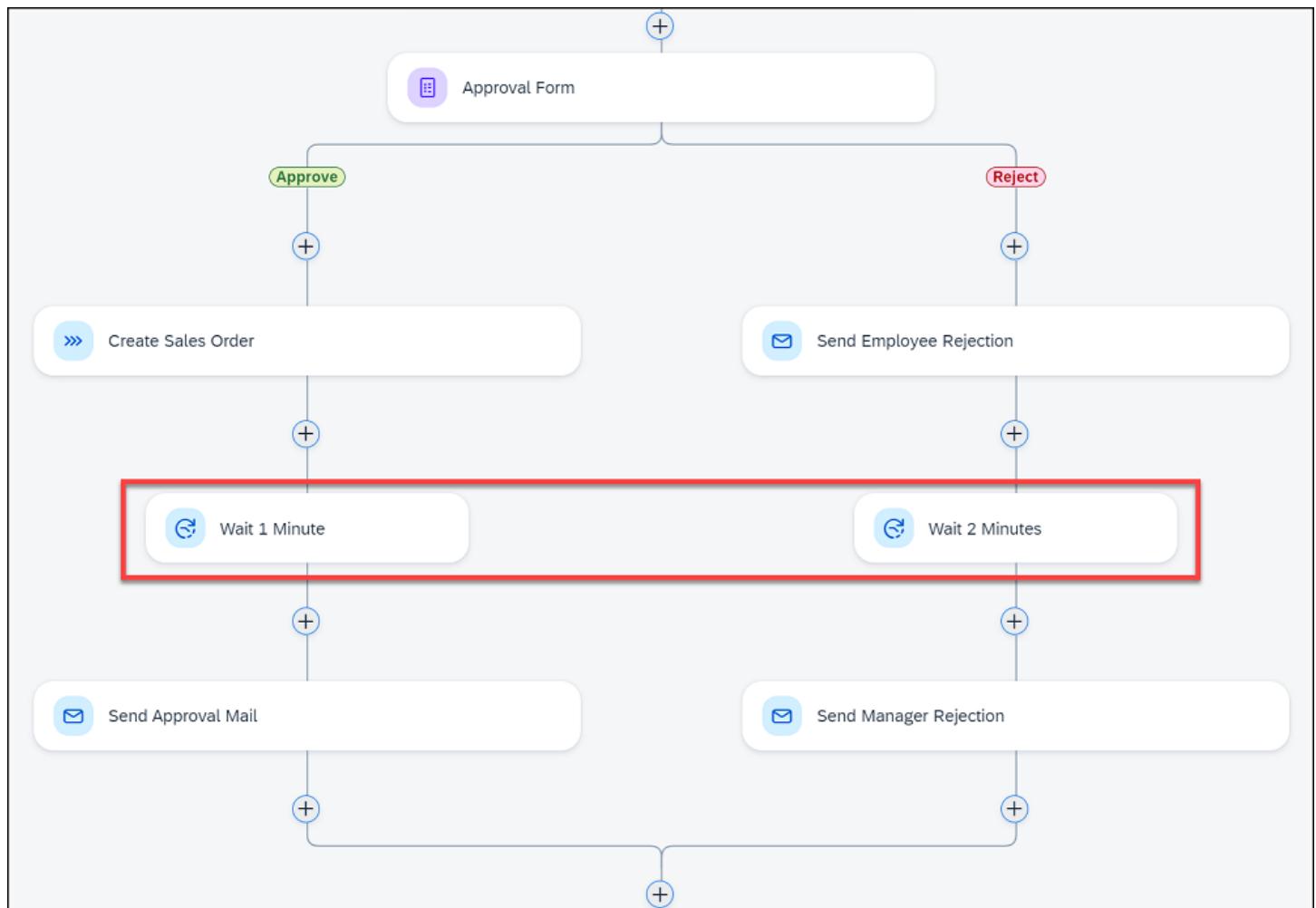
- Minutes
- Hours
- Days
- Months

These intervals then occur from the time the live process first reaches that point. As an example, if the process reaches a 1 hour wait at 09.37, the process continues at 10.37.

In the following example, the process is configured to wait for 5 minutes after an automation has run before the approval is sent:

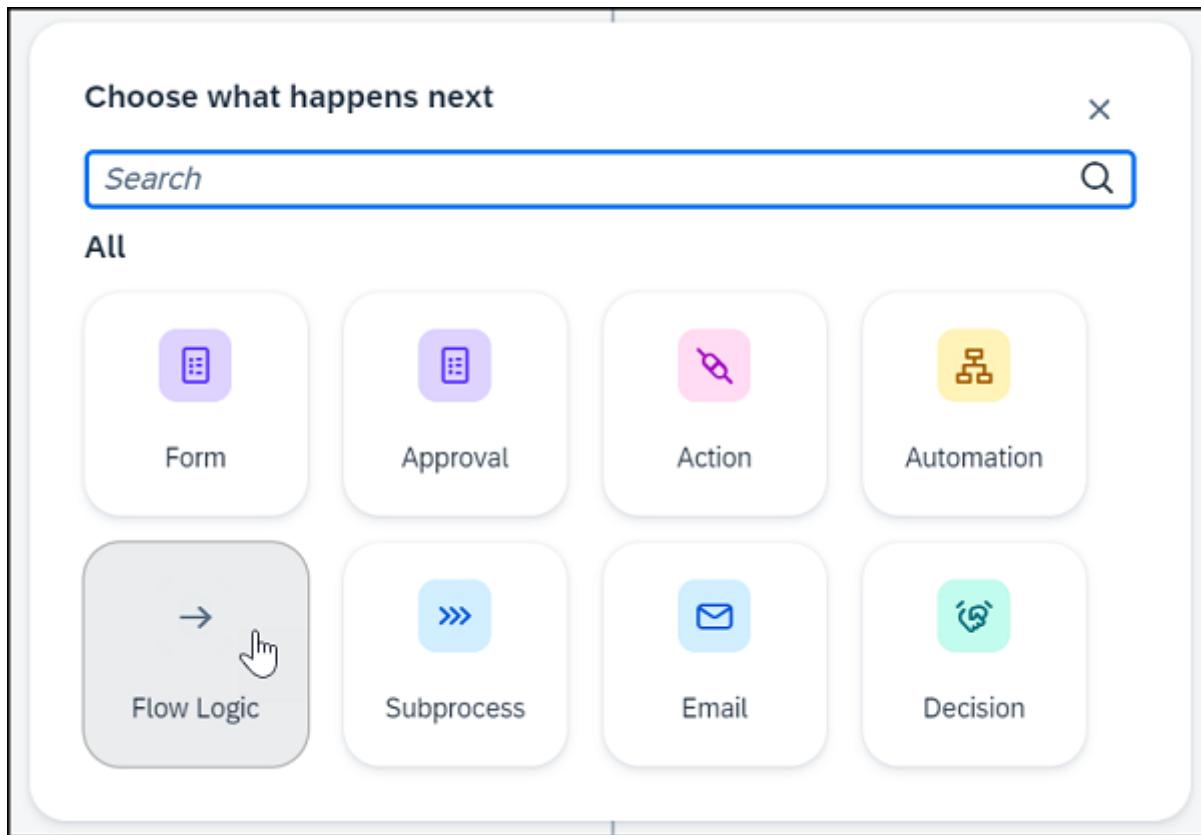


Multiple wait events can be added to the same process, too. In the following example, the process waits for 1 minute after a request has been approved, and 2 minutes after a request has been rejected:



Procedure

1. In the process editor, choose **+ (plus)** **> Flow Logic > Wait for Duration**.



2. Enter a **Step Name**.
3. To define the duration, select a **Unit** of wait (minutes, hours, days, and months) and enter a **Value**.
4. Save your changes.

Create and Design Automations

Automations are composed of a succession of steps you build in SAP Build Process Automation. An automation can orchestrate multiple activities on different applications and screens available on a specific computer.

Automation Tools

To build an automation, you have access to a list of tools from the side panel.

The screenshot shows the SAP Intelligent Robotic Process Automation Cloud Studio interface. On the left, there's a toolbar with various icons for file operations like Overview, Save, and Generate Package. The main workspace displays a simple automation flow: a green Start node at the top connected by a blue arrow to a red End node at the bottom. A blue callout box with white text says, "Start building your automation by dragging activities from the panel to the flow." In the top right corner, there are buttons for Share, Generate Package, and Save. On the far right, a sidebar titled "Automation information" is open, showing sections for Tools, Input/Output, and Info. The Tools section contains a search bar and navigation links for Automations, Activities, Data, and Controls. A red box highlights the "Tools" section of the sidebar.

- *Automations*

You can use a previously created automation from this cloud project, as a tool to include in the flow of another automation.

- *Screens*

An Application Screen is a capture of any application. For more information, see the section about [Capture and Declare Applications](#).

- *Activities*

Activities are used to build the workflow of your automation. They come from the SDK packages imported in the Cloud Studio the first time you create an automation. For more information, see the section about [Automation Activities Provided by SDK Packages](#)

- *Data Types*

A data type is a complex data used to describe a data structure. It can be defined as an input or an output and used throughout your automation. For more information, see [Data Types](#).

- *Controls*

Controls allow you to add tools such as conditions, loops, and scripts to your automation. For more information, see [Add a Control to an Automation](#).

	Condition	Inserts multiple situations defined by an expression, that determines the following step.
	End	Stops the automation.
	For Each	Inserts a sequence of actions to perform on a list of objects such as users.
	Forever	Repeats the step in a loop until it meets the required conditions.
	Repeat	Repeats the step for a defined number of times.

	Loop End	Stops a step set in a loop.
	Screen switch	Inserts multiple situations defined by the screen the user is working on, that determine the following step.
	Custom script	Inserts a step defined by custom Javascript mode.
	Try	Checks if errors occur in a sequence of actions and defines a behavior depending on the error type.
	Stop automation in error	Stops the automation in an error state.

Create an Automation

Automations are composed of a succession of steps you build in SAP Build Process Automation.

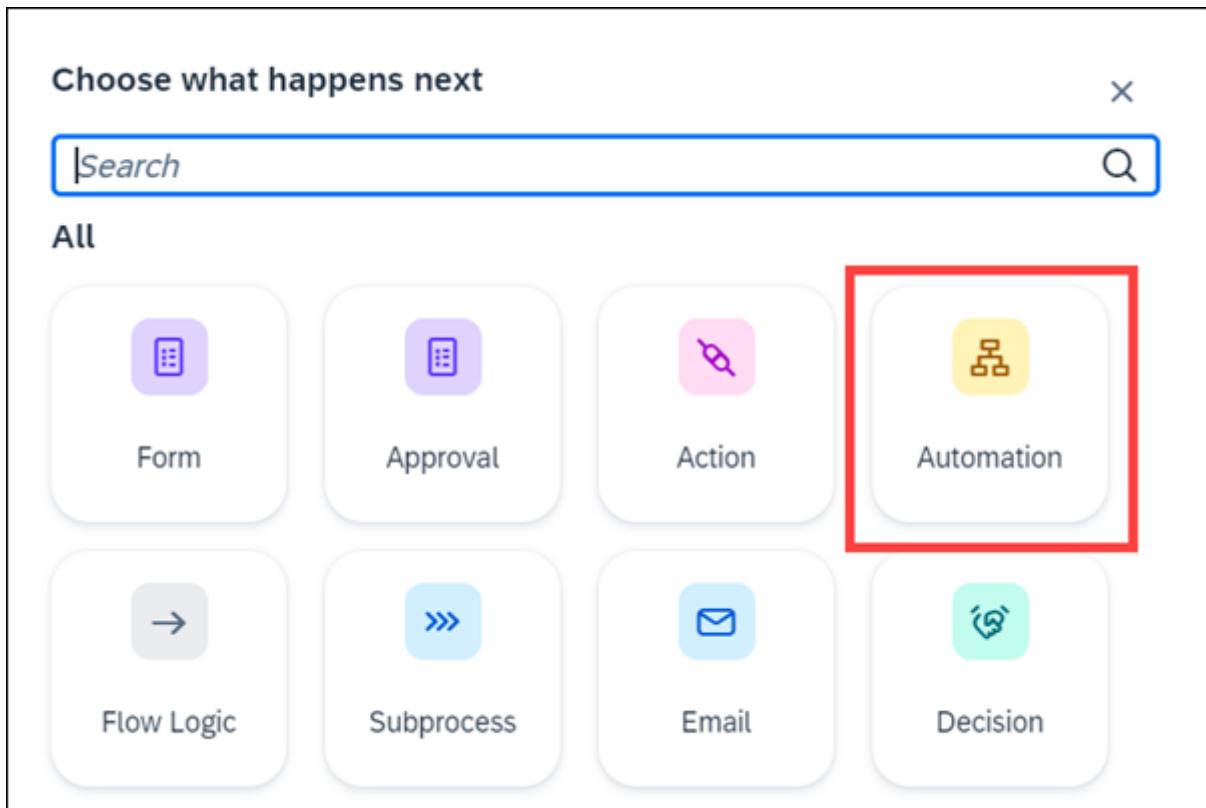
Context

For more information about designing automations, see: [Design Automations](#).

Procedure

1. You have the following options to create an automation:

- o In the **Overview** tab of your business process project, choose  **Create**  **Automation**.
- o From your business process canvas, choose  **(Plus)**  **Automation**  **Blank Automation**.



2. Optional: When adding an automation in the project for the first time, you must define the target agent version. For more information, see [Configure an Agent Version](#).
3. Provide a **Name**, an **Identifier**, and optionally a **Description**.
4. Optional: Enable the **Can only be started from another automation** option. When active, this option prevents you from starting this automation independently. It runs only through another automation inside your project. Attended and unattended triggers will not be available. For more information about triggers, see the dedicated section in the [Factory User Guide](#).
5. Choose **Create**.

The automation editor opens and you can now drag and drop activities, data types, controls, applications, and even other automations from the side panel into the workflow to build your automation.

Create an Alert

Alerts allow you to define business events in SAP Build Process Automation. Once an alert is created, it can be raised from an existing automation. You can use the **Raise Alert** activity from the SDK Core to raise an alert from your automation.

Context

Alerts allow you to raise business events in your automation and pass along information with them for an alert (such as OutOfStock or S4LoginFailed alert). After deploying a package, you can subscribe to these **Alerts** in your environment using **Alert Handlers** to get notifications by email.

Procedure

1. In the **Overview** tab of your business process project, choose **Create > Automation**.
2. Provide a **Name**, **Identifier**, and optionally a **Description**.

i Note

By default, the identifier is the same as the name. An identifier is mandatory for every artifact in the SAP Build Process Automation, so you must manually enter an identifier if it's not already pre-filled.

3. Choose **Create**.

4. A new tab opens in the main panel of the Cloud Studio. In **Parameters** (1) on the left, you can add the parameters in your alert by clicking the **New Parameter** button.

i Note

- Each parameter must have a unique name and must be JavaScript compliant: it must start with an alphabetical character, and subsequent characters can be a letter or a decimal digit. Space characters are not allowed.
- Each parameter must have an identifier (**Name**), and a type (**String**, **Number**, or **Boolean**).

5. In the center **Message** field, define a parameterized message where you can use the parameters defined in the previous step. The value of the parameter will be dynamically injected during the execution of the automation while using a parameter in the message.

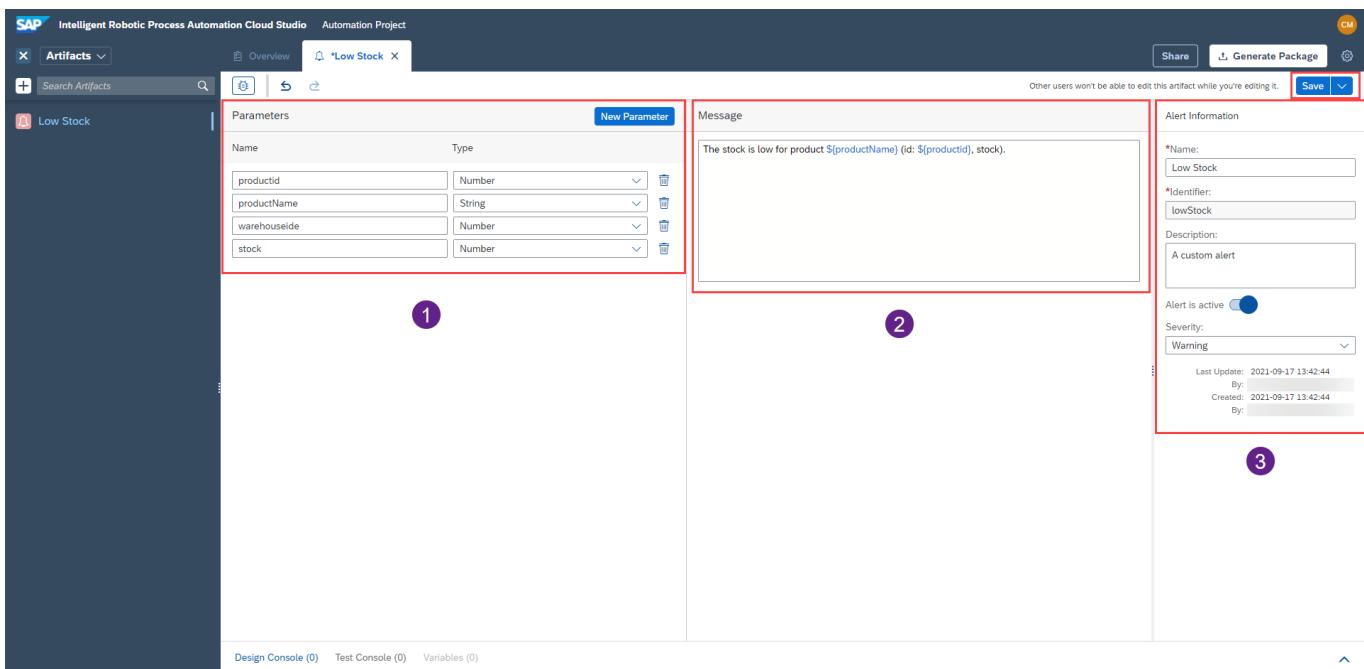
6. In **Alert Information** (3) on the right, **Name** field is already filled with the name of the alert.

7. **Optional:** Edit the **Name** field.

8. **Optional:** Enter a short description about the automation in the **Description** field.

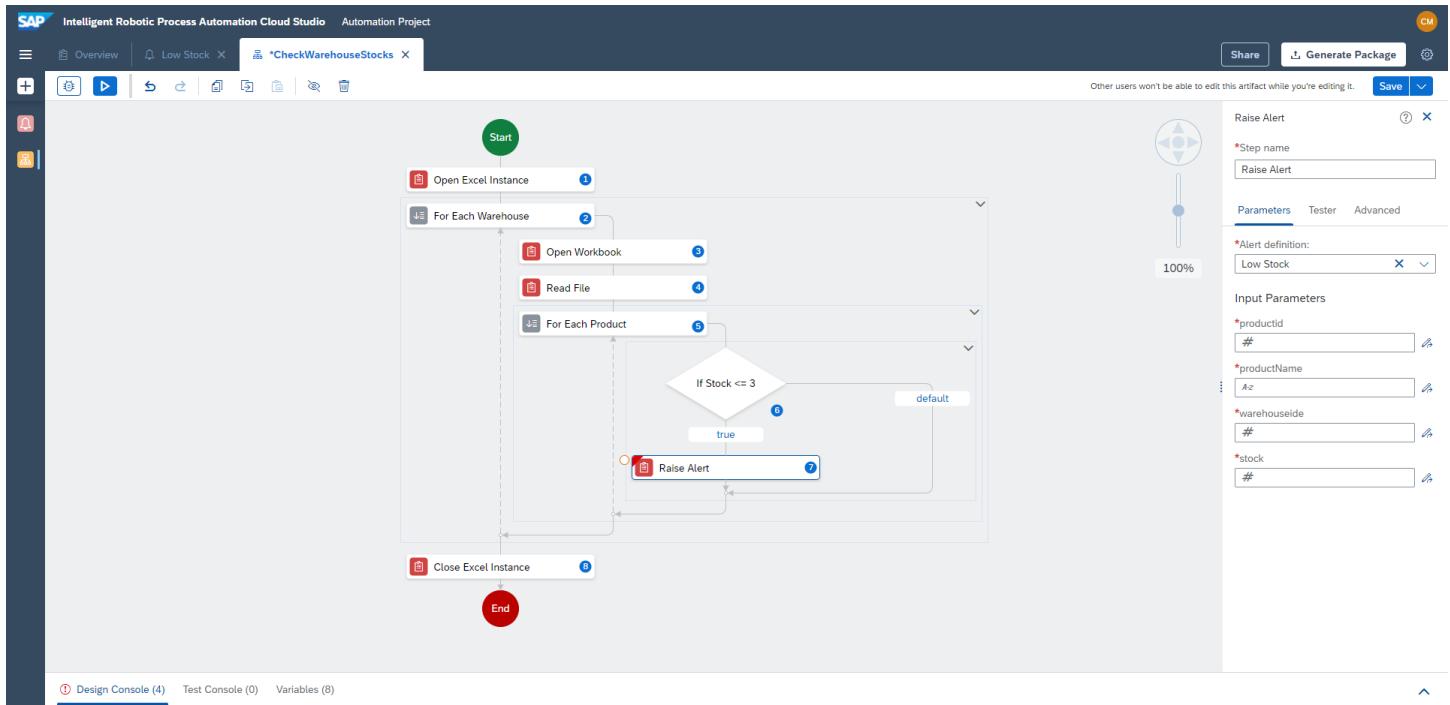
9. Select the type of the **Severity: Information, Warning, or Error**.

10. Click **Save** to save your changes.



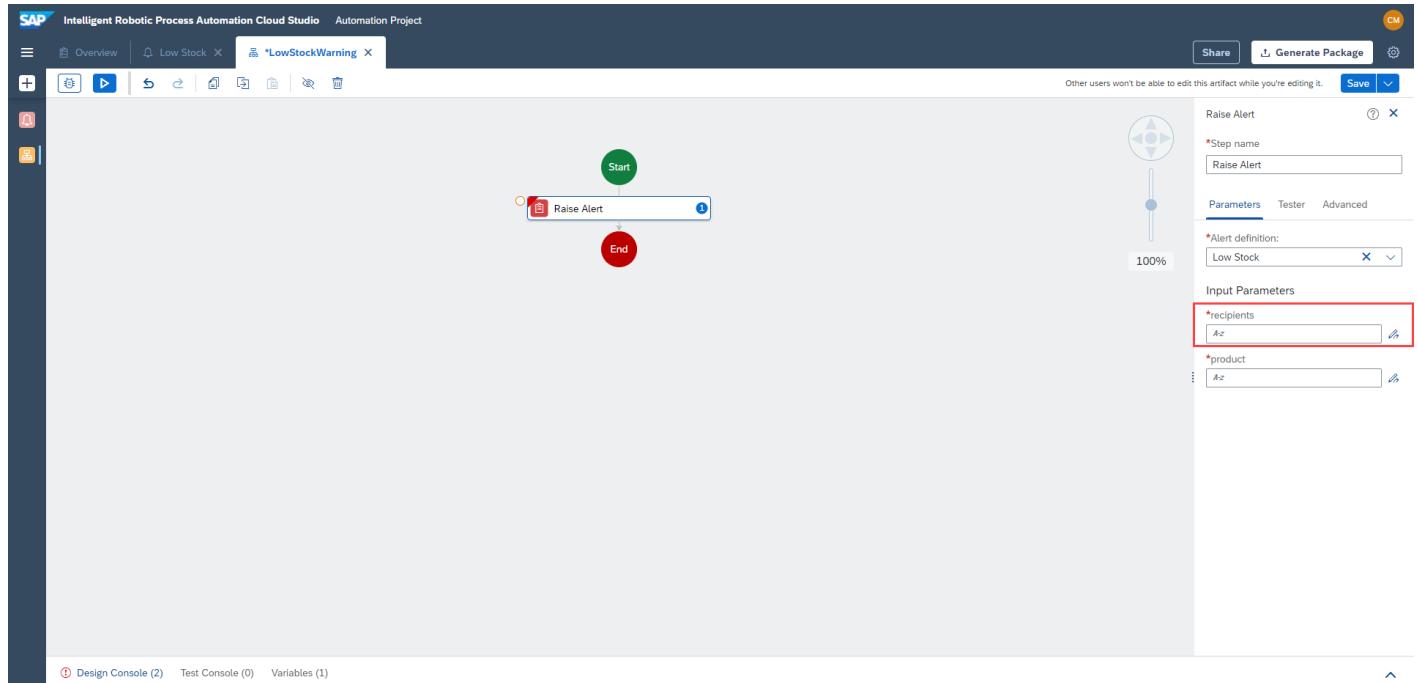
Results

You are now able to use your alert using the **Raise Alert** activity in your automation.



i Note

If you want to use dynamic recipients in your alert handler, you need to create an alert parameter (for example, recipients) that contains the email address(es) of the recipients to which you want to send the email. In your raise alert activity that raises your alert, you need to provide the email addresses separated by a comma, space, or semicolon. The number of total recipients is limited to 50 recipients.



Next Steps

Now that you created an alert in your automation, you need to add an alert handler to SAP Build Process Automation and link it to the alert you've created.

Add an API Trigger to an Automation

You can add an API trigger to an automation in the automation editor.

Context

If you add an API trigger in your automation directly, you can then export the trigger with the project and do not need to add it manually after deployment.

Procedure

1. In the **Trigger** tile, choose *Add trigger* then **API Trigger** and **New API Trigger**
2. In the **Create API Trigger** screen, enter a name and, if required, a description. Choose **Create**.
3. Optionally, right click on the trigger artifact and choose **Edit**. In the **Edit API Trigger** screen you can edit the name or description. If you are using the trigger in an automation, you can configure the trigger settings and the agent attributes. You can also see the input schema for the API from the automation.

Related Information

[Add an API Trigger](#)

[Execute an API Trigger](#)

Add a Process to an Automation

You can add a process to an automation in the automation editor.

Prerequisites

You've created a business process project where the process must expose an input e-mail subject. For more information on how to create a business process project, see [Create a Business Process](#).

Context

Processes within the same project and the dependent project can be added to an automation. The automation only triggers the process. If the process is started correctly, it sends a message during runtime stating that the process must run.

Procedure

1. Create an automation. For more information, see [Create an Automation](#).

In the right-side panel of the automation editor, you can view the **Processes** category.

2. Expand **Processes**. You can see the list of processes that is coming from your project.
3. Drag and drop the process into the workflow of your automation.
4. Select the process.

The step details side panel appears.

In the **Parameters**, add value to the **mailSubject** input parameter field.

5. Click **Release**.
- The **Release Project** pop-up window is displayed.
6. Click **Release** on the **Release Project** pop-up window.

The project is released.

7. Click **Deploy** on the [Overview](#) page.
8. Click **Next** on the [Overview](#) pop-up window.
9. Optionally, configure runtime variables, and click **Next**.
10. Optionally, configure triggers, and click **Deploy**.

The project is deployed.

11. If the project is compatible with automation triggers, you can now [Go to Monitor > Triggers](#) to create a trigger.
 12. On the [Triggers](#) page, go to your trigger, and then click [Actions > Run Now](#) to run your automation.
- If the automation runs successfully, the **Status** of the automation is shown as **Completed**.
13. On the [Process and Workflow Instances](#) page, you can view your process.

The status of the process is shown as **Running**.

14. Go to [My Inbox](#) and select **Approve** or **Reject**.

On the [Process and Workflow Instances](#) page, the status of your process is changed accordingly.

Next Steps

You are now able to trigger the process through automation and receive an e-mail in your inbox.

About Action, Proxy and Certificate

Action

An action is a dependency with predefined rest API interactions. An action group is a collection of individual actions that, once performed, returns a certain response.

Proxy

A proxy server is a software tool that redirects all internet traffic for different purposes, such as concealing your own IP address.

Certificate

A certificate is a digital file that includes information for establishing a secure, encrypted connection to a website. This type of certificate is commonly referred to as an SSL or TLS certificate, an HTTPS certificate, or an SSL server certificate.

On-Premise Action Requirements

The on-premise action needs to be capable of handling proxy challenges and using certificates. By default, the client runtime for the action is unable to handle proxy challenges (such as NTLM or Kerberos) and manage certificates. As a result, the client runtime for the action utilizes the call request SDK core to handle the proxy challenges and manage certificates.

Add an Action to an Automation

Prerequisites

You have created one or several automations, as described in [Create an Automation](#).

Context

You can add an action to the workflow of your automation.

The steps for adding an action to an automation are as follows:

Procedure

1. In the Project Explorer, choose .
2. Choose **Dependencies**.
3. Choose  **Add Dependency**  **Add an Actions project dependency** .

After adding the action to your project, it appears under  **Tools**  **Action Groups**  in the right-hand side panel of your automation.

4. Drag and drop the action into the workflow of your automation.

5. Choose the action.

The step details side panel appears.

Specify a destination for the object and use the result.

6. Save your changes.

Support Proxy to Execute an Action in an Automation

Prerequisites

You have created one or several automations, as described in [Create an Automation](#).

You have added an action to your project, as described in [Add an Action to an Automation](#).

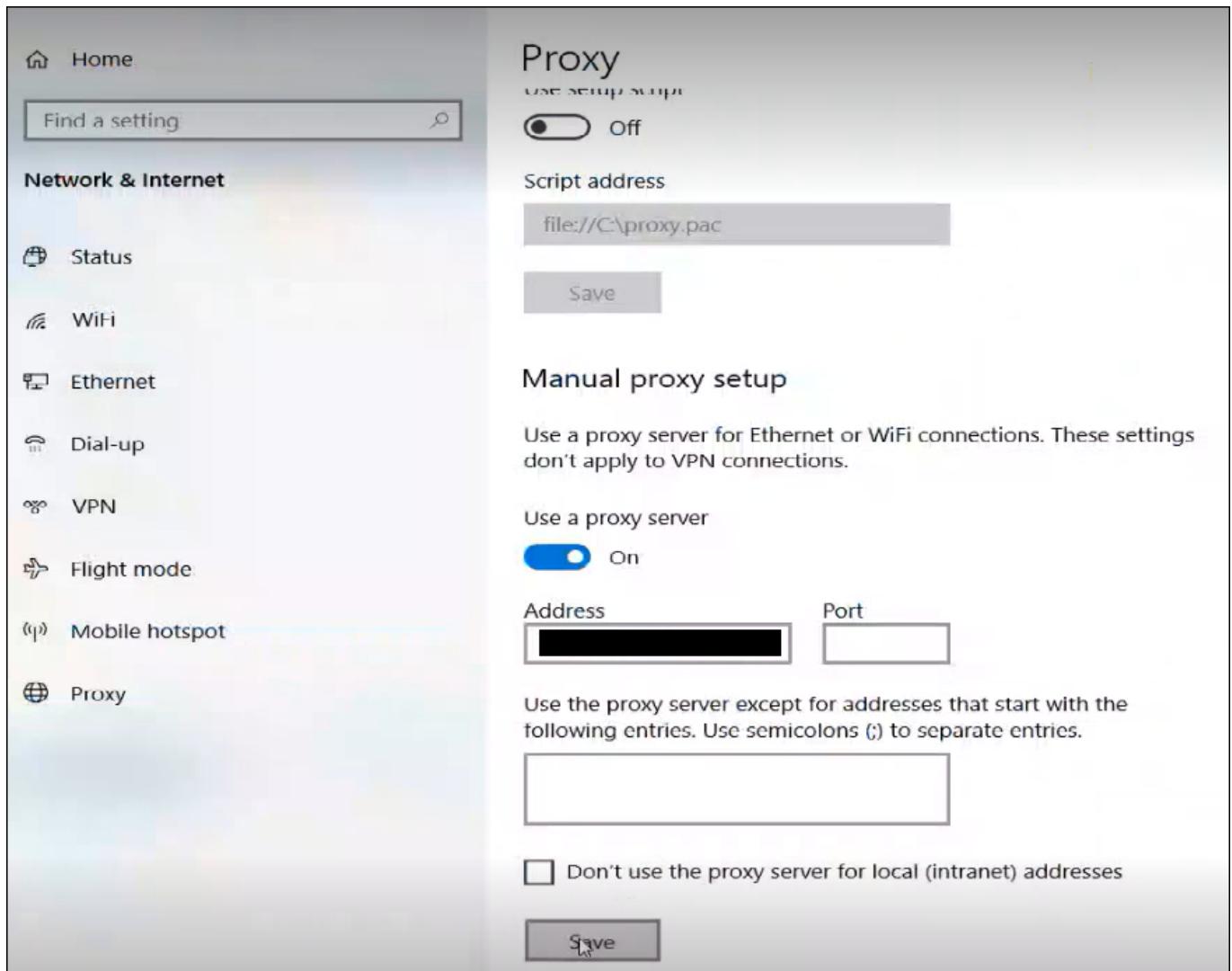
Context

You can use a proxy to execute an action in your automation.

The steps for supporting proxy to execute an action are as follows:

Procedure

1. Open your automation.
2. Drag and drop the action into the workflow of your automation.
3. Choose the action and specify a destination for the object.
4. Activate your proxy server.



5. Save your changes.

Design Automations Quickly with the Recorder

You can automate complex workflows easily using the Recorder in Cloud Studio. It automatically captures applications and designs automations accurately at the same time.

The Recorder records the steps you perform across the screens of an application. Screens, elements, and their underlying metadata are automatically captured and properly declared. Then you can export the recording in the automation designer of the Cloud Studio where a workflow is built. The exported recording steps include automatically recorded application screens, elements, metadata, and the automation with a workflow made up of activities. If required, you can manually edit the captured screens and elements or edit the automation to update the [Step Details](#). For more details about designing automations and capturing applications in the cloud studio, refer to the [Capture an Application](#) section.

Based on your action as a bot developer, the Recorder records a series of interactions consisting of various steps across the screens such as:

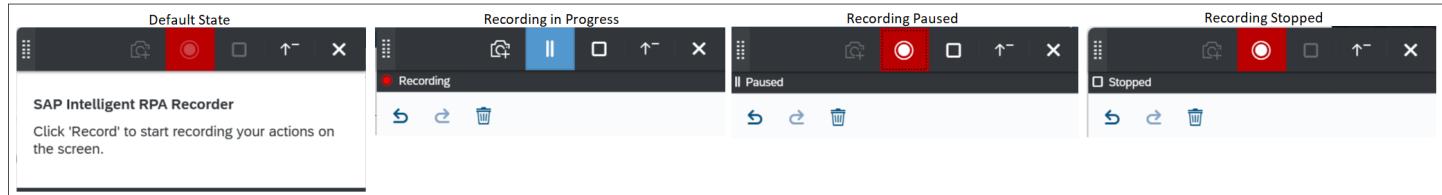
- Entering data
- Mouse clicks and movements
- Keystrokes with object controls including input fields, tables, menus, buttons, check boxes, and screen tabs

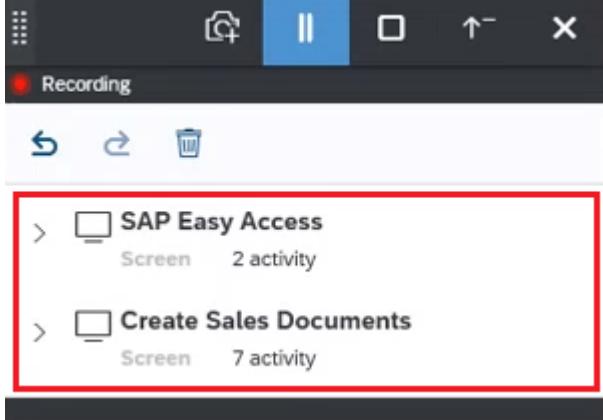
Overview

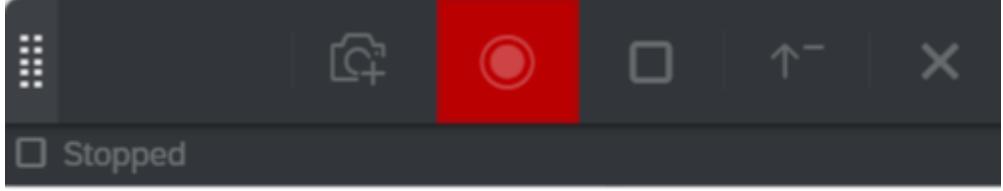
The following screenshot shows the various status of the Recorder.

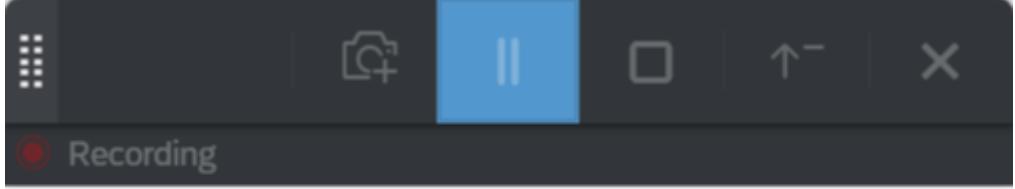
i Note

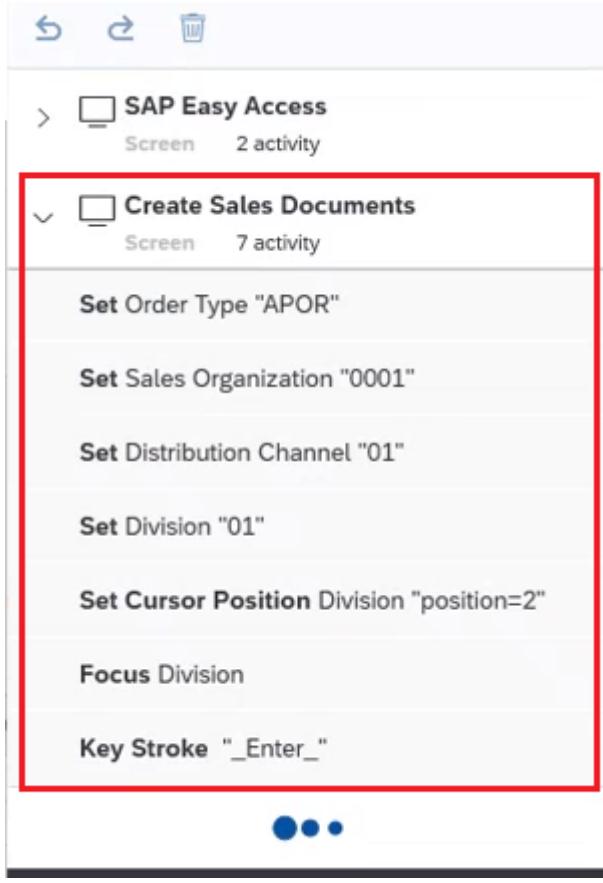
The functionality and UI of the recorder is the same for SAP GUI for HTML, SAP GUI for Windows, and SAP UI5 systems.



Name	Button/Icon	Description
Drag/Move		To move the recorder to a desired location in an application window.
Record		To start the recording. Once you click this button, by default, the active screen will be captured.
Capture		To capture the new screen of an application.
Pause		To pause the recording.
Resume		To resume the recording.
Expand/Collapse	 	<p>To view the Activities List of recorded screens.</p>  <p>To collapse the Activities List.</p>
Recording (Status)	Recording	This indicates that the recording is in progress.
Stopped (Status)	Stopped	This indicates that the recording is stopped.
Undo		To remove unnecessary steps while recording. You can undo only from last recorded step.
Redo		To restore any actions that were previously undone using an Undo function. You can do this only from last recorded step.
Delete All		To delete entire recording. All the steps in the recording widget will be deleted. But WebGUI application being recorded is not rolled back.

Name	Button/Icon	Description
Stop		<p>To stop the recording. Once you stop the recording, by default, the Export button is displayed.</p> <p>Rerecord - If you want to replace the existing recorded steps and start with new recording, then click . The start new recording message is displayed.</p>  <p>Do you want to start a new recording and replace the existing recorded steps?</p> <p style="text-align: right;">OK Cancel</p> <hr/> <ul style="list-style-type: none"> • OK - If you want to start a new recording, then click this button. <p>i Note If you click OK, steps will be cleared, and recording will start from the beginning.</p> <ul style="list-style-type: none"> • Cancel - If you want to continue with the existing recording, then click this button. All recorded steps will be retained, then the Export button is displayed.

Name	Button/Icon	Description
Close		<p>To close the Recorder. Once you click this button, the closing the recorder confirmation message is displayed.</p>  <p>Do you want to export recorded steps before closing the recorder?</p> <p>Export & Close Close Cancel</p> <ul style="list-style-type: none"> • Export and Close - If you want export the recorded steps to Cloud Studio and close the Recording widget, then click this button. • Close – If you want to exit the Recording widget without exporting the recorded steps to Cloud Studio, then click this button. • Cancel – If you want to continue with the recording, then click this button. Retains the same state as before.

Name	Button/Icon	Description
Expand/Collapse	> ▼	<p>To view list of activities of a specific screen.</p>  <p>The screenshot shows the SAP Cloud Studio recorder interface. At the top, there are three icons: a left arrow, a right arrow, and a trash can. Below them is a navigation bar with a left arrow, a monitor icon labeled 'SAP Easy Access', a 'Screen' button, and a '2 activity' indicator. The main area displays a tree structure. The root node is 'Create Sales Documents' (Screen, 7 activity). This node has several children: 'Set Order Type "APOR"', 'Set Sales Organization "0001"', 'Set Distribution Channel "01"', 'Set Division "01"', 'Set Cursor Position Division "position=2"', 'Focus Division', and 'Key Stroke "_Enter_"'.</p>
Export	Export	<p>To collapse the activities list of a screen.</p> <p>To export the recording in the automation designer of the Cloud Studio.</p>

Prerequisites

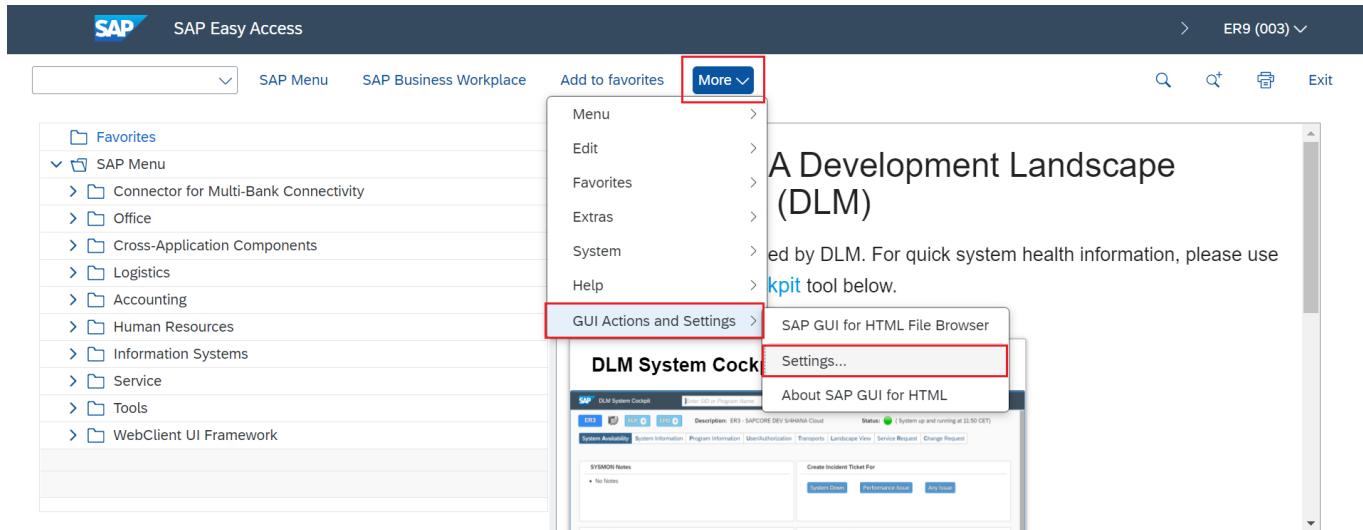
This section describes the necessary prerequisites to use the Recorder in the Cloud Studio with the SAP GUI for HTML (also called WebGUI) and SAP GUI for Windows (also called WinGUI) systems.

Configuring SAP GUI for HTML (WebGUI)

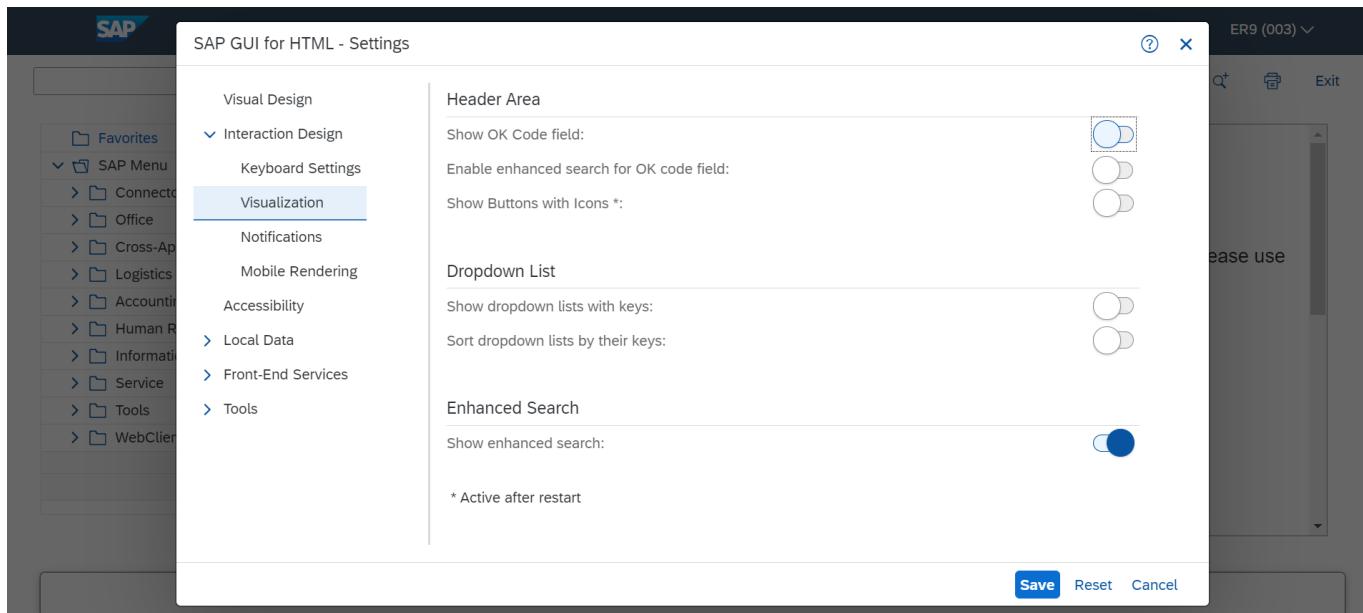
Check the following SAP Note to learn how to activate the WebGUI technology interface of SAP WebGUI: [2853735](#).

Enabling Show OK Code Field

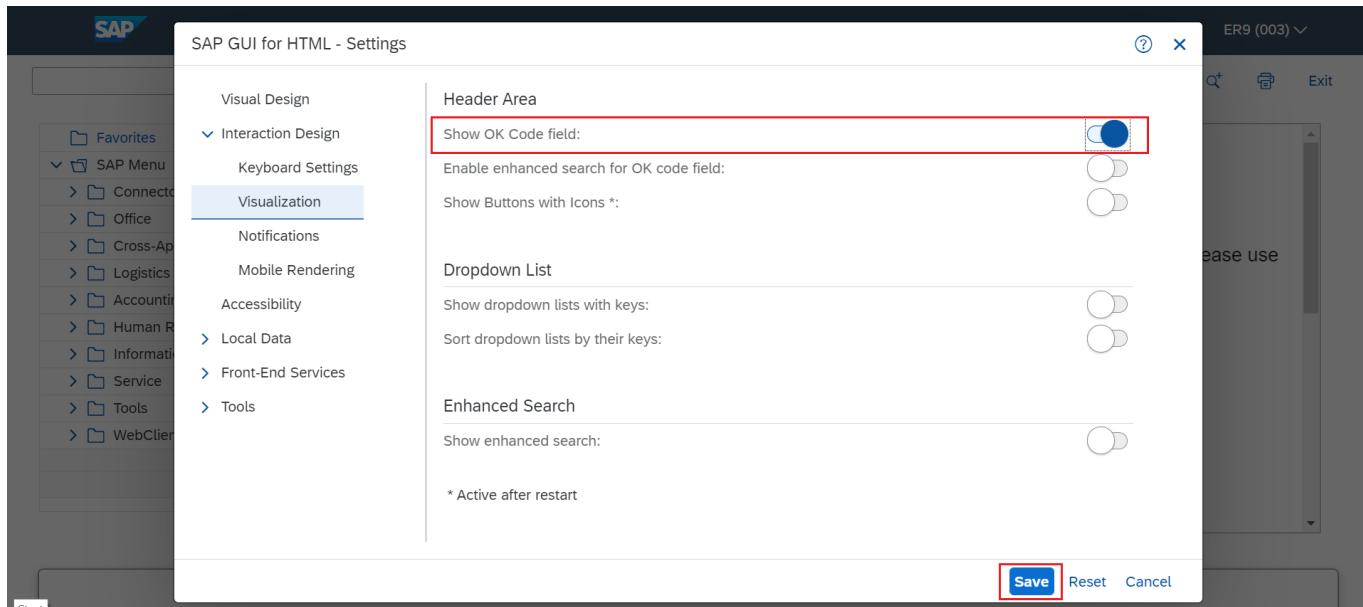
1. In the SAP WebGUI application, Click **More**. The **More** menu is displayed.
2. Click **GUI Actions and Setting** and then click **Settings**.



The SAP GUI for HTML – Settings (by default, **Visualization** is highlighted) screen is displayed.



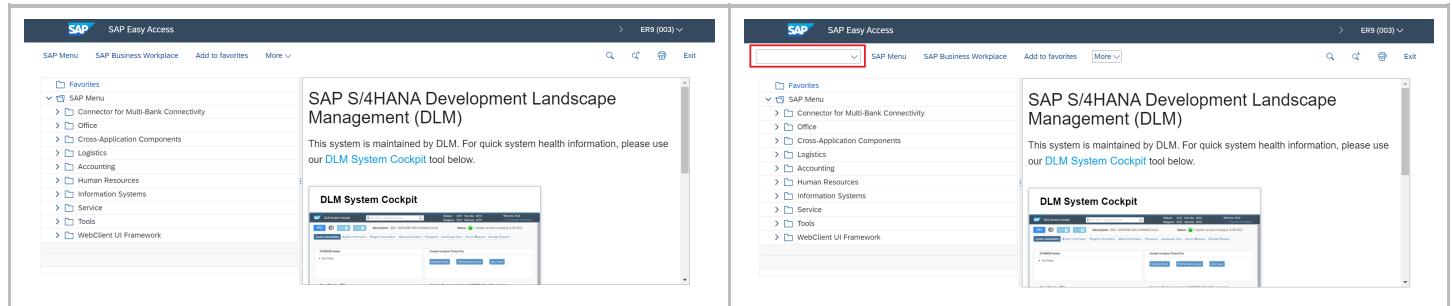
3. Enable the Show OK Code field.



4. Click Save. The Show OK Code field is enabled.

SAP WebGUI before enabling OK Code field.

SAP WebGUI after enabling OK Code field.



i Note

There are cases with the SAP WebGUI application where the capture of the WebGUI application isn't possible, or the actions aren't recorded in the recording widget.

This can happen because of the following reasons:

- The SAP WebGUI client isn't enabled. You can enable the client by applying the SAP note [2853735](#).
- The system isn't set up correctly. You can create a customer ticket on component BC-FES-ITS.

Configuring SAP GUI for Windows (WinGUI)

To use recorder with SAP WinGUI,

- Make sure you use only 32-bit SAP GUI for Windows Application.
- SAP scripting and Recording must be enabled in the SAP WinGUI system.

Enabling Scripting on the Client

To enable SAP scripting on the client, refer to the [Enabling Scripting on the Client Side](#) section.

Enabling Scripting on the Server

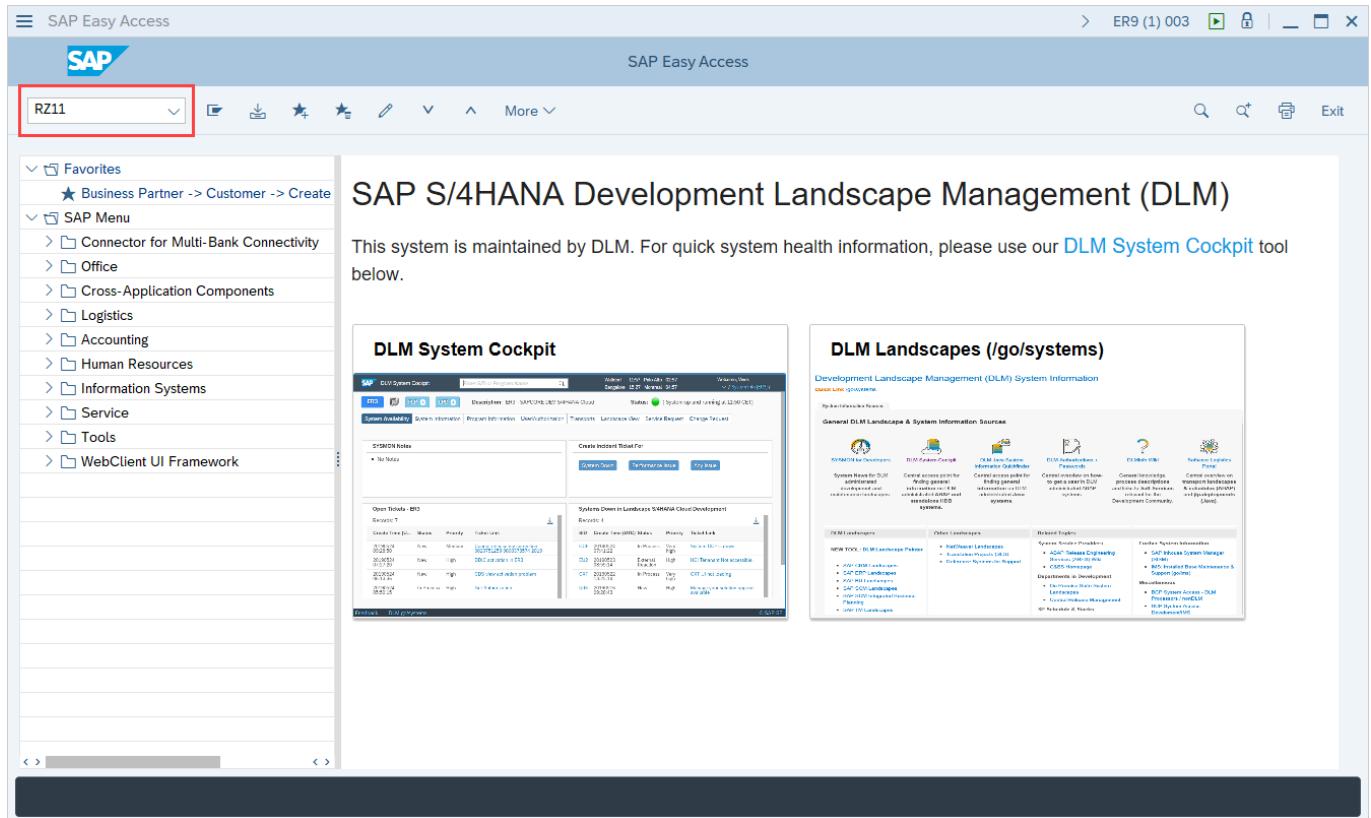
To enable SAP scripting on the server, refer to the [Enabling Scripting on the Server Side](#) section.

Enabling Recording

If you are using an SAP WinGUI application for the first time, then you must enable Recording. Else, the Recorder button will not be enabled in the Cloud Studio.

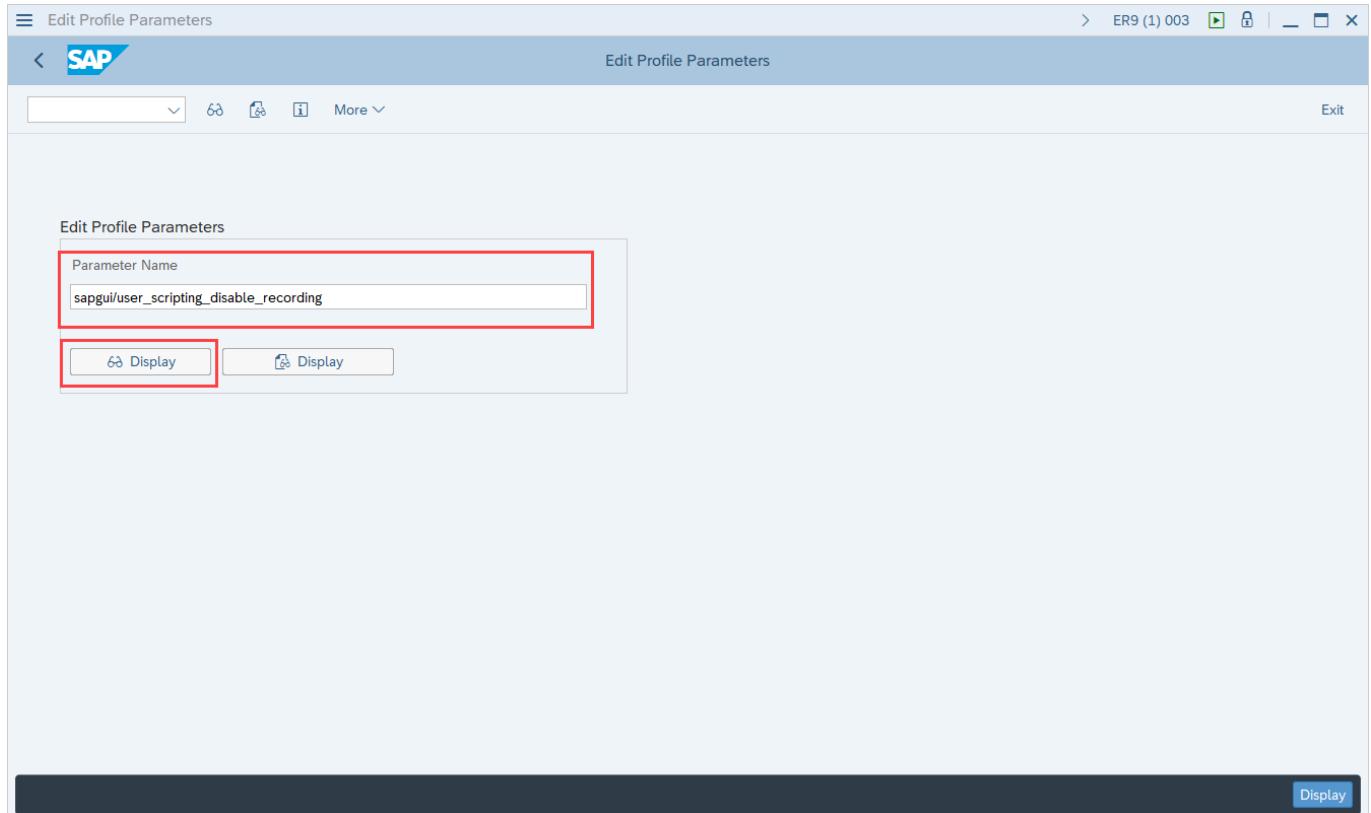
To enable Recording in the SAP WinGUI application, follow the below procedure:

1. Launch SAP Logon and connect to the server.
2. In the **Enter Transaction Code** search field of the **SAP Easy Access** screen, enter the transaction code, **RZ11**.



3. Press **Enter**. The **Edit Profile Parameters** screen is displayed.

4. Enter the parameter name, **sapgui/user_scripting_disable_recording**, in the **Parameter Name** field.



5. Click **Display**. The parameter details screen is displayed.

6. In the **Value of Profile Parameter sapgui/user_scripting_disable_recording** table, check the value of **Current Value**. If the value is **TRUE**, then click **Change Value**.

i Note

If the **Current Value** is set to **FALSE**, the recording is enabled already and you can exit the screen.

Value of Profile Parameter sapgui/user_scripting_disable_recording	
Expansion Level	Value
Kernel Default	FALSE
Default Profile	FALSE
Instance Profile	FALSE
Current Value	FALSE

The **Change Parameter Value** screen is displayed.

- Enter **FALSE** in the **New Value** field. If required, un-check the **Switch on all servers**.

Parameter Values

Parameter Name: sapgui/user_scripting_disable_recording

Kernel Default: FALSE

Default Profile: FALSE

Instance Profile: FALSE

Current Value: TRUE

New Value:

Switch on all servers

Save X

- Click **Save**. The recording is enabled in the SAP WinGUI system.

Automatically Capture Applications with the Recorder

This section describes how to use the Recorder for an SAP GUI for HTML (also known as WebGUI), SAP GUI for Windows (also known as WinGUI), SAPUI5 or Generic Web applications.

Before recording the steps of a workflow, ensure that the application and the Cloud Studio are running in a Chrome browser in two separate windows. The Desktop Agent must be configured to use with the target tenant.

i Note

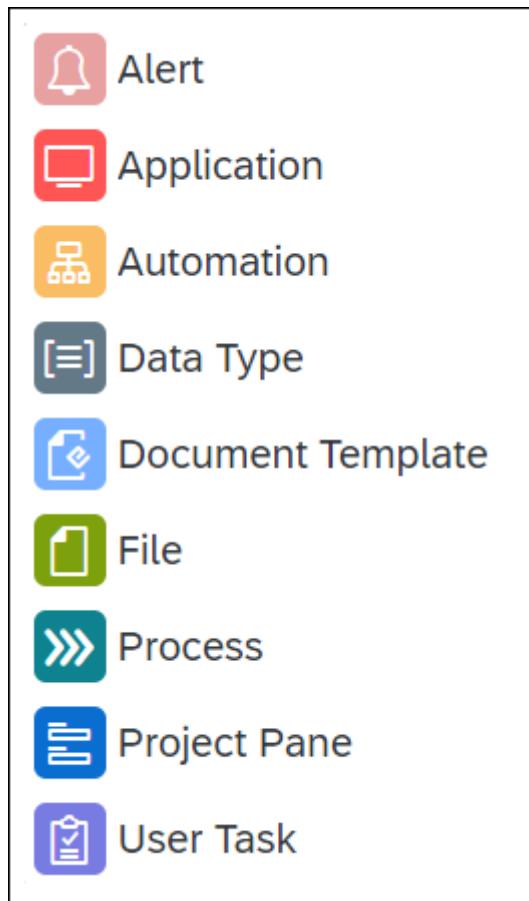
The procedure for recording the steps of a workflow in the Cloud Studio with the SAP WebGUI, SAP WinGUI, SAPUI5 systems, and web applications is same.

The below procedure is performed using the SAP WebGUI system.

The screenshots and data used in this section are only an example for understanding.

Getting Started

To start recording, you must create a project in the Cloud Studio. In the Project Explorer, click  and then click **Create** in the left panel or click **Create** in the Project Explorer main panel. The artifact menu is displayed.



In the artifact menu click **Application**. A new tab labeled **Untitled** opens. Later, this label will change to the name of the captured application. The system starts detecting the applications and their screens currently running on your local machine. When it's done, you'll see a list of screens in the picker panel on the left.

To know more details about capturing an application, refer to the [Capture an Application](#) section.

Select the application you want to record from the list of screens. A preview of the screen is displayed in the capture area.

The screenshot shows the SAP Intelligent Robotic Process Automation Cloud Studio interface. The top navigation bar includes 'Intelligent Robotic Process Automation Cloud Studio' and 'Test38a'. The left sidebar has sections for 'Overview', 'SAP Easy Access', 'Select Screen', and a search bar. The main area displays the 'SAP Easy Access' application with its navigation tree and a preview of the 'SAP S/4HANA Development Landscape Management (DLM)' screen. On the right, there are panels for 'Screen Information', 'Record' (with fields for 'Application Name' set to 'SAP Easy Access' and 'Technology' set to 'SAP Web GUI'), and a 'Record' button at the bottom.

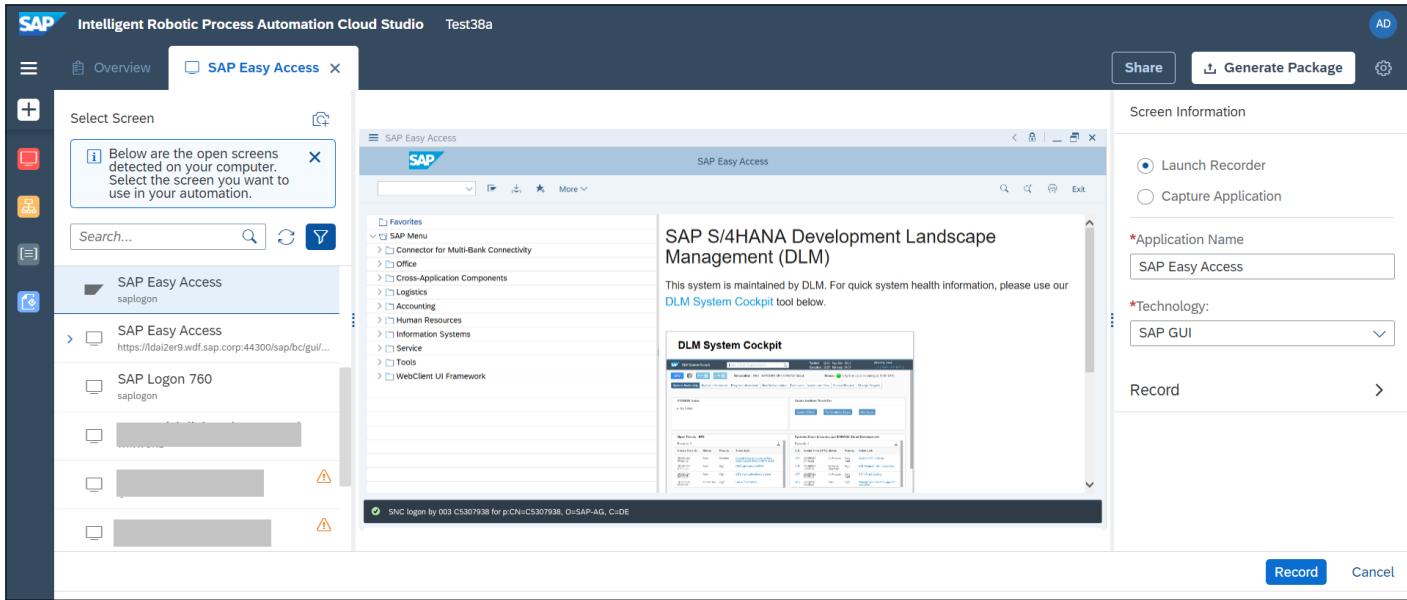
i Note

The recorder button is only enabled if the detected application is an SAP WebGUI, SAP WinGUI, SAPUI5 or Web application.

This is custom documentation. For more information, please visit the [SAP Help Portal](#)

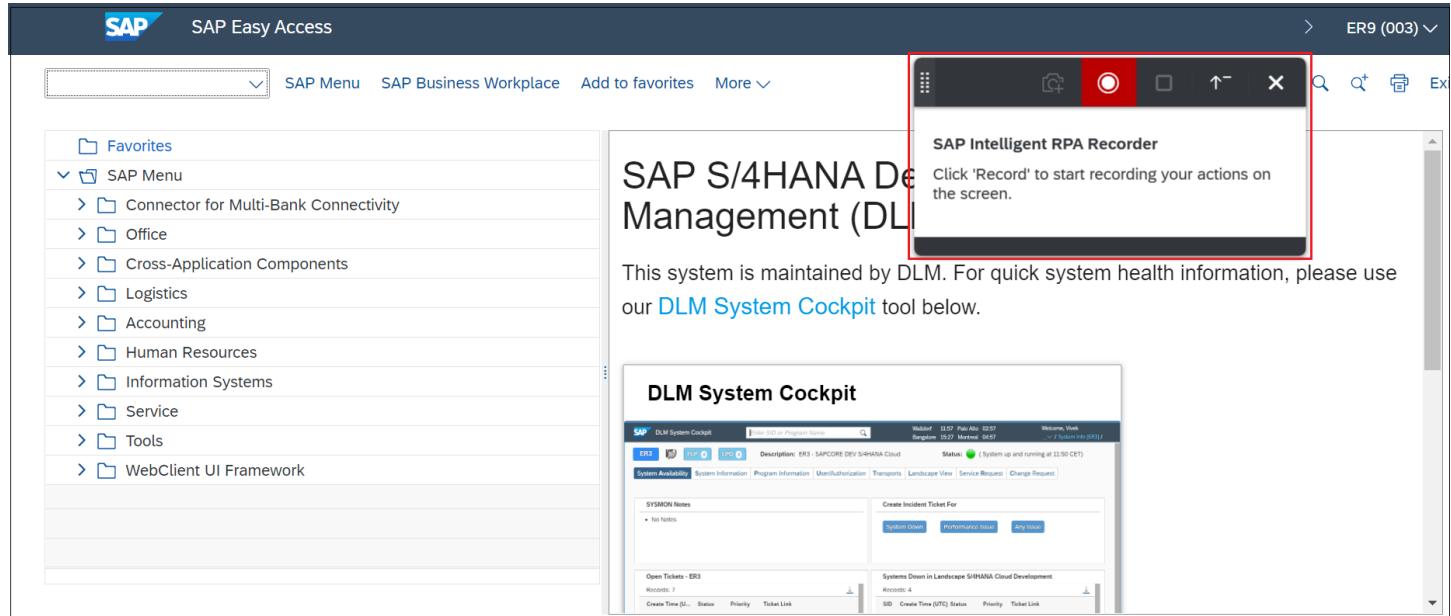
i Note

Below is the WinGUI preview in the Cloud Studio for reference:



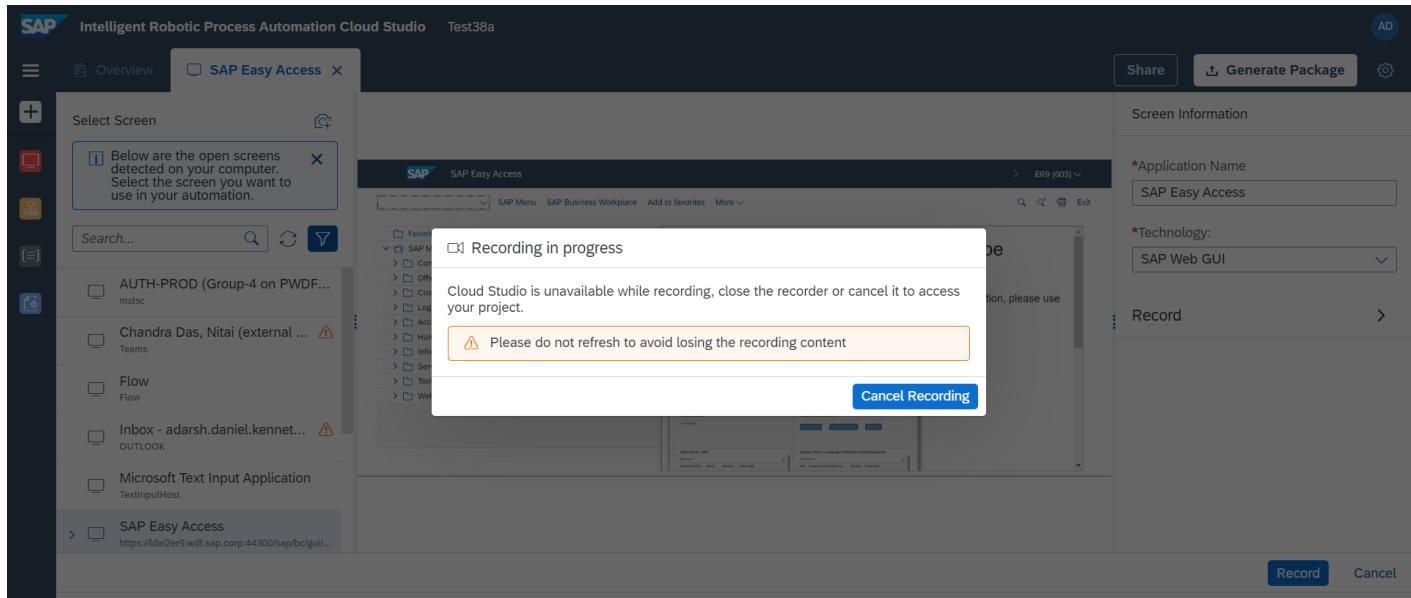
Click **Record**. After you click the **Record** button, you will be directed to the application you want to record.

In the SAP WebGUI, SAP WinGUI, SAPUI5 or Web application, by default, the **Recorder** is enabled.



i Note

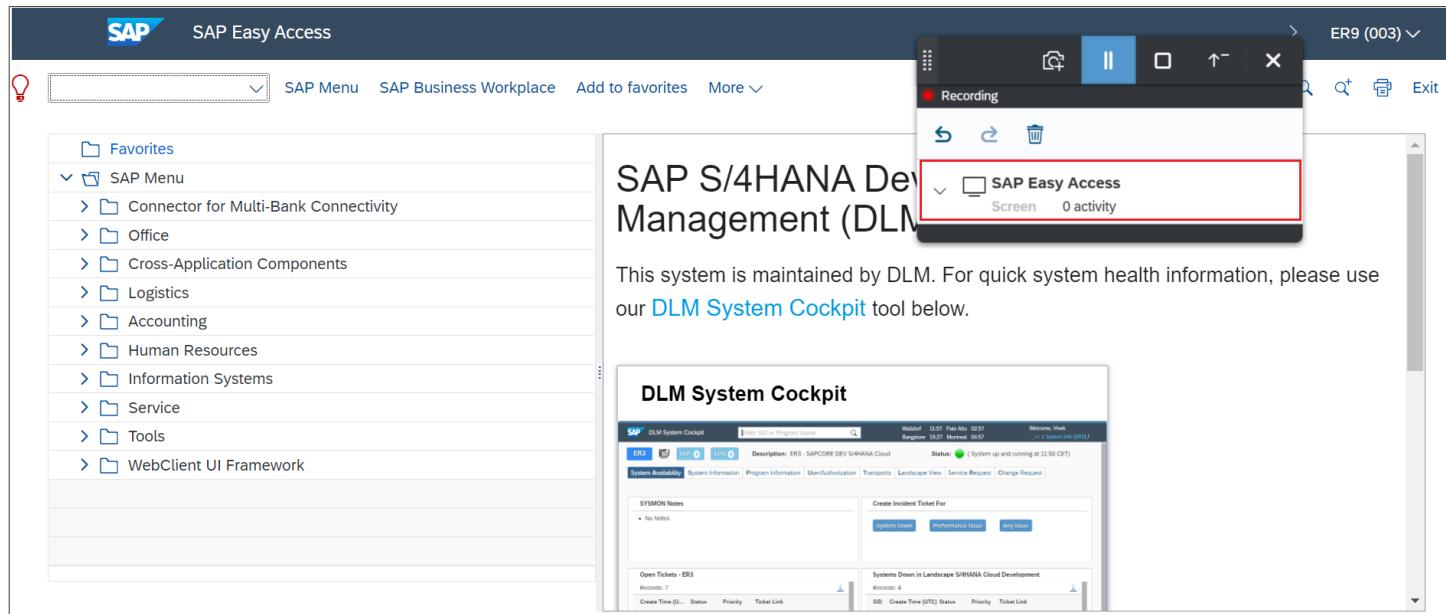
When you click the **Record** button, the Cloud Studio is locked and the **Recording in progress** pop-up message is displayed.



If you want to cancel the recording, click **Cancel Recording**. The **Cancel recording and dismiss all created steps?** warning pop-up is displayed.

Click **OK**. The recording will be cancelled and then Cloud Studio window is displayed.

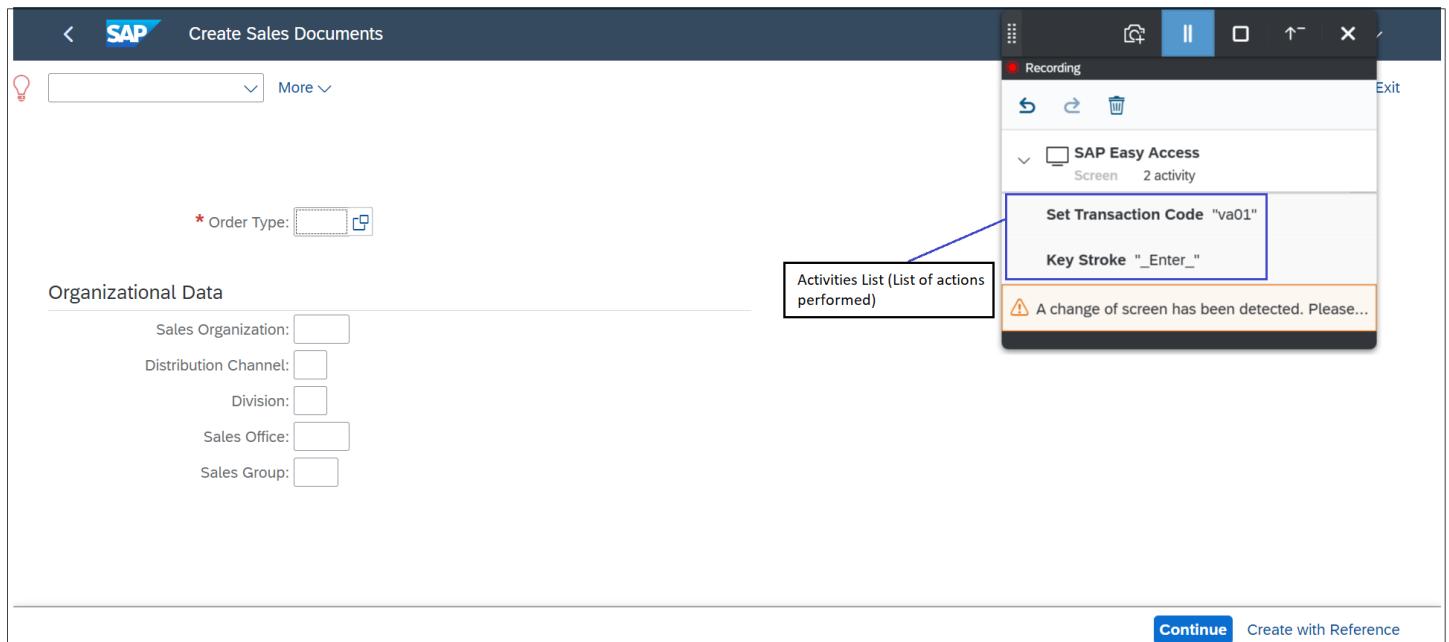
To initiate the recording, you must click  button. The recorder captures the first screen of an application automatically. You can see it in the **Activities** list.



The recorder captures the first screen of an application automatically and subsequent screens must be captured manually by clicking the  button after which the actions you perform on that screen will be recorded automatically.

The steps are recorded even if you do not click the  button manually to capture new screens. However, the generated automation will be incorrect. This is applicable across the workflow recording.

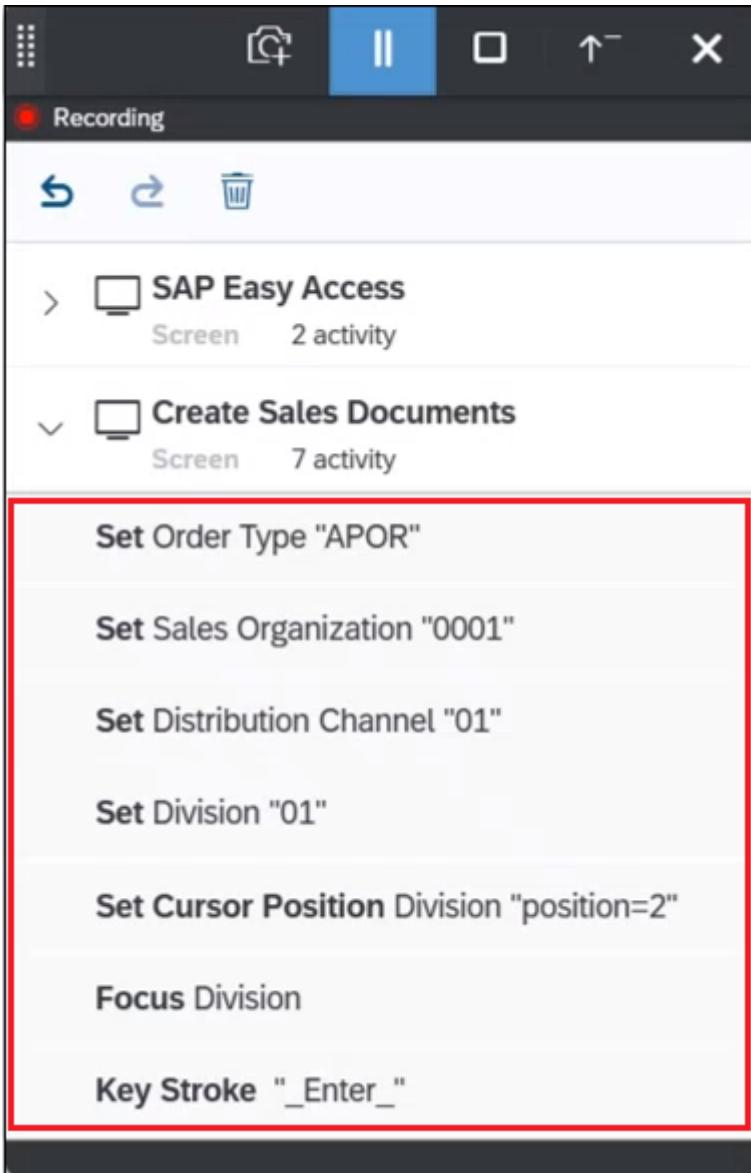
For example, refer to the below screenshot. The recorder automatically records all actions performed on the screen.



i Note

The actions performed on the screen are updated in the **Activities** list only after you press **Enter**.

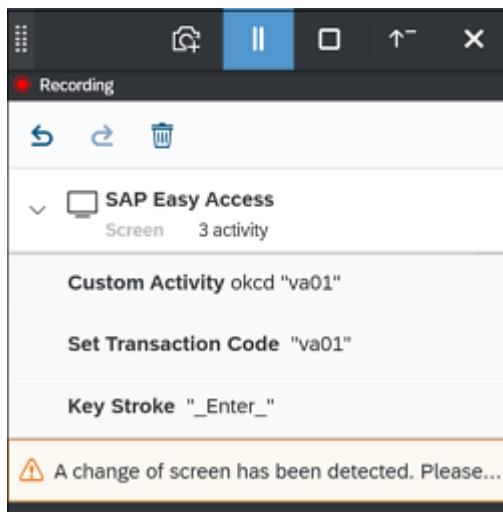
This is because, when you press **Enter**, a server round trip happens and the recording widget is refreshed with data from the server.



While recording, if a new screen appears or if there is a change in UI, then you must first click the button manually and then perform the actions in the application screens. This is applicable across the workflow recording.

Note

Whenever new screen appears or if there is a change in UI, the recorder detects the screen change in most of the cases and displays a hint message. This gives you a hint to capture a screen manually by clicking the button.



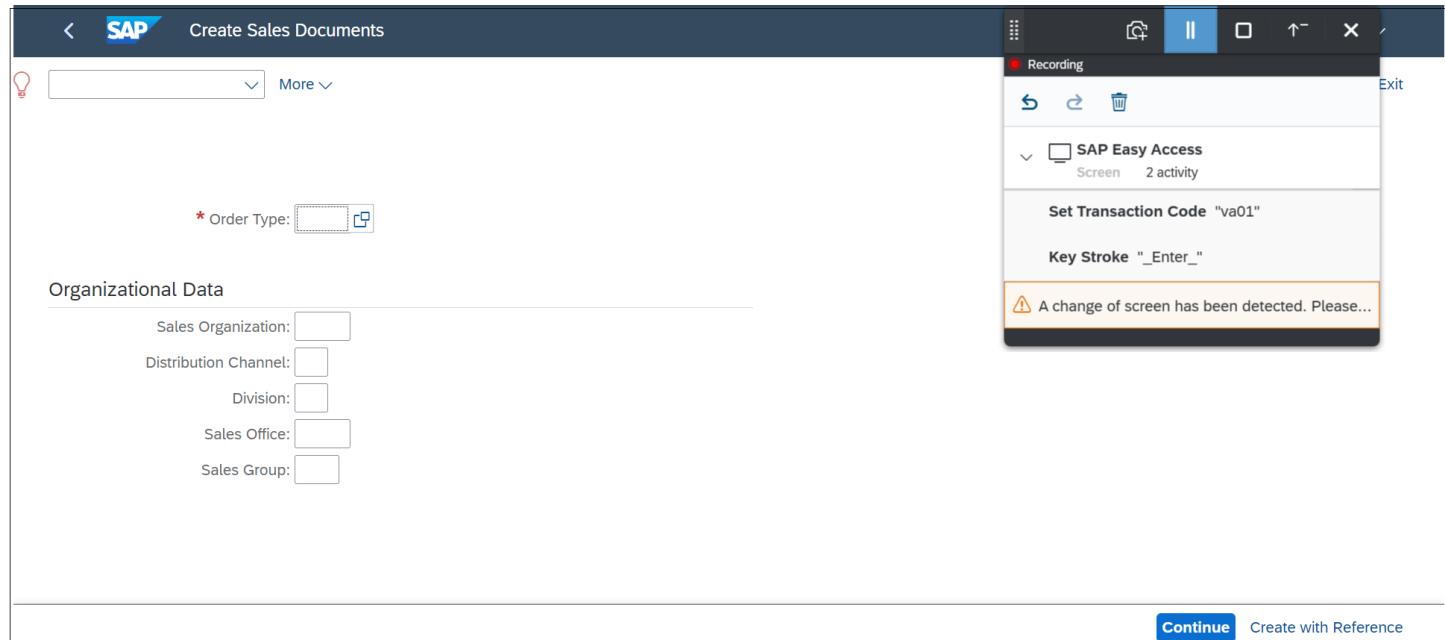
The Recorder may not detect screen changes in all cases. You must click the  button whenever there is a change in the screen or UI.

The hint message disappears when you click the button or perform any action in the screen.

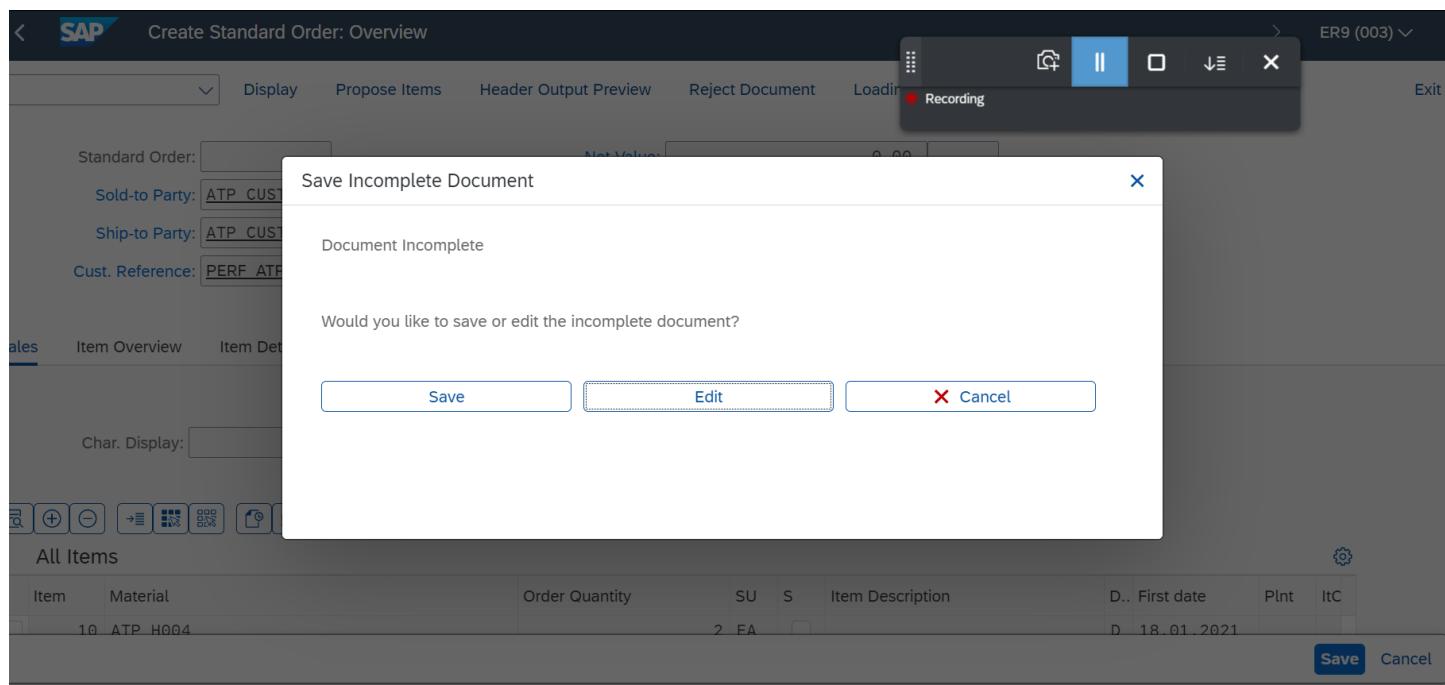
A new screen or change in UI may include application screen, modal dialogs, pop-ups, value helps (F4), UI elements (control, table, and hierarchical list) and many more.

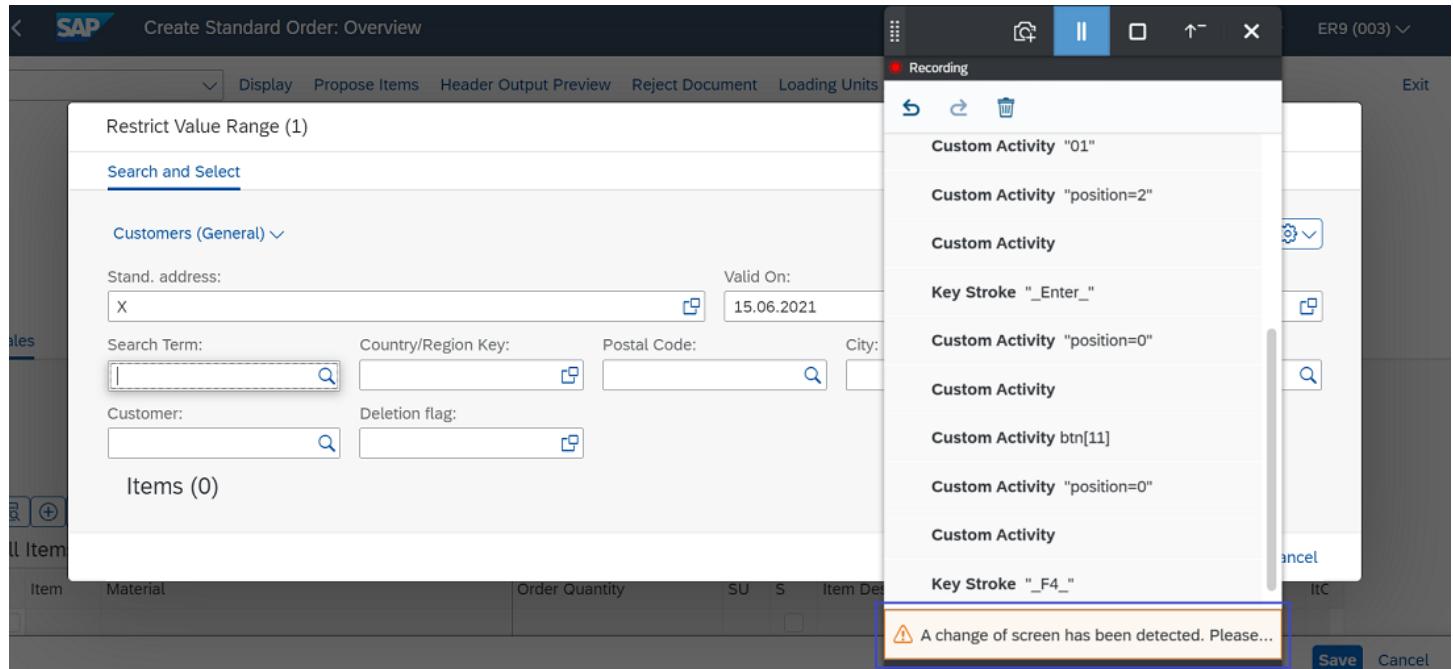
For example, refer to the below screenshot.

New Screen



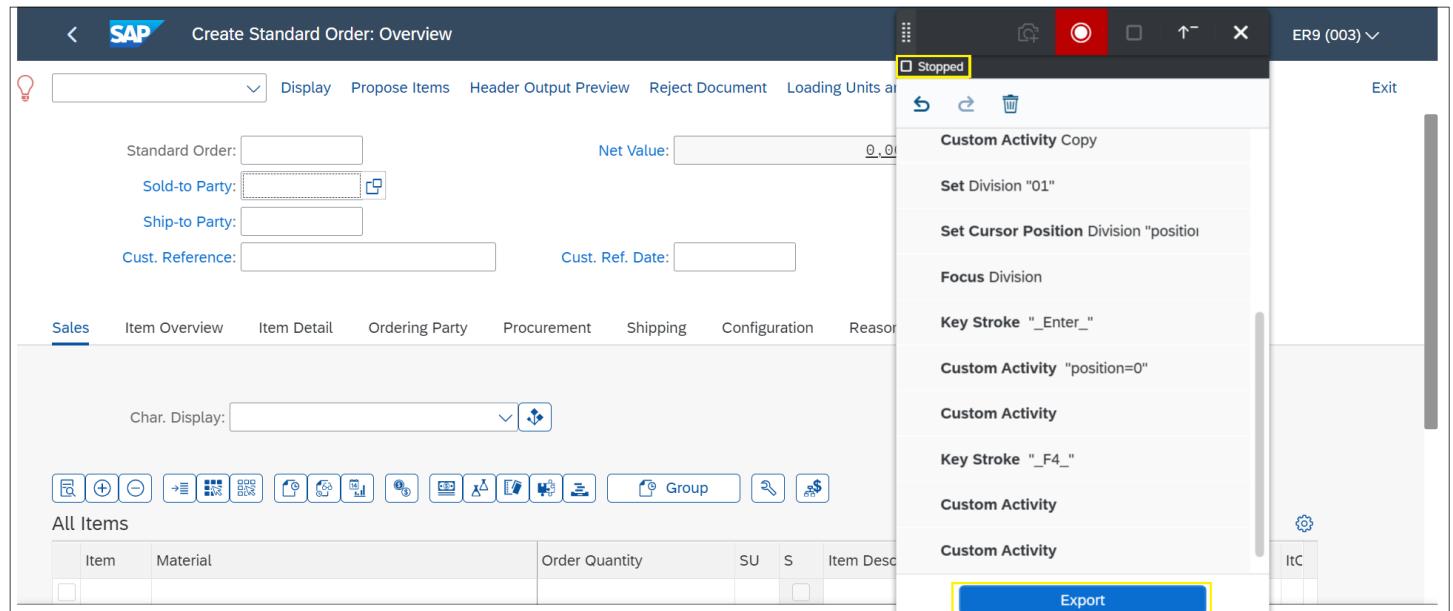
Pop-up Screen





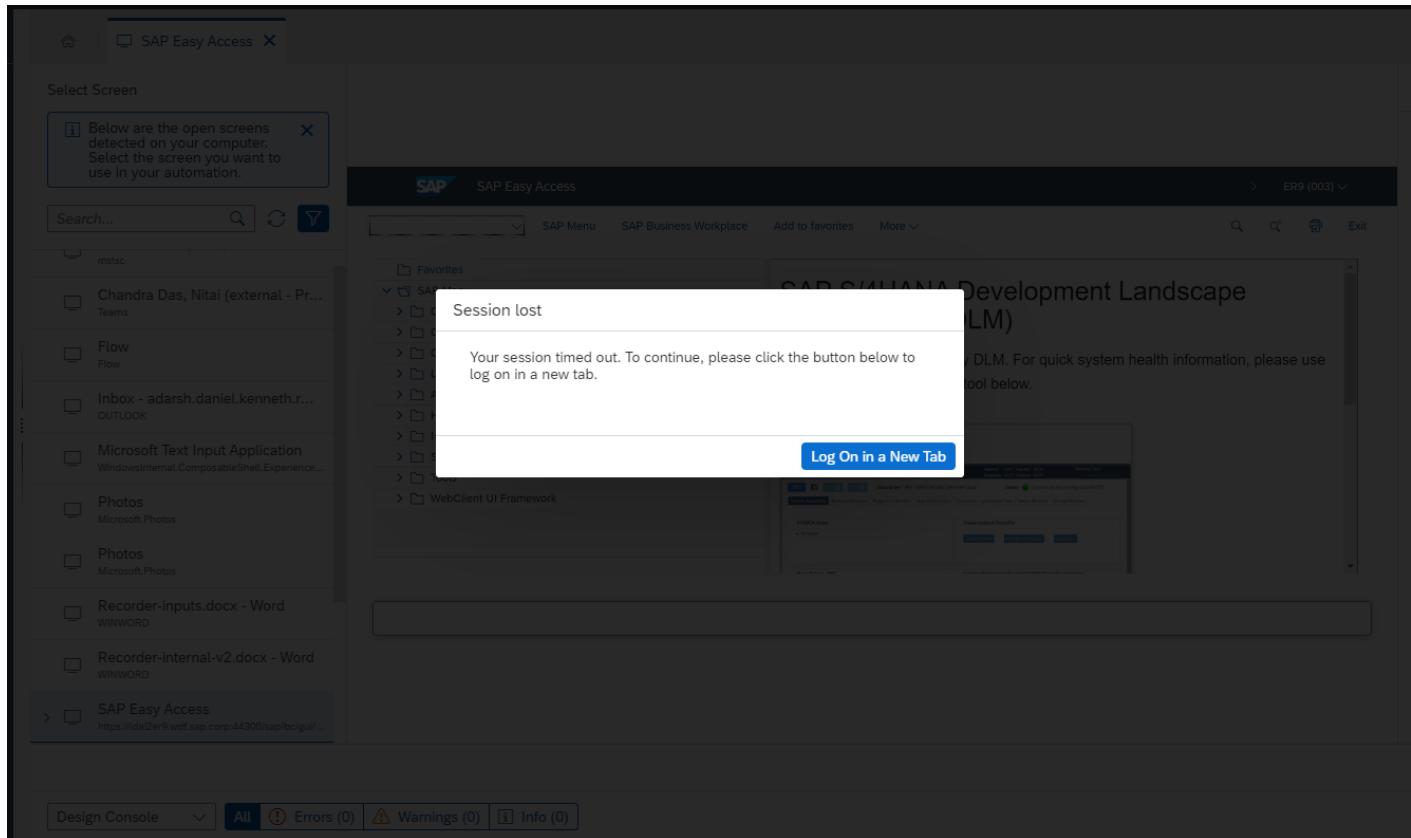
Once you have completed entering the data in the application screens, you can stop recording. Click the button to stop the recording.

After you stop the recording, by default, the **Export** button is displayed.

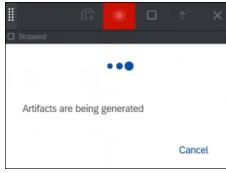
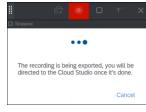
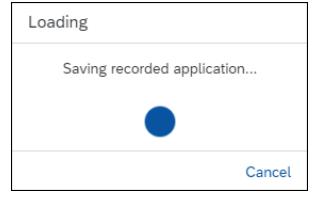
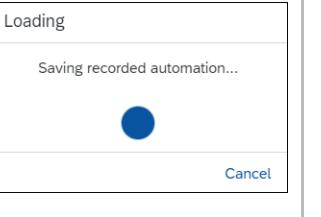


i Note

Before you start exporting the recording to the Cloud Studio, check if the Cloud Studio has not timed out. If it is timed out, you must re-login (do not refresh) to the Cloud Studio and then export the recording.



To export the recorded steps to the Cloud Studio, click **Export**. While exporting is in progress, the system displays the following pop-ups.

Recorder Widget	Recorder Widget	Cloud Studio	Cloud Studio	Cloud Studio
				
Cancel If you do not want to export the steps, then click. The Recorder widget retains the same state as before.	The Cancel is disabled.	If you do not want to export the steps, then click Cancel . The Recorder widget will be closed and Recording successfully cancelled message is displayed.	If you do not want to export the steps, then click Cancel . The Recorder widget will be closed and Recording cancelled message is displayed.	If you do not want to export the steps, then click Cancel . The Recorder widget will be closed and Recording cancelled message is displayed.

If you've recorded this application before, you are prompted whether you want to create a new automation or whether you want to update the existing recording. If you choose **Update an existing automation**, you are further prompted whether to attach the new part either at the beginning or the end of the existing one.

Export Recording

We found automations in your project containing the application you recorded.
Please chose if you want to create a new automation with your recording or if you want to update an existing one.

- Create a new automation
 - Update an existing automation

[Next](#) [Cancel](#)

After successful processing, **Recorder** widget is closed. You are redirected to the Cloud Studio and the confirmation message, **Recording successfully exported** is displayed. All the recorded steps are displayed in the automation designer of the Cloud Studio.

The screenshot shows the SAP Intelligent Robotic Process Automation Cloud Studio interface. The top navigation bar includes the SAP logo, 'Intelligent Robotic Process Automation Cloud Studio', and a project name 'Test38a'. On the left, there's a sidebar with various icons for navigation and management. The main workspace displays a workflow diagram with a green start node and several steps represented by red icons. A context menu is open over one of the steps, showing options like 'SAP Easy Access' and 'Create Sales Documents'. To the right, there's a panel titled 'Automation information' with tabs for 'Tools' (selected), 'Input/Output', and 'Info'. It includes a search bar, a status indicator (78%), and sections for 'Automations', 'Screens', 'Activities', 'Data', and 'Controls'. The bottom navigation bar shows links for 'Design Console (0)', 'Test Console (0)', and 'Variables (2)'.

You can then edit the automation to update the **Step Details**. For more details about designing automations in the cloud studio refer to the [Create an Automation](#) and [Manage Data within an Automation](#) sections.

View Recorded Screens

You can view the recorded screens under [Declared Application](#) and the recorded elements under [Declared Elements](#).

The screenshot shows the SAP Intelligent Robotic Process Automation Cloud Studio interface. In the top navigation bar, there are tabs for Overview, SAP Easy Access 1, and SAP Easy Access (which is currently selected). On the right side of the header, there are buttons for Share, Generate Package, and Settings.

The main workspace displays a captured SAP Web GUI screen titled "Create Sales Documents". This screen includes fields for "Order Type", "Sales Organization", "Distribution Channel", "Division", and "Sales Office". To the right of the screen, there is a "Screen Information" panel with tabs for "Data and Criteria" and "Advanced". Under "Data and Criteria", the "Name" is set to "Create Sales Documents" and the "Identifier" is "s_createSalesDocuments". The "Framework" is set to "SAPWebGui" and the "Technology" is "SAP Web GUI". The "Recognition Order" is set to 2. Below this, under "Recognition Criteria", it says "Screen uniquely identified" and "Recognition Tests: 4".

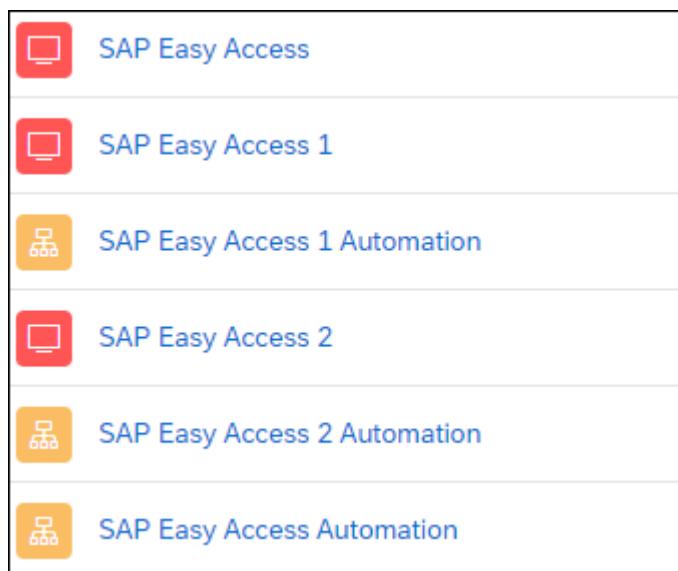
On the left side of the interface, there are two panels: "Declared Application" and "Declared Elements". The "Declared Application" panel lists "SAP Easy Access" and "Create Sales Documents". The "Declared Elements" panel lists "Order Type", "Sales Organization", "Distribution Channel", and "Division". There are also buttons for "Display all on screen" and "Design Console (0)", "Test Console (0)", and "Variables (0)".

If required, you can manually edit the screens to add or remove objects. For more details about editing the screens, refer to the [Declare an Application](#) and [Declare Elements](#) sections.

Capture Multiple Instances of the Same Application

You can declare multiple different applications that target the same running SAP WebGUI or SAPUI5 application. All these declared applications will be recognized independently at runtime without any interference.

In the following screenshot of the SAP WebGUI application, you can see three captured applications and their respective automations that are same in nature as they have captured same applications and pages. All three applications will be recognized independently at runtime, and their respective automations will be executed successfully.



i Note

The multi-application support is available on the manual capture process in the SAPUI5 application.

Prerequisites

The minimum SDK version must be 1.18.38 or above to enable this feature.

Test an Automation

This is custom documentation. For more information, please visit the [SAP Help Portal](#)

After you export the recorded steps to the Cloud Studio and finalize the automation, screens, and its elements, you can test the execution flow of your automation. For more details about testing an automation, refer to the [Test an Automation](#) section.

i Note

Batch Execution of Recorded Actions

In the SAP WebGUI Recorder, you can batch a set of recorded actions that are performed on the same screen and send those actions together to the REST GUI. It allows you to execute your automation faster.

While testing the recorded automation, you must enable the **Optimize for faster execution** option by selecting the following check box on the **Test Automation** window.



You must not use this option for debugging purposes as the actions are executed in a batch mode, and the data in the tester window might not be valid. Therefore, it is recommended to enable this feature once your automation is finalized.

The batch execution is enabled for a project that depends on SAP WebGUI SDK.

Generate Package

Once you have designed your project, you can generate a package in the Cloud Factory. To know more about generating packages, refer to the [Generate Packages](#) section.

i Note

While generating a package, you can also enable the **Optimize for faster execution** option by selecting the following check box on the **Generate Package** window. It will generate a package containing the optimized version of the automation.

Generate Package

*Name:
Test Package

Description:
Enter Package Description

Access rights:


Version Number:
1.0.0

Version Annotation:
Enter a version annotation specific to this version

Optimize for faster execution

Generate Package **Cancel**

Implement Collection in Recorded Elements

You can create an element and declare it as a collection or modify an existing recorded element and declare it as a collection and use them in the automation.

Prerequisites

You've captured at least one screen of the SAP GUI for HTML (also known as WebGUI), SAP GUI for Windows (also known as WinGUI) or SAPUI5 application using the Recorder. For more details about capturing the SAP WebGUI, SAP WinGUI or SAPUI5 application using the Recorder, see [Automatically Capture Applications with the Recorder](#). You've manually edited the screens to add or remove objects. For more details about editing the screens, refer to the [Declare Elements](#) section.

Example

The following scenario describes the usage of collection element while recording a sales order creation in SAP WebGUI application.

In the sales order creation, you will encounter a table where you need to enter the items such as **Material**, **Order Quantity** and so on.

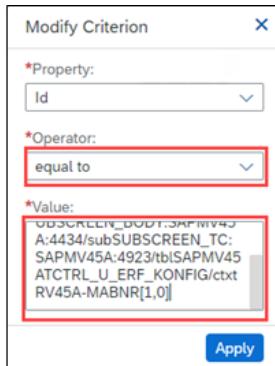
While recording you can just enter one **Material** and **Order Quantity** value in the table like the image mentioned above and finish creating the sales order. An element will be created for both the first cells of the **Material** and the **Order Quantity** columns of the table.

In the recorded application of the sales order creation, two elements `cell[1]` and `cell[1]_1` are created under **Declared Elements**. The `cell[1]` is for the first cell in the **Material** column and the `cell[1]_1` is for the first cell in the **Order Quantity** column.

By default, the criteria for `cell[1]` element is `|d` equals

wnd[0]/usr/tabsTAXI TABSTRIP OVERVIEW/tabbP\01/ssubSUBSCREEN BODY:SAPMV45A:4434/subSUBSCREEN TC:SAP

Here the '0' in the Id represents the first cell in the column.



Context

You can record a dynamic number of rows using the collection control feature in the SAP WebGUI application. This means you can record one entry on a row and export it to the Cloud Studio. In the Cloud Studio, you need to make few modifications to the recorded application (such as modifying the **Criteria**, marking the item as occurs) and the automation (such as adding the **For each** loop) to create a dynamic number of rows. When the bot is executed, all defined number of rows will be filled in automatically, like in any other technology.

To implement collection in recorded elements, follow the below procedure:

Procedure

- Under **Declared Elements**, select the element to be declared as collection, modify the criteria of the element, and then click **Is a collection** () button.

In the sales order creation, select **cell[1]** and **cell[1].1** element respectively, and make these two elements into a collection by changing the criteria in such a way that it can recognize all the cells in that column.

Modify the criteria for **cell[1]** to **Id contains**

`wnd[0]/usr/tabsTAXI_TABSTRIP_OVERVIEW/tabpT\01/ssubSUBSCREEN_BODY:SAPMV45A:4434/subSUBSCREEN_...`

Here you can see that the Id is trimmed till the first index so that it is unique for all the cells in that column.

Modify Criterion

*Property: Id

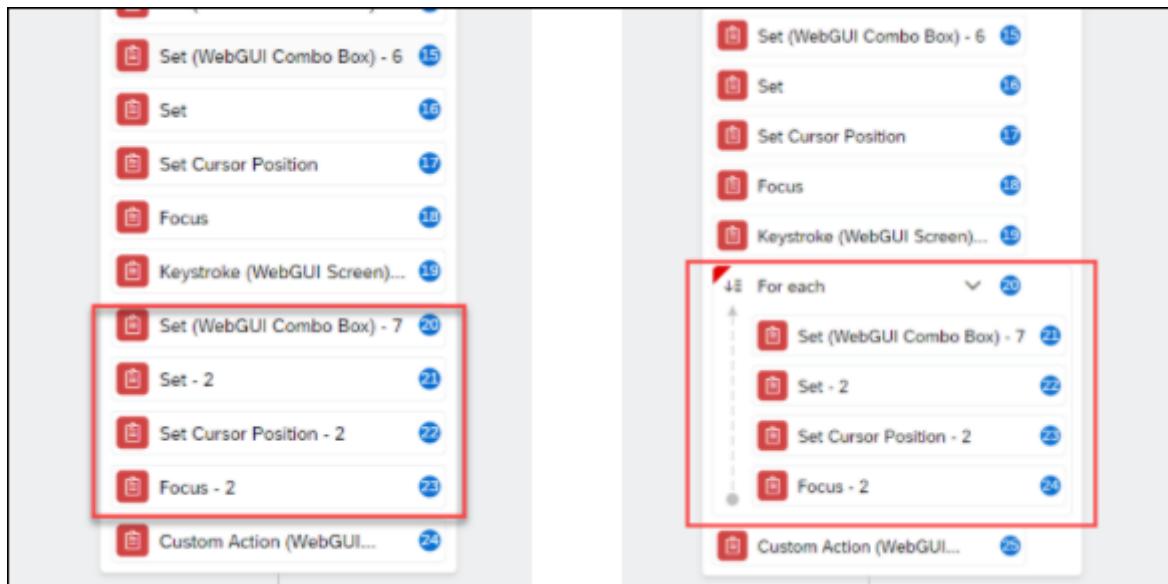
*Operator: contains

*Value:
SUBSCREEN_TC; SAPMV45A:4923/tblSAPMV45 ATCTRL_U_ERF_KONFIG/ctxt RV45A-MABNR[1]

Apply

2. Identify all the steps that have collection element as target in the automation and then embed them inside a **For each** loop.

In the sales order creation, identify all the steps that have **cell[1]** and **cell[1]_1** as target in the automation and then embed them inside a **For each** loop.



3. Modify the **For each** loop in such a way that it will enter the same item values (such as **Material** and **Order Quantity** values in the sales order creation) in all the visible rows of the table.

4. Add a **Custom script** before the **For each** loop and add the relevant code inside it. This code will return all the occurrences of an element in a list during runtime.

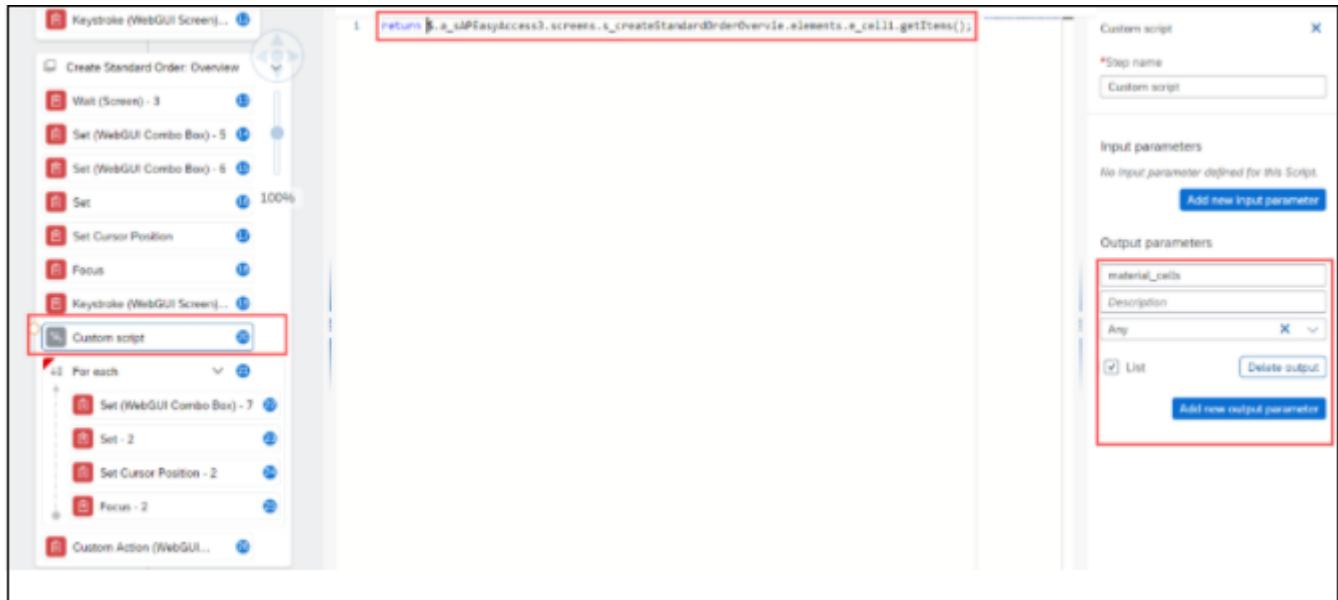
In the sales order creation:

- o Add the following code inside the **Custom script**.

```
return $.[app].screens.[screen].elements.[element].getItems();
```

- o Replace the placeholders with the appropriate application, screen, and element name.
- o The element used in the code mentioned above can be any one of the collection elements you have created (**cell[1]** or **cell[1]_1**).
- o Here, the **cell[1]** element represents the material column in the **Custom script**.

5. Create an output parameter to store the list of elements returned by the **Custom script**.



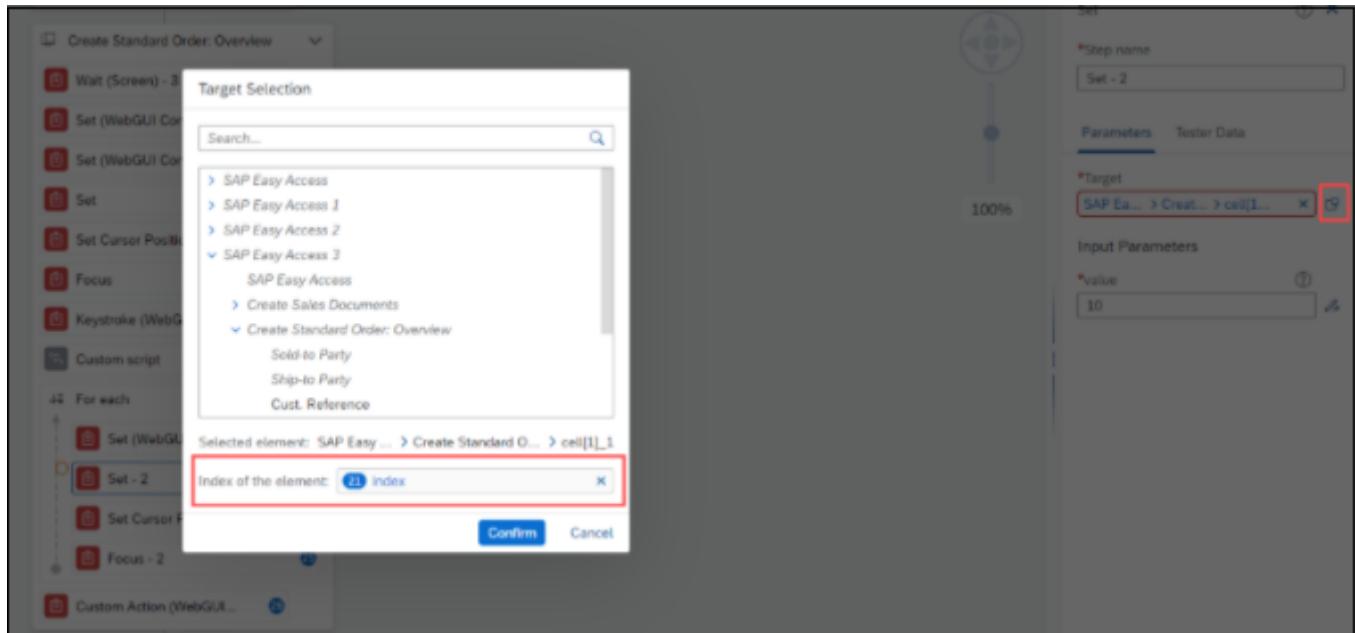
6. Assign the output parameter from the **Custom script** to the **For each** loop as a looping list.

In the sales order creation, you must assign the output parameter **material_cells** as a looping list.

You can see that the **For each** loop provides two **Loop Parameters** which you can use in the steps inside the **For each** loop.

In the sales order creation, **currentMember** loop parameter contains the element from the **materials_list** for that iteration and **index** loop parameter contains the index for the iteration (starts from 0).

7. Use the **index** parameter to update the target index for the steps inside the **For each** loop.



8. Click **Test**. For more details about testing an automation, refer to the [Test an Automation](#) section.

Results

The bot will fill all the visible rows with the same set of values even though the count of the visible rows can vary from system to system. Therefore, you have executed your automation successfully.

Edit a Custom Activity in Generated Automation

While recording application of any technology, if you perform a certain action that is not available in the SDK, a **Custom Action** appears in the automation.

Step	Action
1	Create Standard Order: Overview
2	Wait (Screen) - 3
3	Set (WebGUI Combo Box) - 5
4	Set (WebGUI Combo Box) - 6
5	Set
6	Set (WebGUI Combo Box) - 7
7	Set - 2
8	Set Cursor Position
9	Focus
10	Custom Action (WebGUI...)

The custom activity helps you to execute the automation without missing out on any recorded action.

By default, the **Input Parameters** of the custom activity is not editable as these parameters are mostly technical and therefore, it won't be of much use to you.

However, you must change the parameter of the custom activity in some scenarios. For example, if the **Custom Action** step is for setting a value in a text field, you must change the value to be set in that text field.

To edit the parameters in the custom activity, you must perform the following steps:

- Drag and drop a new empty custom activity on the same **Element/Screen** target and move the activity next to the existing custom activity.

i Note

In any technology, the **Element** or **Screen** type will have its own custom activity.

- Copy the parameters from the existing custom activity to the new custom activity and edit the parameter as per your need.
- Delete the existing custom activity.

Re-recording an Application

When you record an application, you always create a new application and automation at the end of the recording. The **Re-recording of Application** feature allows you to re-record or update an existing application and enhance the existing automation by merging newly recorded artifacts with it.

Prerequisites

In your project, you already have an existing application and automation.

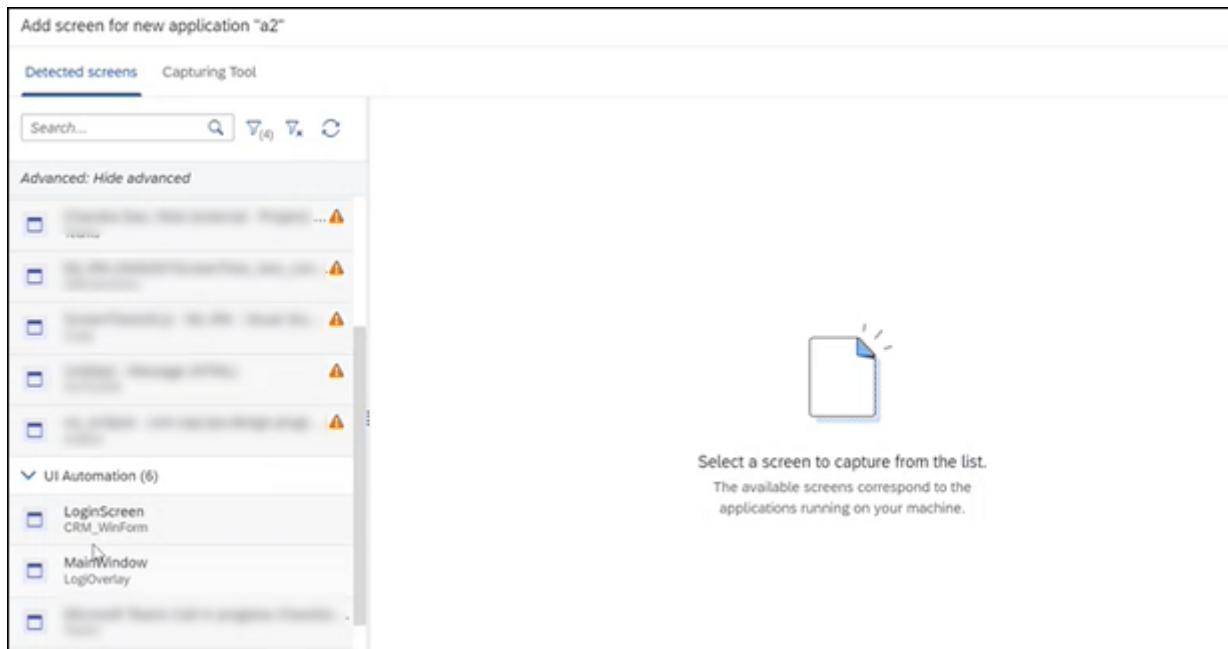
Context

Using this feature, you can perform the following actions inside your project.

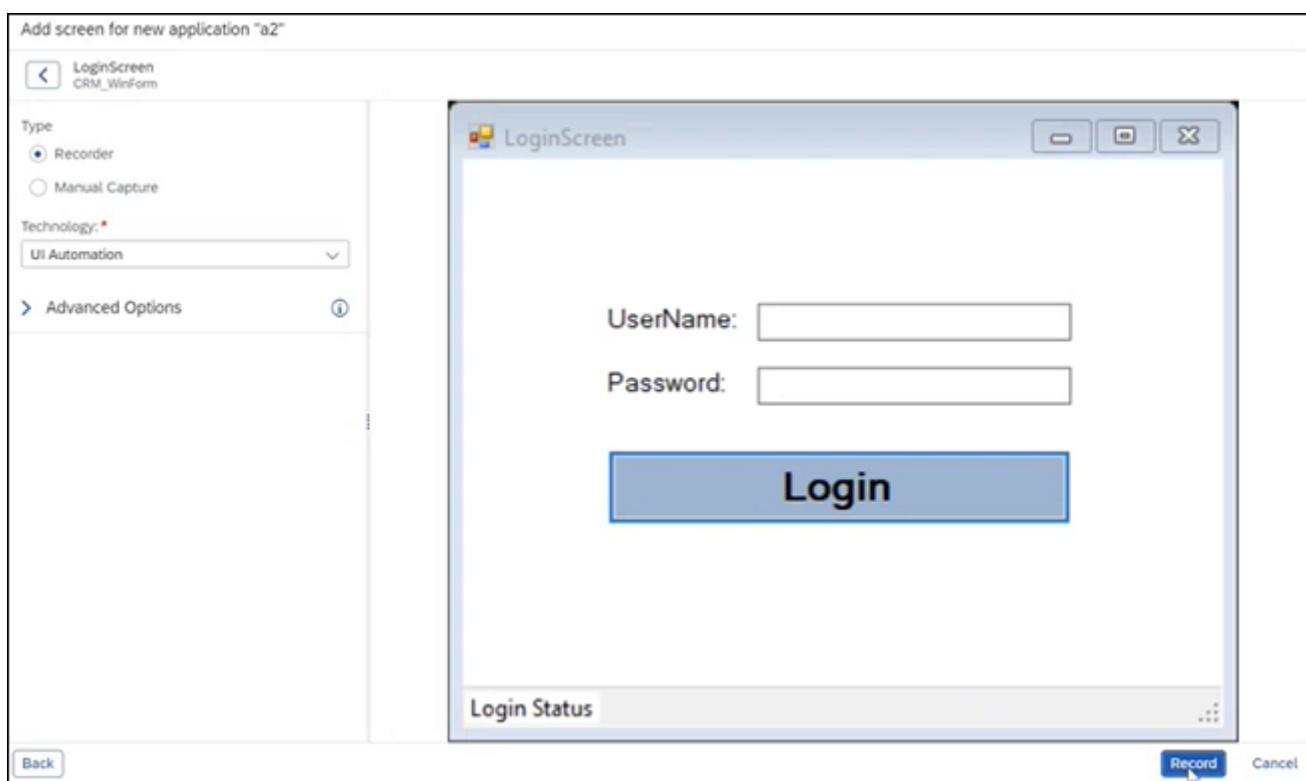
- Create a new application and automation.
- Update an existing application and then create a new automation.
- Create a new application and then update an existing automation.
- Update an existing application and automation.

Procedure

1. **[Optional]:** Create a new project. For more information, see [Create a Project](#).
2. In the project explorer, click **Create**.
3. In the artifact menu, click **Application**. The **Create Application** popup window is displayed.
4. In the **Create Application** popup window, enter the following:
 - Enter a name in the **Application Name** field.
 - Optional. Edit the **Application Identifier** field without using a space.
 - Enter a short description in the **Description** field.
 For more information, see [Capture an Application](#).
5. Click **Create**. A new tab labeled to the name of the captured application is displayed. The system starts detecting the applications and their screens currently running on your local machine. When it's done, a list of screens is displayed in the picker panel on the left.



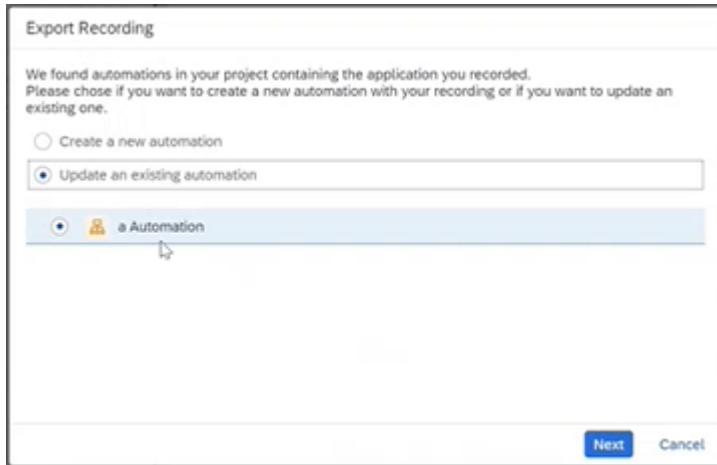
6. Select the application you want to record from the list of screens. A preview of the screen is displayed in the capture area.
7. Click **Next**. This redirects you to the application that you want to record.
8. To start the recorder, click **Record**.



9. To initiate the recording, click the  (Record) button. The Cloud Studio is locked and the selected application is recorded.
10. To stop the recording, click the  (Stop) button.
11. Click the **Export** button.
12. In the **Export Recording** popup window, choose one of the following options:
 - o **Create a new automation**
 If you select this option and click **Next**, the **Successfully created the automation** confirmation message is displayed and you are redirected to the Cloud Studio.

- o Update an existing automation

If you select this option, the existing automation is displayed.



a. Click **Next**.

b. In the **Insert Recording** popup window, decide where to add the new recording:

- **End of current automation**
- **Start of current automation**

c. To insert the steps into your existing automation, click **Insert**.

The **Successfully saved the automation** confirmation message is displayed, and you are redirected to the Cloud Studio.

Screen Capture Modes of the Recorder

There are three screen capture modes available in the Recorder. They are:

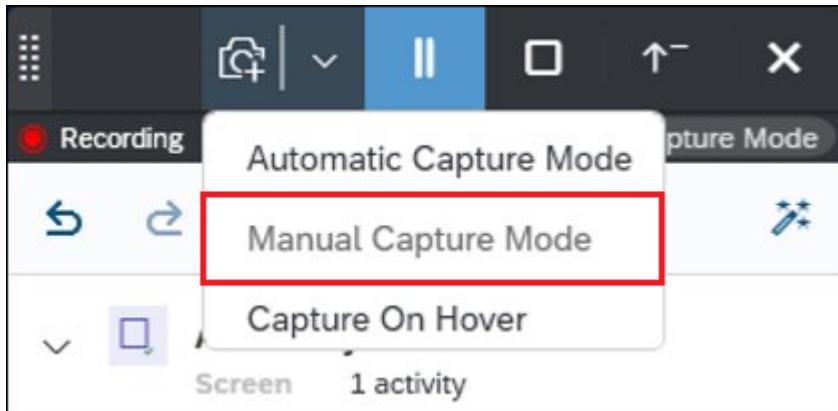
1. [Manual Capture Mode](#)
2. [Capture On Hover](#)
3. [Automatic Capture Mode](#)

i Note

By default, the capture mode depends on the application. You can anytime switch from the **Automatic Capture Mode** to the **Manual Capture Mode** and vice versa.

Manual Capture Mode

This is a standard mode to capture the screen by clicking the camera () button. While recording automation, whenever a new screen appears or if there is a change in the screen, a capture hint message is displayed on the recording widget. Once there is a capture hint message, you must recapture a screen manually by clicking the button and then perform the recording.

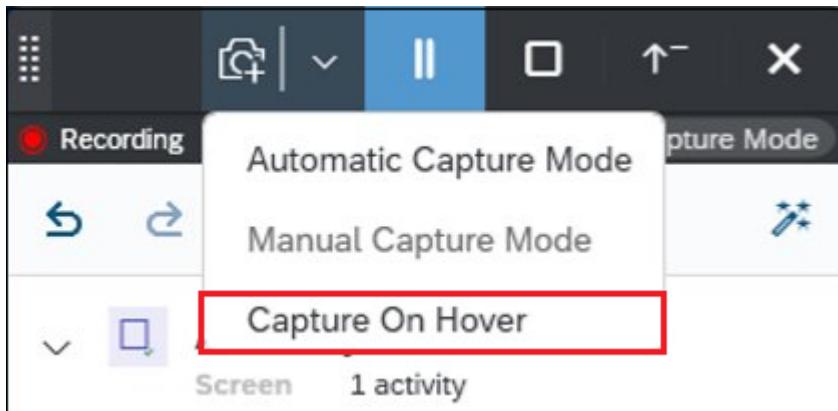


The Recorder captures the first screen of an application automatically and then subsequent screens must be captured manually by clicking the button.

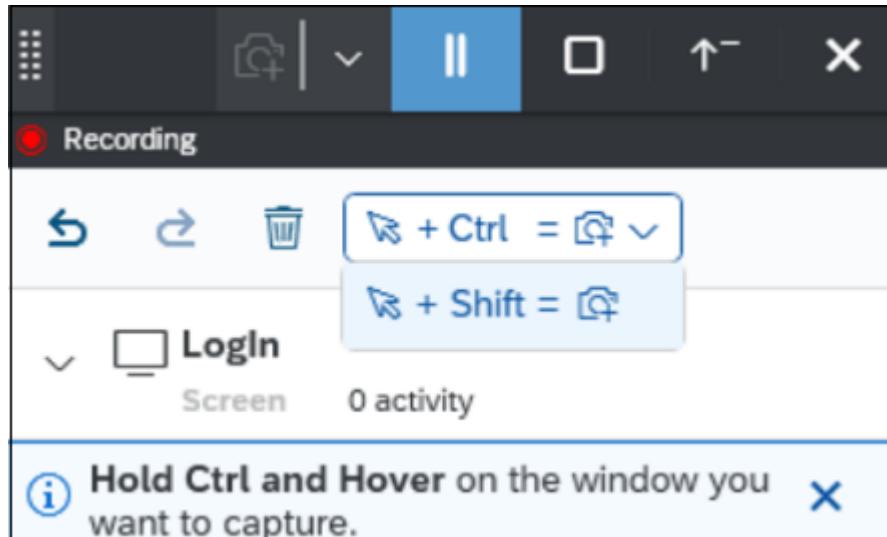
The steps are recorded even if you do not click the button manually to capture new screens. However, the generated automation will be incorrect.

Capture On Hover

The manual capture button takes a screenshot and captures meta-data of the application being recorded. However, this is not possible for some UI components such as sub-menus in a windows application or a web-applications, which become visible only when they are clicked because these UI elements will be hidden when user clicks on manual capture button. In such cases, you can still capture the items such as menu, submenu or any hover control using the **Capture on Hover** mode. This mode enables the capture on **hover + Ctrl** or **Shift** key press.

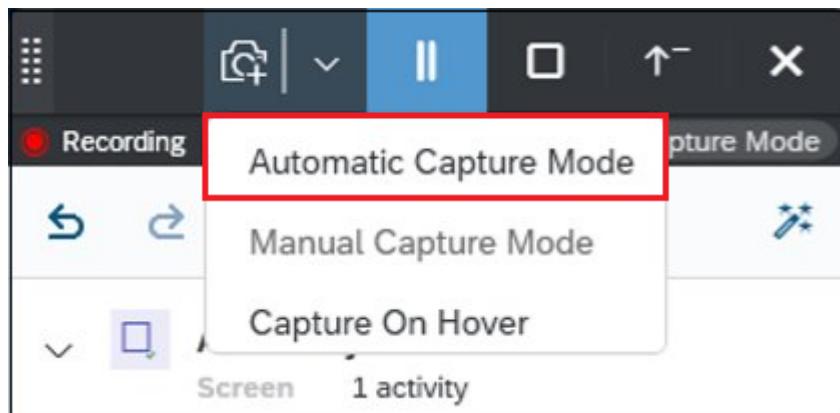


When using the **Capture On Hover** mode, you need to hover over the context menu or window which needs to be captured and then hold the **Ctrl** (or **Shift**) key till the capture begins. The application screen being captured will be highlighted when the capture begins and then you release the **Ctrl** (or **Shift**) key to avoid duplicate captures.



Automatic Capture Mode

The Recorder can also do automatic capture whenever a new screen appears or if there is a change in the screen.



If the automatic capture fails, an error message is displayed on the screen, and then you can switch to the [Manual Capture Mode](#) to capture the screen.

The following list of SAP technologies support the [Automatic Capture Mode](#).

- SAP WinGUI
- UI Automation
- Web SAP Applications
 - SAP WebGUI
 - SAPUI5
 - SuccessFactors
 - Ariba
 - ByDesign
 - Concur

Non-SAP web pages don't support the [Automatic Capture Mode](#).

i Note

For SAPWinGUI, [Automatic Capture Mode](#) works from 770 patch 3 onwards.

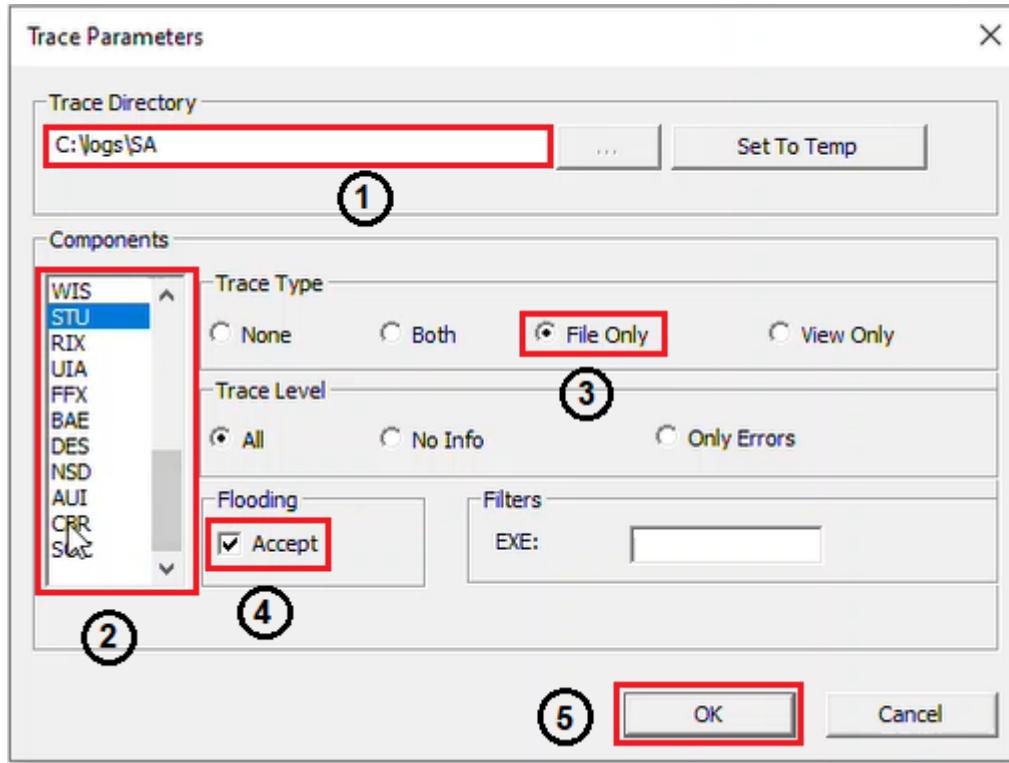
Enable Logs for Recorder in TraceViewer

Context

You need to open the TraceViewer tool and enable the **STU** and **CBR** components to capture logs for the recorder.

Procedure

1. Enter the trace directory path in the **Trace Directory** (1) input box.
2. Select the **STU** component trigram from the **Components** (2) list box.
3. Check the **File Only** (3) radio button under **Trace Type**.
4. Check the **Accept** (4) checkbox under **Flooding**.
5. Select the **CBR** component trigram from the **Components** list box.
6. Check the **File Only** radio button under **Trace Type**.
7. Check the **Accept** checkbox under **Flooding**.
8. Click **OK** (5) to close the **Trace Parameters** dialog box.



i Note

To collect the logs, make sure that the Trace Viewer program is open during your trace session in the recorder. However, you can minimize the program.

Record SAP GUI for Windows Application

You can use the recorder to automate applications built using SAP GUI for Windows (also called SAP WinGUI).

This section provides tips and details to help you to understand and use the recorder appropriately with the SAP WinGUI system.

Prerequisites

To use recorder with SAP WinGUI, make sure you use only 32-bit SAP GUI for Windows Application.

i Note

SAP GUI for Windows applications can be automated in SAP Intelligent RPA using the SAP GUI connector (for manual capture) and the recorder. Both rely on SAP scripting which depends on the version of SAP GUI for Windows client.

We recommend using SAP GUI for Windows 7.40 and above. Please also take into account the SAP GUI for Windows releases as mentioned in the following note to understand the maintenance strategy and deadlines: [147519](#).

Check the [SAP GUI Scripting API documentation](#) to find the SAP Scripting requirements. You also need to take into account that:

- Applications may change from one release to the next. This means scripts for a transaction may work in one release, but not in the other if the application changed.
- Some applications are using controls or logic which are not compatible with scripting (see the limitations in the following note: [587202](#)).

More Information

1. Default Capture Parameters

The default capture parameters in WinGUI recorder are:

- a. Capture Mode = Print Window
- b. Max Capture Height = 16000 Pixels
- c. Max Deepness = 0
- d. Max Siblings = 0

These values are fixed for recording scenario and cannot be changed after recording.

2. WinGUI Recorder Activities

The activities introduced for WinGUI Recorder are:

- a. Open Connection

SAP WinGUI recorder uses Open Connection activity to open connection to SAP GUI system when running an automation. This activity is generated automatically when recording is completed. If you want to connect to another SAP GUI system, then change the connection name parameter of this activity with connection name as shown in the SAP Log On page. Example, Q43 [PUBLIC].

i Note

The connection name must be the same as what you see on your SAP Logon pad.

- b. Close Connection

Closes connection to SAP GUI system that was opened by Open Connection and Open Connection By String.

- c. Click Sync

Recorder generates "Click Sync" activity on GUI Button to perform click synchronously. If Click Sync operation initiates a time-consuming action, then replace with Click.

3. Capture Meta-data

To capture meta-data faster, fewer properties are captured during recording.

Recorder captures meta-data such as Id, Text, Type, Name, ScreenLeft, ScreenTop, Width, and Height properties for a page or item depending on the control type.

If other properties like IconName, Tooltip, LeftLabel, RightLabel, Changeable, and ContainerType are required for recognition, then the page must be recaptured in cloud studio after recording artifacts are exported to Cloud Studio.

4. Two or More Screens of the Same Name

When two or more screens of the same name are added to the SAP GUI Application after exporting the recording, add a unique component from the DOM for each page as a Must Exist element to aid proper recognition.

Initially, with the generated automation, each page sharing the same title will work and not have issues in runtime because of the sequence of the screen appearance in the bot. When the automation is scaled, there could be recognition issues if handling multiple screens with the same criteria process is not in place. To know more details about handling multiple screens with the same criteria, refer to the [Handle Multiple Screens with the Same Criteria in SAP GUI for Windows Recorder](#) section.

5. Password Security

In SAP GUI recorder, if any password field is encountered then its value will be recorded as '*****'.

For the automation to work, the bot developer needs to select the respective step in the automation and change the value of the input parameter from '*****' to your actual password. You can set the value as a parameter and provide the password at runtime.

Refer to the section [Best Practices for Environment Variables of Type Password](#) for best practices for password security.

6. Session Busy

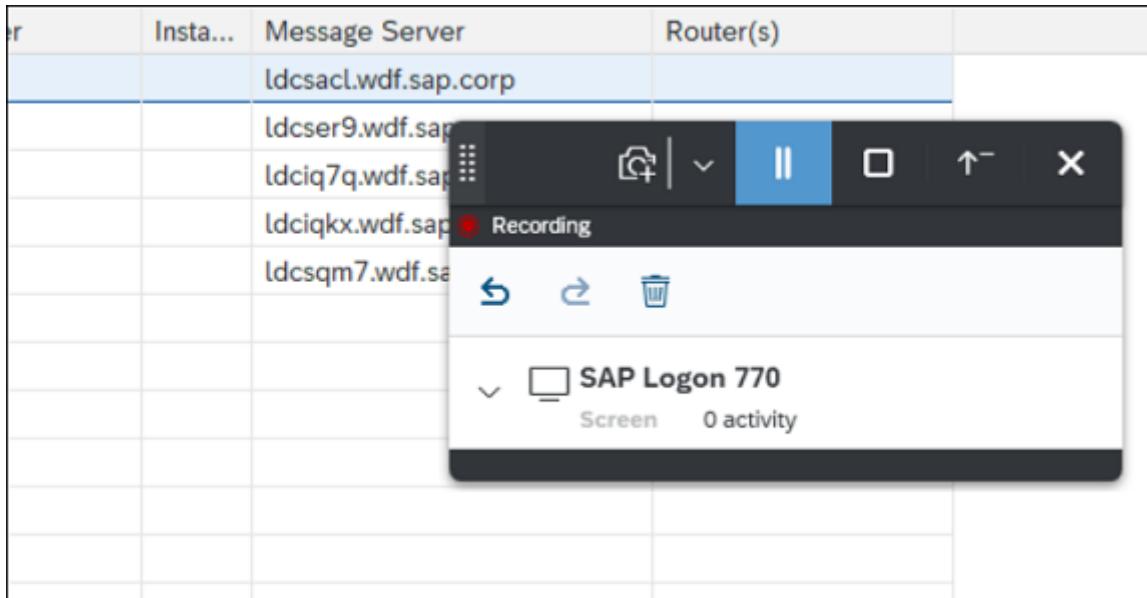
While running the automation, If Current Session is Busy... exception is encountered, user must increase the busyTimeOut value for Set Busy Wait Time activity in the automation.

The Set Busy Wait Time activity helps the SAP WinGUI bot to check if the session is busy before sending each action to the actual SAP GUI application.

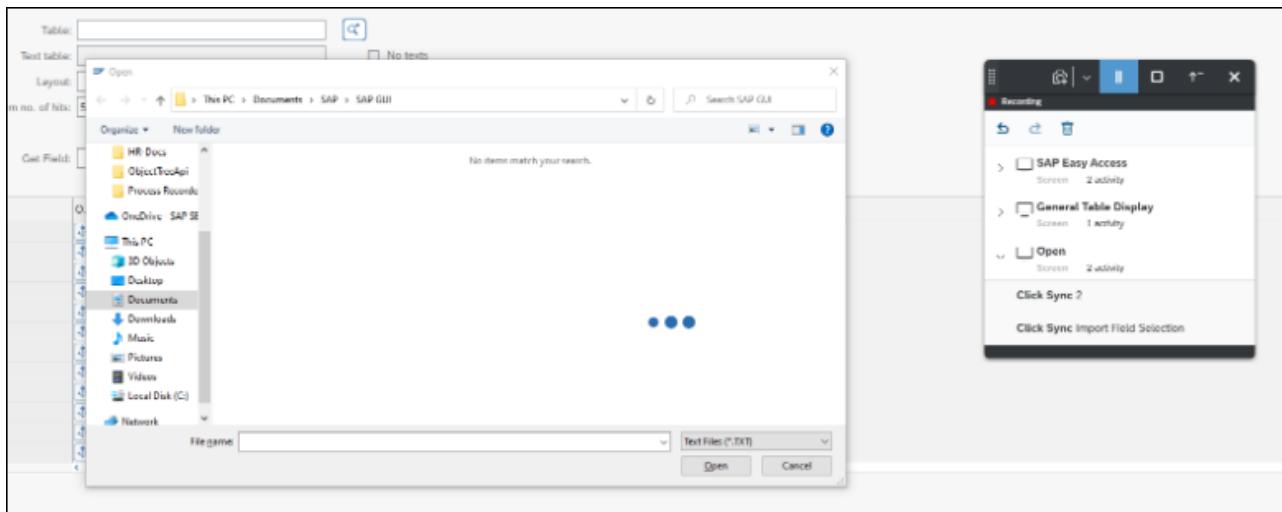
7. Cross Technology Recorder

You can use the Recorder to record applications with a combination of the SAP WinGUI and the UI Automation technology pages.

- Recording can be started from the SAP Logon page. The SAP Logon page will be captured as the UI Automation page and other pages of the SAP WinGUI will be captured as the SAP WinGui pages.



- It is possible to record native Microsoft Windows dialog boxes even if the Recorder is launched for the SAP WinGUI technology. Therefore, it is not required to capture such dialog boxes manually and then integrate them into the project.

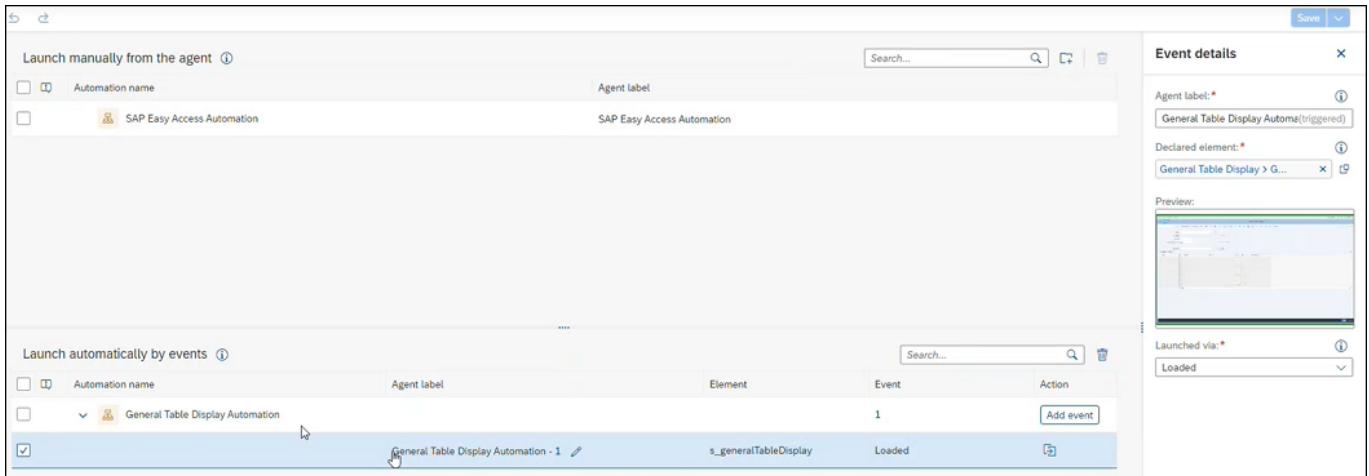


8. Technical Events

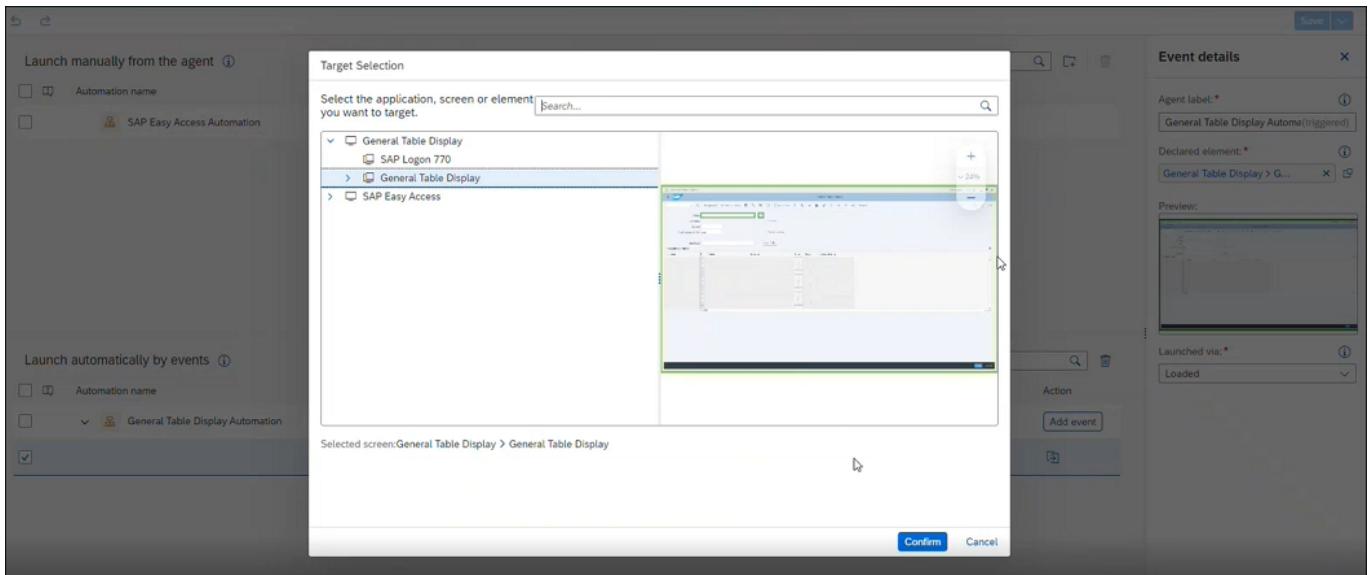
An automation can be launched automatically in attended mode by using Project Launcher by configuring technical events. For more information, see [Trigger Automations Using the Project Launcher](#).

The SAP WinGUI supports the launch of automation automatically in attended mode by using only screen load or unload event. You can define events by selecting either load or unload event under events dropdown. When you load or unload the selected screen, the automation triggers.

In the following example, automation is launched automatically in attended mode by using a screen load event. To define events, **Loaded** event is selected under **Launched via** dropdown.



When the selected screen is loaded, the automation triggers.



Related Topics

- [Overview](#)
- [Prerequisites](#)
- [Automatically Capture Applications with the Recorder](#)
- [Screen Capture Modes of the Recorder](#)
- [Enable Logs for Recorder in TraceViewer](#)
- [Record UI Automation](#)
- [Record Web Applications](#)
- [Error Management](#)
- [Best Practices](#)
- [Read Value of an Element During Recording](#)
- [Set Value of an Element During Recording](#)

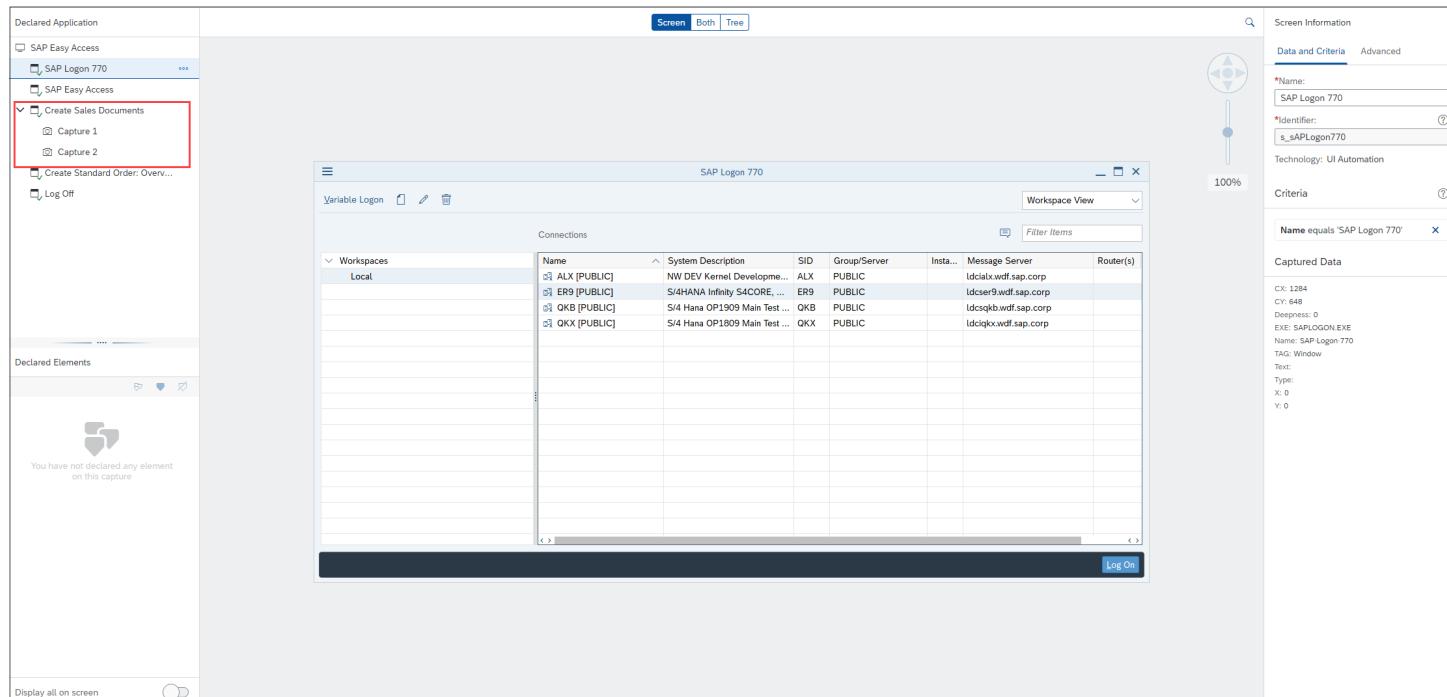
Handle Multiple Screens with the Same Criteria in SAP GUI for Windows Recorder

This section describes how to handle multiple screens with the same Criteria in SAP GUI for Windows (also called SAP WinGUI) recorder.

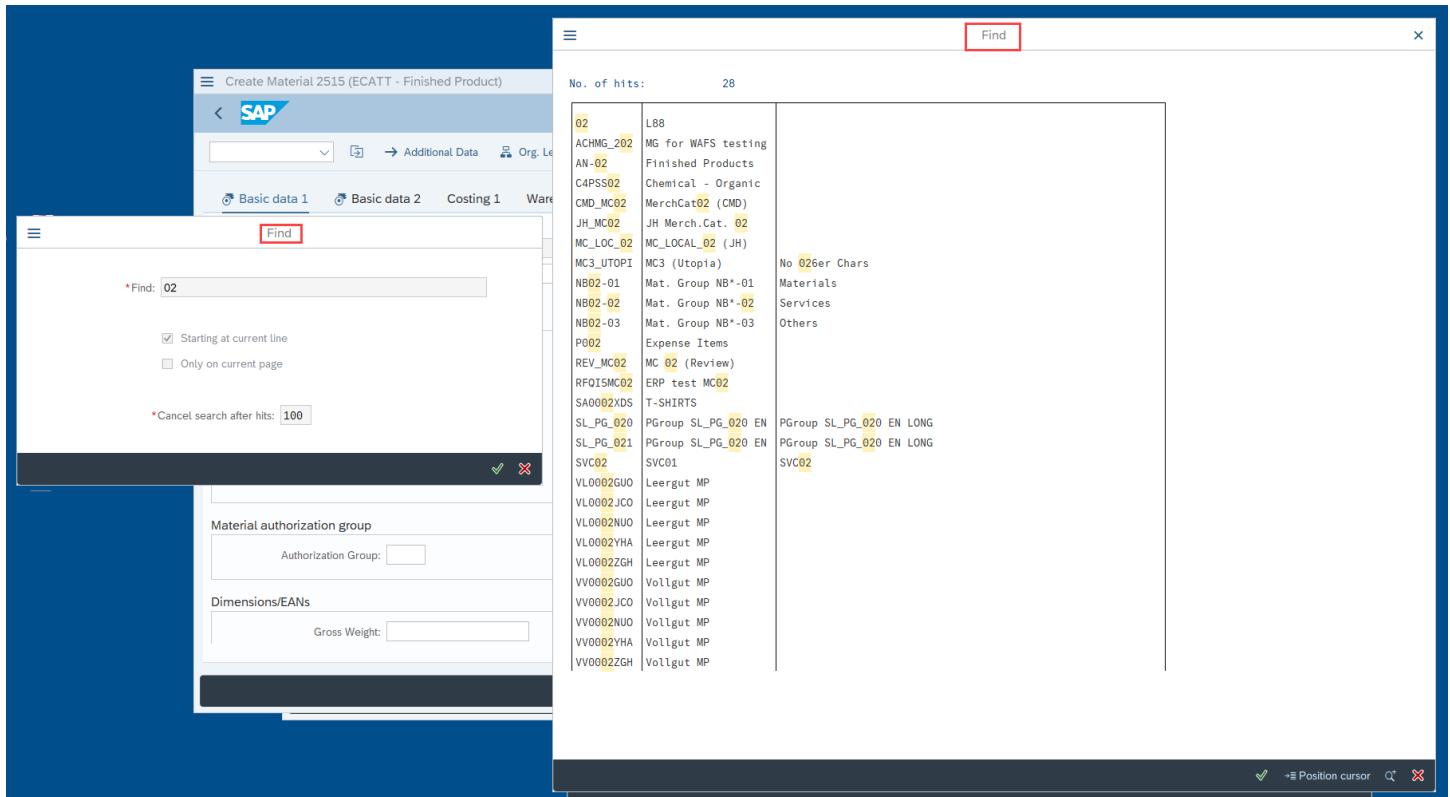
i Note

To know more details about declaring elements, refer to the [Declaration Phase](#) section.

In most of the cases, all the screens with the same title (criteria) will be grouped together as multi-capture by the SAPWinGUI Recorder.



In some cases, two or more screens of the same title may appear simultaneously during the recording process. For example, refer to the below screen.



To recognize (LOAD) all screens simultaneously during runtime, the recorder will choose to declare them as separate captures.

The screenshot shows the SAP GUI Recorder interface. On the left, the "Declared Application" tree includes "SAP Easy Access", "Create Material (Initial Screen)", "Select View(s)", "Organizational Levels", "Create Material 2516 (ECATT ...)", "Material Group 460 Entries", and two entries under "Find" which are highlighted with a red box. The "Declared Elements" section shows "Find" and "Find (Enter)". In the center, a "Find" dialog is displayed with the same search parameters as the previous screenshot. On the right, the "Screen Information" panel shows details for the "Find" dialog, including its name ("Find"), identifier ("s_find"), technology ("SAP GUI"), and captured data (CX: 660, CY: 338, Deepness: 0, EXE: SAPLOGON.EXE, Name: wnd[2], TAG: GuiModalWindow, Text: Find, Type: GuiModalWindow, X: 405, Y: 399). A criteria section indicates "Text equals 'Find'".

Now, the bot works fine as the captures are declared in a proper sequence in the Automation. To make recognition of the screens more robust, the following steps are recommended:

1. In the above scenario, we have two pages with **Find** as the criteria (Title). Let's call these two pages, **Find A** and **Find B**.
2. First, take the screen **Find A** and try to identify a unique element in the Tree (DOM) by comparing it with **Find B**'s Tree and add that as a **Must Exist** element.

The screenshot shows the SAP GUI interface with two 'Find' dialog boxes side-by-side. The left dialog, labeled 'Find A', has a red box around its search input field. The right dialog, labeled 'Find B', has a red box around its search input field. The 'Element Information' panel on the right details the unique element for 'Find A': **GuiTextField** with **Name: *Find**, **Identifier: e_find**, and **Id equals 'usr/txtRSYSF-STR...'**.

In the above image, in **Find A**, the text field declared as a **Must Exist** element is unique to that screen as the element with the same ID is not found in **Find B** after the tree comparison.

Follow the same procedure and identify a unique **Must Exist** element for **Find B** in comparison to **Find A**.

The screenshot shows the SAP GUI interface with two 'Find' dialog boxes side-by-side. The left dialog, labeled 'Find A', has a red box around its search input field. The right dialog, labeled 'Find B', has a red box around its search input field. The 'Element Information' panel on the right details the unique element for 'Find B': **GuiLabel - CMD_MC** with **Name: unique element for FindB**, **Identifier: uniqueElementForFindB**, and **Id equals 'usr/lbl[1,6]'**.

Handle Context Menu Button Activity

This section describes the alternative method for choosing a context menu element after clicking the Context Menu button in the GUI toolbar.

In some cases, the automation can fail because of the dynamic nature of the function code for the context button.

Perform the following steps to fix this issue.

Procedure

1. Delete '[Press Context Button](#)' and '[Select Context Menu Item](#)' steps from the workflow of your automation.

This is custom documentation. For more information, please visit the [SAP Help Portal](#)

i Note

If you check your recording in the Cloud Studio, you can view the recorded context menu button activity which is displayed in two distinct steps: '[Press Context Button](#)' and '[Select Context Menu Item](#)'.

2. Declare the Context Menu button as a new element in the application tab.
3. Go back to the automation tab and choose the declared Context Menu button in the 'Define Screen Activity' window.
4. Drag and drop either '[Select Toolbar MenuItem by Text](#)' or '[Select Toolbar MenuItem by Position](#)'.
5. Enter the value in the input parameter field. If the same value is available in the 'Context Menu button' drop-down list, the click operation will be performed.
6. Save and run the automation.

The following video shows you the alternative method for choosing the context menu element after clicking the Context Menu button in the GUI toolbar.

SAP Intelligent RPA - Handling Context Menu Button Activity



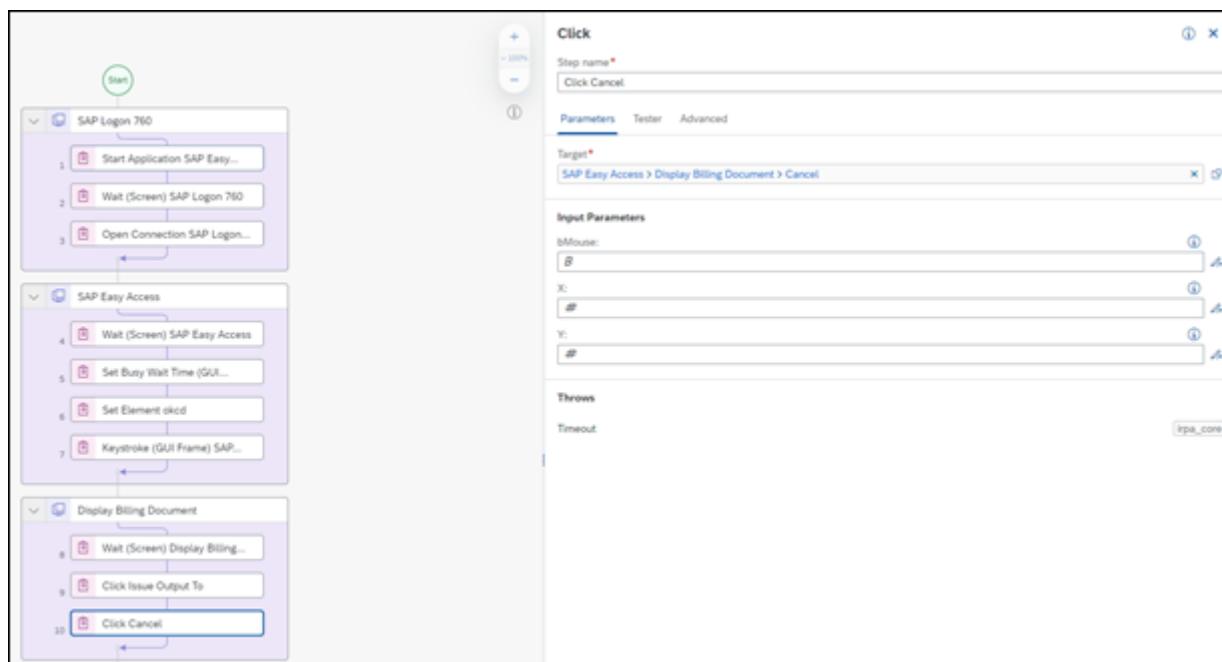
Maintain Menu Items Automation through Tree View

The Recorder can record and generate artifacts that are required to automate menu items in SAP GUI for Windows (also called SAP WinGUI). It encapsulates the steps involved in automating menu items. The automation of menu items can be operated on the main window page itself. It is not required to recognize the menu item dropdown as it appears only after clicking the [More](#) button.

Menu items are available in the [Tree view](#) with the type [GuiMenu](#). It is recommended not to change the actual name of the declared item in recording artifacts so that it can be easily located in the menu items for maintenance purposes. Later, if you want to edit the bot with a new menu item, you can access these menu items through the tree-view and declare it as shown in the following screenshot.

The screenshot shows the SAP GUI Recorder interface. On the left, there is a tree view of the captured application structure under <Display Billing Document>. A specific menu item, 'GuIMenu - Display from archive', is selected and highlighted with a green dot. On the right, the 'Element Details' panel is open, providing information about the selected element. The 'General Information' section includes fields for Name (Display from archive), Identifier (displayFromArchive), and Element Class (GUI Menu). The 'Recognition Criteria' section shows a single criterion: 'Name equals 'Display from ar...''. The 'Captured Data' section displays coordinates and other details: CX: -1, CY: -1, Depth: 3, Id: mbarimenu@ymenu[3], Name: Display from archive, Text: Display from archive, Type: GuIMenu, X: -1, Y: -1.

Then, you must add the **Click** activity in a workflow of automation and add a target for the declared menu item as shown in the following screenshot.



Best Practices

The following is the best practice we recommend in managing recording of workflow using the Recorder.

- Until capture of a page is completed, do not perform any action on SAP GUI page that causes a round trip. If any round-trip action is performed on a page before the capture is completed, the capture fails and the actions are associated with

the previously captured page.

Undo actions associated with previous page and recapture the page.

- While Recording, do not initiate a screen capture when the Session is busy. The screen capture action might not be successful.
- Cloud Studio Timeout

While the recording is in progress, the Cloud Studio may timeout for various reasons. So, before you start exporting the recording to Cloud Studio, check if the Cloud Studio has not timed out. If it is timed out, you must re-login (do not refresh) to the Cloud Studio and then export the recording.

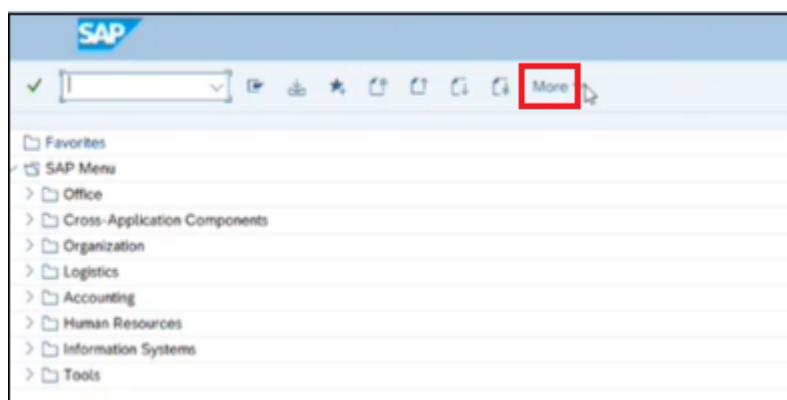
- It is recommended to start recording from the SAP Easy Access page as it adds open connection activity which is more robust than automating the SAP Logon page using the UI Automation.
- **Open Connection** activity requires at least one SAP WinGUI page to be declared in the applications.
- You can use the Recorder to automate the menu items in SAP WinGUI. To automate menu items, it is recommended to perform an action on menu items when the last captured page is the main window page. The automation of menu items can be operated on the main window page itself. It is not required to capture Menu dropdown as it appears only after clicking the **More** button.

Limitations

- While capturing a page, if the SAP scripting does not respond within 7 seconds, a timeout error is generated internally and "wnd[0]" is displayed as title in the recording widget page.

If such captures are seen in the widget, undo the capture and recapture the page.

- The recorder might generate steps named **Custom Action (SAP GUI)** for a few actions during the SAP WinGUI recording. This happens because there is no dedicated activity for this specific action. However, automation will still work in an intended way.
- While recording, if a native Microsoft Windows dialog box is opened/closed, the capture hint isn't displayed.
- Click on the **More** button isn't supported in SAP WinGUI automation, and it is not required in a workflow of automation.



Record UI Automation

UI Automation Recorder is used for recording desktop applications. You can automate complex workflows easily using the UI Automation Recorder.

The UI Automation Recorder records the steps you perform across the screens of an application. Screens, elements, and their underlying metadata are automatically captured and properly declared. Then you can export the recording in the automation

designer of the Cloud Studio where a workflow is built and activities and parameters are set, once again automatically. If required, you can manually edit the captured screens and elements or edit the automation to update the Step Details.

The way you use the recorder for UI Automation is similar to other technologies, like SAP GUI for HTML (also known as WebGUI) or SAP GUI for Windows (also known as WinGUI) for instance. For more information about the general use of the recorder, refer to the [Automatically Capture Applications with the Recorder](#) section.

i Note

Before using UI Automation recorder, it is recommended to refer the UI Automation recorder [Best Practices](#), [Limitations](#), and [More Information](#) sections.

Related Topics

- [Overview](#)
- [Prerequisites](#)
- [Automatically Capture Applications with the Recorder](#)
- [Screen Capture Modes of the Recorder](#)
- [Enable Logs for Recorder in TraceViewer](#)
- [Record SAP GUI for Windows Application](#)
- [Record Web Applications](#)
- [Error Management](#)
- [Best Practices](#)
- [Read Value of an Element During Recording](#)
- [Set Value of an Element During Recording](#)

Best Practices

The following is the best practice we recommend in managing the recording of a workflow using the UI Automation Recorder.

- Whenever the DOM is changed in the provider application, you must capture it by clicking the **Capture** button in the recording widget.

For Example,

Opening of a pop-up, expansion of a tree, selection of a tab, and opening a new window.

If you missed a capture during the recording, you must fix it after the export:

- Add the missing captures.
- Fix element declarations. Indeed when an element is not part of the captured DOM, the declaration points by default to the last element of the DOM. You must correct this by redeclaring the element.
- Fix the generated automation: add the **Refresh Screen** and **Wait Element** activities before every captured screen in the automation workflow.
- The **Wait** activity is required between consecutive clicks

- The click activity performs click asynchronously. If multiple consecutive asynchronous click activities are used, the sequence of the clicks performed on the target application cannot be guaranteed. So, the **Wait** activity must be used between consecutive click activities.
- A new synchronous **Click** activity has been added and is used by the recorder, so that the recorded automation does not have the issue associated with multiple consecutive asynchronous clicks.

For more information, see the [Best Practices for Wait Activities](#) section.

- Capture hint while performing manual capture
 - If an action on the first screen (which was already captured) leads to the second screen and the very first action on the second screen (which was not captured) leads to the third screen, then the user will be on the third screen when the capture hint is displayed. In this case, if the user captures the screen on seeing the hint, then the third screen will be captured and the second screen will not be captured. The action performed on the second screen also will not be recorded. So, in such cases, user should pause the recording, go back to the second screen, resume the recording, capture the screen, and then continue the recording.
 - If an action on the first screen (which was already captured) modifies the DOM such that some part of the screen is unchanged and then an action is performed on the part of the screen which did not change, then the capture hint will not be displayed. In this case, user can either capture on seeing a change in the DOM or add refresh / wait activities in the exported automation as required to ensure that the final automation is robust.
- Effect of screen resolutions on the UI Automation Recorder

The UI Automation Recorder works best with 100% DPI (dots per inch) resolution. It also works fine with other resolution scales (for example, 150% DPI) for most Windows applications. However, in a few applications, the UI Automation framework gives the wrong element based on the mouse pointer, and this might interrupt the bot execution. Therefore, it's recommended to use the recorder for UI Automation with a 100% DPI resolution.

Limitations

- If context menu item/menu items are not part of the DOM, such items cannot be captured with **Capture** in the recording widget.
 - This is due to the fact that these pop-ups would disappear when **Capture** is clicked in the recording widget. Hence, they may not be available in the DOM.
 - If context menu/menu items have shortcut keys, those can be recorded using the **Keystroke** activity. Alternatively, export to Cloud Studio and capture menu items using **Ctrl + Hover** option.
- No support to record right-click:
 - Typical cases of right-clicks are used to open a context menu. As capturing of a context menu and right-clicks are also not supported.
- No support for mouse wheel and mouse drag and drop:
 - Scrolling can be performed using the mouse wheel. Scrollbar is not supported. Hence, mouse wheel is not supported.
- No support for recording on controls:

[Document](#), [Scrollbar](#), [Splitbutton](#) and [Slider](#)

- These are the controls that the current SDK supports. Recording these controls is not supported.
- However, recording widget/Cloud Studio displays the control that UI Automation framework provides and any activity performed is mapped to the **Click Sync** activity.

- Compound **Keystroke**:

- Recording compound key strokes is currently supported at screen level but not at an element level. If such key stroke is required for an element, the activity must be added after exporting the content to the Cloud Studio.

- Capture hint

To verify if the target element on which the action was performed is present in the last captured screen, runtime ID provided by UI Automation framework is used as an identifier. So, if the framework provides same runtime IDs for different elements, then the capture hint may not be displayed. This behaviour is observed in certain cases in Outlook.

More Information

- Password fields:

Passwords fields will be recorded, however they would be displayed as "****" in the recording widget. After export, you must update this value.

- By default, the recorder captures the window in focus. If there are multiple windows, you must bring the window to be captured in focus and then capture.

- Support for Additional Controls:

The UI Automation Recorder records the activities on some additional controls such as Document, Scrollbar, Slider, and Split Button.

The following activities can be recorded using the UI Automation Recorder:

Controls	Activities Supported
Document	Set Element
Scrollbar	Click / Double Click
Slider	Click / Double Click
Split Button	Click / Double Click

- In UI Automation, capture hint behavior is different from other technologies due to the nature of the UI Automation technology.
 - When a window / pop-up is opened / closed, the capture hint would be displayed. After the hint message is displayed, the user can capture and perform the actions.
 - For scenarios such as tab selection and expansion of tree that caused DOM change, capture hint is not displayed immediately after the DOM change. It is displayed when an action is performed on the new DOM content. This action is not recorded in the recording widget before the page is captured. After page capture, the activity is restored automatically in the recording widget.
 - If the user ignores the capture hint and performs multiple actions, only the action that led to the capture hint will be restored and displayed in the recording widget after the user captures the screen and all other actions that were performed after the capture hint was displayed will be ignored.

i Note

Scroll activity (for example, Mouse Wheel Scroll, Drag Thumb) is not supported on the scrollbar and slider controls.

Record Web Applications

You can now use the Recorder to record non-SAP Web pages, SAPUI5, SAP GUI for HTML (also known as WebGUI), SAP SuccessFactors, and SAP Ariba screens in the same recording session. The recorder can intelligently detect the change in nature of the screens and then generate activities accordingly.

Prerequisites

To record an SAP WebGUI application, check the following SAP Note to get required access to the authorization object "S_WEBGUI": [2853198](#).

Desktop Agent and Extension Version

To record non-SAP Web pages, SAPUI5 and SAP WebGUI screens in the same recording session:

- Desktop agent version must be equal to or greater than 2.0.25
- Chrome or Edge extension version must be equal to or greater than 2.0.25

For more information on how to use the Recorder, see [Automatically Capture Applications with the Recorder](#).

Related Topics

- [Overview](#)
- [Prerequisites](#)
- [Automatically Capture Applications with the Recorder](#)
- [Screen Capture Modes of the Recorder](#)
- [Enable Logs for Recorder in TraceViewer](#)
- [Record SAP GUI for Windows Application](#)
- [Record UI Automation](#)
- [Error Management](#)
- [Best Practices](#)
- [Read Value of an Element During Recording](#)
- [Set Value of an Element During Recording](#)

Record SAP GUI for HTML Application

You can use the Recorder to easily automate complex workflows within SAP GUI for HTML (also known as WebGUI) system.

i Note

SAPWebGui is not a technology anymore. It is a Framework under **Web** technology. The newly added screens will have **Web** as Technology and **SAPWebGui** as Framework.

Framework:	SAPWebGui	
Technology:	Web	

- However, older bots recorded with **SAPWebGui** as Technology will continue to work.
- You can still add new **SAPWebGui** screens to the old Application declaration of **SAPWebGui** Technology and use it in the automation.

Related Topics

- [Overview](#)
- [Prerequisites](#)
- [Automatically Capture Applications with the Recorder](#)
- [Screen Capture Modes of the Recorder](#)
- [Enable Logs for Recorder in TraceViewer](#)
- [Record SAP GUI for Windows Application](#)
- [Record UI Automation](#)
- [Record Web Applications](#)
- [Error Management](#)
- [Best Practices](#)
- [Read Value of an Element During Recording](#)
- [Set Value of an Element During Recording](#)

Record Embedded SAP GUI for HTML Applications

You can record SAP GUI for HTML (also known as WebGUI) applications embedded inside S/4HANA iframe or frame.

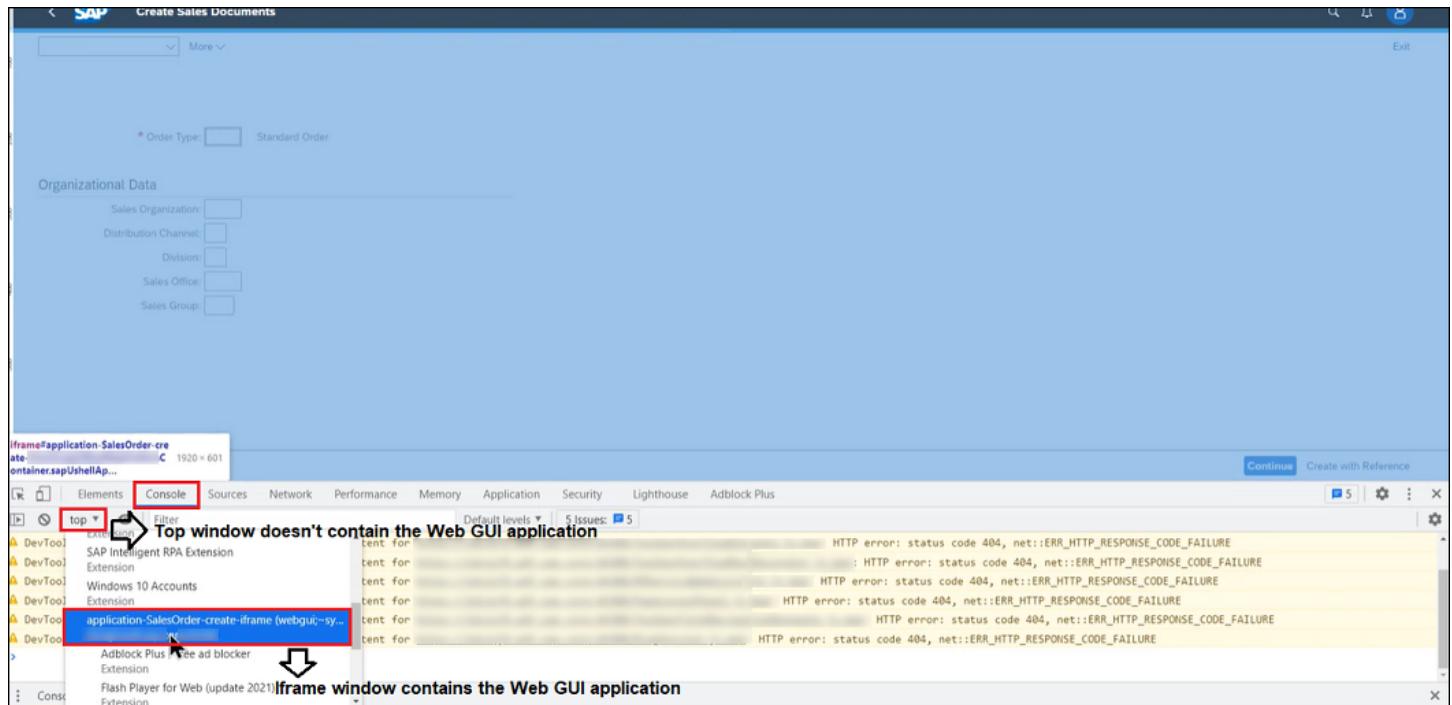
Currently, this feature is supported in all SAP WebGUI applications embedded inside S/4HANA on-premise and cloud systems.

i Note

An iframe is a mechanism to display a web application or page within a web application or page.

A web page contains an iframe if:

- you are not able to find the requested web element using the selectors or Xpaths.
- you see an option other than **top** in the Execution Context Selector window after opening the browser's developer tool by clicking the  icon > More Tools > Developer tools or holding down the **[Ctrl]** + **[Shift]** + **[I]** on the login screen.



Prerequisites

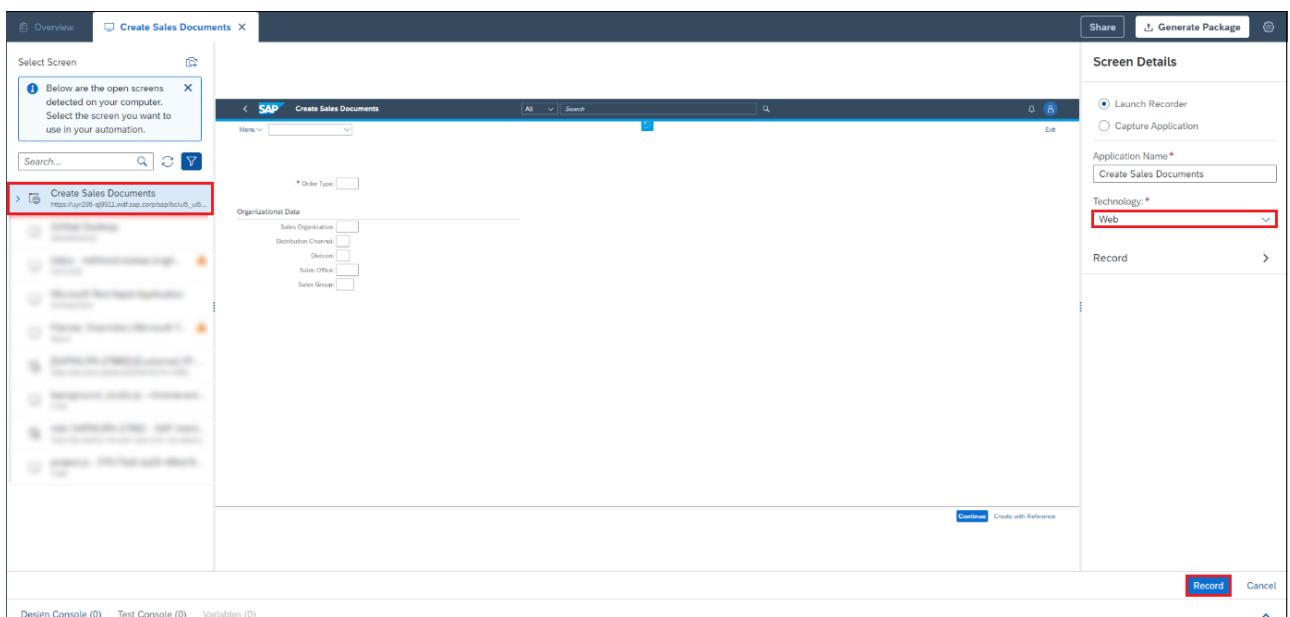
Check the following SAP Note to get required access to the authorization object "S_WEBGUI": [2853198](#)

Procedure

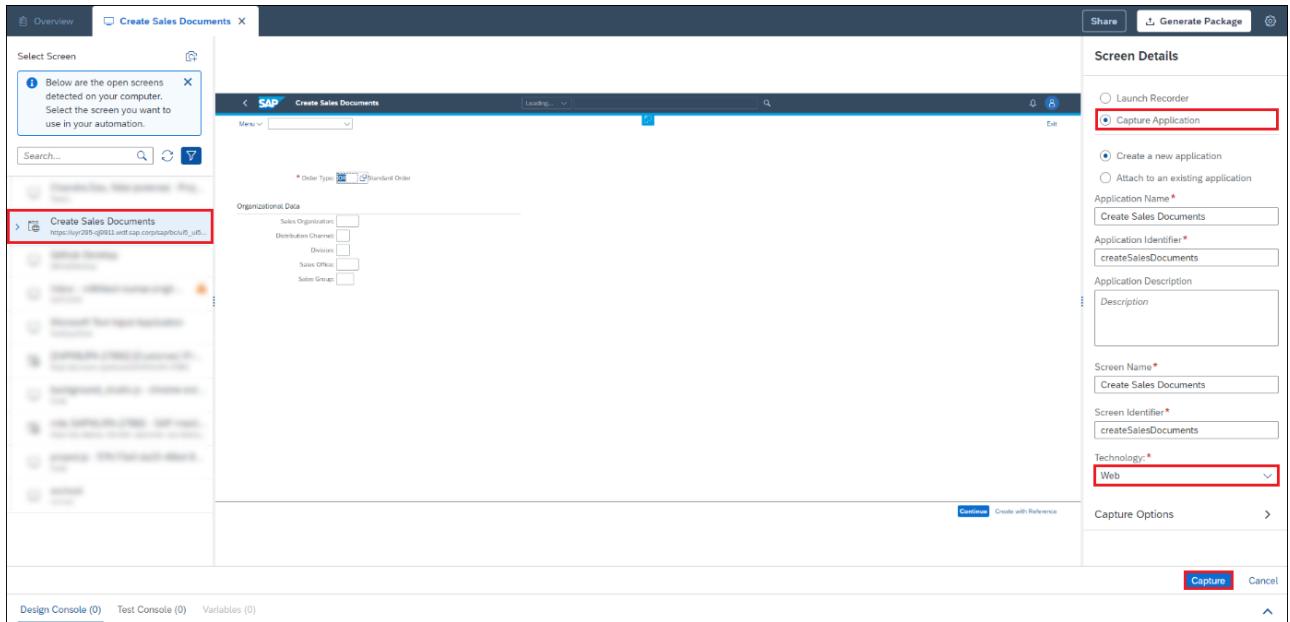
1. In the Project Explorer, click and then click **Create** in the left panel or click **Create** in the Project Explorer main panel. The artifact menu is displayed.
2. In the artifact menu click **Application**. A new tab labelled **Untitled** opens. Later, this label will change to the name of the captured application. The system starts detecting the applications and their screens currently running on your local machine. When it's done, you'll see a list of screens in the picker panel on the left.
3. Choose the application you want to capture from the list of screens. For more information about capturing an application, see [Capture an Application](#).

If you select the top window or browser application, the following options will be enabled:

- o **Record:** By default, the **Record** option will be enabled, and the **Technology** will be selected as **Web**. This option supports the recording of SAP WebGUI applications embedded inside S/4HANA iframes.

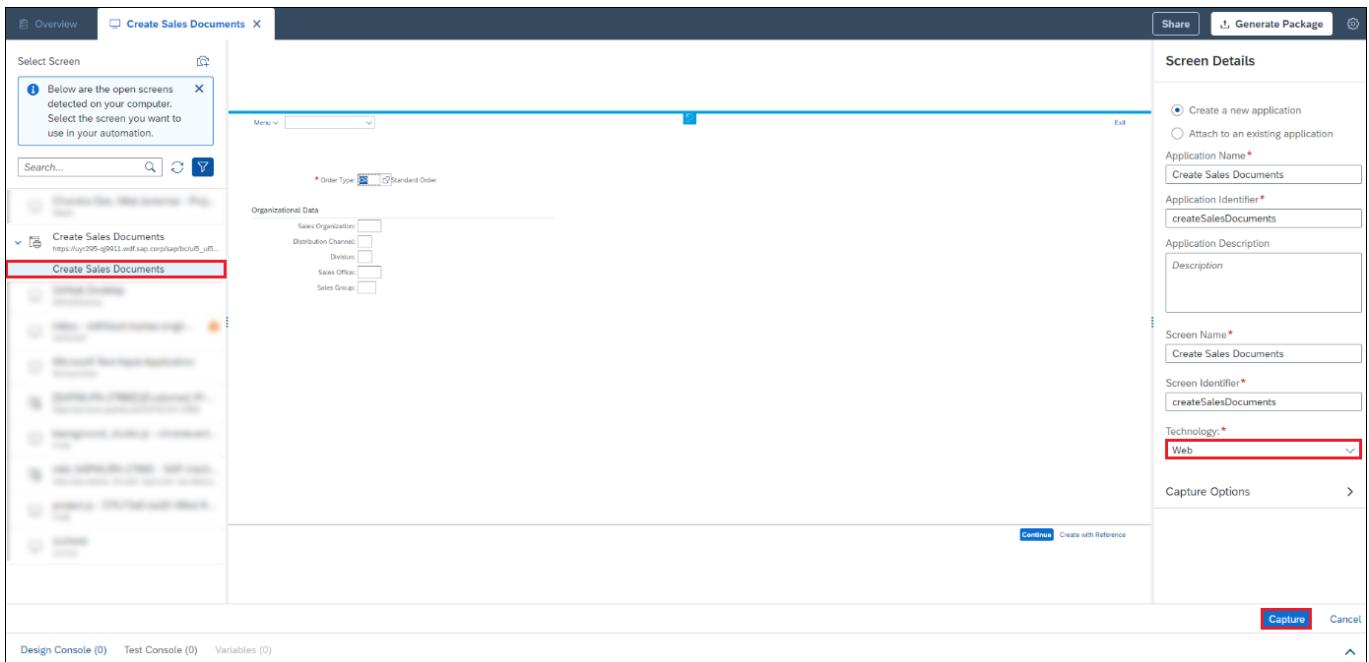


- o **Capture:** If you want to do a manual capture, select **Capture Application**. The **Capture** option will be enabled, and the **Technology** will be selected as **Web**.



4. Select the iframe window or SAP WebGUI application by expanding the listed result of your application. The following option will be enabled:

Capture – By default, the **Capture** option will be enabled, and the **Technology** will be selected as **Web**.



The newly added screens will have **Web** as Technology and **SAPWebGui** as Framework.

Framework:
SAPWebGui
Technology: Web

5. Click **Record**. For more information on how to use the Recorder for an SAP WebGUI application, see [Automatically Capture Applications with the Recorder](#).

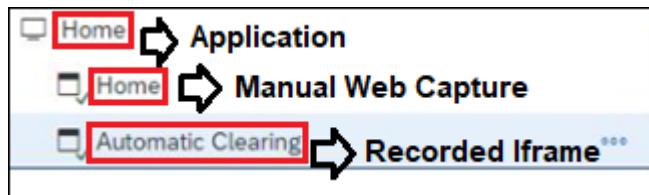
View Recorded Screens

You can view the recorded screens of the SAP WebGUI application under **Declared Application**. If you select a recorded iframe screen, the **MAINFRAME** will be displayed as **false** in the application criteria.

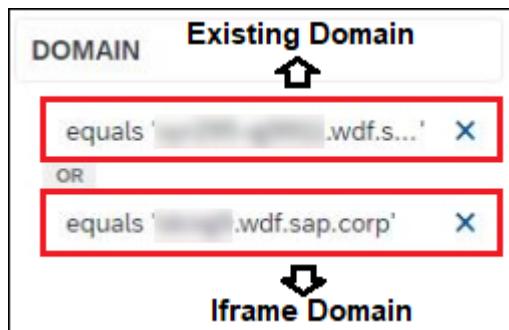
The screenshot shows the SAP Application Recorder's 'Create Sales Documents' screen. In the 'Recognition Criteria' panel, there is a condition 'MAINFRAME is false' which is highlighted with a red box and has a cursor pointing at it.

i Note

If you want to record an iframe inside an already captured web application (such as SAPUI5 application), you must manually add the iframe domain in the application criteria.



The iframe domain must be added as an **OR** condition with the existing domain's condition.



Best Practices

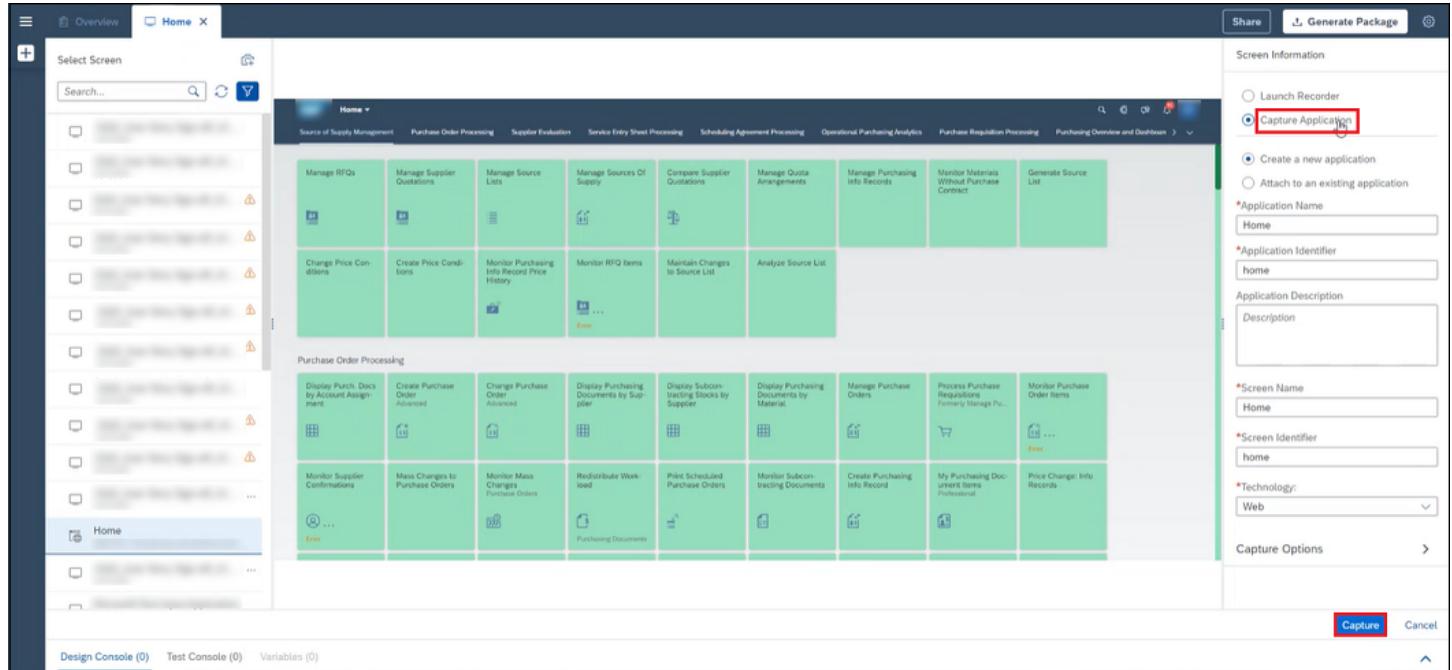
It is recommended not to add URL as a criterion for a screen. However, you can use **TITLE** and **MAINFRAME** together as criteria. By default, **TITLE** and **MAINFRAME** are added as criteria to the screens when you record an application.

Record SAPUI5 Application

You can use the Recorder to easily automate complex workflows within SAPUI5 system. For more information on how to use the Recorder for an SAPUI5 application, see [Automatically Capture Applications with the Recorder](#).

The recording feature is available since 2111 release. However, you can still capture UI5 screens manually.

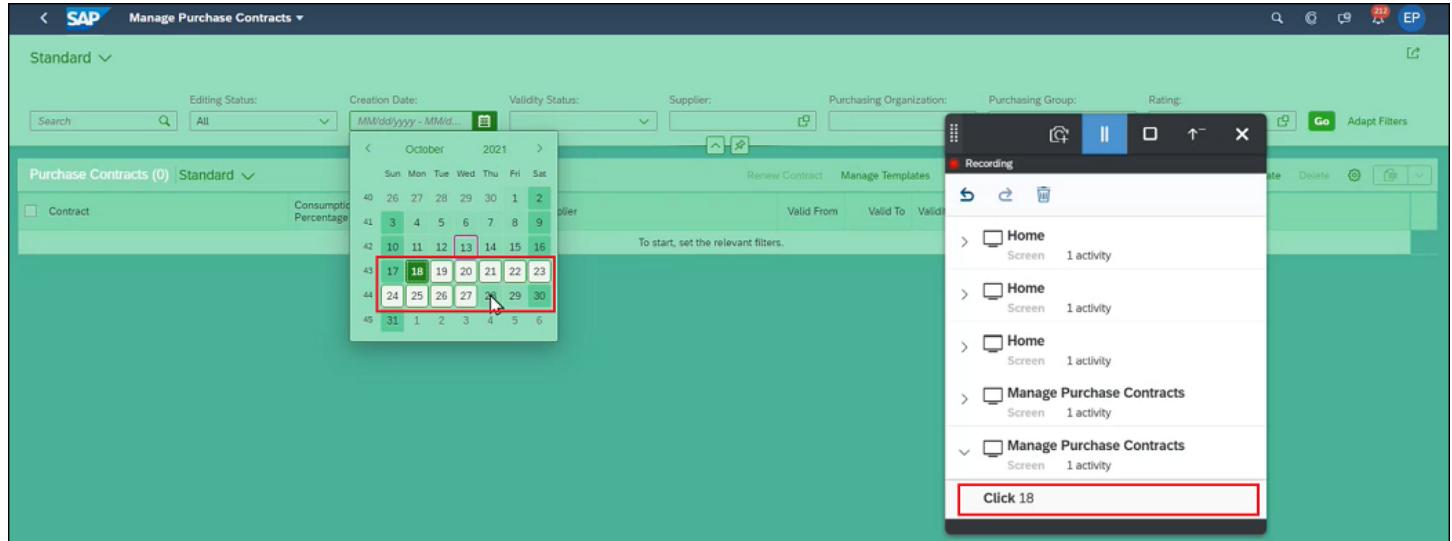
This is custom documentation. For more information, please visit the [SAP Help Portal](#)



More Information

Select Date Using Date Picker

While recording, you can open date picker and then select a single or a range of dates as required. All the actions will be recorded as a click activity.



However, it is still recommended to set the date directly in the input field as you can easily change the date if required in the future.

If you select the date using date picker, there will be a dependency on the UI elements. In case you want to change the date in the future, then you must declare the UI elements from scratch.

Criteria Optimization

Criteria are optimized for elements where the parent is added to the criteria. The parent that helps in recognition is added to the criteria. It helps automation to run faster and makes element declaration more readable.

Related Topics

- [Overview](#)
- [Prerequisites](#)
- [Automatically Capture Applications with the Recorder](#)
- [Screen Capture Modes of the Recorder](#)
- [Enable Logs for Recorder in TraceViewer](#)
- [Record SAP GUI for Windows Application](#)
- [Record UI Automation](#)
- [Record Web Applications](#)
- [Error Management](#)
- [Best Practices](#)
- [Read Value of an Element During Recording](#)
- [Set Value of an Element During Recording](#)

Best Practices

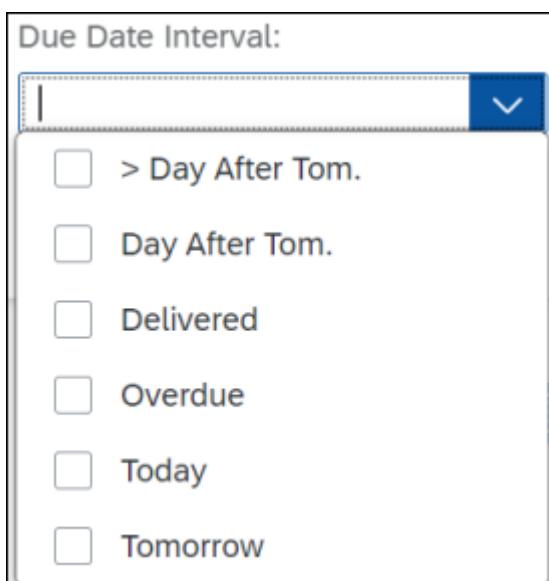
The following best practices are recommended to record a workflow using the SAPUI5 recorder:

- You must start recording on a refreshed screen.
- **Capture Screen**

You must capture a screen by choosing the camera () button manually if:

- new controls are loaded into the screen once the previous capture is completed. You must capture the screen before interacting with the newly appeared controls.
- there is a scroll to a specific section on the screen. Instead of using the scroll bar and capturing the screen, we recommend that you use the navigation bar to scroll to the relevant section on the screen when necessary.
- there is a drop-down list or a pop-up message on the screen. We recommend that you set the value into all types of input boxes (such as ComboBox or Multi-ComboBox) from the keyboard instead of selecting from the list or pop-up message and capturing the screen.

In the following example, you must capture the screen before selecting a value from the drop-down list.



- o a new table row is created on the screen before entering values into the input boxes of the table row.

In the following example, you must capture the screen after a new table row is created.

Item	Category	Account Assignment Category	Material	Target Quantity	Net Order Price	Price Unit
10	Standard (0)			0,000		0,00

- o the table results are updated after choosing **Go**.
- o the input box state changes between valid and error state after adding values to the input boxes of the table row.

- **Input Suggestion/ Date control**

While recording, you must set the input values by using the keyboard instead of choosing the suggestions that come from the input boxes.

- **Input Boxes**

You must choose the **Tab** or **Enter** key to record actions after entering the values into the input boxes.

i Note

- o If you want to set an empty value to an input box and record action, you must make changes in the recorded artifacts.
- o In case of Fiori Smart filter bar input controls, the **Enter** key is not supported. Therefore, choose the **Tab** key to record action.

- **Collection**

Before implementing collection on the recorded elements, it is important to first check the criteria. If the criteria includes the parents, the topmost parent must be set as the collection instead of the base parent.

i Note

If any of the input boxes within a collection of declared input fields change to a different control after the input value is entered, the loop does not function correctly due to the dynamic nature of the UI5 application.

- **Criteria**

If there is any discrepancy in the criteria that appears in orange or red mark, you can select the criteria manually.

- **Handle Fiori Login Page**

Using the **UI5 Login** activity, you can log into UI5 applications. You can also capture login screen manually along with the recorder.

For more information, see the [UI5 Login](#) activity section.

Record SuccessFactors Application

You can use the Recorder to easily automate workflows of the SAP SuccessFactors application.

For more information on how to use the Recorder, see [Automatically Capture Applications with the Recorder](#).

Related Topics

- [Overview](#)
- [Prerequisites](#)
- [Automatically Capture Applications with the Recorder](#)
- [Screen Capture Modes of the Recorder](#)
- [Enable Logs for Recorder in TraceViewer](#)
- [Record SAP GUI for Windows Application](#)
- [Record UI Automation](#)
- [Record Web Applications](#)
- [Error Management](#)
- [Best Practices](#)
- [Read Value of an Element During Recording](#)
- [Set Value of an Element During Recording](#)

Best Practices

The following best practices are recommended to record the SAP SuccessFactors application.

- While recording a SuccessFactors application, if you come across any input box that has autocomplete drop-down list, it is recommended to type the complete value in the input box and then click on the list item.
- After exporting the recorded artifacts, it is recommended to check the recognition criteria of the declared elements in all the screens of the application before executing the automation. If an element has conflicting or invalid criteria, it must be fixed for the successful execution of the automation.

Record Ariba Application

You can use the Recorder to easily automate workflows of the SAP Ariba application.

For more information on how to use the Recorder, see [Automatically Capture Applications with the Recorder](#).

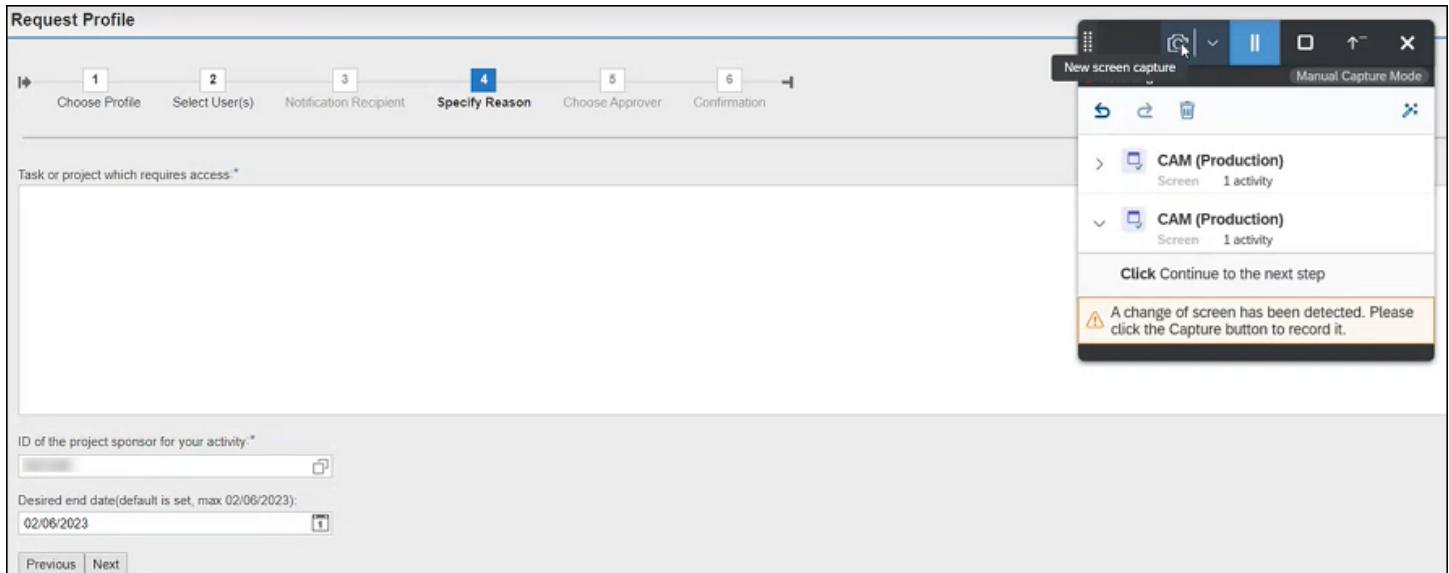
Related Topics

- [Overview](#)
- [Prerequisites](#)
- [Automatically Capture Applications with the Recorder](#)

- [Screen Capture Modes of the Recorder](#)
- [Enable Logs for Recorder in TraceViewer](#)
- [Record SAP GUI for Windows Application](#)
- [Record UI Automation](#)
- [Record Web Applications](#)
- [Error Management](#)
- [Best Practices](#)
- [Read Value of an Element During Recording](#)
- [Set Value of an Element During Recording](#)

Record Web Dynpro Application

You can use the recorder to easily automate workflows of the Web Dynpro application that is part of an iFrame or a standalone Web Dynpro.



However, you can also capture the Web Dynpro applications manually. The Web Dynpro application is supported using the technology as "WEB", and the recorded screen is detected as framework "WEB".

For more information on how to use the recorder, see [Automatically Capture Applications with the Recorder](#).

Related Topics

- [Overview](#)
- [Prerequisites](#)
- [Automatically Capture Applications with the Recorder](#)
- [Screen Capture Modes of the Recorder](#)
- [Enable Logs for Recorder in TraceViewer](#)
- [Record SAP GUI for Windows Application](#)

- [Record UI Automation](#)
- [Record Web Applications](#)
- [Error Management](#)
- [Best Practices](#)
- [Read Value of an Element During Recording](#)
- [Set Value of an Element During Recording](#)

Record UI Automation Pages from the Web Recorder

Use the recorder to record Web and Windows (UI Automation technology) pages seamlessly. The recorder generates two separate applications for each technology while exporting the recording in the Cloud Studio.

i Note

- The recording must start from the Web page only.
- Whenever the Windows dialog or popup (UI Automation page) is encountered, new page capture is required even though there is no capture hint on the recording widget.
- Also, capture the page once the interaction with Windows dialog or popup is completed and it gets closed.
- The recorder can also record the file browsers.

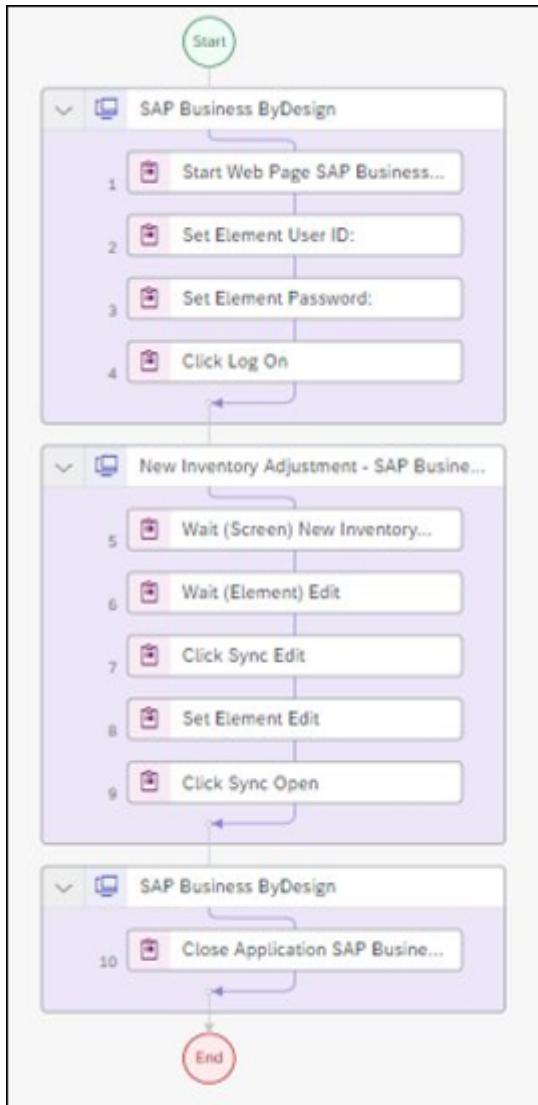
In the following screenshots, you can see that the generated artifacts contain two separate applications (one for WEB technology and another for UI Automation technology) and one automation.

The screenshot shows the SAP Launchpad interface. In the center, a 'New Inventory Adjustment' screen is displayed. A file selection dialog box is open, showing a list of files in the 'Downloads' folder. The dialog includes buttons for 'Open' and 'Cancel'. To the right of the screen, a 'Screen Details' panel is open, containing sections for 'General Information', 'Recognition Criteria', and 'Captured Data'. The 'General Information' section shows the name as 'New Inventory Adjustment - SAP Bus...' and the identifier as 's_newInventoryAdjustmentSAPB'. The 'Recognition Criteria' section includes a condition 'Name equals 'New Inventory ...'' and a 'Deepness ≤ 0' condition. The 'Captured Data' section shows coordinates CX: 1550 and CY: 934.

The screenshot shows the 'Artifacts' list in SAP Launchpad. It displays three entries: 'SAP Business ByDesign', 'SAP Business ByDesign Automation', and 'SAP Business ByDesign_1'. Each entry includes columns for Name, Description, Type, Last edited, Last edited by, and Created On. The 'Created On' column shows dates from September 1, 2022.

Name	Description	Type	Last edited	Last edited by	Created On
SAP Business ByDesign	SAP Business ByDesign	Application	2 minutes ago	[redacted]	September 1, 2022
SAP Business ByDesign Automation	No value.	Automation	2 minutes ago	[redacted]	September 1, 2022
SAP Business ByDesign_1	SAP Business ByDesign	Application	2 minutes ago	[redacted]	September 1, 2022

The generated automation contains steps for both applications in sequence.



Related Topics

- [Overview](#)
- [Prerequisites](#)
- [Automatically Capture Applications with the Recorder](#)
- [Screen Capture Modes of the Recorder](#)
- [Enable Logs for Recorder in TraceViewer](#)
- [Record SAP GUI for Windows Application](#)
- [Record UI Automation](#)
- [Record Web Applications](#)
- [Error Management](#)
- [Best Practices](#)
- [Read Value of an Element During Recording](#)
- [Set Value of an Element During Recording](#)

Best Practices

The following are the best practices we recommend to record web applications.

- Always start recording on a refreshed page.
- When a scenario is recorded in one domain and executed in another domain, the following requirements must be met:
 - The application criteria must be updated with the new domain using an OR condition.

For example,

The screenshot shows the 'Application Details' screen in Cloud Studio. The 'Data and Criteria' tab is selected. Under 'Recognition Criteria', there is a section titled 'DOMAIN' with two entries separated by an 'OR' operator. Both entries are enclosed in a red box. The first entry is 'equals 'eu1.concursolutions.com'' and the second is 'equals 'testdomaineu1.concursol...''. Below this, under 'Captured Data', the following information is listed: DOMAIN: eu1.concursolutions.com, TITLE: SAP Concur Home, and URL: https://eu1.concursolutions.com/home.asp.

- None of the screens should have domain in their criteria.
- When a screen title is empty, then **mainframe** and **URL** are generated as criteria. If the **URL** is changing dynamically during runtime, use part of the **URL** (using contains operator) or **Domain** to recognize a screen.
- If an error occurs while running the bot, check the declaration and criteria for the application, screen, or elements of the created application.
- If facing any difficulty in selecting the element in a screenshot, use **Both** or **Tree** option in Cloud Studio to select an element.
- Input Boxes:
 - After entering the value in the input box, press the **tab** or **enter** key to get the recorded action.
- Drop-down or pop-ups:
 - Use the keyboard to set the values in all types of input boxes (ComboBox and Multi ComboBox) instead of selecting from a list or pop-up. While selecting the values from the keyboard, capturing a screen is not required.
 - If you select the values from a list or pop-up, then you must capture a new screen before selecting a value from a list or popup.

Capture Screen and Capture Hint

- Ensure that the application is loaded completely before capturing an application screen.
- If a screen capture fails, refresh the browser tab being recorded and try again.
- If you see a capture hint on the recording widget and change in the UI, it is recommended to perform a capture ().

- If you see a change in the UI and there is no capture hint on the recording widget, it is recommended to perform a capture ().
- If you expect a change in the UI, it is recommended to wait for at least a second for the capture hint to appear on the recording widget.
- If a capture hint is ignored and you forget to capture a screen, there is a possibility that some of the actions recorded after the previous action are not generated in the automation. In this scenario, the capture of the application screen containing that element is to be added as a multi-capture. Declare the element and manually add the activity in the automation.

Limitations

- Mouse Hover

If the **SubMenu** closes on clicking of the menu item then you will have to click again to open the **SubMenu**. In this situation, there will be two click actions on the recording widget. You will have to delete one of them manually.

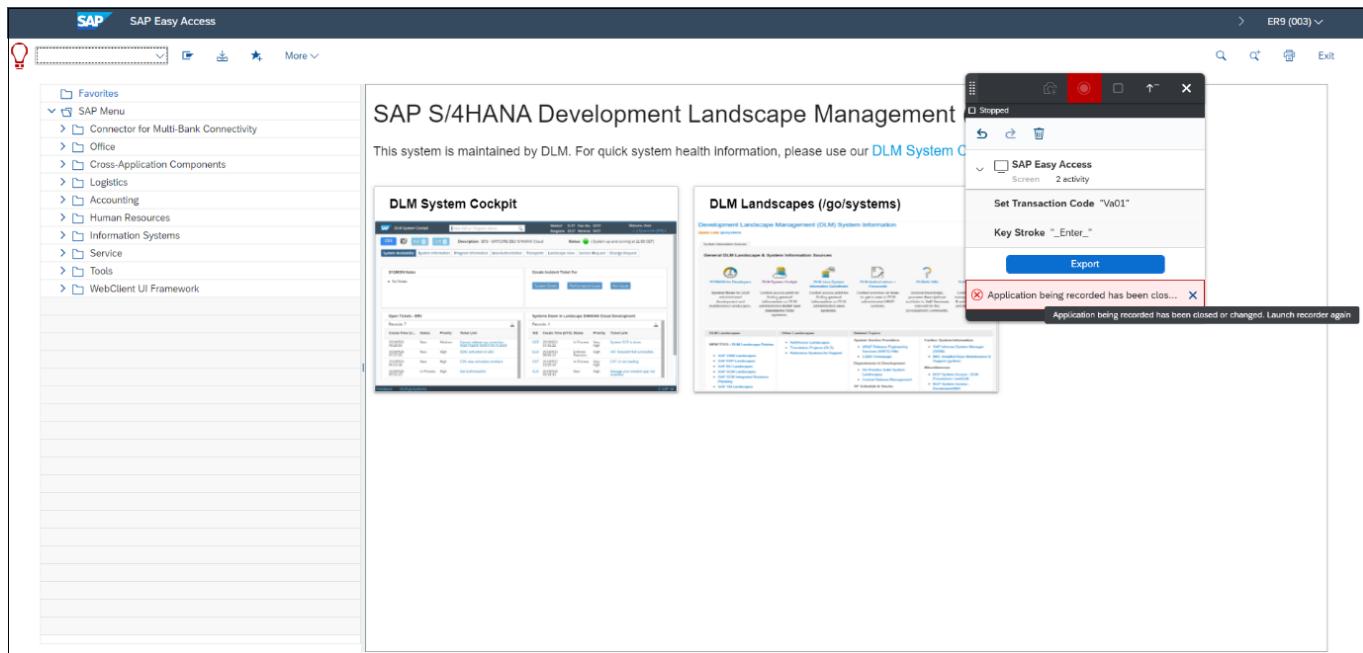
- A headless browser (a browser without a graphical UI) is not supported for a bot execution.

Error Management

This section provides details about errors and recommendations to address the errors that may occur during recording.

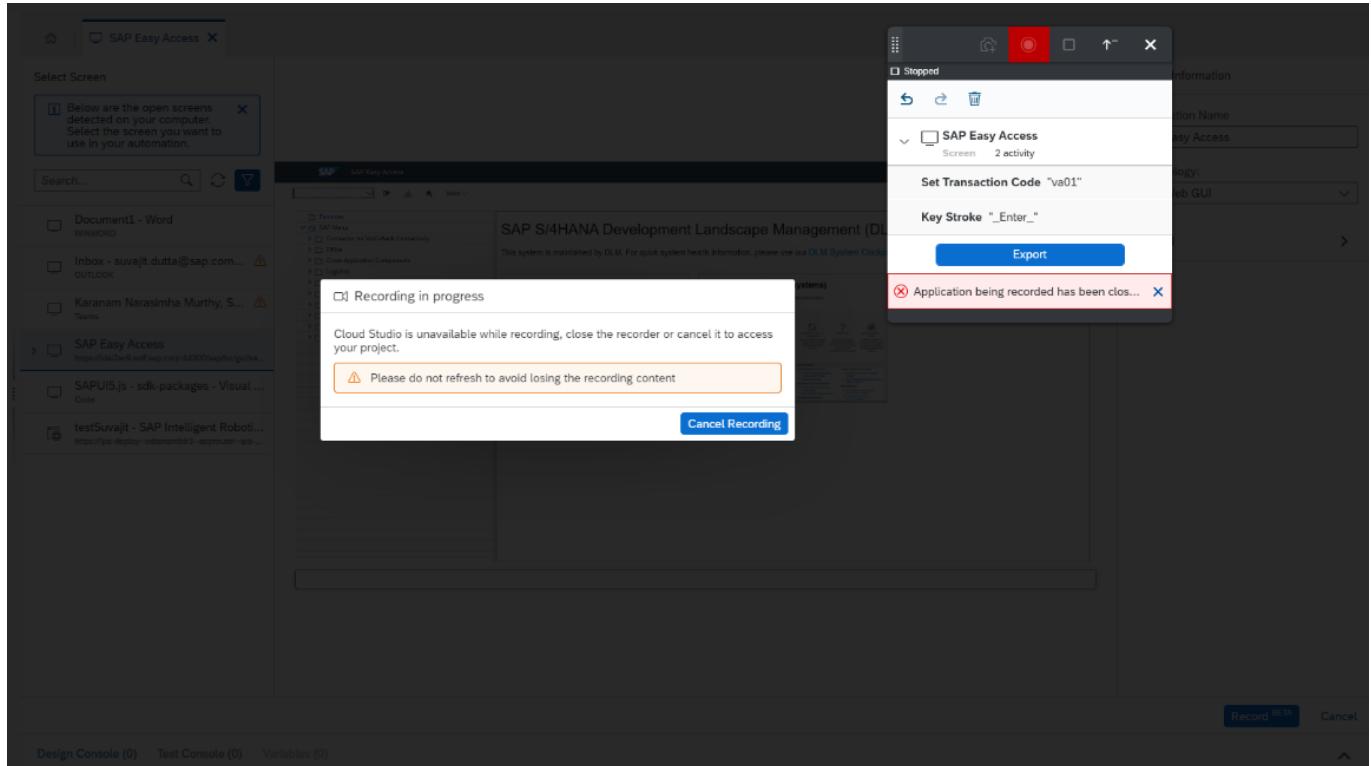
The error messages are same for all technologies. The error message examples shown below are captured using SAP GUI for HTML (also called SAP WebGUI) technology.

- When you launch the recorder and start the recording and if you refresh the SAP WebGUI screen in between the recording, then the Recorder shows an exception message.



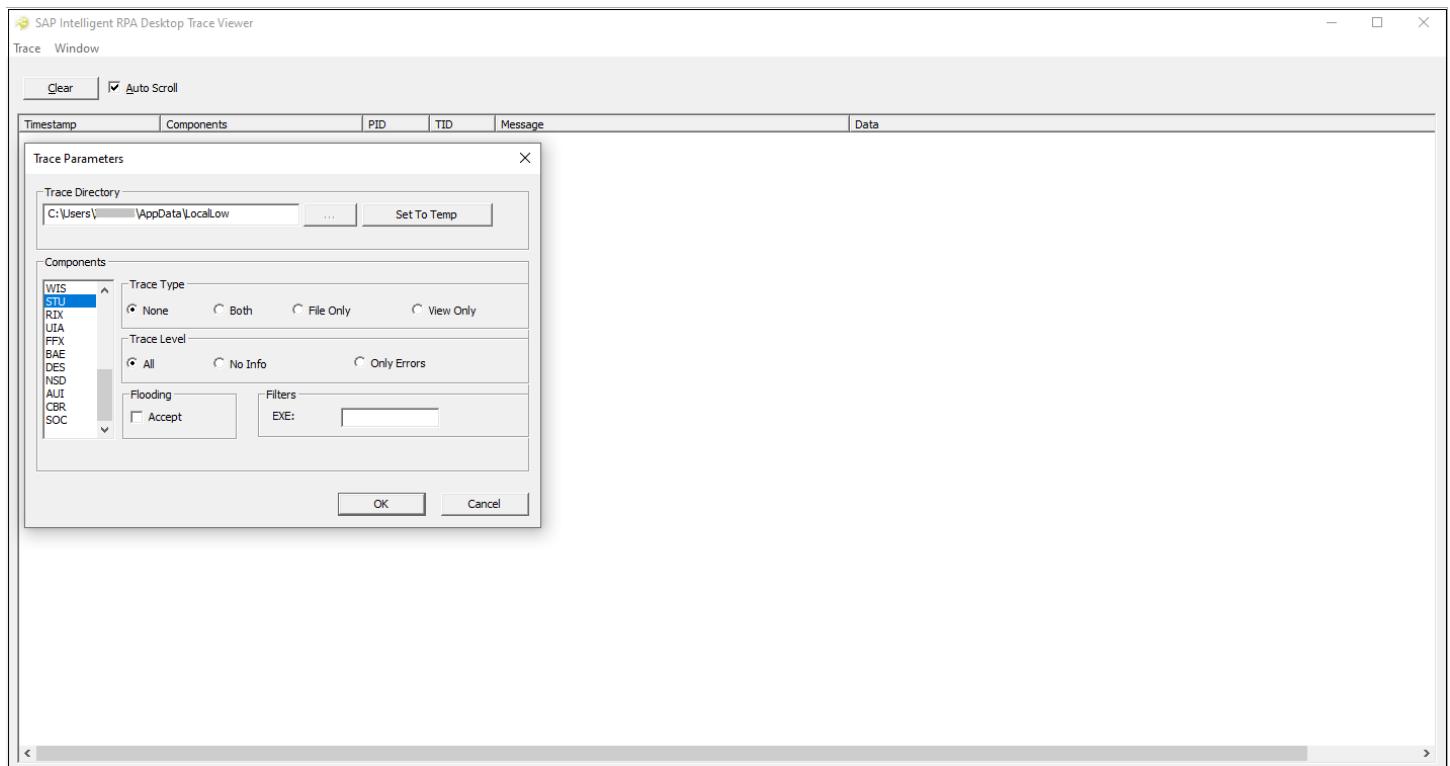
You must click the **X** button to exit the error pop-up.

- When you launch the recorder and start the recording and if you close the SAP WebGUI screen in between the recording, then the Recorder shows an exception message.



You must click the **X** button to exit the error pop-up.

To get more details on the error, you can keep CxTraceViewer.exe by selecting STU option.



SAP Intelligent RPA Desktop Trace Viewer						
Trace		Window				
<input type="button" value="Clear"/>		<input checked="" type="checkbox"/> Auto Scroll				
Timestamp	Components	PID	TID	Message	Data	
(i) 21/02/18 12:48:24:231	StudioAgent.in	0	18420	CxSAPWebGuProvider - [Transmitter = ChromeCapture][Target = stu...		
(i) 21/02/18 12:48:24:233	StudioAgent.in	0	18420	CxSAPWebGuProvider - [Transmitter = ChromeCapture][Title = send...		
(i) 21/02/18 12:48:30:315	StudioAgent.in	0	18420	[Transmitter = studioAgent][Target = chromeBroker][Title = GetPagePr...	JSON = {"id":3,"method": "getScreenPreview","context": "capture","transmitter": "studioAgent","target": "cloudStudio"}	
(i) 21/02/18 12:48:30:500	StudioAgent.in	0	18420	[Transmitter = chromeBroker][Target = studioAgent][Title = LaunchRe...	JSON = {"method": "LaunchRecorder","context": "cloudApplicationId", "target": "cloudApplicationId"}	
(i) 21/02/18 12:48:30:502	StudioAgent.in	0	18420	[Transmitter = studioAgent][Target = chromeBroker][Title = Ack][Cloud]	JSON = {"id":4,"method": "acknowledgement","context": "record","transmitter": "studioAgent","target": "cloudStudio"},	
(i) 21/02/18 12:48:30:524	StudioAgent.in	0	18420	StudioAgent RecorderMessageHelper - [Cloud]		
(i) 21/02/18 12:48:30:533	StudioAgent.in	0	18420	StudioAgent RecorderMessageHelper - [Cloud]		
(i) 21/02/18 12:48:30:534	StudioAgent.in	0	18420	CxSAPWebGuProvider - FocusApplication started. [Cloud]		
(i) 21/02/18 12:48:30:535	StudioAgent.in	0	18420	CxSAPWebGuProvider - [Transmitter = studioAgent][Target = Chrom...		
(i) 21/02/18 12:48:30:536	StudioAgent.in	0	18420	CxSAPWebGuProvider - Timer started - ChromeCapture - RECORD...		
(i) 21/02/18 12:48:30:537	StudioAgent.in	0	18420	CxSAPWebGuProvider - [Transmitter = ChromeCapture][Target = stu...		
(i) 21/02/18 12:48:30:538	StudioAgent.in	0	18420	CxSAPWebGuProvider - Timer stopped - ChromeCapture - RECORD...		
(i) 21/02/18 12:48:30:539	StudioAgent.in	0	18420	CxSAPWebGuProvider - [Transmitter = ChromeCapture][Title = send...		
(i) 21/02/18 12:48:30:540	StudioAgent.in	0	18420	CxSAPWebGuProvider - FocusApplication broker response received...		
(i) 21/02/18 12:48:30:541	StudioAgent.in	0	18420	CxSAPWebGuProvider - FocusApplication finished. [Cloud]		
(i) 21/02/18 12:48:30:542	StudioAgent.in	0	18420	StudioAgent RecorderMessageHelper - LaunchRecorder finished. [C...		
(i) 21/02/18 12:48:30:543	StudioAgent.in	0	18420	record - Recording widget is trying to launch [Cloud]		
(i) 21/02/18 12:48:30:544	StudioAgent.in	0	18420	record - User language is set = en-US [Cloud]		
(i) 21/02/18 12:48:30:545	StudioAgent.in	0	18420	record - Recording widget launched successfully [Cloud]		
(i) 21/02/18 12:48:30:546	StudioAgent.in	0	18420	StudioAgent RecorderMessageHelper - RecorderLaunched started ...		
(i) 21/02/18 12:48:30:547	StudioAgent.in	0	18420	StudioAgent RecorderMessageHelper - RecorderLaunched finished ...		
(i) 21/02/18 12:48:33:527	StudioAgent.in	0	18420	[Transmitter = studioAgent][Target = chromeBroker][Title = Laun...	JSON = {"context": "record","id":4,"method": "startRecording","status": "started","target": "cloudStudio","transmitter": "cloudStudio"}	
(i) 21/02/18 12:48:33:528	StudioAgent.in	0	18420	StudioAgent RecorderMessageHelper - StarRecording started. [Clo...		
(i) 21/02/18 12:48:33:552	StudioAgent.in	0	18420	CxSAPWebGuProvider - StarRecording started. [Cloud]		
(i) 21/02/18 12:48:33:553	StudioAgent.in	0	18420	CxSAPWebGuProvider - [Transmitter = studioAgent][Target = Chrom...		
(i) 21/02/18 12:48:33:554	StudioAgent.in	0	18420	CxSAPWebGuProvider - Timer started - ChromeCapture - RECORD...		
(i) 21/02/18 12:48:33:555	StudioAgent.in	0	18420	StudioAgent RecorderMessageHelper - StarRecording finished. [C...		
(i) 21/02/18 12:48:36:633	StudioAgent.in	0	18420	CxSAPWebGuProvider - [Transmitter = ChromeCapture][Target = stu...		
(i) 21/02/18 12:48:36:635	StudioAgent.in	0	18420	CxSAPWebGuProvider - Timer stopped - ChromeCapture - RECORD...		
(i) 21/02/18 12:48:36:640	StudioAgent.in	0	18420	CxSAPWebGuProvider - [Transmitter = ChromeCapture][Title = send...		
(i) 21/02/18 12:48:36:645	StudioAgent.in	0	18420	CxSAPWebGuProvider - StarRecording broker response received ...		
(i) 21/02/18 12:48:36:648	StudioAgent.in	0	18420	CxSAPWebGuProvider - StarRecording finished. [Cloud]		
(i) 21/02/18 12:48:36:660	StudioAgent.in	0	18420	CxSAPWebGuProvider - START_RECORDING - REC_EXTENSION_3...		
(i) 21/02/18 12:48:36:665	StudioAgent.in	0	18420	StudioAgent RecorderMessageHelper - StartRecordingCallback start...		
(i) 21/02/18 12:48:36:669	StudioAgent.in	0	18420	record - ERROR - Request denied. [Cloud] -> System.Exception: R...		
(i) 21/02/18 12:48:36:683	StudioAgent.in	0	18420	record - ERROR - StudioAgent RecorderMessageHelper - StartRec...	JSON = REC_EXTENSION_3003	

Best Practices

This section provides the best practices for recording with respect to all technologies.

- Cloud Studio Timeout

During recording, the Cloud Studio may timeout for various reasons. So, before you start exporting the recording to Cloud Studio, check if the Cloud Studio has not timed out. If it is timed out, you must log in again (do not refresh) to the Cloud Studio and then export the recording.

- Default Activities in Automation after Re-recording

Once the re-recording is completed, the exported automation has some default activities related to starting and terminating the application. Remove those activities if they are not needed. Refer to the following default activities with respect to each technology:

- Web Automation

- Start Web Page
 - Close Application

◦ UI Automation

- Start Application
 - Terminate Application

SAP GUI

By default, the SAP Logon screen is captured and the following activities are added to it:

- Start Application
 - Wait (Screen)
 - Open Connection

- Close Connection
- Terminate Application

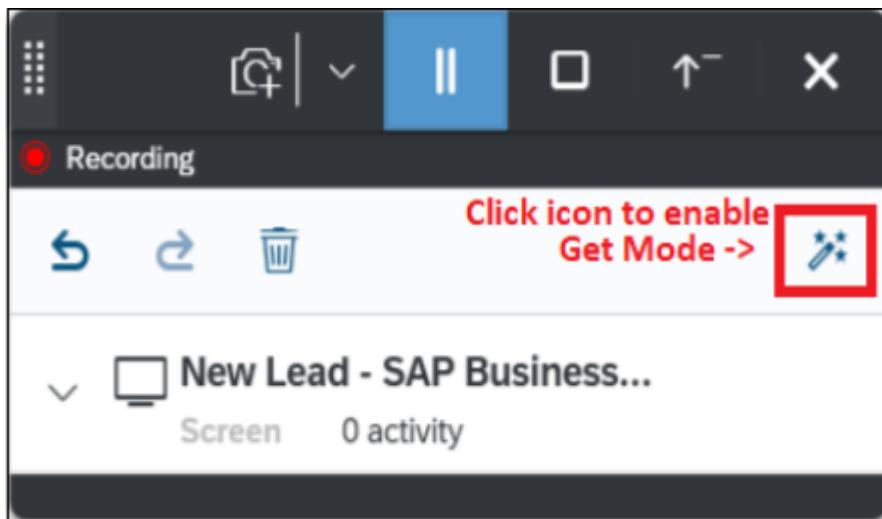
Read Value of an Element During Recording

Now, you can read the value of an element during the recording of the steps.

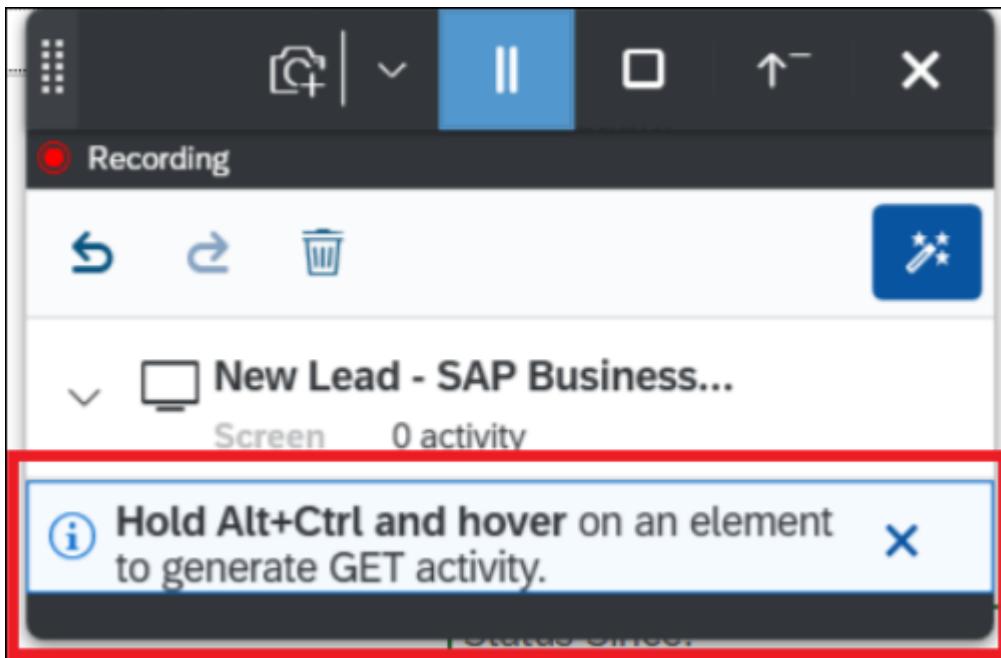
Procedure

Refer to the following steps to get the value of an element during the recording of the steps.

1. You can read the value of an element only in enabled **Get** mode. Click the  icon to enable the **Get** mode.

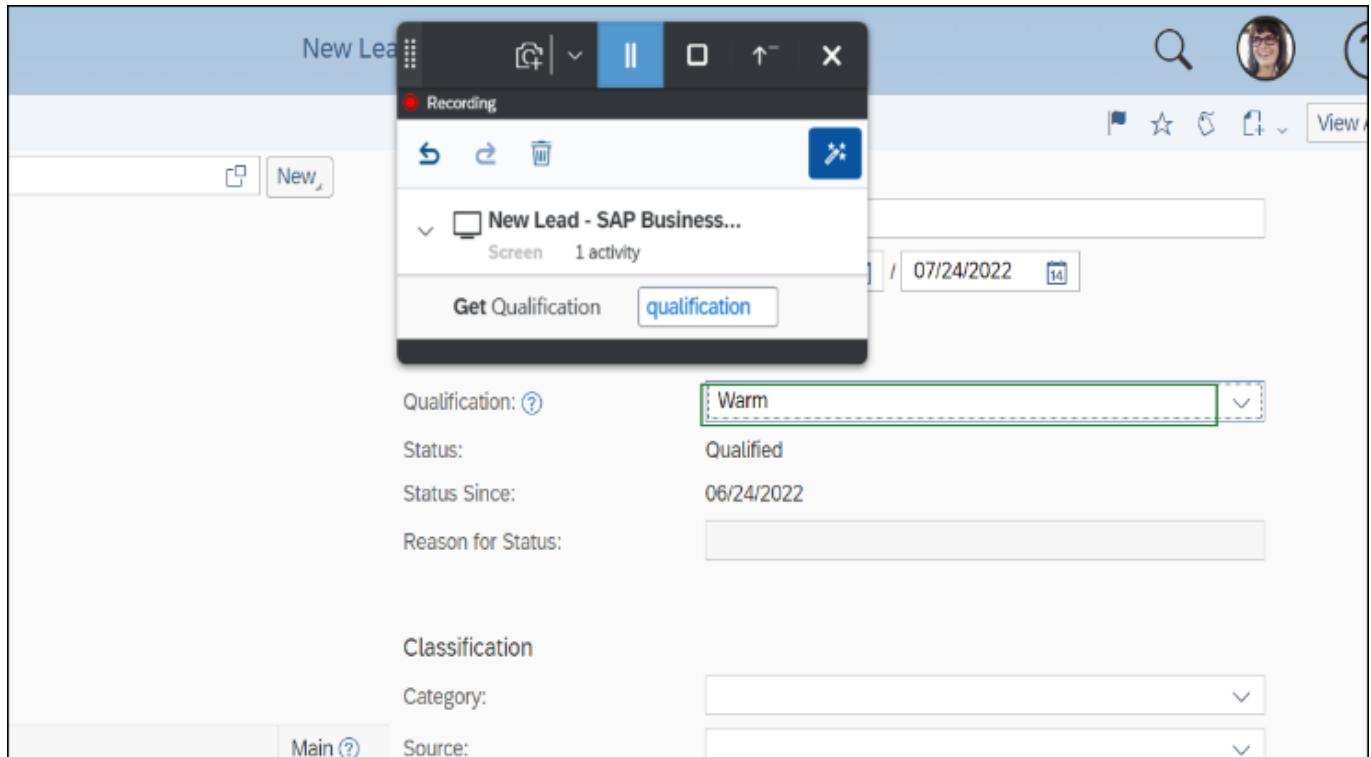


Once the mode is enabled, the icon is highlighted as shown in the following screenshot and a hint message is displayed at the bottom of the recording widget.

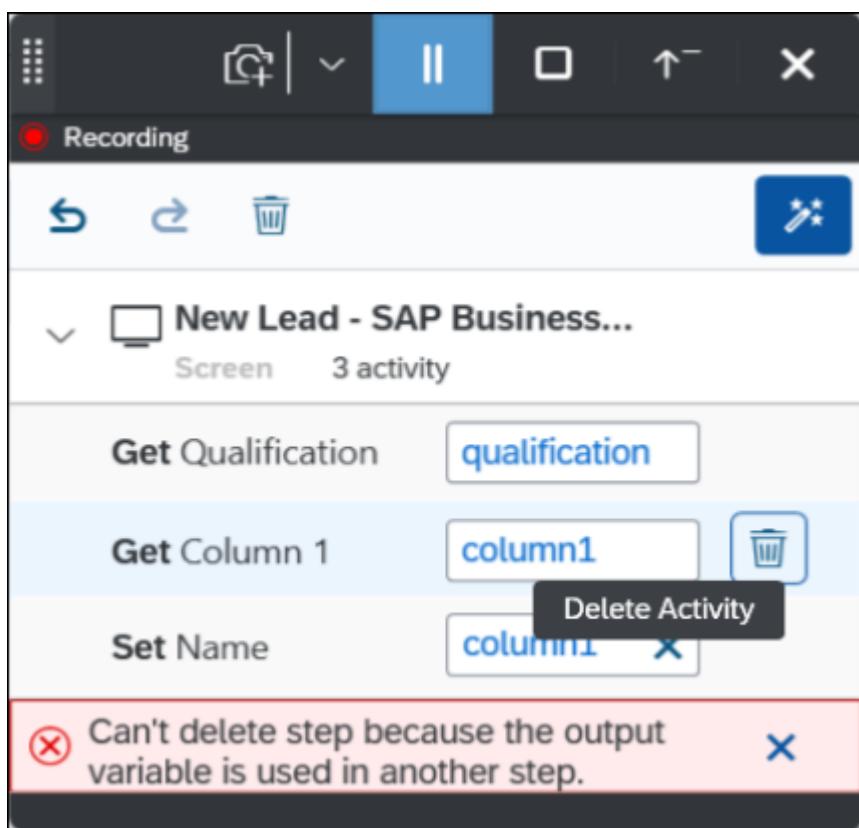


2. To generate the **Get** activity, place the cursor at the targeted element and the selection is highlighted with the green bounding rectangle. If the bounding rectangle targets the correct element, press the keyboard shortcut Alt + Ctrl to generate the activity, otherwise, move the cursor further for the correct selection.

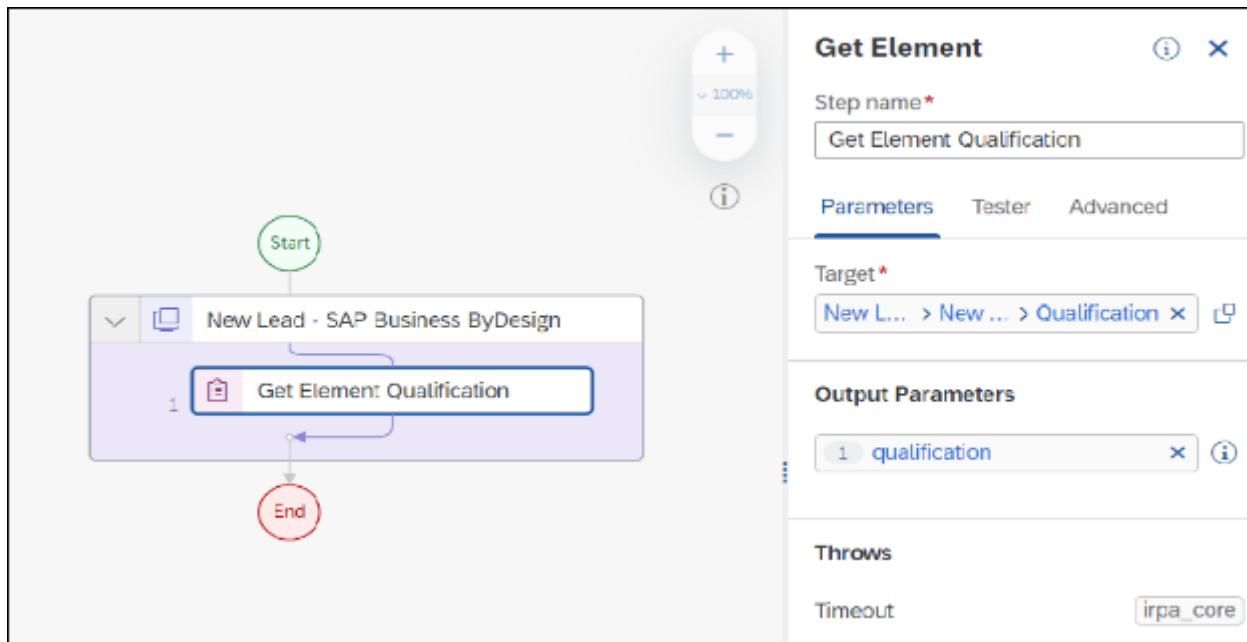
The **Get** activity is generated with the output variable as shown in the following screenshot. You can further utilize this output variable according to your use case.



3. If you try to delete the **Get** activity, whose output variable is referenced to some other activity, the following error message is displayed at the bottom of the recording widget.



4. Once the recording is exported, the generated automation will have the **Get** activity with the same output parameter as shown during the time of recording in the recording widget.



i Note

In SAP GUI for Windows (also called SAP WinGUI) when,

- you press the keyboard shortcut Alt+Ctrl, the **More** option can be triggered to pop up. This behavior is expected as the pressing of the Alt key is mapped to open the pop-up window of the **More** option. Press anywhere in the SAP GUI application after reading the value of an element, and the pop-up window will be disappeared.
- the SAP GUI application runs in the monitor scaled to values other than 100%, the green border around the element can appear in different locations.

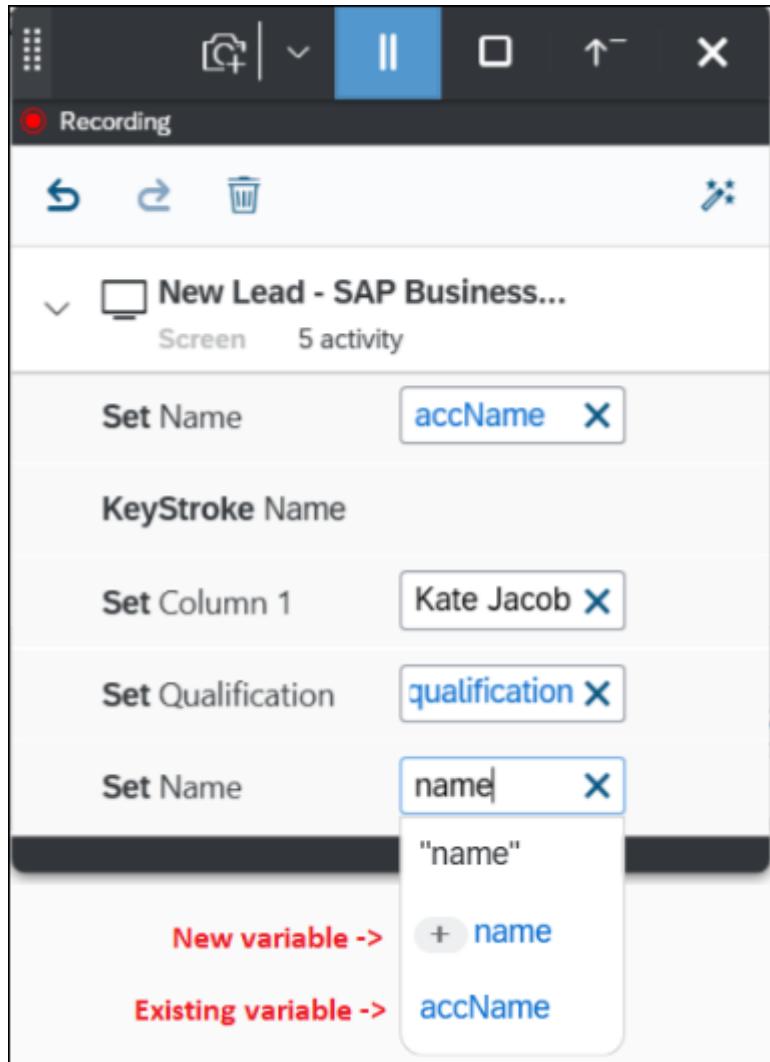
Set Value of an Element During Recording

Now, the recording widget is enhanced to support the editing of input to the **Set** activity.

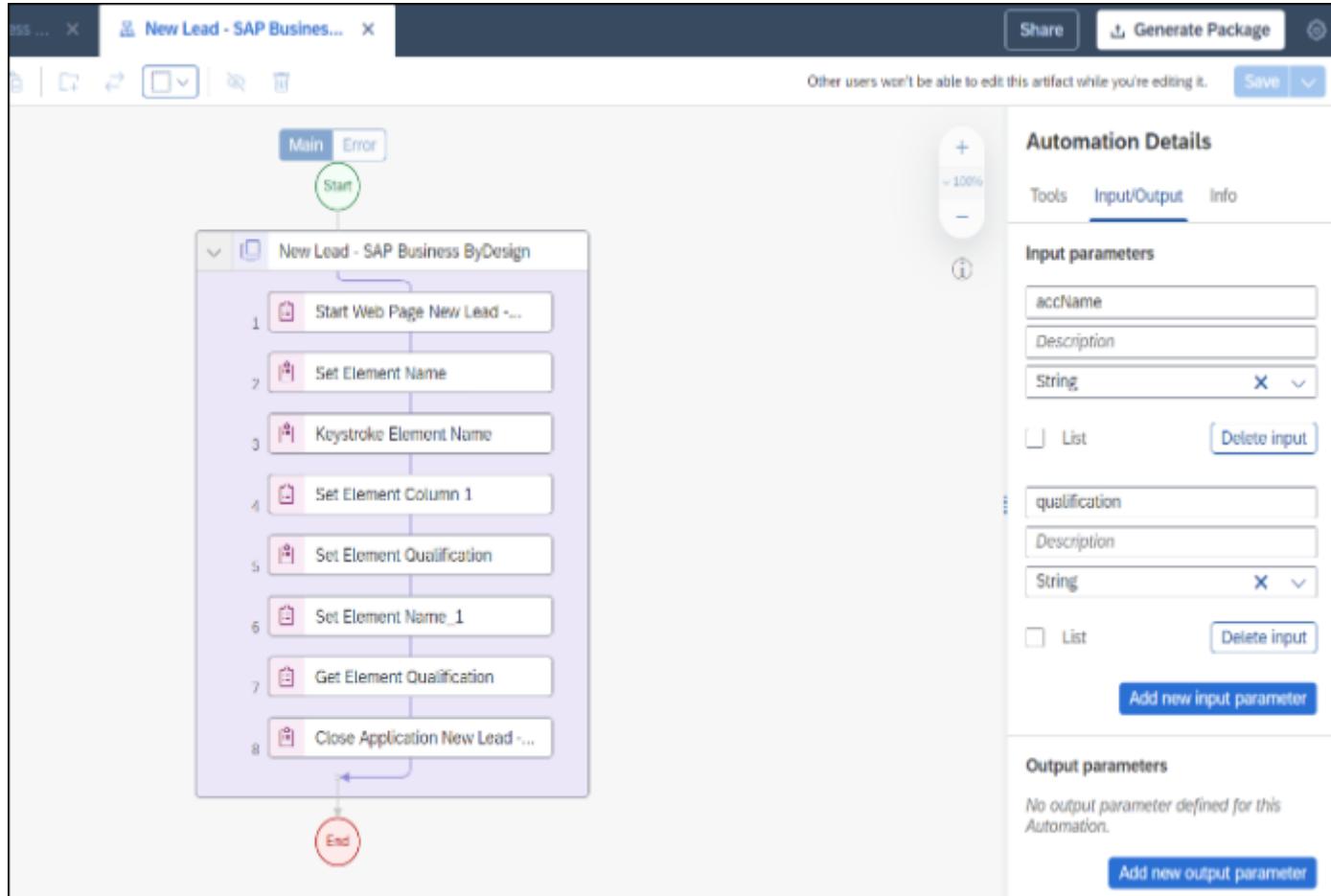
Once the **Set** activity is generated, you can change the value of the input if required.

The input can be of three types. You can provide:

1. hard coded value which is shown inside double quotes.
2. create and assign new variable which is shown with a prefixed '+' symbol.
3. select an input from a list of pre-existing variables.



Once the recording is exported, the newly created variables will be displayed in the Cloud Studio.



Capture and Declare Applications

Capturing and declaring applications involves identifying the applications you want to control in an automation. Even though applications are based on different technologies, they share many features in common.

Capturing Applications

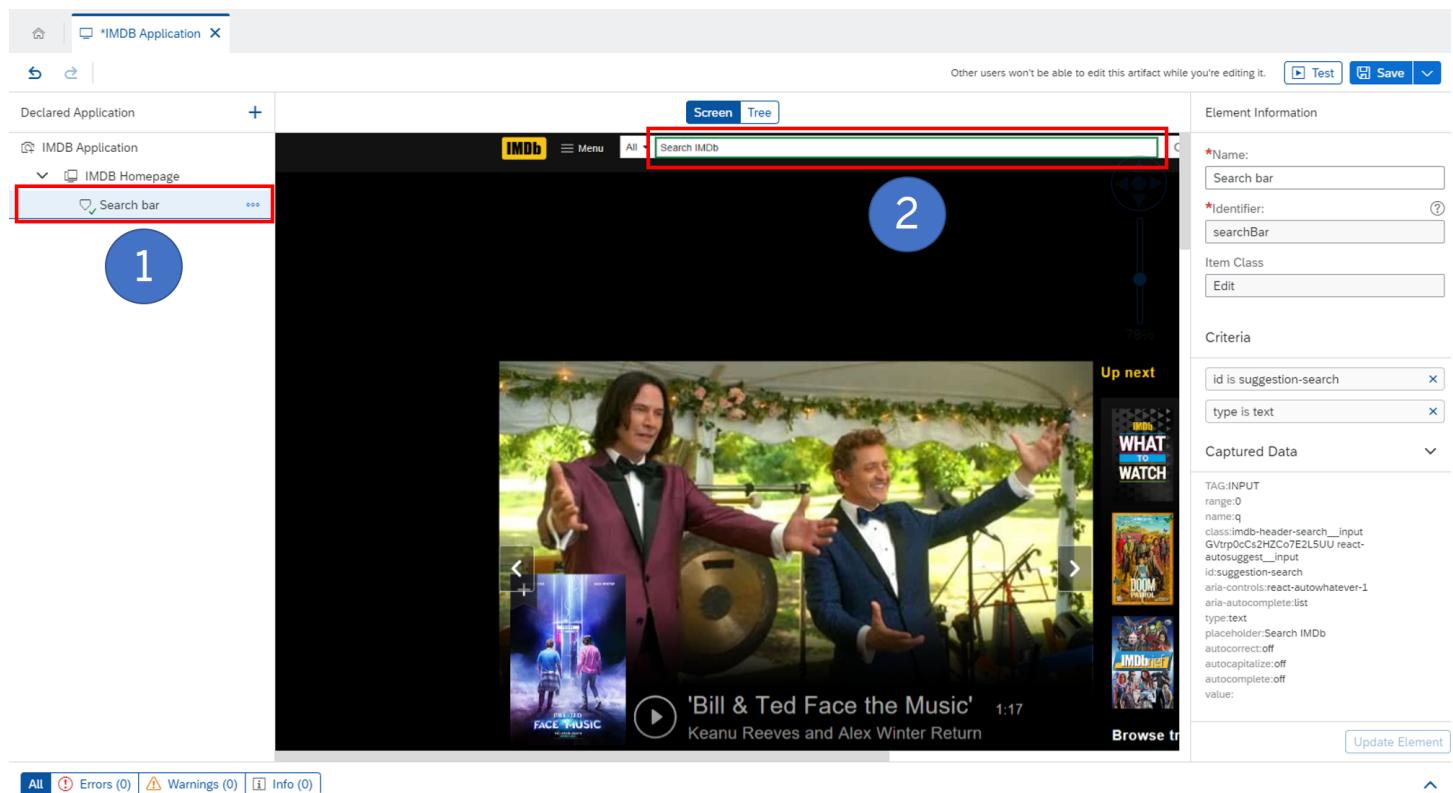
The application editor allows you to capture an application running on your local machine, select and capture screens, and recognize UI elements on those screens for use in automations.

The screenshot shows the SAP Intelligent Robotic Process Automation Cloud Studio application editor. The top navigation bar includes 'Intelligent Robotic Process Automation Cloud Studio' and 'test38'. The main area is divided into several sections: 'Overview' (with a 'Share' and 'Generate Package' button), 'Select Screen' (listing open screens like 'Document1 - Word', 'Document2 - Word', and 'Flow'), a preview window showing the IMDb website, and 'Screen Information' on the right. Under 'Screen Information', there are options to 'Create a new application' or 'Attach to an existing application'. The 'Application Name' field is set to 'IMDb: Ratings, Reviews, and Wh...', and the 'Application Identifier' field is set to 'iMDBRatingsReviewsAnd'. A large blue 'Capture' button is at the bottom right.

For more information about capturing applications, see: [Capturing Applications](#)

Declaring Applications

Once an application has been captured, you then must declare it. Declaring means selecting the criteria that the system will apply in order to correctly identify an application, screen or element when running your automation.



For more information about declaring applications, see: [Declaring Applications](#)

Capture Applications and Design Automations

[Capture and Declare Applications](#)

The Cloud Studio artifact of type *Application* lets you capture an application running on your local machine, select and capture screens, and recognize UI elements on those screens for use in automations.

[Design Automations](#)

Automations are composed of a succession of steps you build in the Cloud Studio with the automation designer. An automation can orchestrate multiple activities on different applications and screens available on a specific computer.

[Trigger Automations Using the Project Launcher](#)

The project launcher allows you to launch your automations in attended mode, either manually from the agent or automatically using events as triggers.

[Manage Files](#)

In Cloud Studio, you can create, import, and edit a file.

[Business Activity Monitoring \(BAM\)](#)

Capture and Declare Applications

The Cloud Studio artifact of type *Application* lets you capture an application running on your local machine, select and capture screens, and recognize UI elements on those screens for use in automations.

Capturing and declaring applications means identifying the applications you want to control in an automation. Even though applications are based on different technologies, they share many features in common.

You start by locating the screens that you want to use. Within a screen, you then look at the UI elements you want to control - such as input fields, output fields, and buttons.

Another key task is declaration, which means choosing the recognition criteria for the application, screens and UI elements that will be used at run time to correctly identify them.

When you have captured all screens and declared the application, screens and UI elements, you can then test the application before including it in an automation.

i Note

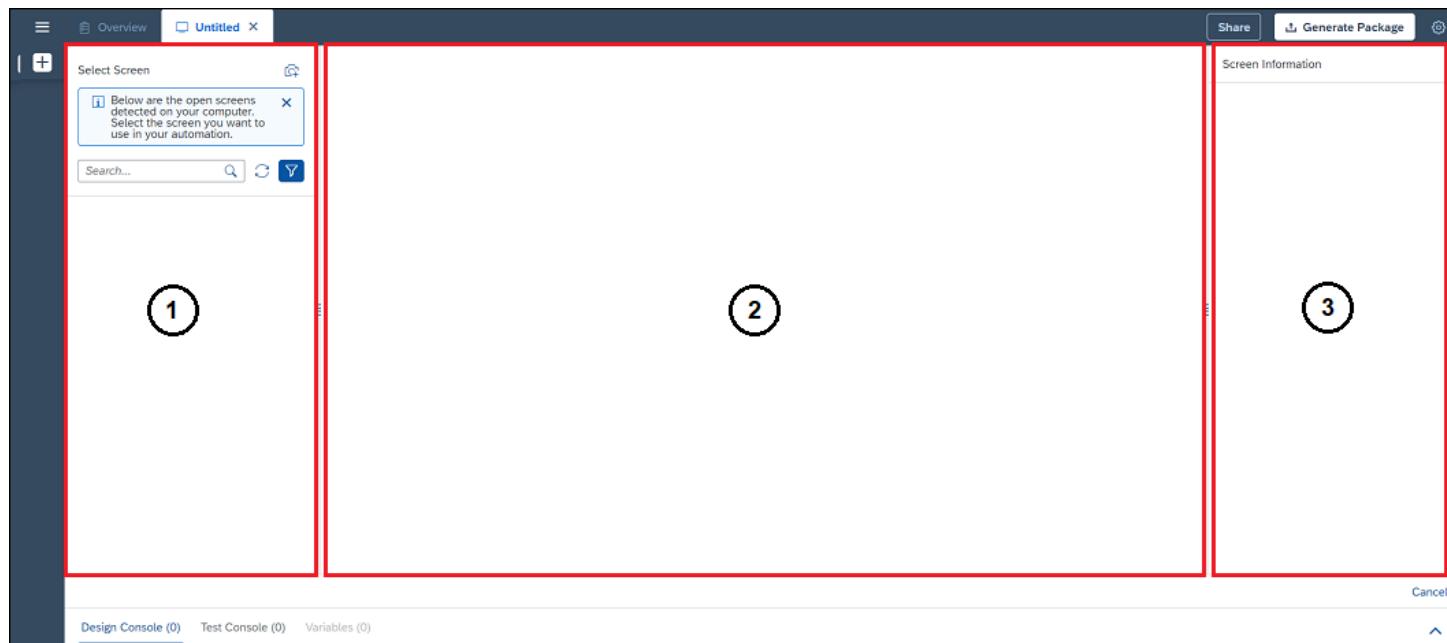
You can't create an application in a project if the project already contains a Desktop Package.

Overview

The application editor allows you to capture an application running on your local machine, select and capture screens, and recognize UI elements on those screens for use in automations.

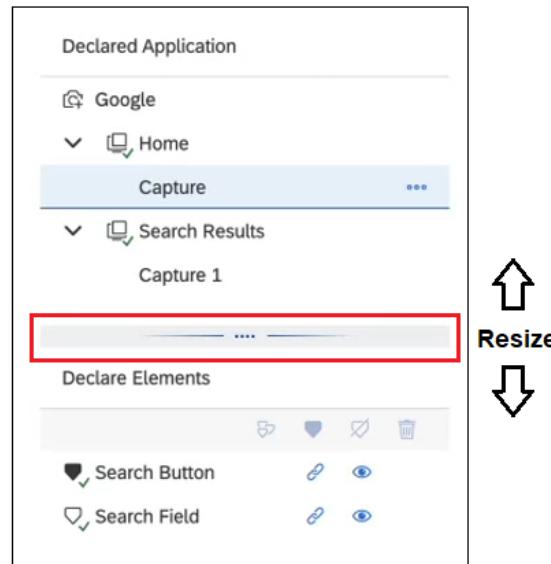
The application editor consists of three parts:

- picker panel on the left side (1)
- main area in the center (2)
- screen information on the right side where criteria are defined (3).



Resizing the Picker Panel

The application editor's picker panel splits into two sections - **Declared Application** and **Declare Elements**. These two sections can be resized so that you can give more space to any section. To resize, move the splitter up or down accordingly as shown in the following screenshot.



Capture an Application

Prerequisites

Capturing requires SAP Intelligent Robotic Process Automation Factory and the Cloud Studio to be running in a Chrome browser. It also requires the browser extension installed and activated, and the Desktop Agent configured to use with the target tenant.

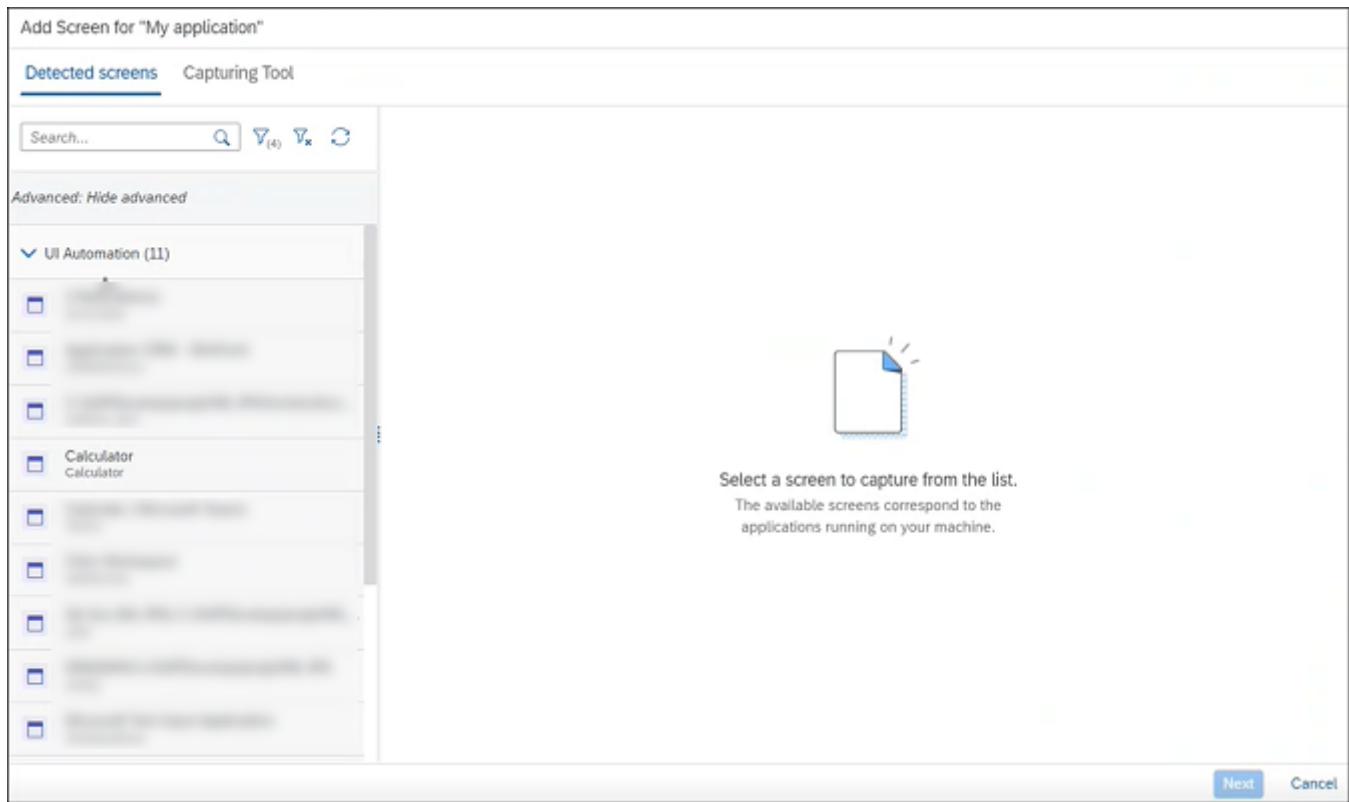
i Note

For more information about browser extension, see [Configuring Web Browsers](#)

To capture an application, you start by capturing one of its screens. An application can be a browser app, a Windows application, an SAP application, and so on.

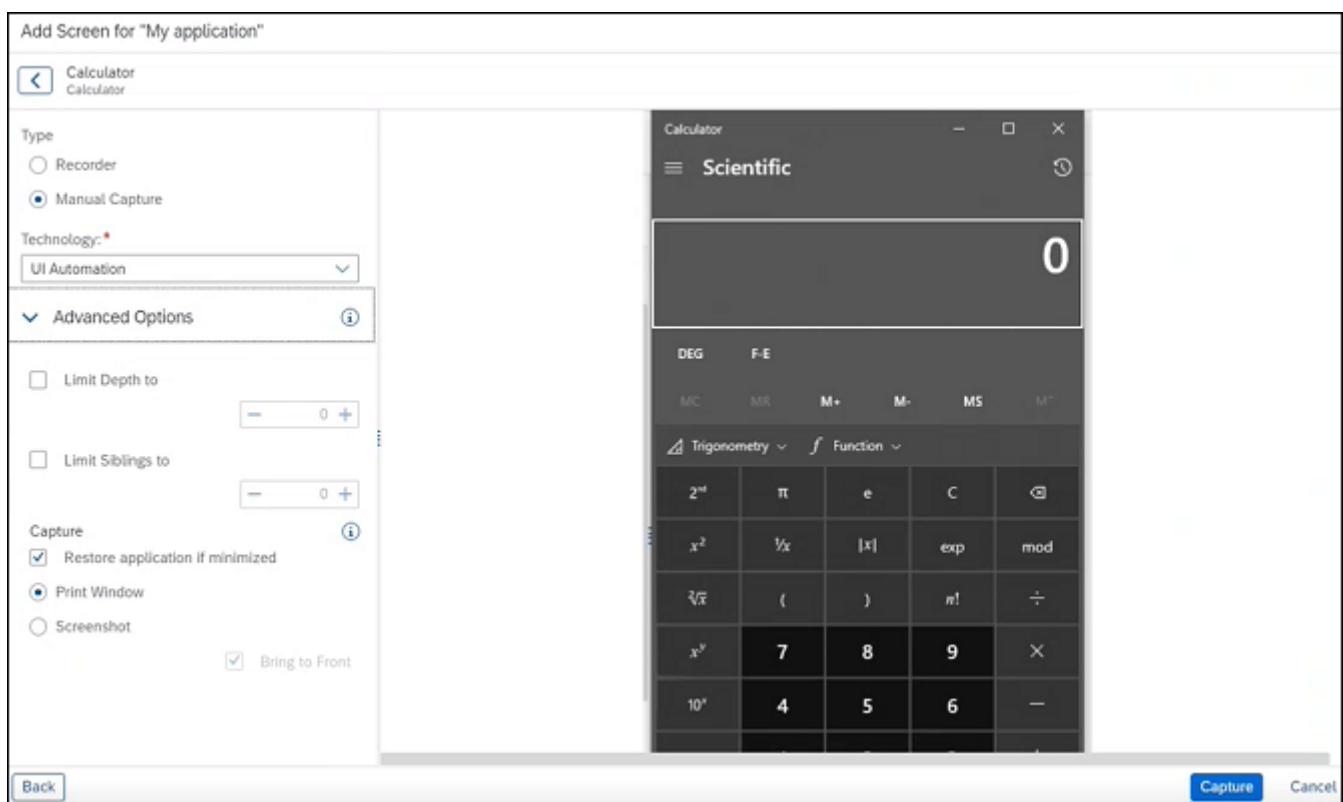
Procedure

1. Create a new project. For more information, see [Create and Manage Projects](#).
2. In the project explorer, click and then click **Create** in the left panel or click **Create** in the project explorer main panel. The artifact menu is displayed.
3. In the artifact menu , click **Application**. The **Create Application** popup window is displayed.
4. In the **Create Application** popup window, you have the following options:
 - Enter a name in the **Application Name** field.
 - Edit the **Application Identifier** field (without space) (optional).
 - Enter a short description in the **Description** field.
5. Click **Create**. A new tab labeled to the name of the captured application is displayed. The system starts detecting the applications and their screens currently running on your local machine. When it's done, a list of screens is displayed in the picker panel on the left.



6. Select the application you want to record from the list of screens. A preview of the screen is displayed in the capture area.

7. Click **Next**. This redirects you to the application that you want to capture.



8. Select the **Type** as **Manual Capture**.

9. **(Optional):** Select the technology by choosing from the **Technology** dropdown list.

i Note

The Cloud Studio offers various technologies for capturing applications. For example, Web is used to capture a browser application, Win for Windows applications. UI Automation technology can be used to capture a wide range of applications.

⚠ Caution

We recommend keeping to the detected technology in most cases.

10. Define **Advanced Options**. For details, [Capture Options](#).

11. Click **Capture** to start the screen capture. You've now captured an application by capturing one of its screens.

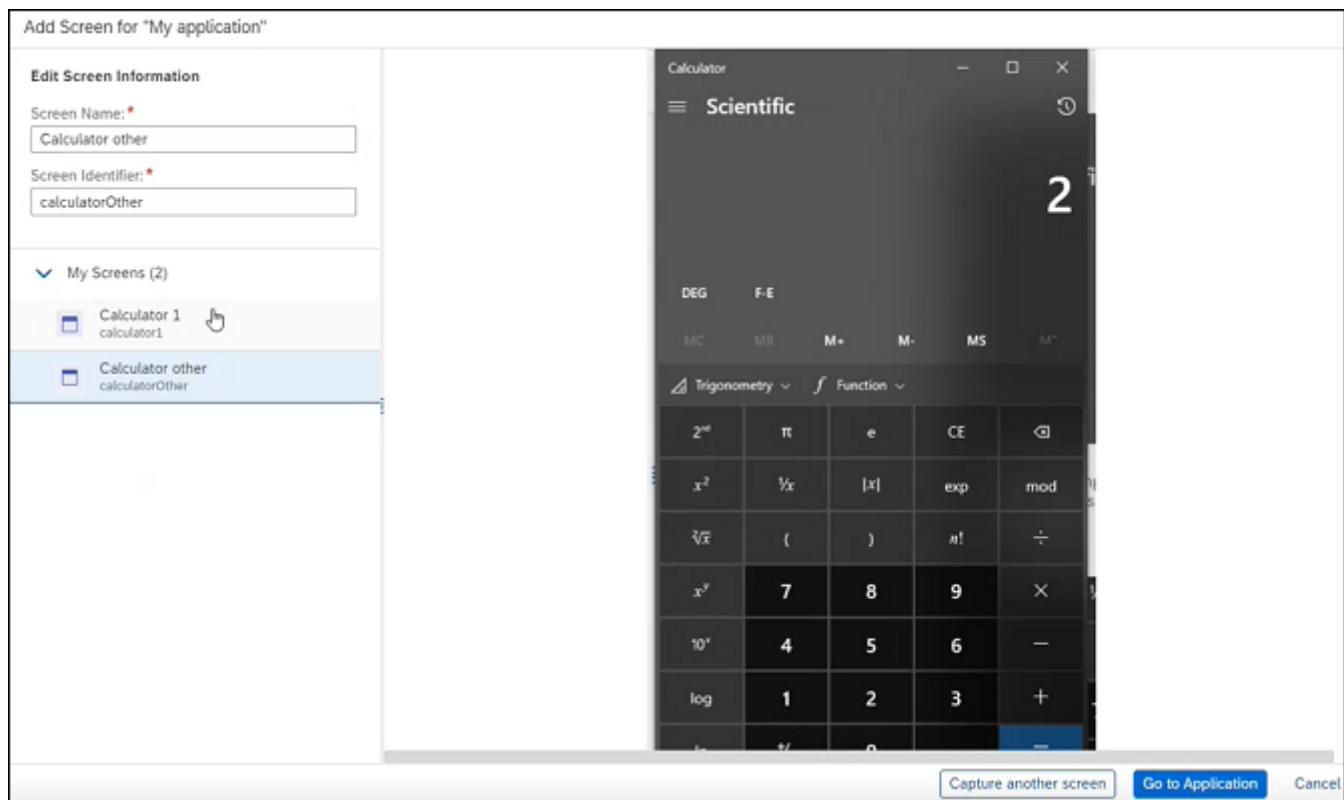
i Note

If you haven't imported SDK packages in your project yet, the mandatory SDK packages are automatically downloaded.

In the left-hand navigation panel, the captured screen information is displayed. You can edit the captured screen information. The captured screens are listed under the **My Screens** list.

12. **(Optional):** Click **Capture another screen** to capture new screen.

13. Click **Go to Application** once screen capture is completed.



Your application is created and displayed in a new tab. In the left-hand navigation panel of your application, you see a list of captured screens under the Declared Application. Once the screen is captured, the image in the capture area changes and becomes interactive. You can zoom in and out. The application recognition criteria help your bot to recognize the screen correctly. When your automation runs, the application recognition criteria are automatically selected. You can change the criteria when you declare the application.

Next Steps

You can now capture as many screens as you want to the declared application. You also need to declare your application, its screens, and the elements on the screens (like buttons and input fields) that you want to use in your automation project.

Best Practice

When you start a capture with Chrome and Firefox in Cloud Studio, make sure the window zoom factor is the same on every screen (whatever the browser zoom factor).

If you have some issues capturing screen elements, you may be able to solve the problems by setting the zoom factor to 100%. Go to the Scale and layout settings of your laptop and make sure that your zoom factor is set to 100%, then restart your session to apply the change.

In the Capture window, a warning message lets you know if the zoom level of the target page is not set to 100%.

The following issues can be avoided if the zoom factor is set to 100% in the runtime phase:

- Incorrect position of highlighting of elements.
- Incorrect coordinates during a mouse click event and its failure.

Capture another Screen

Prerequisites

You've captured an application and at least one screen as described in [Capture an Application](#).

Context

You might need to capture the same screen multiple times, for example to capture different popup areas.

Procedure

1. From a project in Cloud Studio, click on **Applications** in the **Project Explorer** if necessary, and choose the application for which you want to capture another screen.
2. Click  next to the declared application and click **Add Screen** to add a screen capture.
3. Choose the screen you want to capture from the list of open screens detected.

The icons in front of the screen names indicate the technology, such as Web.

You can refresh the list any time via  - for instance, when the applications currently running on your desktop have changed.

- a. **Optional:** Enter a term in the search field to find your screen faster.

Your search term is compared with the screen names, executable names, and where relevant, URLs of the detected open screens.

- b. **Optional:** If the list is very long, filter it via  before choosing a screen.

Filter running screens

Technology

- HLLAPI
- UI Automation
- Web
- Win
- SAP GUI
- SAP Web GUI

Advanced

- Hide advanced
- Hide minimized

Filter

Cancel

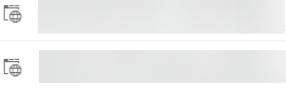
A preview image of the screen is displayed in the capture area. The screen name and identifier are automatically entered in the field in the right-hand panel. You can change them manually.

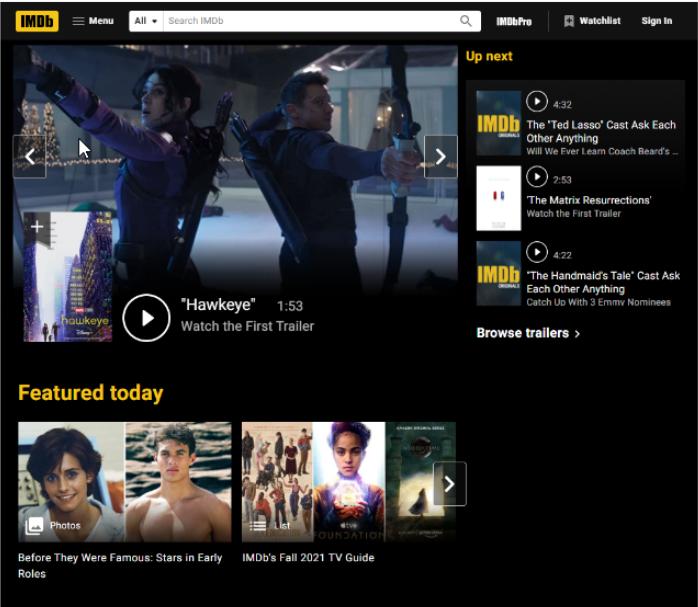
Add Screen

Select Screen +/-

Below are the open screens detected on your computer. Select the screen you want to use in your automation.

↻
Y





Google
<https://www.google.fr/>

IMDb: Ratings, Reviews, and Where...
https://www.imdb.com/?ref_=nv_home

Screen Information

*Application Name: IMDbApp

*Screen name:

*Screen Identifier:

Technology:

Capture Options

Max capture height (in pixels)

Disable Auto Scrolling (Web)

Capture ?

Restore application if minimized

Print Window

Screenshot

Bring to Front

Capture **Cancel**

4. Optional: Select the technology by choosing from the dropdown list.

Cloud Studio offers various technologies for capturing applications. For example, Web is used to capture a browser application, Win for Microsoft Windows applications. UI Automation technology can be used to capture a wide range of applications.

⚠ Caution

We recommend keeping to the detected technology in most cases.

5. Optional: Define one or more **Capture Options**.

This is custom documentation. For more information, please visit the [SAP Help Portal](#)

For details, see [Capture Options](#).

6. Choose **Capture**.

You've now captured an additional screen for your application. Once the screen has been captured, the preview in the capture area changes and becomes interactive. You can zoom in and out.

The new screen appears in the picker list on the left.

If you have captured the same screen multiple times, a number starting with 1 is automatically appended to the names and identifiers of all but the first capture.

Click the application in the picker list and you're shown tiles representing each captured screen. Click a tile to view the corresponding screen.

7. Save your work.

i Note

Delete a screen by choosing the three dots after the screen name in the picker panel and selecting **Delete**.

Next Steps

Next, you need to declare the screen.

Related Information

[Declare a Screen](#)

Warning Messages when Capturing a Screen

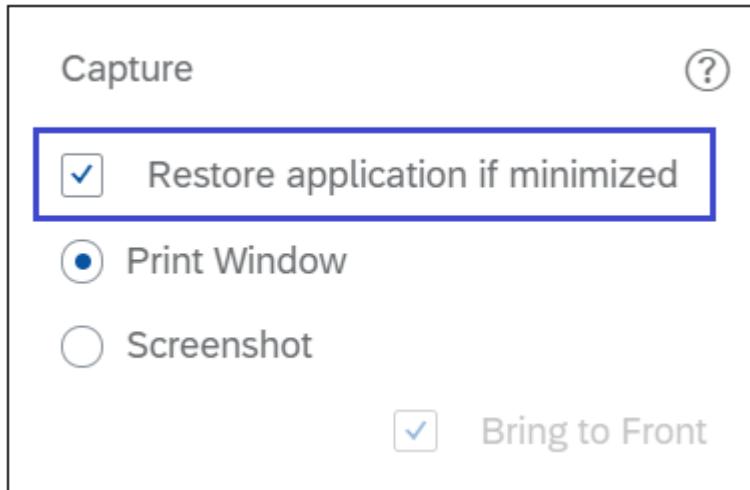
Context

When browsing available screens in the **Search screen** panel, you select entries from the list. If the preview screen is empty during the application capture, several warning messages can be displayed. These warning messages explain why the screen cannot be captured.

List of Warning Messages

- **The screen is minimized and will be restored during capture.**

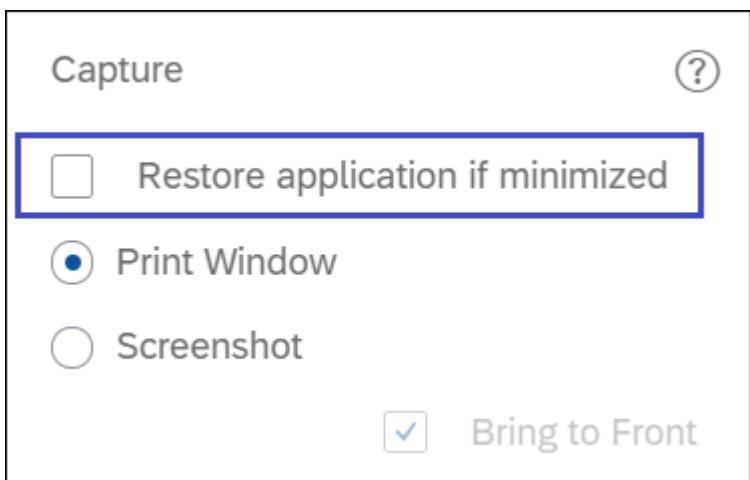
This warning message is displayed when the target screen is minimized and the **Restore application if minimized** option is selected.



If the **Restore application if minimized** option is selected, then the target application screen that is minimized will be restored while capturing.

- **The screen is minimized. Check the option to restore it on capture.**

This warning message is displayed when the target screen is minimized and the **Restore application if minimized** option is **not** selected.



If the **Restore application if minimized** option is **not** selected, then the target application screen that is minimized will **not** be restored while capturing.

- **This screen is a frame of a web application and does not have a user interface to preview.**

This warning message means that the screen is a frame of a web application that is not visible in the viewport.

Duplicate Captured Application Screen

In the Cloud Studio, you can duplicate the captured application screens one or more times. The duplicate screen represents the contents of the original screen.

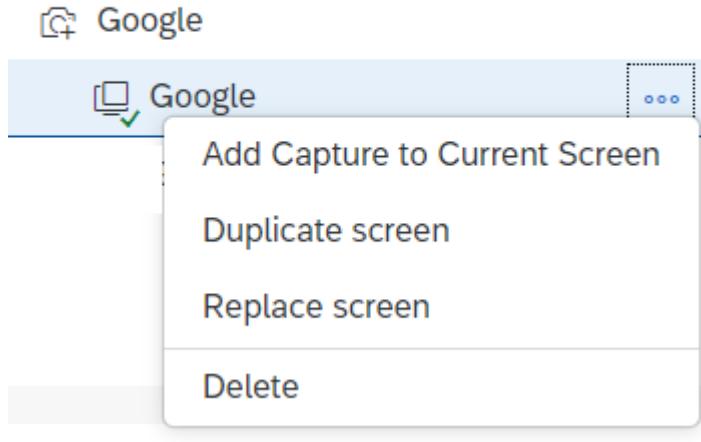
In the duplicated screen, you can declare new elements, or you can modify the existing declared elements of the screen.

To duplicate a screen, follow the below procedure.

1. In the IRPA Cloud Studio, under **Declared Application**, click the icon of a captured screen.

More Options are displayed.

Declared Application



2. Click **Duplicate screen**.

The **Duplicate screen** pop-up screen is displayed.

Duplicate Screen

*Name	Google - Copy
*Identifier	googleCopy

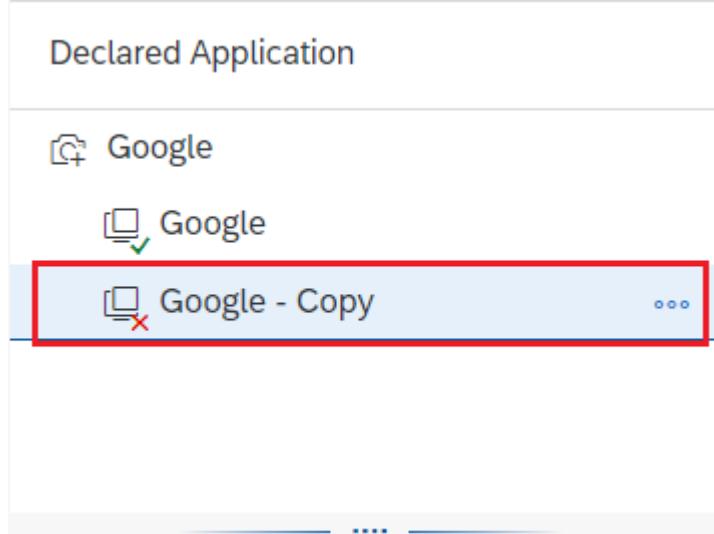
Duplicate [Cancel](#)

Note

If required, you can update the name of a screen.

3. Click **Duplicate**.

The duplicate screen is created and displayed under **Declared Application**.



List the Application Frames

This is custom documentation. For more information, please visit the [SAP Help Portal](#)

You can list all the application frames and subframes that appear on the screen. You can select the relevant frame of the captured screen. During an element declaration, the corresponding area of the selected frame is highlighted with respect to the rest of the screen. Therefore, you can easily distinguish between the different frames and find the different components that are selected on the screen.



In the following screenshot, you can observe that only the area of the selected frame is highlighted and the rest of the screen which is not part of the selected frame is faded. Thus, you can easily find the element available on the highlighted screen and select it accordingly.

This screenshot shows the SAP Application Capture interface with a faded background. In the center, there's a video player window with a man's face. An arrow points to this window with the text 'Highlighted Frame'. To the right, there's a 'Screen Information' panel with various configuration options, including 'Name: Test Capture (Frame 2)', 'Identifier: https://images-na.ssl-images-amazon.com...', 'Framework: None', 'Technology: Web', and 'Recognition Order: 2'. At the bottom right of the panel, there's a 'Declare Element' button.

You can also select the elements within the frame without any offsets as shown in the following screenshot.

This screenshot shows the SAP Application Capture interface with a highlighted frame. The 'Element Information' panel on the right is open, showing details for an element named 'A' with identifier 'a', element class 'Hyperlink', and recognition tests. The 'Captured Data' section shows 'clickthrough href: https://www.patreon.com/c/gaming target: _blank'. At the bottom right of the panel, there's a 'Declare Element' button.

Capture Options

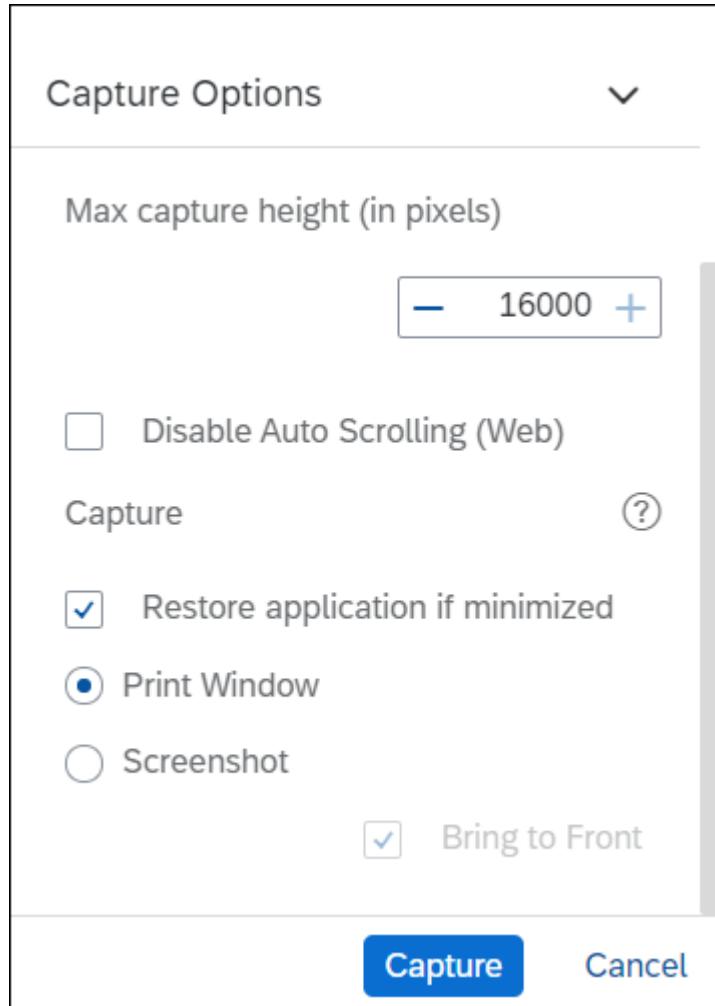
SAP Intelligent RPA Cloud Studio supports multiple application capture options such as **Max capture height (in pixels)**, **Disable Auto Scrolling (Web)**, **Print Window**, **Screenshot (Bring to Front)**, and **Screenshot**. These options help the user to capture various scenarios of the application screens.

You can find the capture options at the bottom of the left panel in the Cloud Studio.

The screenshot shows the SAP Intelligent RPA Cloud Studio interface. On the left, there's a sidebar titled 'Select Screen' with a message: 'Below are the open screens detected on your computer. Select the screen you want to use in your automation.' It lists several open applications: Book2 - Excel, Document1 - Word, Flow, Google, IMDb: Ratings, Reviews, and Wh..., Inbox - dsharsh.daniel kenneth.r..., Microsoft Text Input Application, and CAD IDDA Document Review. The 'Google' entry is selected. The main area shows a Google search results page for 'Google'. On the right, there's a 'Screen Information' panel with fields for 'Application Identifier' (set to 'google'), 'Application Description' (set to 'Description'), 'Screen Name' (set to 'Google'), 'Screen Identifier' (set to 'google'), and 'Technology' (set to 'Web'). A 'Capture Options' button is highlighted with a blue border. At the bottom right of the main area are 'Capture' and 'Cancel' buttons.

To see the application capture options click **Capture Options**. The capture options are displayed.

Before capturing a screen, you can define one or more of the following capture options, if necessary.



Capture Option	Description
Max capture height (in pixels)	<p>This option limits the height of the screen captured. It's useful when capturing very large web pages. Please note that the entire tree structure of the screen's elements (fields, buttons, and so on) is captured.</p> <p>After capturing a screen, you can see this tree structure by switching the view in the center pane from Screen to Tree.</p>
Disable Auto Scrolling (Web)	It disables auto scrolling of web application screen.
Restore application if minimized	<p>If the Restore application if minimized option is selected, then the target application screen that is minimized will be restored while capturing.</p> <p>If the Restore application if minimized option is not selected, then the target application screen that is minimized will not be restored while capturing.</p>
Print Window	<p>By default, the target application screen is captured using the Print Window mode.</p> <p>i Note If the resulting capture is a black bitmap, the Cloud Studio automatically switch to the Screenshot mode.</p>
Screenshot	This mode is the same as the Screenshot mode, except that it is up to you to bring the target screen to front. This can be useful in some specific cases where an application has a child window part (pop-up screen).
Screenshot with Bring to Front	<p>If the Cloud Studio do not detect that the resulting capture is black, you can manually switch to the Screenshot (Bring to Front) mode by selecting Screenshot (Bring to Front) option.</p> <p>i Note In this case, the Cloud Studio automatically brings the target screen to front.</p>

Related Topics

- [Capture an Application](#)
- [Warning Messages when Capturing a Screen](#)

Application Multi-Captures

You can capture multiple screens of the same application.

The captures represent variations of the same screen.

At any point in time you can switch between the captures and delete a capture. You can define the Capture Name and Description (Optional).

Elements defined in one of the captures will be highlighted on all the other captures based on the defined criteria.

Add a Capture to an Existing Screen

You can capture multiple screens of the same application. The captures represent variations of the same screen.

Prerequisites

- You've captured an application and at least one screen as described in [Capture an Application](#).

- You've defined the required elements for a screen capture. For more information about declaring an element, see the [Declare Element Icons](#) and [Declare an Element](#) sections.

Context

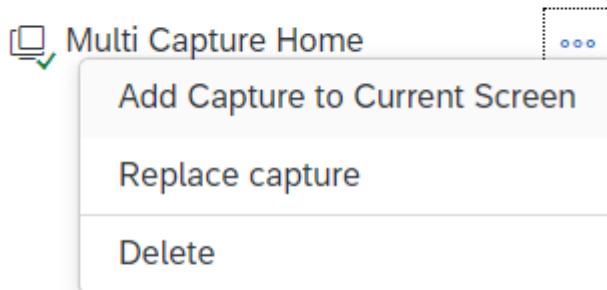
To capture multiple screens of the same application, complete these steps:

Procedure

1. In the SAP Intelligent RPA Cloud Studio, under **Declared Application**, click the  icon of a captured screen to display more options.

Declared Application

Multi Capture



Declare Elements

2. Click **Add Capture to Current Screen**.

3. The **Add Capture** screen shows a list of open screens detected by your agent. Choose the screen that you want to add and click **Next**.

The selected screen is displayed in the preview area.

4. Select the options for the capture.

The **Link all elements found on-screen** check box is selected by default. This fetches all declared elements from the screen and only links the elements that can be found based on the criteria in the current capture. If you don't want to link all elements found on the screen, deselect the **Link all elements found on screen** check box.

5. Click **Capture**.

Note

If the new capture isn't compatible with the criteria for the existing screen, you can choose to add the capture but you need to check the screen recognition criteria.

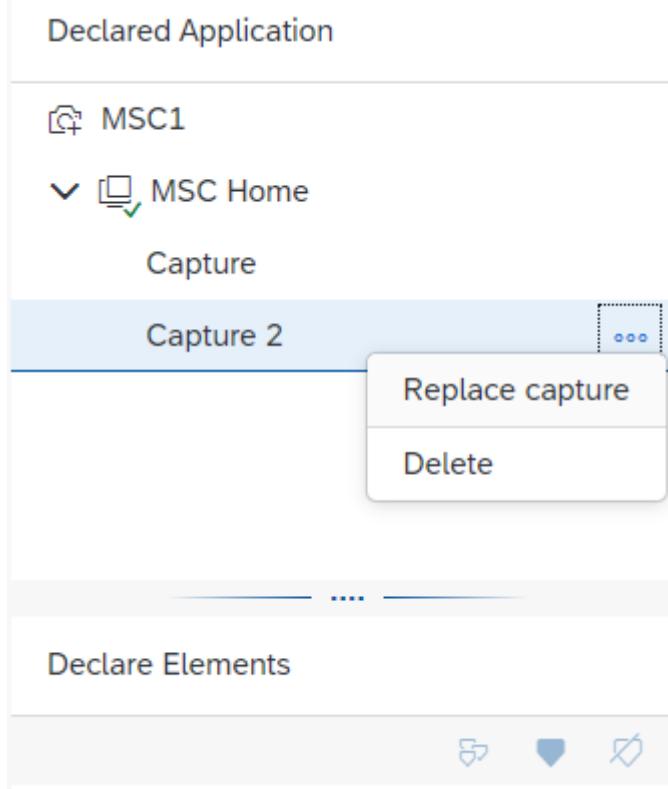
Delete a Capture

At any point in time you can delete a capture.

To delete a capture, follow the below procedure:

1. In the IRPA Cloud Studio, under **Declared Application**, select the desired capture and click the  icon.

The more options are displayed.



Declared Application

MSC1

MSC Home

Capture

Capture 2

Replace capture

Delete

....

Declare Elements

2. Click **Delete**

The warning pop-up message is displayed.

Warning

Are you sure you want to delete "Capture 2"?

Delete **Cancel**

3. Click **Delete**

The capture will be deleted.

Replace a Capture

To replace a capture, follow the below procedure:

1. In the IRPA Cloud Studio, under **Declared Application**, select the desired capture and click the  icon.

The more options are displayed.

Declared Application

The screenshot shows the SAP Application Management interface. At the top, it says 'Declared Application'. Below that is a tree view with 'MSC1' expanded, showing 'MSC Home' and 'Capture'. Under 'Capture', there is a sub-item 'Capture 2'. A context menu is open over 'Capture 2', with two options: 'Replace capture' and 'Delete'.

2. Click **Replace capture**.

The **Replace Screen** is displayed.

3. In the open screens list detected from your agent, select the desired screen you want to replace with the previous capture.

The selected screen is displayed in the preview area.

The screenshot shows the 'Replace Screen' dialog box. On the left, there's a list of detected screens: 'IMDb: Ratings, Reviews, and Where... (https://www.imdb.com/)', 'Google Search (https://www.google.com/search?q=hello+world&s...)', and 'SAP IRPA Document Review Track... (https://sap.sharepoint.com/:x/r/sites/120365/_ley...)''. In the center, a preview area shows a screenshot of the IMDb website for 'Sipsmith Gin'. On the right, there are 'Screen Information' and 'Capture Options' tabs. The 'Screen Information' tab shows 'Application Name: GoogleLC', 'Screen name: Google', 'Screen Identifier: google', and 'Technology: Web'. The 'Capture Options' tab has a 'Capture' button at the bottom.

4. Click **Capture**.

After successful capture of the application screen, a preview screen is displayed in the capture area.

Capture on Hover

You can capture an application screen using Capture on Hover.

i Note

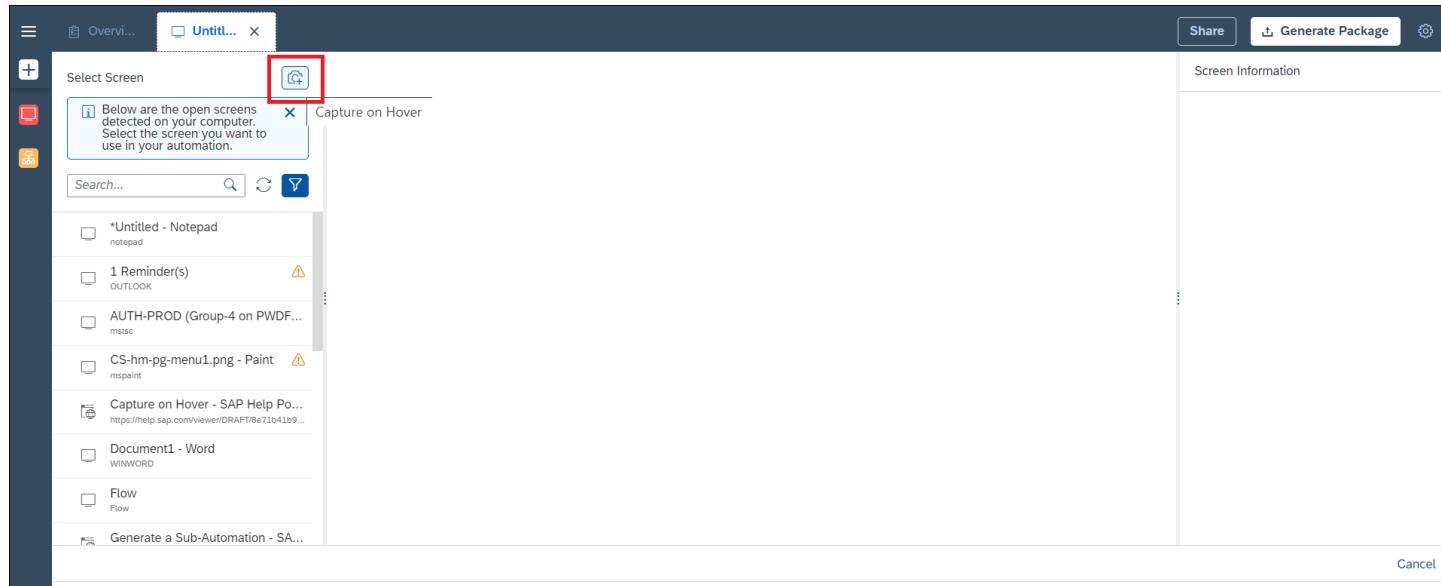
This is custom documentation. For more information, please visit the [SAP Help Portal](#)

The capture on hover is only available for Desktop Agents [version 2.0.12](#) and above.

Sometimes, elements can disappear from the screen when you lose focus on the screen to be captured. In such cases, you can still capture the element such as drop-down menus using the **Capture on Hover** option. To achieve this, you must navigate to the application you want to capture and hover over the screen, part of the screen, or drop-down menu to capture.

Using Capture on Hover

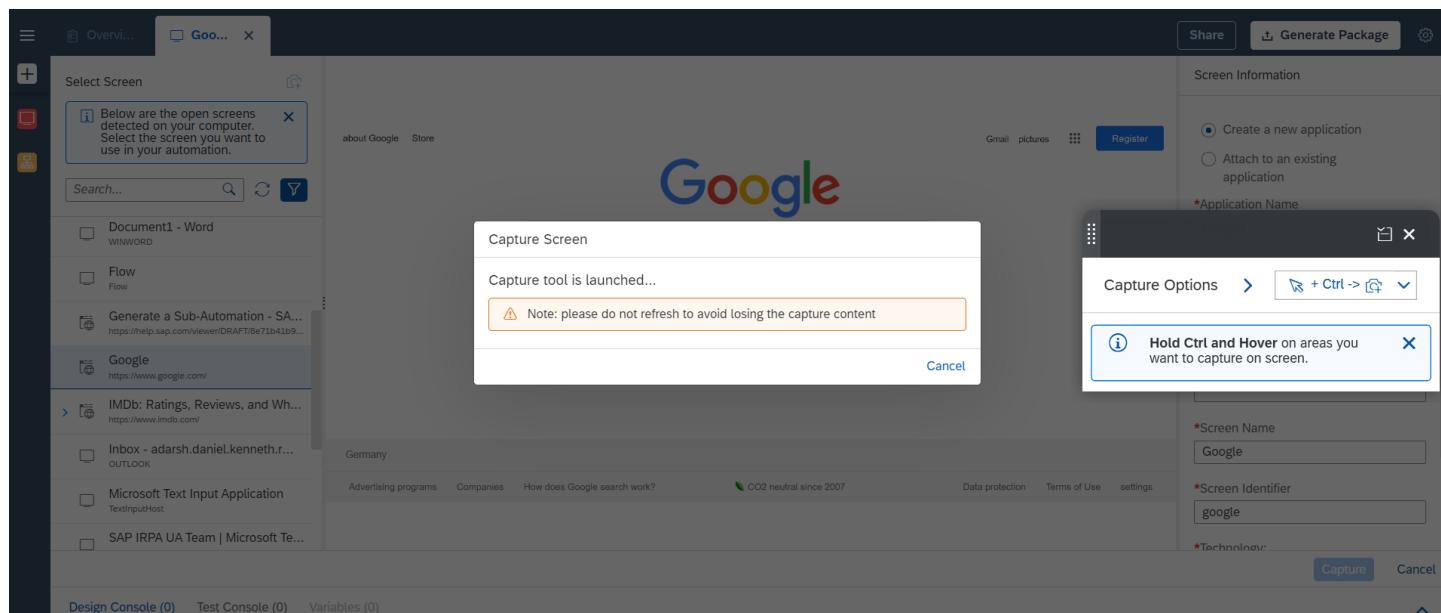
In the Cloud Studio, you can find the **Capture on Hover** option at the right side of the **Select Screen** panel.

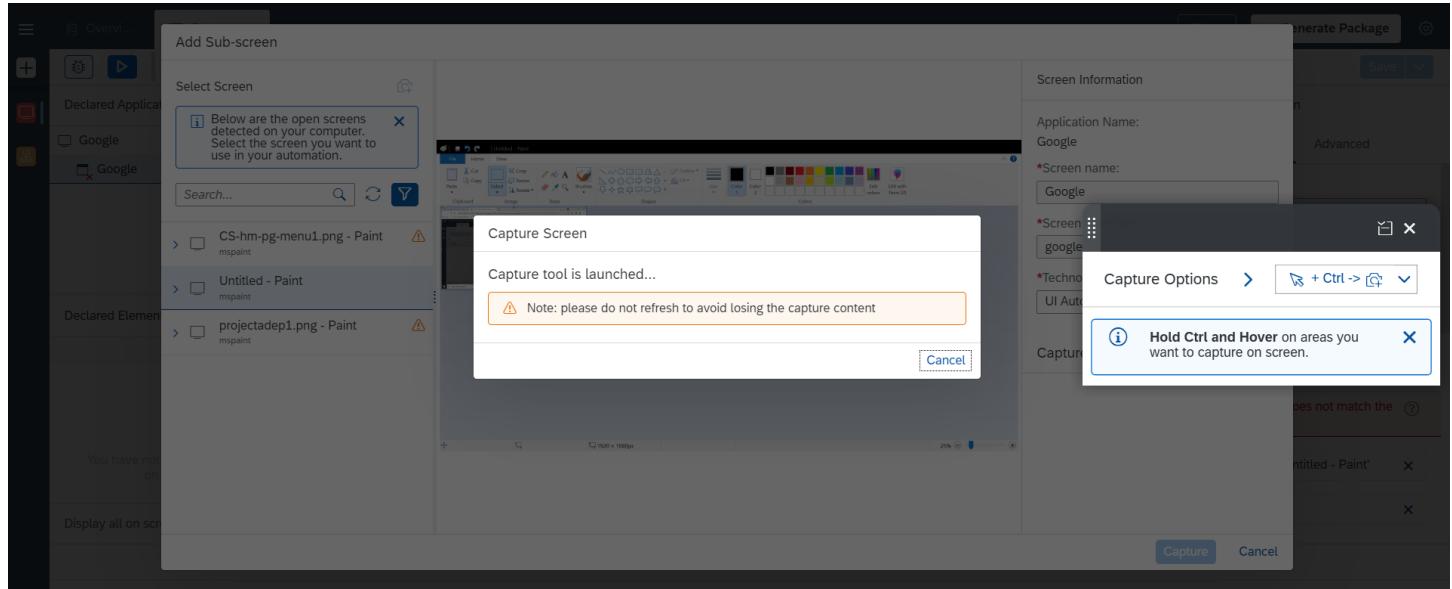


i Note

In the Cloud Studio, the **Capture on Hover** option is available when capturing the first screen of an application, adding capture, and replacing a screen.

You can capture an application by selecting a screen in the Cloud Studio or without selecting a screen. When you click the **Capture on Hover** button, the Cloud Studio is locked and the Capture tool is launched. For example, refer to the following screenshot.



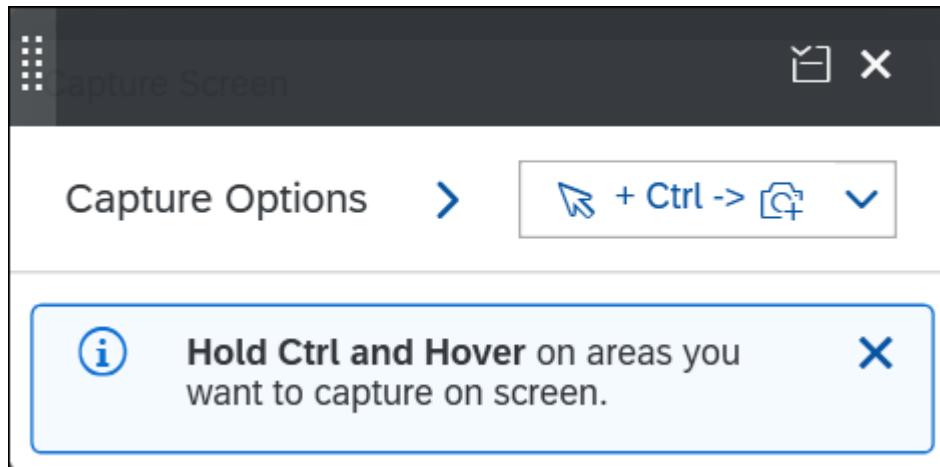


i Note

If you cancel the capturing or Cloud Studio cancels the capturing, then the Cloud Studio is unlocked.

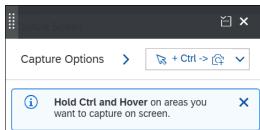
The capture tool will be closed if you refresh or close the Cloud Studio.

Capture Tool



The capture tool includes the two capture options **ctrl + hover** and **shift + hover**. By default, the **ctrl + hover** option is enabled.

The function of these two options are the same. The only difference is the keys (ctrl and shift) you use to capture an application screen.

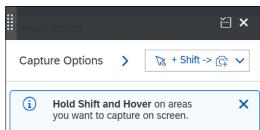
ctrl + hover

Press and hold the **ctrl** key and then hover the mouse over the entire screen, part of the screen, or drop-down menu. The hovered area is highlighted with a red border and then starts capturing the highlighted area.

The screenshot shows a Google search results page for 'batman'. A red border highlights the search results and the sidebar information about Batman, which includes images, a summary, and links to related articles.

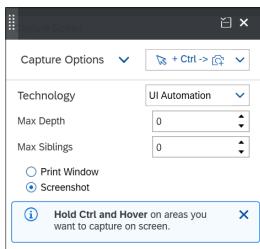
The screenshot shows a Microsoft Excel spreadsheet with the 'Insert' tab selected in the ribbon. A red box highlights the 'Insert Sheet' option under the 'Cells' section of the ribbon.

After successful capture, the capture tool is closed automatically and you will be redirected to the Cloud Studio.

shift + hover

Press and hold the **shift** key and then hover the mouse over the entire screen, part of the screen, or drop-down menu. The hovered area is highlighted with a red border and then starts capturing the highlighted area.

After successful capture, the capture tool is closed automatically and you will be redirected to the Cloud Studio.

More options**Technology:**

This option allows you to change the technology. You can change technology only when you are on the UIAutomation screen. The web can be captured in UIAutomation or in a Web screen. But when you are on a web screen, you cannot change the technology and must capture with the Web technology only.

The screenshot shows the 'Capture Options' dialog with the 'Technology' dropdown set to 'UI Automation'. Other options like 'WEB' are also listed. A tooltip at the bottom says 'Hold Ctrl and Hover on areas you want to capture on screen.'

Disable auto scrolling:

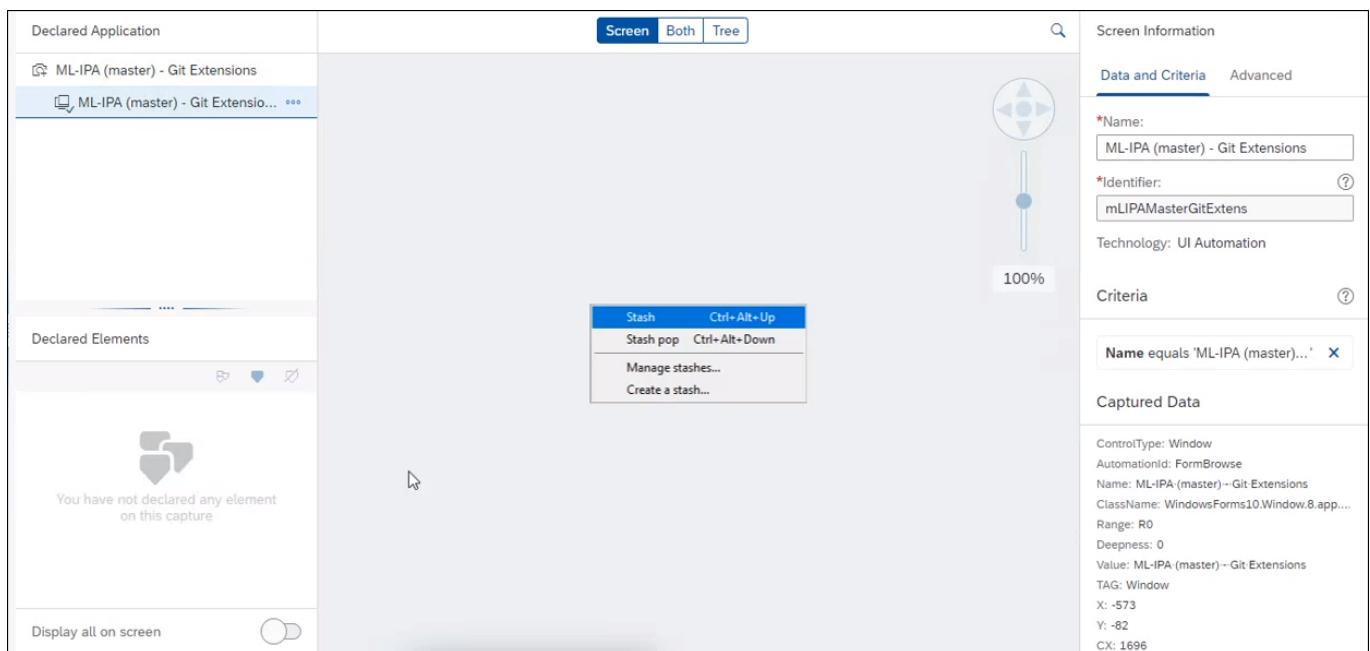
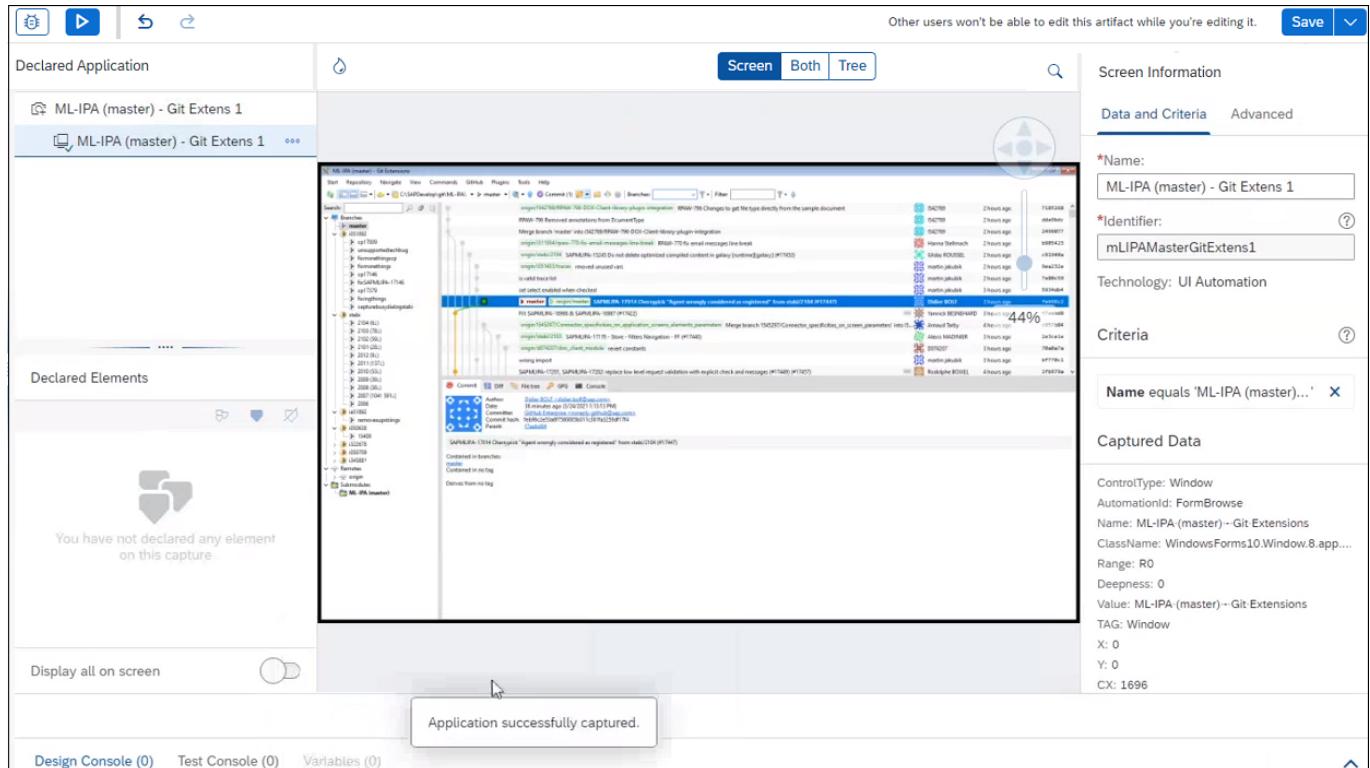
This option disables auto scrolling. It is useful when capturing a pop-up menu.

Sometimes, when you scroll, you may lose focus on the to be captured area. In such cases, use the **Disable auto scrolling** option and then capture the required part or menu of the web page.

To know more details about the capture options, refer to the [Capture Options](#) section.

Procedure

- Click **Capture on Hover**. The Cloud Studio is locked and the capture tool is launched.
- Go to the application you want to capture.
- Press ctrl and then hover on the application screen you want to capture. On the capturing tool, the **Capturing Screen to Studio** message is displayed.
- Go back to the Cloud Studio. You can see the captured screen. For example, refer to the following screenshots.



The screenshot shows the Cloud Studio interface with a declared application named "Book1 - Excel". The application window displays an Excel spreadsheet with a green cell selected. A context menu is open over the cell, with the "Insert" option highlighted. To the right of the application window is a "Screen Information" panel containing details about the application, such as its name, identifier, technology (UI Automation), and captured data.

If required, you can add a new screen or replace the current captured screen using [Capture on Hover](#).

Capture Sub Screen

In some automation processes, it is required to declare a sub screen. Cloud Studio supports sub screen capture for UI Automation only.

This sub screen relationship ensures that the required controls are recognized. However, if not declared correctly, it prevents the required controls from being recognized. A sub screen is declared in the same way as an application screen. The only difference is that it must be declared with the corresponding parent screen.

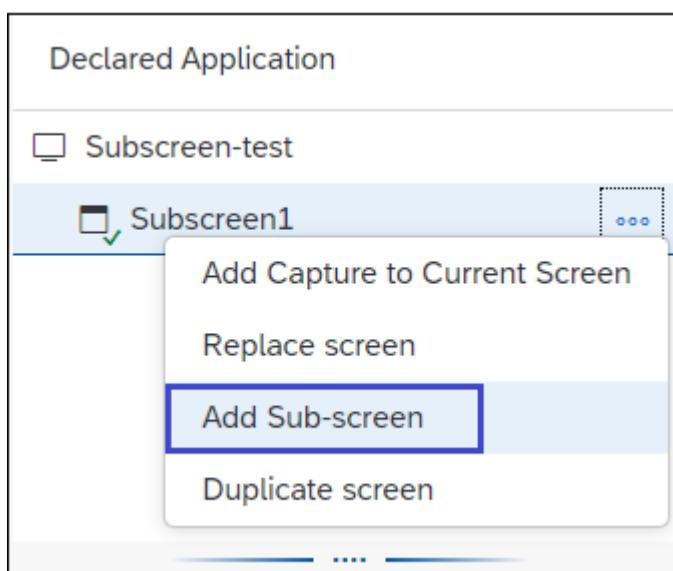
i Note

The SDK version 1.12.0 or higher and the Desktop Agent Version 2.0.14 or higher are supported for sub screens.

Capturing a Sub Screen

1. In Cloud Studio, under **Declared Application**, click the icon of an application screen for which you want to capture the sub screen.

More Options are displayed.



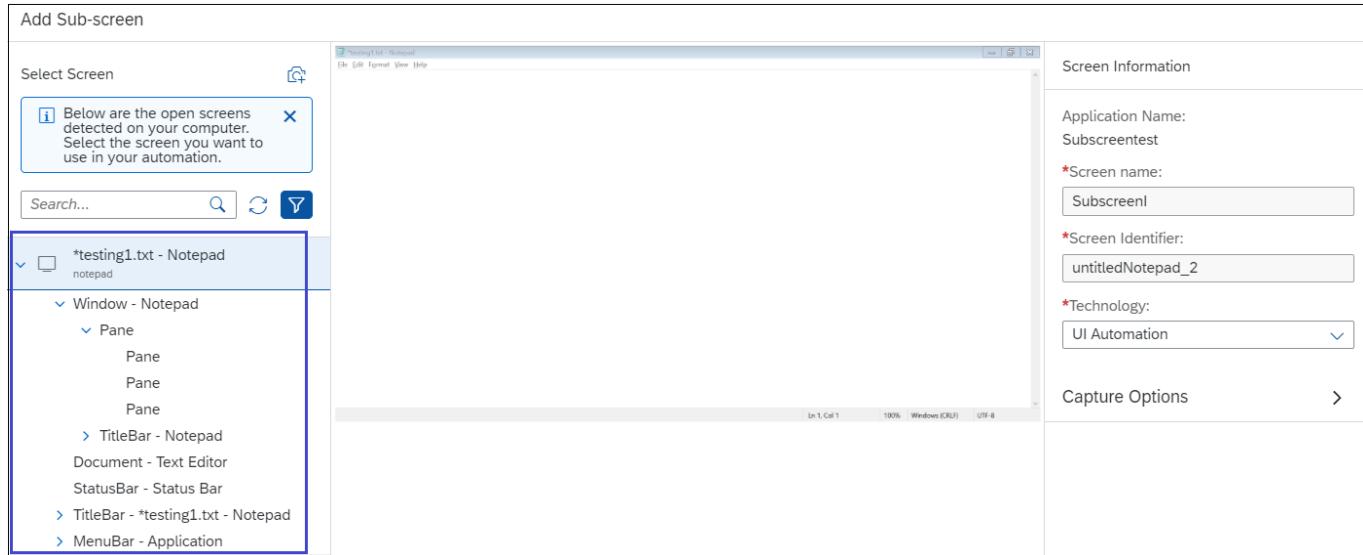
2. Click **Add Sub-screen**.

This is custom documentation. For more information, please visit the [SAP Help Portal](#)

The **Add Sub-screen** pop-up screen is displayed.

- In the recognized screens list from your agent, select the appropriate parent screen.

The screen structure is displayed. For example, refer to the below screenshot.



i Note

In the screen structure, select the screen or an item to preview in the main pane.

- Select the to be captured screen or screen item.
- Update the screen name in the **Screen name** field, if required.
- Click **Capture**.

After successful capture, a preview is displayed in the main pane of Cloud Studio.

All declared screens are listed under **Declared Applications**.

i Note

You can capture multiple sub screens.

You can capture a sub screen of a sub screen.

You can capture child sub screen of a sub screen.

Manage Screen Ordering

In Cloud Studio, you can manage the order of applications, screens, and sub-screens. This helps the technology interfaces to recognize the screens in the right order during runtime.

Prerequisites

Capture at least one application and two screens. For more details about capturing an application and screens, refer to the [Capture an Application](#) section.

In the left panel, under the **Declared Application**, when you mouse hover with an item (application, screen, or sub-screen) at the targeted place, the straight line is highlighted with a dotted line. This helps you to drop an item at an appropriate place.

Declared Application

- Screen Ordering
- Google Screen II
 - Capture I
 - Capture II
 - Capture III**
- DuckDuckGo — Screen I
 - Capture 1
 - Capture 2
 - Capture 3

Declared Elements

i Note

When you reorder a screen, the associated captures and sub screens will also be moved.

Declared Application

- Screen Ordering
- Google Screen II
 - Capture I
 - Capture II
 - Capture III**
- DuckDuckGo — Screen I
 - Capture 1
 - Capture 2
 - Capture 3

Declared Application

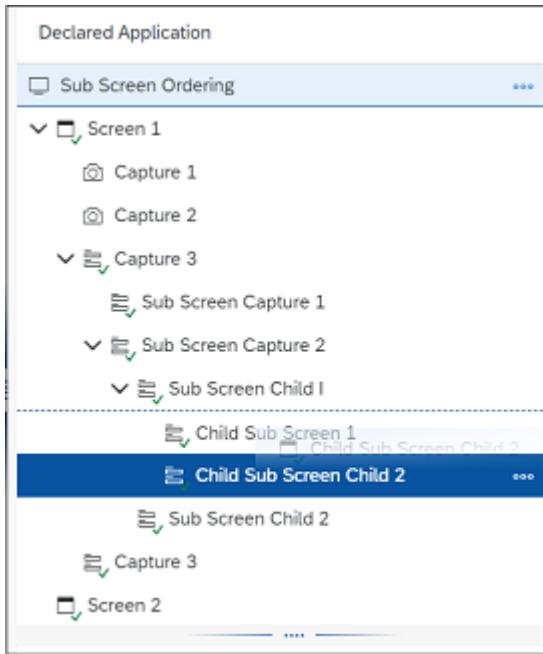
- Screen Ordering
- DuckDuckGo — Screen I
 - Capture 1
 - Capture 2
 - Capture 3
- Google Screen II
 - Capture I
 - Capture III
 - Capture II

You can reorder captures in the same level only under its screen.

Declared Application

- Screen Ordering
- DuckDuckGo — Screen I
 - Capture 1
 - Capture 2
 - Capture 3**
- Google Screen II
 - Capture I
 - Capture III
 - Capture II

You can reorder sub screen only in the same level. Moving to a parent or its child screen is not allowed.



Surface Automation

Image recognition and Optical Character Recognition (OCR) are used to automate user interfaces whose underlying technology (including the document object model or DOM) is not accessible at design time or runtime. This technique is known as surface automation.

The Cloud Studio supports surface automation capture for the **UI Automation** technology. Surface Automation can be used as an alternative when technology connectors are unavailable. Surface automation is the process of creating an automation according to images shown on a screen.

You can use surface automation only when a sub screen is declared in the same way as an application main screen. A sub screen must be declared as a parent screen corresponding to the main application screen.

i Note

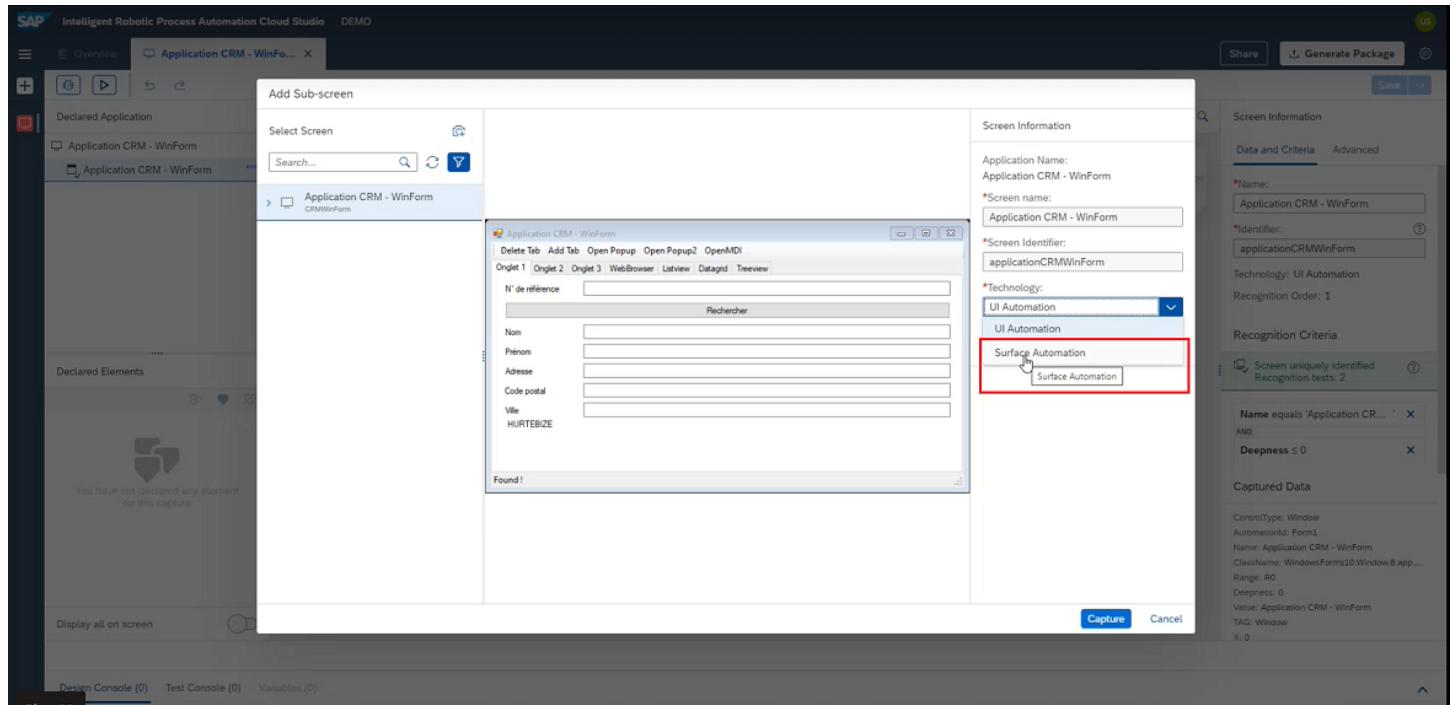
Currently, capturing sub screen of a sub screen (child node) in an application using **Surface Automation** technology is not supported and you cannot select a child node with Surface Automation.

i Note

If the design and runtime scaling factors are different then Surface Automation will not be supported.

Selecting Surface Automation

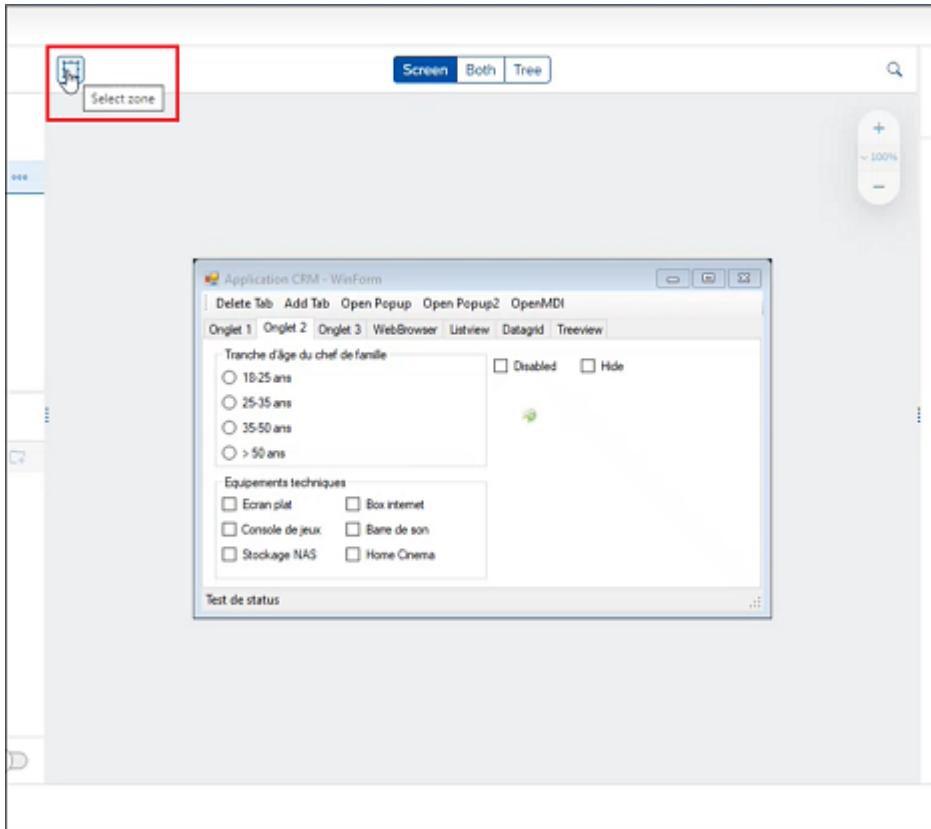
Select a sub screen and then select **Surface Automation** from the **Technology** field in the right panel under **Screen Information**.



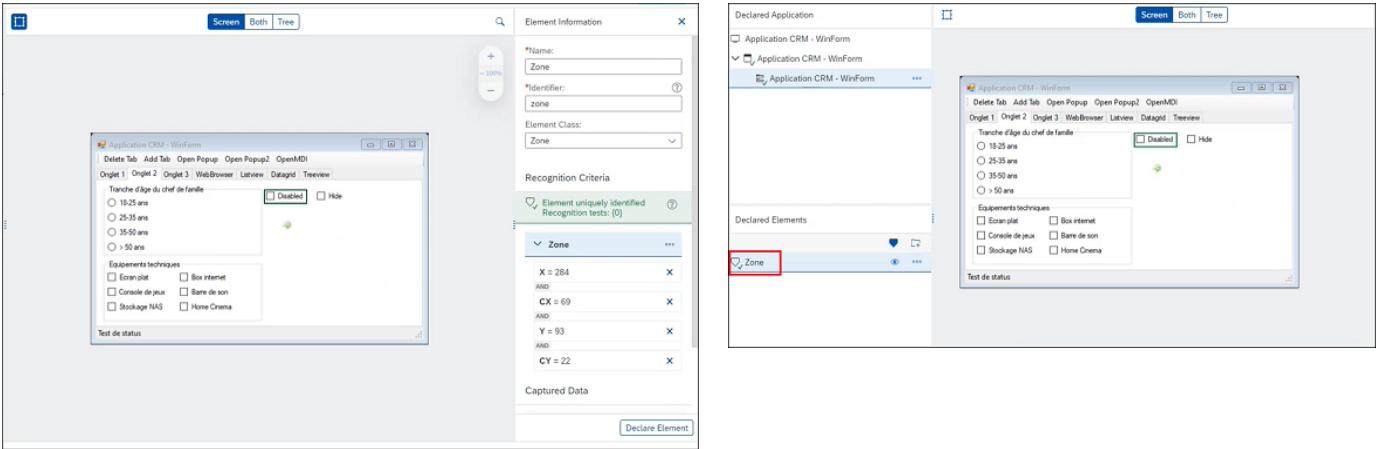
After capturing a sub screen, you can declare the required elements from the screen.

i Note

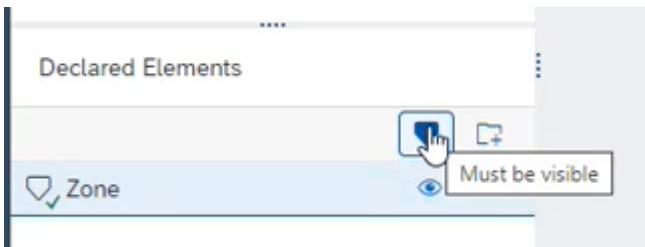
You can declare an element type of zone (any area in the captured screen) using the **Select zone** option.



You must select the **Select zone** option and in the captured screen select the area to be declared as zone and then declare the element. For example, see the below screenshots.



The only declare element icon available for the zone you have selected on the captured screen is **Must be visible** because the zone has no relation to the other elements in the DOM tree structure.



For more information on declaring elements, see [Declare an Element](#).

To view more recognition options, click the **Advanced** tab.

Screen Information

Data and Criteria **Advanced**

Language: (?)

English ▼

OCR Polling Delay (milliseconds): (?)

— 0 +

Trade-off: (?)

Default ▼

Segmentation mode: (?)

Sparse text ▼

Capture mode: (?)

Print Window ▼

Name	Description
Language	<p>You can select the preferred language.</p> <p>i Note</p> <p>Currently supported languages:</p> <ul style="list-style-type: none"> • English (En) • French (Fra) • German (Ger)
OCR Polling Delay (milliseconds)	The time interval in which the screen or image is recognised during runtime.
Trade-off	<p>The OCR recognition accuracy or output quality.</p> <p>Fast - The recognition is fast and accuracy maybe low.</p> <p>Accurate - The recognition maybe slow but accurate.</p> <p>Default - The recognition accuracy between fast and accurate.</p>

Segmentation mode	<p>This is related to recognizing text from an image. The text recognition parameters of Segmentation mode are derived from OCR.</p> <p>Sparse text - You can select this option to recognize more text from an image.</p> <p>Single word - You can select this option to recognize the image as a single word.</p>
Capture Mode	<p>The capture mode helps you to capture various scenarios of the application screens. For more information, see Capture Options.</p>

Related Topics

- [Capture an Application](#)
- [Capture Sub Screen](#)
- [Declare an Application](#)
- [Declare a Screen](#)
- [Declare Elements](#)
- [Capture Options](#)

Declaration Phase

To use applications and its screens and elements in your automation project, you first need to capture and declare them. Declaring means selecting the criteria that the system will apply in order to correctly identify an application, screen or element when running your automation.

An application is composed of one or several screens. As for the screens, they are made of elements. Elements are buttons, input fields, and other controls on a screen.

Declare an Application

Prerequisites

You've captured an application and at least one screen as described in [Capture an Application](#).

Context

To declare an application, you need to set recognition criteria - used to identify the application correctly when your automation runs.

Procedure

1. The application recognition criteria - used to identify the application correctly when your automation runs - are automatically selected during the capture.

Application Information

Data and Criteria Advanced

*Name:

IMDb Application

*Identifier:

iMDbApplication

Framework:

None

Description:

Description

Application is active

Last Update: 2021-09-17 11:02:29

By:

Created: 2021-09-17 11:02:29

By:

Technology: Web

Recognition Criteria



DOMAIN equals 'www.imdb.com' X

If necessary, click a criterion in the **Recognition Criteria** panel to change its properties.

Modify Criterion

*Property:

DOMAIN

*Operator:

equal to

*Value:

www.imdb.com

→ Tip

You cannot declare two different applications with the same criteria (same domain for instance), otherwise only the first one will be recognized, and the second application will be hidden.

2. **Optional:** To add other criteria, click any data displayed in **Captured Data**: the selected data is then automatically added as a criterion.

Captured Data

DOMAIN: www.imdb.com

3. Once you defined your recognition criteria, click **Save**.

Application Criteria

When capturing application with UIAutomation and SapGui technology, the criterias **X**, **Y**, **Range** and **Deepness** follow the absolute and relative rule. When you select a criteria, the absolute value is the value displayed in the **Captured Data** panel, and this value can change to a relative value depending on the context: if the criteria has a parent or is located in a subscreen.

For example, refer to the following screenshot.

The screenshot shows the SAP Quality Center interface with the 'Captured Data' and 'Element Information' panels open.

Captured Data Panel: Displays the 'Advanced Password Generator' application window. The 'Options' section shows 'Min. size: 8' and 'Max. size: 8'. The 'Result' section shows a generated password 'APWG 1.1'. The 'Element Information' panel on the right shows the following details:

- Data and Criteria:** Deepness ≤ 1
- Edit:**
 - Range equals 'R11'
 - AND X = 83
 - AND Y = 97
- Captured Data:**
 - AutomationId: 24494832
 - Name: 8
 - ClassName: Edit
 - Range: R0R1R11
 - Deepness: 2
 - Value: 8
 - TAG: Edit
 - X: 96
 - Y: 153
 - CX: 34
 - CY: 23
 - IsRequiredForForm: false
 - IsKeyboardFocusable: false
 - IsOffscreen: false
 - .IsEnabled: false

A red box highlights the 'Edit' section of the 'Element Information' panel, specifically the range criteria. Another red box highlights the 'X' and 'Y' coordinates in the 'Captured Data' section.

You can see that selected criterias **X**, **Y**, **Range** have different values in both panels. The values in the **Recognition Criteria** panel are the correct value to properly identify your element.

If you select the absolute value, the element cannot be uniquely identified and is marked as invalid as it cannot be detected during the testing of your application.

Properties

When capturing an application using UIAutomation technology, two properties are provided in the captured data of the application:

- EXE: name of the process
- TITLE: title of the main associated window

If the TITLE property is set as a recognition criterion in the Application criteria, the UIAutomation technology ignores it. This criterion is only used if you declare a web screen inside a UIAutomation application. The TITLE property is not available in the recognition criteria of the different screens captured for an application.

Declare a Screen

Prerequisites

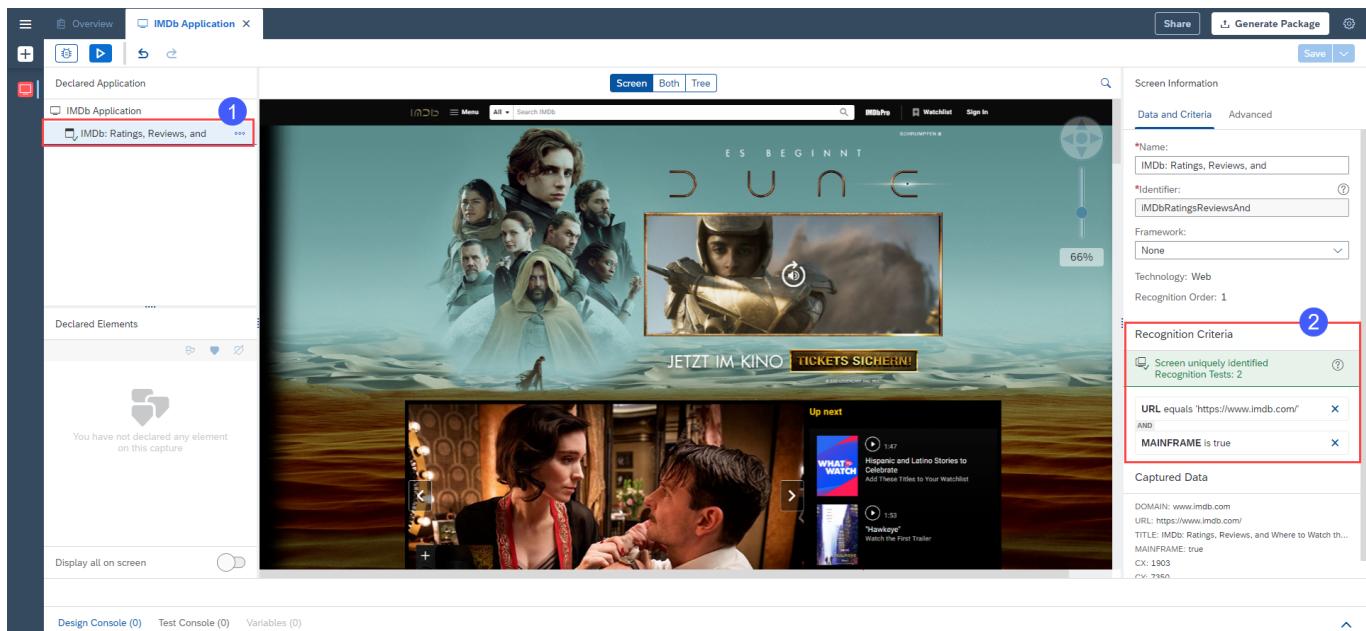
You've captured an application and at least one screen as described in [Capture an Application](#).

Context

To declare a screen, you need to set recognition criteria - used to identify the screen correctly when your automation runs.

Procedure

- Once a screen has been captured, select it in the left-hand navigation panel (1). Some criteria have been automatically selected during the capture (2).



- Optional:** Select to remove a criterion.

Recognition Criteria

Screen uniquely identified
Recognition Tests: 2

URL equals 'https://www.imdb.com/' X

AND

MAINFRAME is true X

Captured Data

DOMAIN: www.imdb.com
 URL: https://www.imdb.com/
 TITLE: IMDb: Ratings, Reviews, and Where to Watch th...
 MAINFRAME: true
 CX: 1903

3. **Optional:** To add other criteria, click any data displayed in **Captured Data**: the selected data is then automatically added as a criterion. In the example below, the captured data "MAINFRAME" and "TITLE" (1) have been assigned as criteria (2).

Recognition Criteria

Screen uniquely identified
Recognition Tests: 2

URL equals 'https://www.imdb.com/' X

AND

MAINFRAME is true X

Captured Data

DOMAIN: www.imdb.com
URL: https://www.imdb.com/
TITLE: IMDb: Ratings, Reviews, and Where to Watch th...
MAINFRAME: true
 CX: 1903

4. **Optional:** Click a criterion in the **Criteria** panel to change its properties.

Modify Criterion X

*Property: URL

*Operator: equal to

*Value: <https://www.imdb.com/>

Apply

5. Once you've defined your recognition criteria, click **Save**.
6. **Optional:** After you've declared one or more elements of the screen, you can define an element as must exist, must not exist or as a collection. Select the three dots after the element name in the picker panel.

Declared Application +

⌚ IMDb: Ratings, Reviews, and

⌚ IMDb: Ratings, Reviews, and

⌚ Search bar

⌚ Search

⌚ Find - IMDb

✓ Must exist
✗ Must not exist
● Is a collection
Delete

Must exist means that the screen is not recognized unless the element or elements defined as must exist are detected.

Must not exist means that the screen is not recognized if an element defined as must not exist is detected.

Is a collection means that the element consists of multiple parts - for example, a list with multiple entries.

Declare Elements

An application is composed of one or several screens. As for the screens, they are made of elements. Elements are buttons, input fields, and other controls on a screen.

Declare Element Icons

Before defining the elements for a capture, go through the declare element icons and its descriptions. This will help you to define element for a capture appropriately.

Icon	Icon Name	Description	Procedure
	Is a collection.	You can define an element as a collection. Is a collection means that the element consists of multiple parts - for example, a list with multiple entries. For more details about declaring an element as a collection refer to the Declare an Element as a Collection section.	Select the desired element and click the  icon to define the element as a collection.
	Must exist.	Must exist means that the screen is not recognized unless the element or elements defined as must exist are detected.	Select the desired element and click the  icon to define an element as Must exist .
	Must not exist.	Must not exist means that the screen is not recognized if an element defined as must not exist is detected.	Select the desired element and click the  icon to define element as Must not exist .
	Link.	This includes various options such as Link to selected element , Unlink , Learn more , and Appears in .	Select the desired element and click the  icon to view available link options.
Link to selected element	Link to selected element.	This will remove the existing link and link to the selected element in the current capture or previous capture.	Select the desired element and click the  icon and then click Link to selected element to remove the existing link and link to the selected element.
Unlink	Unlink.	At any point in time you can manually unlink the element in the capture.	Select the desired element and click the  icon and then click Unlink to unlink the element in the capture.
Appears in	Appears in.	Displays the captures in which the element appears.	Select the desired element and click the  icon and then click Appears in to view the captures in which the element appears.
Learn more	Learn more.	Navigates to the help topic.	Select the desired element and click the  icon and then click Learn more to go to the help topic.
	Ignore Element on this capture.	The element will be ignored in the capture and the element will not be linked when you select the option Link all elements found on screen .	Select the desired element and click the  icon to ignore the element in the capture.
	Do not ignore element on this capture.	The element will not be ignored in the capture.	Select the desired element and click the  icon to ensure that the element is not ignored in the capture.

	More options.	This includes more options such as Add to criteria and Delete .	Select the desired element and click the icon to view more options.
Add to criteria	Add to criteria.	The element is added in the side panel hierarchy as a parent/child element to the criteria of the target element. You can also modify the criteria for this parent/child element.	Select the desired element and click the icon and then click Add to criteria .
Delete	Delete.	This allows you to delete an element.	<p>Select the desired element and click the icon and then click Delete.</p> <p>The warning pop-up message is displayed.</p> <p> Warning</p> <p>Are you sure you want to delete "Text Editor"?</p> <p style="text-align: right;">Delete Cancel</p> <p>Click Delete. The element is deleted from the capture.</p>
Duplicate	Duplicate.		

Capture Operators

Describes the capture operators.

Operators for String

Cloud Studio	Description
equals	The property value is strictly equal to the argument.
not equals to	The property value is strictly NOT equal to the argument.
contains	The property value contains the argument.
doesn't contain	The property value does not contain the argument.
starts with	The property value starts with the argument. For Web-based only (*).
doesn't start with	The property value does not start with the argument. For Web-based only (*)
ends with	The property value ends with the argument. For Web-based only (*)
doesn't end with	The property value does not end with the argument. For Web-based only (*)
is empty	The property is empty. No argument.
is not empty	The property is NOT empty. No argument.
matches	The property value matches the argument. For more information, see Match Operator .
doesn't match	The property value does not match the argument.

<Property> <operator> <argument>: AutomationID equals to 'num4ID'

<Property> <operator> <argument>: AutomationID is empty

Operators for Numbers

Cloud Studio	Description
=	The property value is strictly equal to (=) the argument.
≠	The property value is strictly NOT equal to the argument.
>	The property value is strictly greater than the argument.
<	The property value is strictly less than the argument.
≥	The property value is greater or equal than the argument.
≤	The property value is less or equal than the argument.
≤	The property value is less or equal than the argument.
≠	The property value is not strictly greater than the argument.
≠	The property value is not strictly less than the argument.
≠	The property value is NOT greater or equal than the argument.
≠	The property value is NOT less or equal than the argument.

<Property> <operator> <argument>: CX = 42

Other Operators (Boolean and List)

- A Boolean property has one value that can be selected or unselected.
- An enumeration is a collection (of Boolean).

Cloud Studio	Description
Selected or Unselected Checkbox	The property value (a Boolean) is selected or not.
Selected or Unselected Checkboxes	The property values (Boolean) are selected or not.

<Property> <operator> <argument>: isFourtyTwoTheAnswer = true

Match Operator

The **Match** capture operator allows you to find a string on a screen.

You must enter a regular expression in the **Value** field. This regular expression must respect the POSIX Basic Regular Expression syntax.

You must enter your regular expression without adding any specific delimiting characters.

Please note that the regular expression pattern of the **Match** operator is case-insensitive.

Example 1: calculat.r

For example, you can enter `calculat.r` in the **Value** field to detect all buttons that have a name containing these characters.

Modify Criterion X

*Property: Name Name

*Operator: match match

*Value: calculat.r calculat.r

Apply

Screen Both Tree

Element Information

*Name: Minimize
Identifier: minimize
Element Class: Button

Recognition Criteria

3 elements are identified.
Recognition tests (first element): 5

Name match 'calculat.r'

Captured Data

AutomationId: Minimize
Name: Minimize Calculator
Range: R0R0R1
Deepness: 2
Value: Minimize Calculator
TAG: Button
X: 272
Y: 1

You can notice that three buttons with a matching name have been detected.

Example 2: \d

You can also enter `\d` to find text elements that have a value containing digits.

Modify Criterion

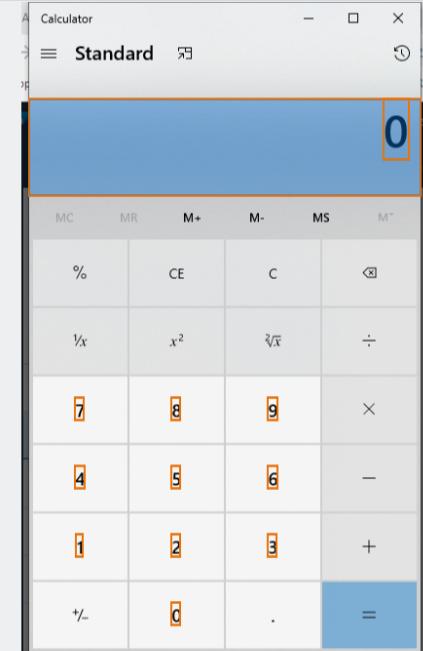


*Property:

*Operator:

*Value:

Apply



Screen
Both
Tree

+ <100%
-

Text
Group
Custom
Pane
TitleBar

Display Conflicts

Element Information
Save

*Name:	<input type="text" value="CalculatorResults"/>
*Identifier:	<input type="text" value="calculatorResults"/>
Element Class:	<input type="text" value="Text"/>

! 12 elements are identified. ?

Value match '\d'

Recognition Criteria
X

Text

Value match '\d'

Captured Data
Declare Element

AutomationId: CalculatorResults
Name: Display is 0
Range: R0R1R1R1
Depth: 3
Value: Display is 0
TAG: Text
X: 8
Y: 100

You can notice that 12 texts with a matching value have been detected.

Declare an Element

Prerequisites

You've captured an application and at least one screen as described in [Capture an Application](#). You've also declared a screen as explained in [Declare a Screen](#).

Context

To declare elements on a screen, you need to set recognition criteria - used to identify the elements correctly when your automation runs.

Procedure

- Once a screen has been captured and declared, select it in the left-hand navigation panel.
- Click an element to select it.

A green border is shown around the selected element - in this example, it's the search bar.

The screenshot shows the SAP Application Management interface with the 'IMDB Application' selected in the left navigation. The main view displays a movie poster for 'Bill & Ted Face the Music'. The top navigation bar includes 'IMDb', 'Menu', 'All', and a search bar labeled 'Search IMDb'. A green border highlights the search bar. To the right, there's an 'Element Information' panel with fields for 'Name' (set to 'INPUT'), 'Identifier' (set to 'INPUT'), 'Item Class' (set to 'Edit'), and a 'Criteria' section containing a note to add a criterion. Below this is a 'Captured Data' section with detailed HTML code for the search input field, and a 'Declare Element' button at the bottom.

→ Tip

If you find it difficult to highlight the right screen element in the image view, choose **Tree** and select it in the tree view instead.

- In the **Element Information** side panel, enter a name and an identifier for the element.

The **Element Class** is automatically detected. *Edit* means the element is editable - it's an input field.

Element Information

***Name:**

Search bar

***Identifier:** [?](#)

searchBar

Item Class

Edit

- Optional:** Choose the criteria you want to assign to the element.

- Remove a criterion via [X](#).

Criteria

TAG is INPUT	
--------------	--

b. To add further criteria, click any data displayed in **Captured Data**.

The selected data is then automatically added as a criterion. In the example below, the captured data "id" and "type" have been assigned as criteria.

Criteria

id is suggestion-search	
type is text	

Captured Data



```

TAG:INPUT
range:0
name:q
class:imdb-header-search__input
GVtrp0cCs2HZCo7E2L5UU react-
autosuggest__input
id:suggestion-search
aria-controls:react-autowhatever-1
aria-autocomplete:list
type:text
placeholder:Search IMDb
autocomplete:off
autocapitalize:off
autocomplete:off
value:

```

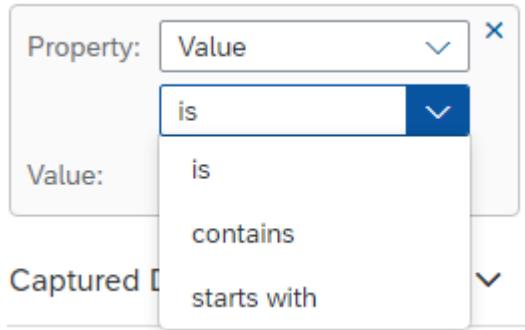
c. **Optional:** To add another value to a criterion, click again in **Captured Data** the data you already assigned as a criterion. This other value is displayed in the input panel below the operator **or**.

Criteria									
▼ INPUT <table border="1"> <tr> <td>id</td> <td></td> </tr> <tr> <td>is 'suggestion-search'</td> <td></td> </tr> <tr> <td>OR</td> <td></td> </tr> <tr> <td>is 'search bar'</td> <td></td> </tr> </table>		id		is 'suggestion-search'		OR		is 'search bar'	
id									
is 'suggestion-search'									
OR									
is 'search bar'									

Then click the criterion to change its value (see next step).

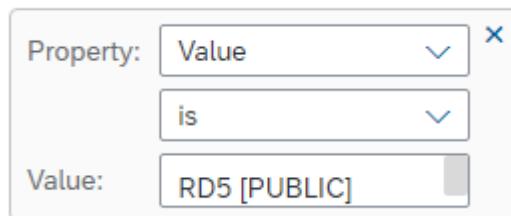
5. **Optional:** Click a criterion in the **Recognition Criteria** panel to change its properties.

A field value property, for example can be changed by choosing a different comparator instead of "equal to" and altering the value.



❖ Example

Change "Value equal to Search" to "Value starts with Search".



6. Once you've added and defined your recognition criteria, click **Declare Element**.

i Note

By default, if you create a new capture, all declared element of the screen are linked to the new capture. But, if you declare a new element, it is linked only to the currently displayed capture. So, after declaring new element, you must link it to other captures, if required.

i Note

You can drag and drop the declared elements within the left-side panel. This functionality allows you to reorganize your declared elements.

When an element has been captured, it is displayed in the **Declared Elements** panel (1). When you select the element on the left-side hand panel, it is automatically highlighted on the screen (2).

7. Save your work.

i Note

Delete an element by choosing the three dots after the element's name in the picker panel and selecting **Delete**.

Duplicate Declared Element

In the Cloud Studio, you can duplicate the declared elements. The duplicate element represents the characteristics of the original element.

Prerequisites

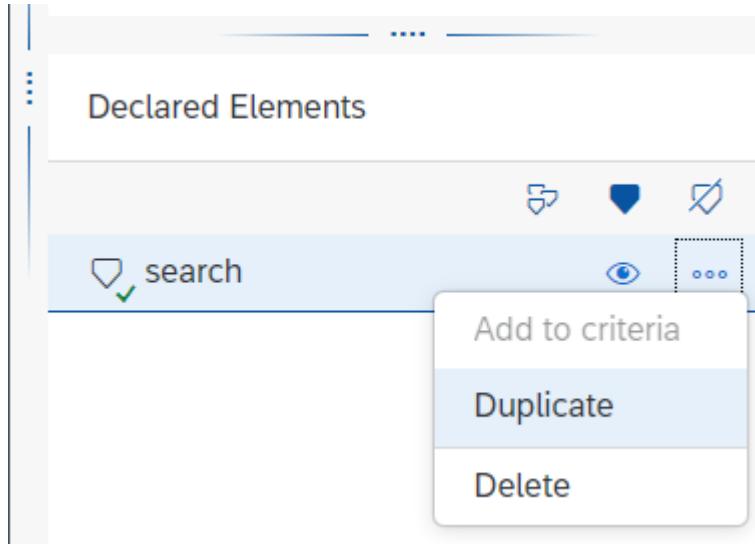
You've declared at least one element as described in [Declare an Element](#).

Context

Procedure

1. In the IRPA Cloud Studio, under **Declared Application**, select the application screen for which you want to declare a duplicate element.
2. Under **Declared Elements**, click the  icon of a element.

More Options are displayed.



3. Click **Duplicate**.

The **Duplicate Element** pop-up screen is displayed.

? Duplicate Element

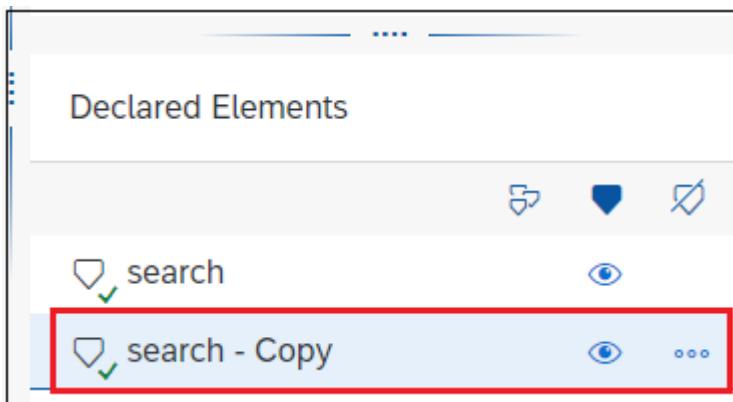
*Name	search - Copy
*Identifier	searchCopy
Duplicate Cancel	

i Note

If required, you can update the name of an element.

4. Click **Duplicate**.

The duplicated element is created and displayed under **Declared Elements**.



Declare a Parent as a Collection

Context

You can set the parent element of a declared element as a collection for an easier pattern identification.

Procedure

1. Once you have selected the parent element and its child, remove all criteria that have been automatically selected by clicking the □ icon.

Element Information X

*Name:
Cruella (2021)

*Identifier: ?
cruella2021

Element Class:
Hyperlink ▼

Recognition Criteria

▽ Element uniquely identified ?
Recognition Tests: 490

▼ TD ...

Text equals 'Cruella (2021)' X

▼ A ...

Text equals 'Cruella' X

Captured Data

Text: Cruella (2021)
class: result_text
nth-child-tag: 1
contentEditable: inherit

Declare Element

2. On the **Criteria** panel, next to the parent element, click the X icon.

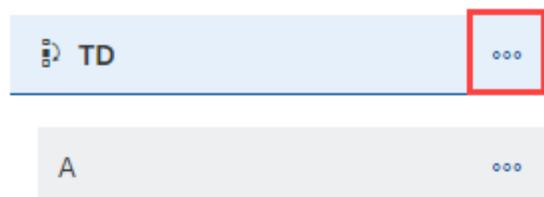
Element Information X***Name:*****Identifier:** (?)

Element Class:

 ▼

Recognition Criteria

⚠ 12 elements are identified. (?)
Recognition Tests (1st element):
487



Captured Data

Text: Cruella (2021)
class: result_text
nth-child-tag: 1
contentEditable: inherit
style: [object CSSStyleDeclaration]

Declare Element

3. Click **Set as collection**.

The parent element of the declared element is now set as a collection. As a consequence, the declared element has also become a collection.

The parent element of the declared element is now set as a collection. As a consequence, the declared element has also become a collection.

i Note

Once you have set an element as a collection, you can undo this action by clicking the icon and selecting **Set as single element**.

Deepness Criteria

After you have captured the application and screens, identify the relevant UI elements. Elements are buttons, input fields, and other controls on a screen. For more details on declaring an element, refer to the [Declare an Element](#) section.

By default, the **Deepness** criteria is added when you select an element. This is applicable for UI Automation and SAP GUI as these technologies use depth capture data. Deepness criteria can only be used with the less than or equals to (\leq) operator.

i Note

You cannot edit the **Operator** of Deepness criteria.

Labeled by

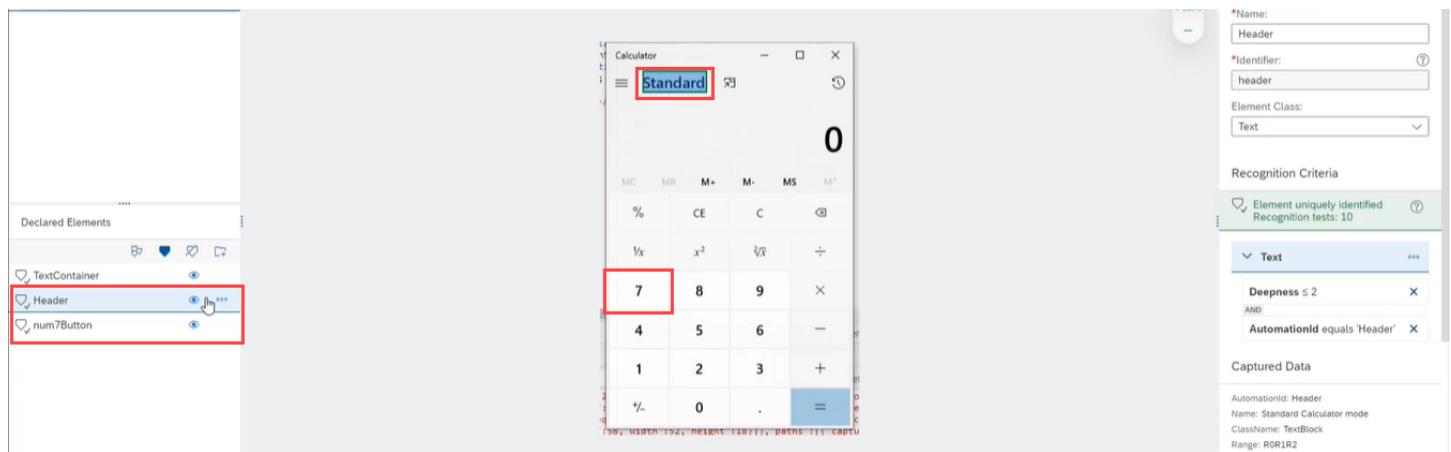
The **Labeled by** parameter allows you to declare dynamic elements on a captured screen by indicating their related position with another static element.

i Note

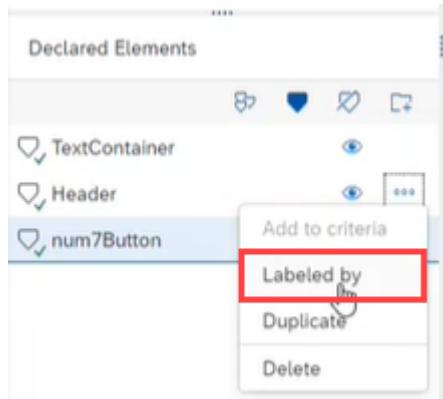
This feature is supported by all technologies except Web applications.

With the **Labeled by** parameter, you have the possibility to declare a label element as criteria to a target element, meaning that the identifier of the label element is saved as recognition criteria to the target element.

For example in the following screenshot, the header and the button number seven are declared elements on the calculator..



Now we select the button seven element and click on the **Labeled by** option of the header element.



In the **Recognition Criteria**, the header element has been added as label to the button seven element.

The screenshot shows the 'Recognition Criteria' panel in SAP Cloud Studio. At the top, there's a green header bar with a checkmark icon and the text 'Element uniquely identified' and 'Recognition tests: 46'. Below this, a tree structure is displayed under a 'Button' node. The first node is labeled 'Header' and has a red border around it. The tree continues with 'AND', 'Deepness ≤ 4', another 'AND', and 'AutomationId equals 'num7...''. Each node has a small 'X' icon to its right.

Group Declared Elements

In the Cloud Studio, you can group your declared elements. Grouping your declared elements allows you to facilitate their organization and management.

Prerequisites

You've declared several elements as described in [Declare an Element](#).

Procedure

1. On the left-hand side panel, under **Declared Elements**, select the elements you want to group by keeping **Ctrl** pressed and clicking each element.
2. Click to group these elements.

The screenshot shows the 'Declared Elements' panel in SAP Cloud Studio. At the top, there are several icons: a download arrow, a blue heart, a shield, and a group icon (a folder with a plus sign) which is highlighted with a red box. Below these are two sections: 'Names' (with a plus sign icon) and 'Data'. The 'Data' section is expanded and contains four items: 'Company', 'Contact', 'Country', and 'City', each with a small eye icon to its right. The entire 'Data' section is also highlighted with a red box.

i Note

You can drag and drop elements inside and outside of a group, or into another group.

Note that if you remove the last element contained in a group, the group will be deleted.

i Note

You can drag and drop a group up or down in the list of declared elements on the **Declared Elements** left-hand side panel.

3. Optional: Change the **Name** of the group on the **Group Details** right-hand side panel.

Within the declared elements of a same screen, a group must have a unique name.

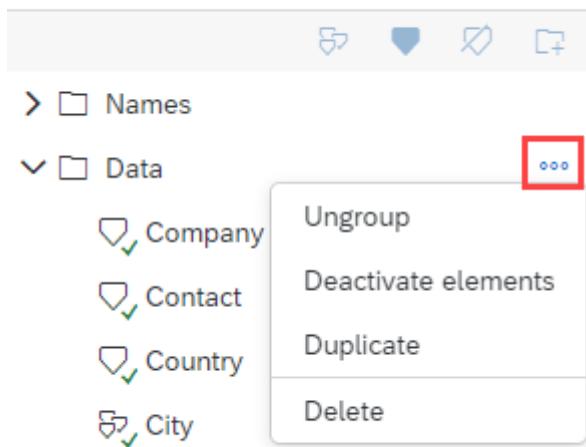
4. Optional: Click the **□** icon.

You can:

- **Ungroup** elements of a group.
- **Deactivate elements** if you want to ignore a group and the elements it contains.
- **Duplicate** a group and its elements.
- **Delete** a group.

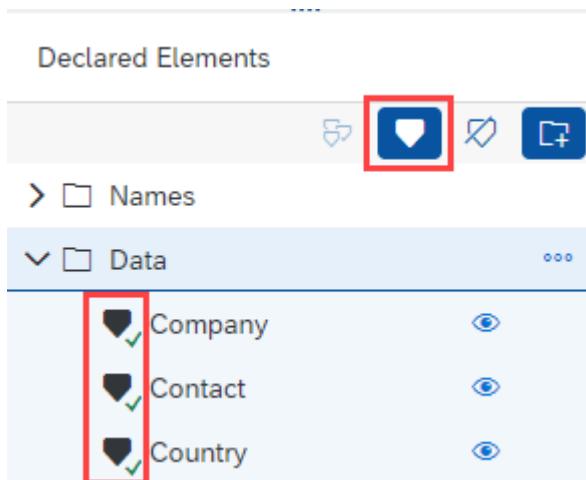
If you click **Delete**, the **Delete group** panel will appear, allowing you to either **Delete** or **Ungroup** the group.

Declared Elements



5. Optional: Apply the same settings to all elements in the same group.

While selecting a group, click the **Must exist** and **Must not exist** buttons to apply these settings to all elements of the group.



Declare an Element as a Collection

A collection consists of multiple or list of elements of same type. For example, a list of titles, URLs, summary, and many more.

Prerequisites

You must capture an application and at least one screen. To know more details about capturing an application screen refer to the [Capture an Application](#).

Context

You can declare an element as a collection using the **Is a Collection** button.

Declare an Element as a Collection Using the Is a Collection Button

1. Declare an element as described in [Declare an Element](#).
2. Under **Declared Elements** (1) select the element (2) to be declared as collection and then click **Is a collection** button (3).

The screenshot shows the SAP Intelligent Robotic Process Automation Cloud Studio interface. In the left sidebar, under 'Applications', 'Batman' is selected. In the main area, the 'Declared Elements' section is open, showing a list of elements. One element, 'Title', is highlighted with a red border. Next to it is a blue circular button with a white icon, which is the 'Is a collection' button. The central panel, labeled 'Screen', displays a search result for 'Batman' with several video thumbnails. One of the video cards for 'THE BATMAN Official Trailer #1 (NEW 2022)' has a green border around its title and thumbnail, indicating it is selected. The right panel, 'Element Information', shows details for the selected 'Title' element, including its name, identifier, element class (Hyperlink), and criteria. A feedback section indicates that the element cannot be uniquely identified.

After the element is declared as a collection, the **□** symbol is displayed next to the declared element and the **Is a collection** button turns blue. By default, all related or similar elements of a screen are added as a collection.

This screenshot shows the same SAP interface after the 'Title' element has been declared as a collection. The 'Title' element in the 'Declared Elements' list now has a green border and a blue background, indicating it is selected. The 'Is a collection' button next to it is also blue. The central 'Screen' panel and the right 'Element Information' panel remain largely the same, showing the search results and element details respectively.

When an element is selected under **Declared Elements**, you can observe that the matched element gets highlighted with green border (2) and the selected element gets highlighted with blue background (3) in the **Screen** central panel. In the **Tree** view, the selected element gets highlighted with a blue background and a blue line at the bottom (4) of the

element. You can also find one green dot (4) which refers to the corresponding initial element that you select under **Declared Elements** and then build all **Criteria** accordingly.

i Note

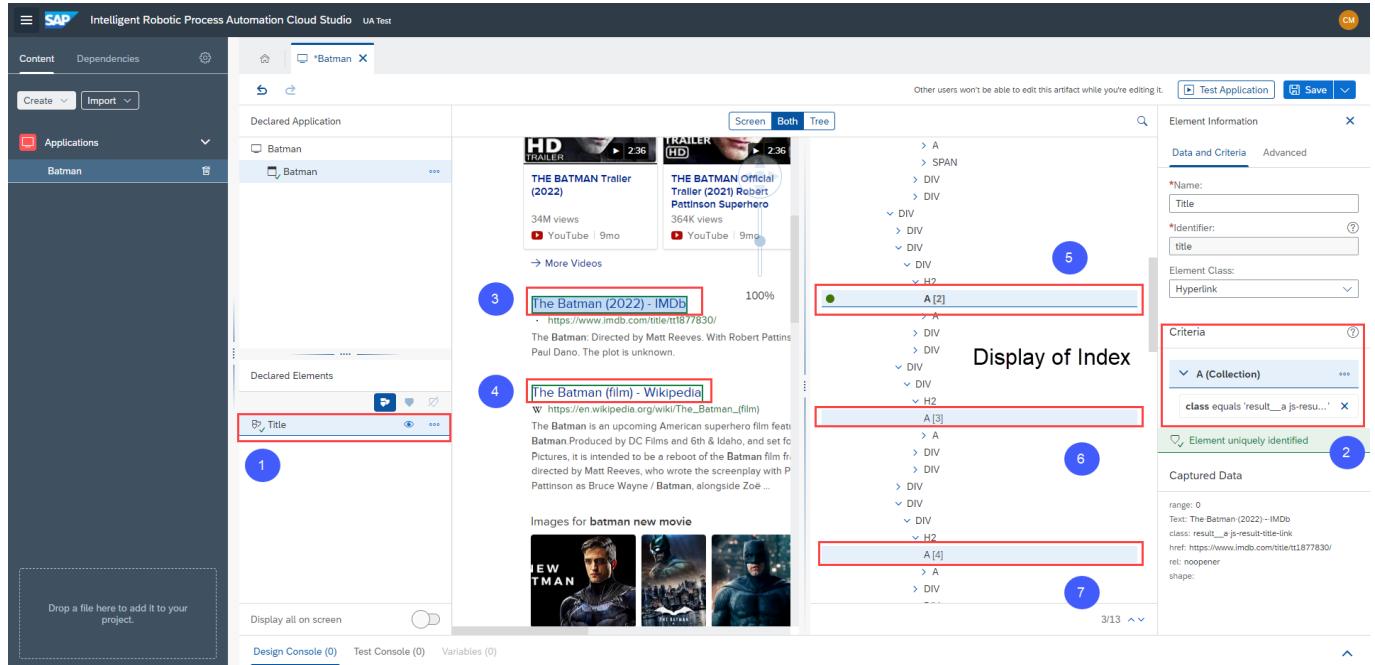
In the **Tree** view, you can move from one matching element to another the arrows (5) at the bottom of the screen.

If the selected element under **Declare Elements** is not declared as a collection and it matches multiple elements, the selected element gets highlighted with a blue background (1). All other matched elements that have conflict with the selected element get highlighted with red border (2) in the **Screen** central panel. In the **Tree** view, the selected element gets highlighted with a blue background and a blue line at the bottom (3) of the element and all other elements that have conflict with the selected element become red font (4).

i Note

The **Display Conflicts** toggle button appears, and you can turn it on to expand the whole tree. If you turn this toggle button off, all child elements of the tree collapse.

When an element (1) is declared under **Declared Elements** and a class (2) of the element is selected as a **Criteria**, you can observe that the selected element gets highlighted with a blue background and a green border (3) and the index of all matched elements gets highlighted with a green border (4) in the **Screen** central panel. In the DOM **Tree** view, the selected element gets highlighted with a blue background and a blue line at the bottom (5) and the index of all matched element gets highlighted with a blue background (6)(7).



i Note

By default, if you create a new capture, all declared element of the screen are linked to the new capture. But, if you declare a new element, it is linked only to the currently displayed capture. So, after declaring new element, you must link it to other captures, if required.

Advanced Declaration

During the declaration phase, you can't always identify elements with a simple declaration of criteria. You identify these elements with the structure of the page: you add hierarchical elements (parent/child) to their recognition criteria.

Prerequisites

You've captured an application and at least one screen as described in [Capture an Application](#). You've also declared a screen and an element as explained in [Declare a Screen](#) and [Declare an Element](#).

1. In the **Screen** central panel, click the target element of the captured screen.
2. Switch to the **Both** or **Tree** view.
3. In the tree, hover over the element you want to add as a parent/child element and click .

i Note

Similar to the concept of *ancestor* from the V1 release in the **Desktop Studio**, you can also select an element that has already been declared and add this element as parent/child element.

In the **Declared Elements** section, hover over the element you want to add as a parent/child and click .

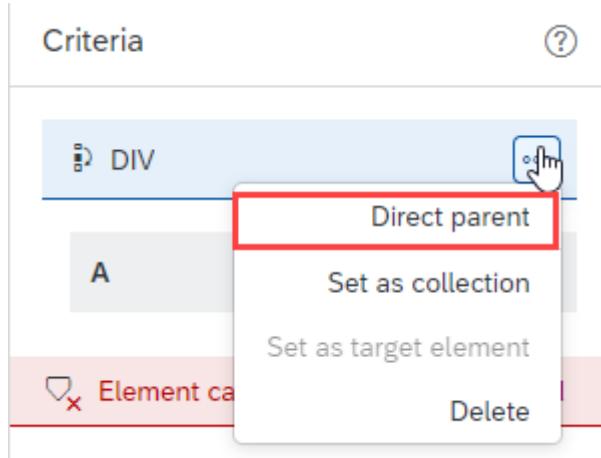
i Note

You can't add as parent/child an element that is not a parent/child in the hierarchy tree.

4. Click **Add to criteria**.

The element is added in the side panel hierarchy as a parent/child element to the criteria of the target element. You can now also modify the criteria for this parent/child element.

5. To specify that a hierarchical element must be a direct parent of the target element, click on this hierarchical element in the **Criteria** panel and select **Direct parent**.

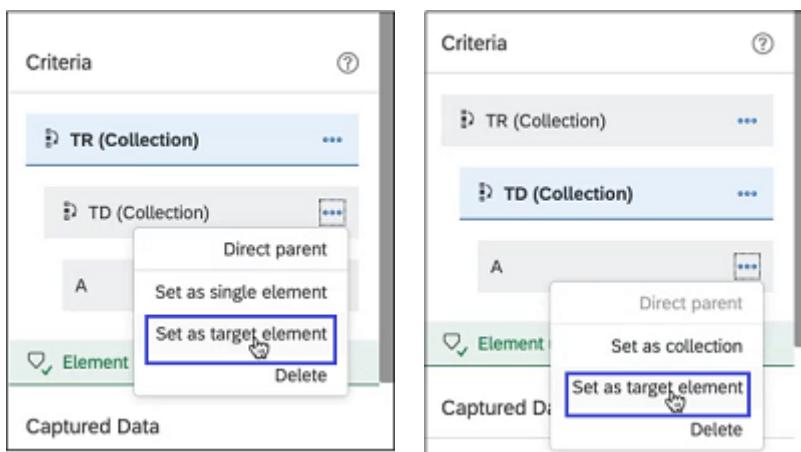


Set as Target Element

You can set a target element from the screen **Criteria**.

You can identify elements in the complex structure of a screen and then define hierarchy for the elements (parent/child) to their recognition criteria. This involves one or more elements to be recognised, where one of these elements is the element to be targeted.

You can edit the defined hierarchy for the elements (parent/child).



Related Topics

- [Declaration Phase](#)
- [Declare Elements](#)
- [Declare an Element as a Collection](#)

Recognition Tests

In the Cloud Studio, recognition tests are a performance indicator used to identify the number of elements, screens and sub screens associated with a captured application screen.

To view possible recognition tests, you must select a captured screen or declared element. In the following screenshot, the possible recognition tests are represented with a numeric value.

The screenshot shows the SAP Cloud Studio interface with the 'Calculatrice' application open. The 'Element Information' panel on the right displays the following details for the selected 'Window' element:

- Name:** Calculatrice
- Identifier:** calculatrice
- Element Class:** Window

The 'Recognition Criteria' section shows one criterion: "Element uniquely identified" with a value of 1. The 'Captured Data' section shows the following details for the window element:

- Name: Calculatrice
- ClassName: ApplicationFrameWindow
- Range: R0
- Depth: 0
- Value: Calculatrice
- TAG: Window
- X: 0
- Y: 0

i Note

The possible recognition tests may vary based on the declared element and its recognition criteria.

Recognition Tests Details

The recognition of an element depends on the position of the element in the DOM tree. If an element is not uniquely identified in a screen, the recognition tests will only be performed on the first element that appears on the list. The recognition tests can display a warning (in orange) when you select an element not uniquely identified from a list of multiple elements with the same criteria.

If you declare an element not uniquely defined as a collection, the sum of tests required to identify all elements of the collection will be displayed.

For example, refer to the following screenshots.

The screenshot shows the SAP Fiori Modeler interface for a 'Calculatrice' application. The main view displays a calculator screen with a numeric keypad and various function keys like MC, MR, M+, M-, MS, and M*. A red box highlights the 'Minimize' button in the top right corner of the calculator window. On the left, the 'Declared Elements' sidebar shows a 'Minimize' entry with a blue icon. The 'Element Information' panel on the right details a recognition test for the 'Minimize' button, specifying a deepness of 2 and an AutomationId of 'Minimize'. The 'Recognition Criteria' section indicates that 2 elements were identified. The bottom right corner of the interface shows a 'Captured Data' section with specific details about the 'Minimize' element.

This screenshot shows the SAP Fiori Modeler interface again, focusing on the 'Minimize' element. A red box highlights the 'Minimize' entry in the 'Declared Elements' sidebar. The 'Element Information' panel shows a recognition test for the 'Minimize' button with a deepness of 2 and an AutomationId of 'Minimize'. The 'Recognition Criteria' section indicates that 1 element was identified, labeled as 'List uniquely identified'. The bottom right corner shows captured data for the 'Minimize' element.

Parent/child element

For a parent/child element, the recognition tests works together with the deepness criteria. With the deepness criteria, the number of elements to test on the application screen is reduced and the performance is optimized. For more information , see [Deepness Criteria](#).

For example, when you select a child element with a deepness criteria less than or equal two ≤ 2 , the recognition tests identify all the potential parent elements and once a parent element is selected, the recognition tests only count the elements in between the child and parent elements you have selected.

In the following screenshot, the parent/child elements are **Group** (parent) and **Button** (child) . When you click on the **Group** element with a deepness criteria of ≤ 3 , the recognition tests identify 14 elements.

The screenshot shows the SAP Fiori Launchpad interface. On the left is a screenshot of a calculator application window titled "Calculatrice Standard". The calculator has a numeric keypad from 0 to 9, arithmetic operators (+, -, ×, ÷), and various function keys like MC, MR, M+, M-, MS, and M-. The display shows "0". On the right is the "Element Information" panel. It includes tabs for "Screen", "Both", and "Tree". The "Tree" tab is selected, showing a hierarchical tree of UI elements. A specific node, a "Group" element, is highlighted with a red border. In the "Recognition Criteria" section, there is a green checkmark next to the text "Element uniquely identified" with the note "Recognition tests: 14". The "Captured Data" section shows details about the selected element, including its AutomationId, Name, ClassName, Range, Deepness, Value, TAG, and X coordinate. At the bottom right of the panel is a "Declare Element" button.

Variables (0)

Now, when you click on the **Button** element with a deepness criteria of ≤ 4 , the recognition tests only count the four **Button** elements with the same deepness criteria, and uniquely identify 18 elements.

This screenshot is similar to the one above, showing the SAP Fiori Launchpad interface with a calculator application. The right side shows the element tree and configuration details. However, in this version, a specific "Button" element within the "Group" node is highlighted with a red border. In the "Recognition Criteria" section, there is a green checkmark next to the text "Element uniquely identified" with the note "Recognition tests: 18". The "Captured Data" section shows the same details as the previous screenshot. The "Declare Element" button is also present at the bottom right.

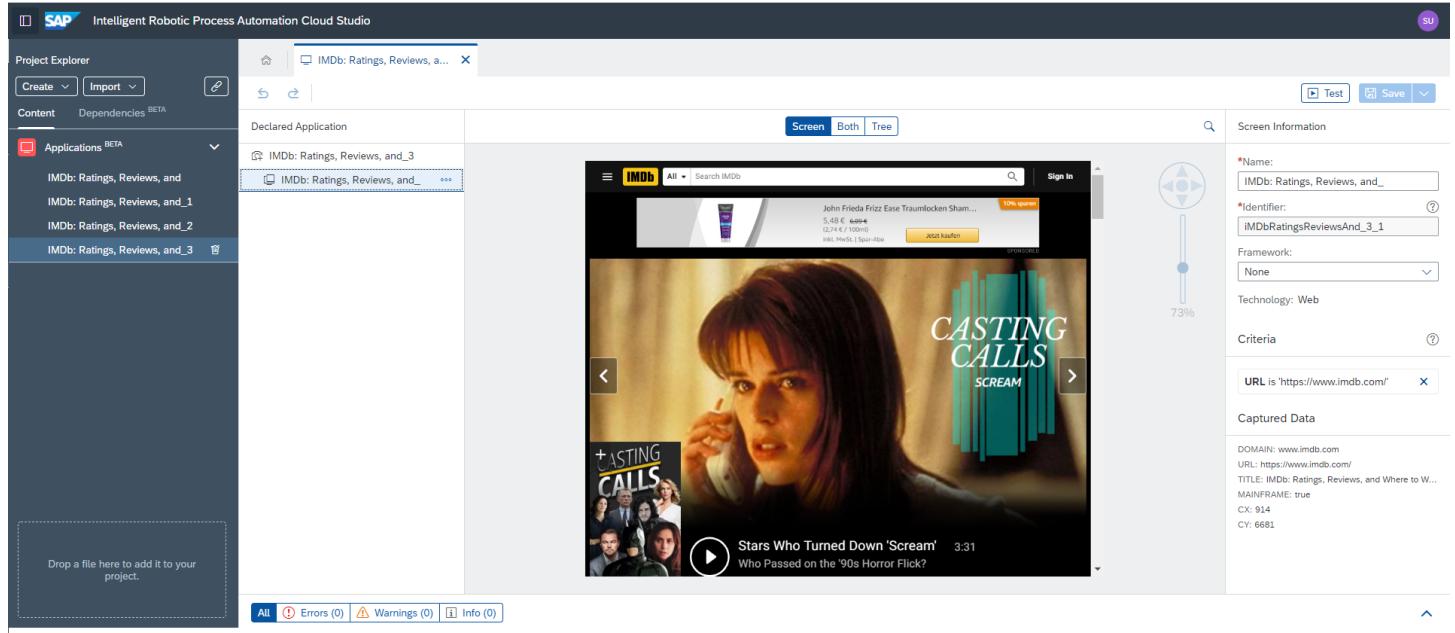
Search Option

Search

The search option allows you to find the terms (text, buttons, and input fields) in the screens and tree control types.

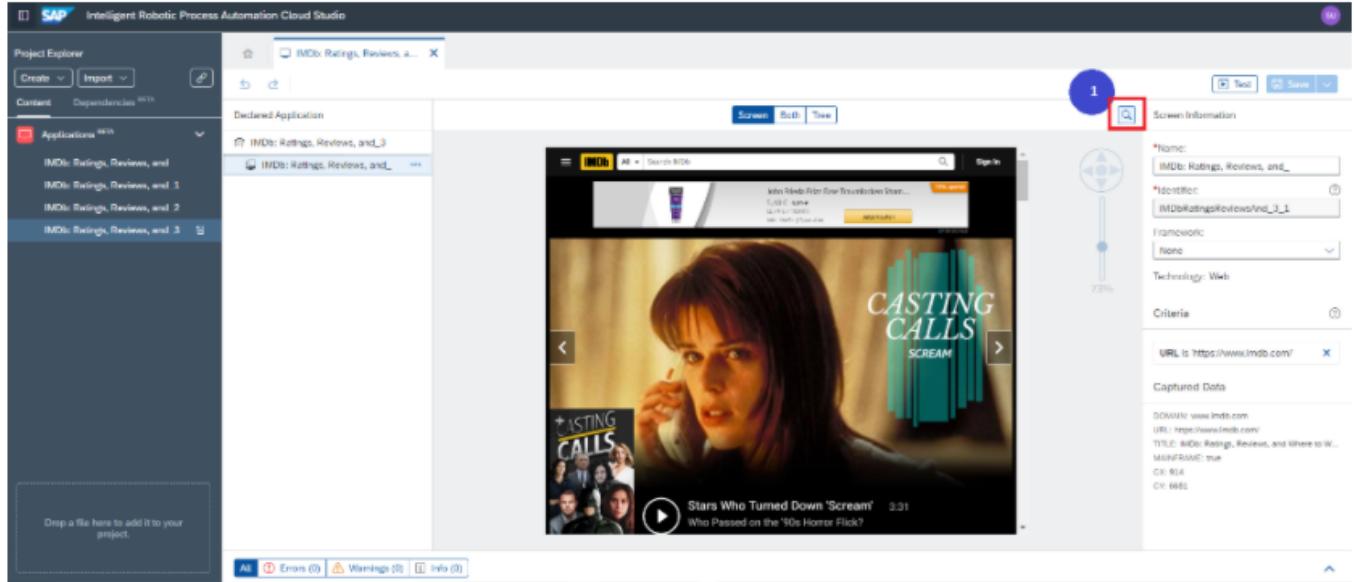
Context

After successful capture of the application screen, a preview screen is displayed in the capture area. For more information about capturing the screen, see the [Capture an Application](#) section.



Procedure

1. Click the **Search icon** (1).



The search field is expanded (2).

The screenshot shows the SAP Intelligent Robotic Process Automation Cloud Studio interface. On the left, there's a Project Explorer with a 'Content' section containing several application entries. In the center, a 'Declared Application' section shows a screenshot of an IMDb search results page for 'SCREAM'. The search bar at the top of the screenshot has a red box around it. To the right of the screenshot is a 'Screen Information' panel with various configuration options like 'Name', 'Identifier', 'Framework', and 'Technology'. At the bottom, there are tabs for 'All', 'Errors (0)', 'Warnings (0)', and 'Info (0)'. A blue circle with the number '2' is located in the top right corner of the main workspace.

2. Enter a term in the **Search** field.

3. Click **Search** or press **Enter**. The search result count is displayed to the left side of the **Search** field.

Screen

In the **Screen** view, the search results are highlighted in yellow colour.

This screenshot is similar to the one above but shows the search results. The search bar in the top right now displays '6 results'. The rest of the interface, including the project explorer, declared application, and screen information panel, remains the same.

Tree

In the **Tree** view, the search results are highlighted in blue colour.

The screenshot shows the SAP Intelligent Robotic Process Automation Cloud Studio interface. The left sidebar contains a 'Project Explorer' with a 'Create' and 'Import' button, and a 'Content' section with 'Applications' listed. The main workspace has tabs for 'Screen' and 'Tree', with 'Screen' currently selected. The 'Screen' tab displays a screenshot of the IMDb website, specifically a movie page for 'John Frieda Fritz Eise Traumflocken'. The 'Tree' tab shows a hierarchical tree structure of the application's UI elements. To the right of the tree, there is a 'Declared Application' section with a search bar and a table for 'Screen Information' and 'Captured Data'. The 'Captured Data' table includes columns for DOMAIN, URL, TITLE, MAINFRAME, CX, and CY.

Both

This view displays both the **Screen** and **Tree** views in split screen mode.

This screenshot is identical to the one above it, showing the 'Both' view of the SAP Intelligent Robotic Process Automation Cloud Studio. It displays the same project structure, application declaration, and captured data as the first screenshot, but with both the 'Screen' and 'Tree' tabs active simultaneously.

Test an Application

You can test an application before you include it in an automation.

Prerequisites

- You've captured an application as described in [Capture an Application](#), and you've declared all the screens and UI elements you want to use in an automation.
- You've created an environment in Cloud Factory. For more information, see [Create an Environment](#).

Context

By testing an application before using it in an automation, you ensure that you have declared all of the elements correctly. At runtime, the elements are correctly identified and can be controlled as intended.

Procedure

1. In Cloud Studio, open the application and choose [Test Application](#).

2. Choose [Test](#).

The system generates and deploys a test package. Then it launches the [Application Tester](#).

The applications declared in your project are displayed in the [Declared Applications](#) section.

3. Choose a declared application to expand the screens to test.

If you have entered different values for more than one instance, the tester differentiates the instance you choose by displaying the corresponding values as input in the [Declared Element](#) section.

4. Choose a screen or an instance to go to the declared elements and activities.

Choose a declared element to display its collection. For more information, see [Declare an Element as a Collection](#).

If you have created a parent-child hierarchy and added a collection as direct parent to another collection, the elements of the child collection are displayed in sublists on deeper levels.

i Note

This behaviour does not apply to web-declared applications. For a web-declared application, the elements of a child collection are displayed on a single line.

On the right side of each declared element is an activity matching its type, for example a [Click](#) activity for a button, or an input field.

5. Test each screen element.

a. Choose an element from the list.

A list of activities you can execute on the element is displayed at the bottom of the application tester.

b. Choose an appropriate activity.

c. Optionally, switch to the corresponding application to check whether your action was successful.

You've defined an element named [searchBar](#) in the application's screen. It's an input field:

Make sure that the right screen is open on your desktop. On the application tester, choose [searchBar](#). In the activities section, choose [Set Element](#). Enter a value for the inputs and click [Execute](#). The value appears in the text bar of the declared element. Switch to the application to check whether your value appears in the search bar.

On the application screen, there is also a button named [searchButton](#). Choose the [Click](#) activity on the right side of the search button. If your action is successful, both the application and the application tester change to a new screen.

6. Optionally, choose the next screen and test every screen.

7. Optionally, you can stop testing by closing the application tester window or by returning to the Cloud Studio and choosing [Stop](#) in the [Test Application](#) window.

Results

Your application is ready to be used in an automation.

Next Steps

While you are testing your application, you can navigate through the different artifacts in your Cloud Studio project and choose an element in read-only mode.

The [Design Console](#) indicates which tester is currently running in your project.

The screenshot shows the SAP Design Console interface. At the top, there are tabs for 'Design Console (0)', 'Test Console (0)', and 'Variables (0)'. On the right side, there is a status bar with 'Application test mode' and a dropdown arrow. Below the tabs, there are buttons for 'All', 'Errors (0)', 'Warnings (0)', and 'Info (0)'. A red box highlights the 'Quick Fix' button, which is located on the far right of the error/warning/info buttons.

Related Information

[Add an Application Screen to an Automation](#)

Quick Fix

The quick fix feature allows you to restore or delete a missing element in your captured application.

Context

In some captures, the screens contain corrupted elements like a missing parent for a child element. The quick fix feature allows you to restore or delete a missing element in your captured screen.

Follow the procedure to fix the error in your application.

Procedure

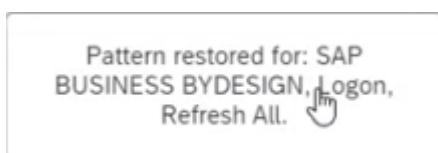
1. In the **Design Console**, see the error that displays the **Quick fix** button.

The screenshot shows the SAP Design Console interface. At the top, there are tabs for 'Design Console (1)', 'Test Console (0)', and 'Variables (0)'. On the right side, there is a status bar with 'Application test mode' and a dropdown arrow. Below the tabs, there are buttons for 'All', 'Errors (1)', 'Warnings (1)', and 'Info (0)'. A red box highlights the 'Quick Fix' button, which is located on the far right of the error/warning/info buttons. A message at the bottom left says 'The application contains elements that have a corrupted pattern.'

2. Click on **Quick fix** and select one of two options:

- o **Restore**

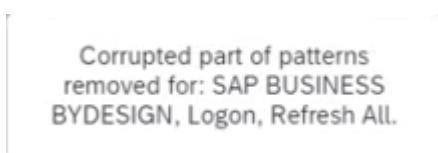
The following message is displayed: **Pattern restored for: SAP element**.



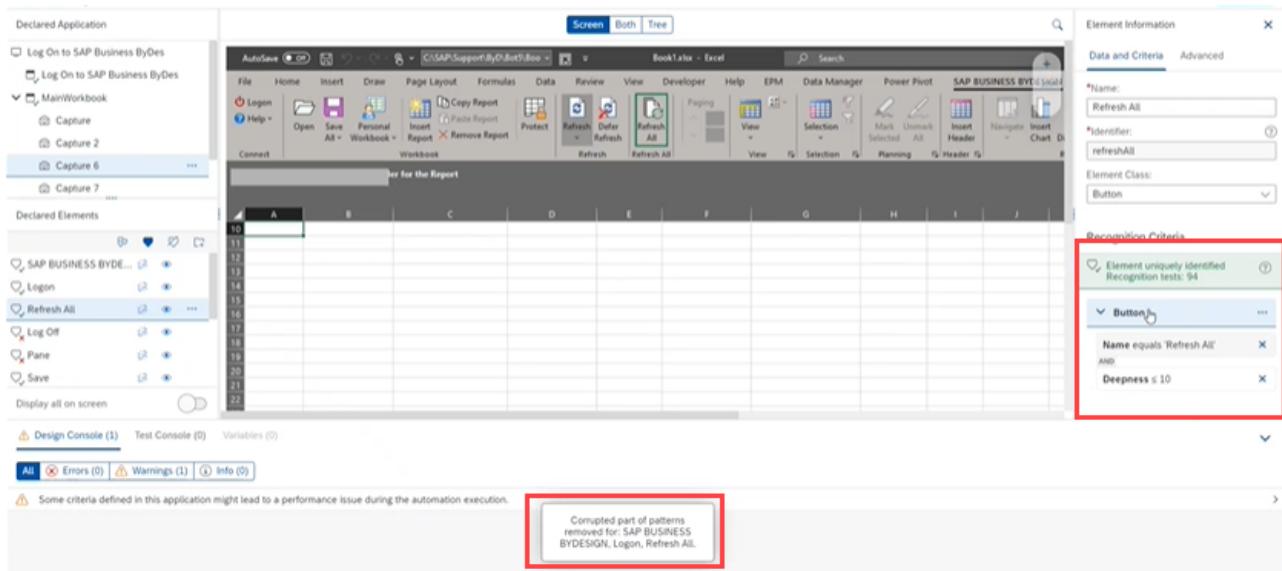
Now if you look in the **Recognition Criteria** section, the missing element has been restored.

- o **Delete**

The following message is displayed: **Corrupted part of patterns removed for: SAP element**.



Now if you look in the **Recognition Criteria**, the corrupted part of your screen has been deleted and the element is uniquely identified.



Next Steps

Once you have fixed your error, click on your username on the top right-hand corner and select **Message History**. A message history window opens where you can display in more details the content of the quick fix.

The image displays two screenshots of the SAP Cloud Studio Message History window. The left screenshot shows a list of messages with categories: Errors (0), Warnings (0), Success: 0, and Info (1). The right screenshot shows a detailed view of a single message: 'Quick fix applied on Log On to SAP Business ByDes'. It includes a bullet point: 'Pattern restored for: SAP BUSINESS BYDESIGN, Logon, Refresh All.' and a date: 'Date: 2021-11-10 16:04:25'.

Supported Interface Technologies

Capturing and declaring applications means identifying the applications you want to control in an automation. To do this, the Cloud Studio and the Desktop Agent provide a set of interface technologies for controlling purposes.

The Web Technology

For more information, see [The Web Technology](#).

The UI Automation Technology

For more information, see [The UI Automation Technology](#).

The SAP GUI Technology

For more information, see [The SAP GUI Technology](#).

The SAPUI5 Technology

SAPUI5 is an extension of the Web technology. It's a framework that includes a collection of libraries you can use to build applications which run in a desktop or mobile browser – while only maintaining one code base.

The Web Technology

Using the Web technology, you can capture HTML pages displayed in web browser tabs. You can use the following web browsers:

- Internet Explorer
- Google Chrome
- Firefox
- Microsoft Edge

Screen Recognition

When the Web technology detects the opening of a Web screen:

- It looks for the first declared Web application with matching criteria.
- If it finds one, it looks for the first declared screen of the application with matching criteria.
- Otherwise, it ignores the Web screen.

Application Instance Management

An application instance is a running instance of an application. The Web technology handles applications instances as follows:

- A Web application loaded in one Web browser tab is considered as one instance of the Web application.
- If a Web application is loaded twice in two different Web browser tabs, the Web technology regards them as two instances of the same Web application.

i Note

If a Web application instance navigates to another Web application in the same tab, the Web application instance is considered to be ended.

Frame Capture

If a screen is composed of different frames, you must separately capture the frame of your choice as a new screen to get the content of this screen. If you only capture the main frame, you will not be able to capture the other frames.

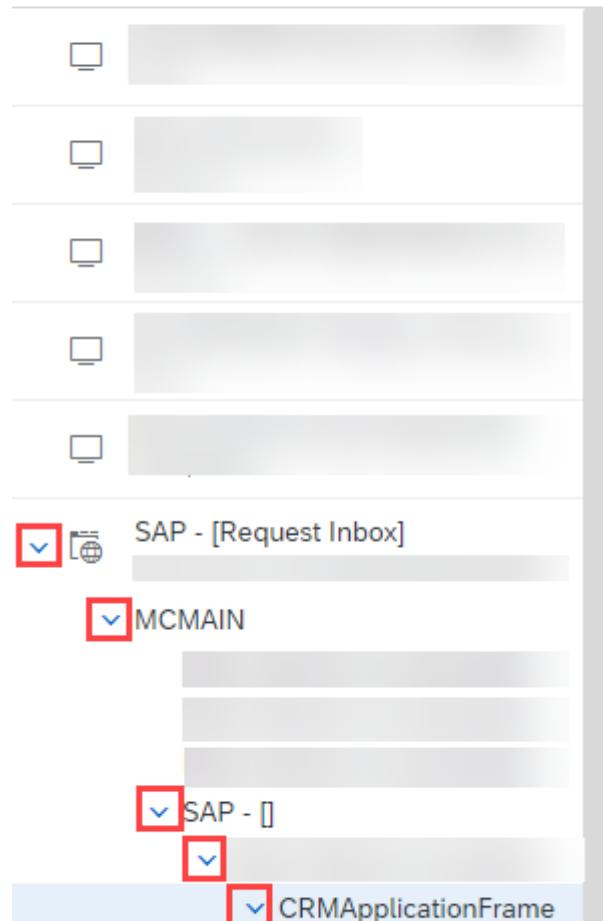
- Click the arrow next to the main screen and deploy the tree view. Select the frame that you want to capture and click **Capture**.

You can only capture frames that appear in the tree view on the **Select Screen** panel. Please note that some frames cannot be captured because they do not have a user interface.

Select Screen 

Below are the open screens detected on your computer. Select the screen you want to use in your automation.

Search...   



SAP - [Request Inbox]
MCMAIN
SAP -
CRMApplicationFrame

SAP - [Request I...
blank

- You can see that the frame you have just captured has the following recognition criterion: **MAINFRAME is false**.

Screen Information

Data and Criteria Advanced

*Name:

CRMApplicationFrame

*Identifier:



cCRMApplicationFrame

Framework:

None



Technology: Web

Recognition Order: 1

Recognition Criteria

Screen uniquely identified

Recognition tests: 2

URL equals 'https://sfp.bss.net...'

AND

MAINFRAME is false

If the frame is not recognized by the application:

Make sure that the application screen shares common recognition criteria with the frame.

You must add the recognition criteria of the frame to the application screen to make sure that the frame is recognized. For instance, add the **Domain** or **URL** recognition criterion of the frame to the application screen.

For more information on how to add recognition criteria to an application, see [Declare an Application](#).

Identify Correct Frame Using FRAMEINDEX Parameter

If a screen has multiple frames and the metadata for all frames are the same, it will be difficult for you to identify the correct frame at runtime.

Now, the **FRAMEINDEX** parameter can be used to improve the recognition capability of frames. It will uniquely identify the correct frame even though there are identical frames that exist on a screen.

It is recommended to use the **FRAMEINDEX** parameter at design time when other parameters like **DOMAIN**, **TITLE**, or **MAINFRAME** do not work.

In the following screenshot of a captured frame, you can see the **FRAMEINDEX** parameter under **Captured Data** section.

i Note

The **FRAMEINDEX** parameter is not applicable for a captured main frame.

Declaring an Application Using the Web Technology

For general information on this topic, see [Declare an Application](#).

Set Application Criteria

Application criteria are used by the Web technology as follows:

When the Web technology detects the opening of a Web screen:

- It looks for the first declared Web application whose criteria match.
- If it finds one, it looks for the first declared screen of that application whose criteria match. If it finds a screen, it associates the new Web screen with that page.
- Otherwise, it ignores the Web screen.

That means that application criteria must be:

- Broad enough to match all the Web screens of the Web application.
- But not too broad, to ensure they do not match the Web screens of other Web applications.

You set recognition criteria using the Cloud Studio:

- You can use operators to define the value of properties.

For example, you can use operators to define: **DOMAIN (Property) equal to (Operator) help.sap.com (Value)**.

Modify Criterion X

*Property:

DOMAIN

*Operator:

equal to

*Value:

help.sap.com

Apply

- You must declare **at least one** recognition criterion.
- You can set multiple criteria on the same property, which are then connected by a logical **OR**.

Criteria ?

DOMAIN

is not empty X

OR
 equals 'help.sap.com' X

Usually, you set a recognition criterion for the **DOMAIN** property.

Advanced Parameters

You can use one of the following advanced parameters when declaring an application using the Web technology:

One instance per thread (Only for IE)

Application Information

Data and Criteria Advanced

Application Path

https://www.google.com/

One instance per thread (Only for IE) ?

Yes No

Polling Delay (in ms)

250

The Cloud Studio communicates with the technology which instance ID to select.

- Select **No** to use the **Process ID**.
- Select **Yes** to use the **Thread ID** to distinguish different instances of an application.

i Note

This option is used only with the Internet Explorer (IE).

Polling Delay (in ms)

The Cloud Studio waits for an application to check for its readiness or state. This option sets the polling timer frequency in milliseconds (polling mechanism cannot be stopped). By default, the polling delay is 250 milliseconds.

- Click **+** to increase the polling timer frequency.
- Click **-** to decrease the polling timer frequency.

Declaring a Screen Using the Web Technology

For general information on this topic, see [Declare a Screen](#).

Set Screen Criteria

The Web technology applies screen criteria as follows:

When the Web technology detects the opening of a Web screen:

- It looks for the first declared Web application whose criteria match.
- If it finds one, it then looks for the first declared screen of the application whose criteria match. If such a screen is found, the technology associates the new Web screen with that screen.
- And so on.

That means that screen criteria must be:

- Broad enough to match the desired Web screen of the Web application.
- But not too broad, to make sure not to match the other Web screens of the Web application.

When you set recognition criteria on a screen using the Web technology:

- You can use operators to define the values of properties. For example, you can use operators to define: **URL (Property) equal to (Operator) https://help.sap.com/viewer/index (Value)**.

Modify Criterion

*Property: URL

*Operator: equal to

*Value: https://help.sap.com/viewer/inde

Apply

- You must declare **at least one** recognition criterion.
- You can use **OR** to define a combination of criteria for each property.

TITLE

starts with 'SAP' X

OR

equals 'SAP Help Portal' X

Usually, you will set a recognition criterion on the screen part of the **URL** property.

Advanced Parameters

You can use of the following advanced parameter when declaring a screen using the Web technology:

Capture Cached Elements

Screen Information

Data and Criteria Advanced

Launch URL
https://www.google.com/search?q=batm:

Capture Cached Elements ?

Yes No

This option allows to capture screen elements automatically and their latest value being available for every event. By default, this option is set to **No**.

- Select **Yes**, if screen elements must be captured automatically and their latest value being available for every event.

- Select **No** if you do not want to capture screen elements automatically by the system.

Declaring an Element Using the Web Technology

For general information on this topic, see [Declare an Element](#).

Set Element Criteria

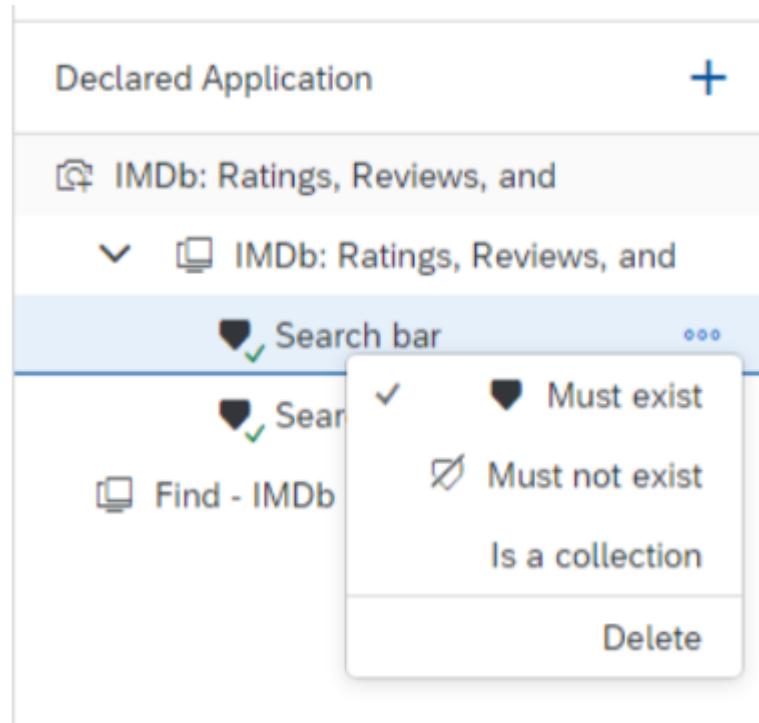
Element criteria are used by the Web technology in the following way:

Once the Web technology recognizes a Web screen as a declared screen, it can search the screen DOM to find the target of each element:

- An element targets the first DOM component that matches its criteria.
- If an element is multiple, it will target all of the DOM components that match its criteria.

That means that element criteria must be:

- Precise enough to match the desired component of the Web screen.
- Broad enough to be able to handle as far as possible Web screen evolutions.
- **Must exist** means that the screen is not recognized unless the element or elements defined as must exist are detected.
- **Must not exist** means that the screen is not recognized if an element defined as must not exist is detected.
- **Is a collection** means that the element consists of multiple parts - for example, a list with multiple entries.

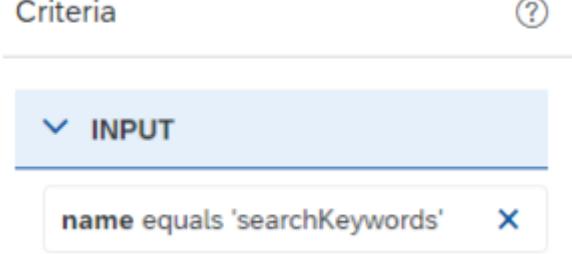
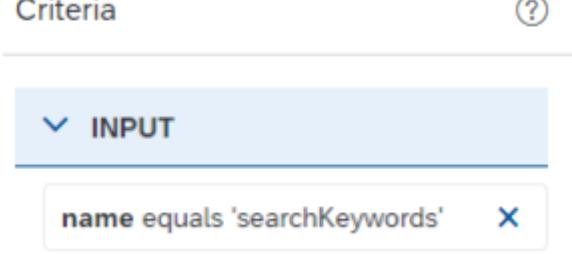
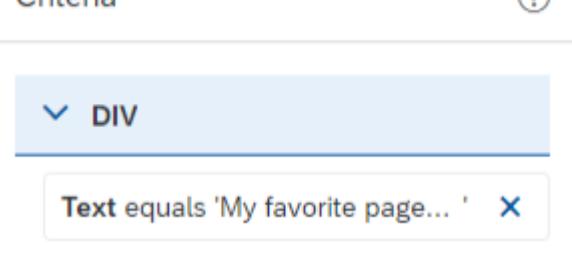


When you set recognition criteria on an element using the Web technology:

- You can use operators to define the values of properties.
- You must declare **at least one** recognition criterion. Elements are usually declared for controlling purposes. You can also use elements to help screen recognition by defining elements with specific options:

- You can use **OR** to define a combination of criteria for each property.

Usually, you will set recognition criteria for the following properties:

TAG	<p>Elements are usually declared for controlling purposes. The Web technology requires a criterion for the HTML name (TAG) of the component: INPUT, DIV, SPAN, and so on. When you create an element from a Web component, the Cloud Studio automatically sets this criterion. It cannot be edited or removed.</p> <p>Example:</p> 
id or name	<p>It is advisable to set a criterion for the properties id or name of the component, if present and unique. These properties are usually set by Web developers and do not change. When an element is created, the Cloud Studio automatically sets a criterion for one of these two properties if the criterion is suitable.</p> <p>Example:</p> 
Text	<p>With some components (such as SPAN or TH), if the id or name properties are not usable, you can set a criterion for the Text property.</p> <p>Example:</p>  <p>i Note</p> <p>This property corresponds to the HTML property, InnerText. So, if a component has a Text property equal to My Text, all ancestor components also have a text property containing My Text.</p>

Advanced Declaration

During the declaration phase, you cannot always identify elements with a simple declaration of criteria. You identify these elements with the structure of the page: you add hierarchical elements (parent/child) to their recognition criteria. For example,

when declaring an element with a criterion on the nth-child-tag property, you should also declare a direct parent for this element. For more information, see [Advanced Declaration](#).

i Note

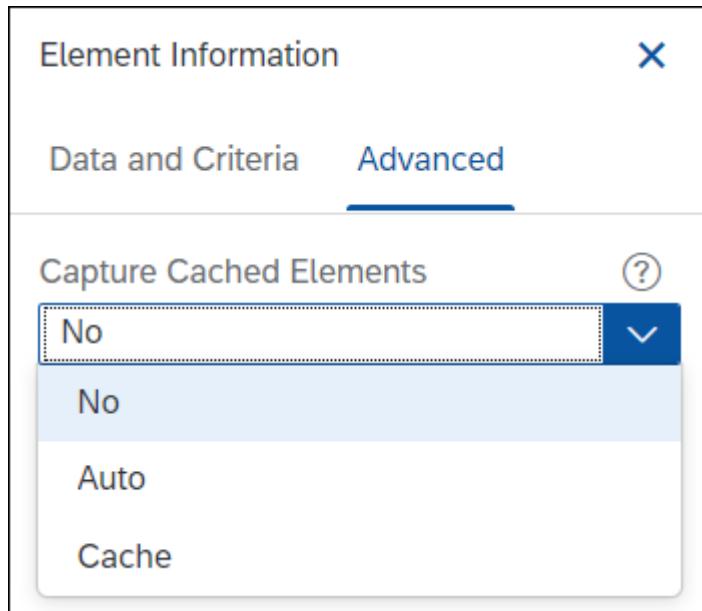
The Range property is replaced with nth-child-tag property in the web technologies.

The Range property is available in old captures.

In an old capture, when you select the range criteria, the **Is a collection** property is disabled.

Advanced Parameters

Capture Cached Elements



By default, the Capture Cached Elements values is **No**.

- **No** - Select **No** if you do not want to retrieve the element values automatically.
- **Auto** - Select **Auto** to retrieve the element values automatically for every event.
- **Cache** - Select **Cache** to retrieve all cached element values in the memory for every event.

Best Practices for Establishing Recognition Criteria for Web Applications

The following are the best practices for establishing recognition criteria for Web applications.

- When automating a web application, make sure to choose **WEB** as the **FRAMEWORK** and **DOMAIN** as the **Recognition Criteria**. If there are multiple domains, add them using an **OR** condition.
- When capturing a web page (by adding a new screen or a new capture), make sure to select the framework of the application that you are trying to capture as the framework of that screen. The framework can fall into one of the following categories:
 - **ARIBA**
 - **UI5**
 - **SAPWebGUI**

- SuccessFactors

If it does not fit into any of these categories, it is categorized as **WEB**. Choose **TITLE**, **URL**, or **MAINFRAME** as the recommended **Recognition Criteria** for the screen. Use the following conditional operators:

- Contains
- Starts with
- Ends with

- If a scenario is recorded in one domain and then executed in a different domain, the application criteria must be updated with the new domain using an **OR** condition.
- To execute an automation scenario for an application with multiple domains, the application criteria must be updated to include all the domains using an **OR** condition.
- Use a title for a screen. If you do not provide a title, the **MAINFRAME** and **URL** should be used as criteria. In cases where the URL changes dynamically while the program is running, use a portion of the URL (using the **Contains** operator) or domain to identify a screen.
- If a bot encounters an error during execution, check the declaration and criteria of the application, screen, or elements that are created within the application.
- Make sure that two screens do not have identical or connected criteria. For instance, if the title of Screen1 begins with 'ABCD' and the title of Screen2 begins with 'ABC', either screen can be identified during runtime.
- Use **Multi Capture** when capturing the same screen twice, such as popups or dropdowns.
- Make sure to use the **Start Web Page** activity to initiate the automation of a web application.
- Use the **Wait (Screen)** activity to ensure that the automation pauses until the screen of the web application (such as an angular application) has finished loading.
- Make sure to add the **Wait (Element)** activity before using the **Close Screen** or **Close Application** activity so that the automation waits for the log off screen to finish loading before proceeding to close the application.
- When you are automating using web automation, make sure that the **Element Class** comes from the **WEB** SDK or any of the web related SDKs. Choose one of following recommended **Recognition Criteria** for elements:

- NAME
- CLASS
- TEXT
- TITLE
- ARIA-LABEL
- TYPE

Do not use ID as a recognition criterion. Use the following conditional operators:

- Contains
- Starts with
- Ends with

- If the **Set Element** activity is not effective, we recommend that you use the **Keystroke Element** activity to perform the search control.
- Use the **Mouse Click** activity to perform the upload, download, or browse control. When you use the **Mouse Click** activity, the browser does not enforce security policies, allowing you to interact with your local file system during these operations.

Best Practices for Automating the SAP ByDesign Application

The following are the best practices for automating the SAP ByDesign application.

- Use only a **WEB** framework to automate the SAP ByDesign application. You must not use the **SAPUI5** framework for automation.
- Make sure that all elements use the **Element Class** that is provided by the **WEB** SDK.

- Use **Multi Capture** when capturing the same screen twice, such as popups or dropdowns.
- Select the **WEB** search type as the target element when using the **Keystroke Element** activity within the SAP ByDesign application.
- Use Recorder when automating the SAP ByDesign applications.
- Select the **Element Class** as the **WEB Element** for any combo box.
- Make sure that you use the **Starts with** conditional operator for any text property when using a combo box.

Limitations

This section provides the limitations while capturing screens or frames using the Web Technology.

- If a screen has frames that have no src tag, or about:blank url, that screen can't be captured or recorded as the listing of these frames is empty.

If a screen has dynamic iFrames, make sure that you capture the screen manually instead of using the Recorder to capture the screen.

i Note

An iFrame, also known as Inline Frame, is an element that loads another HTML element inside of a web page.

Dynamic iFrames are the iFrames that appear on a web page dynamically.

- The capture of web pages with a DOM larger than 60MB is not supported due to a limitation imposed by Chrome's policies.

i Note

While designing or executing automation for the Web technology, you may encounter the following error messages:

- **Uncaught (in promise) Error: No tab with id: xxxxxxxx.**

[← Errors](#) [Clear all](#)

! Uncaught (in promise) Error: No tab with id: 1303339407.

Context
background.js

Stack Trace

```
background.js:1 (anonymous function)
```

1 `((()=>{var e={306:e=>{e.exports=class{constructor(e){this.oFrameDict={},this.oMessage=e,this.iTimeo`

- `TypeError: Cannot set properties of null (setting 'iFramesChangedBy')`

[← Errors](#) [Clear all](#)

! `TypeError: Cannot set properties of null (setting 'iFramesChangedBy')`

Context
background.js

Stack Trace

```
background.js:1 (anonymous function)
```

1 `((()=>{var e={306:e=>{e.exports=class{constructor(e){this.oFrameDict={},this.oMessage=e,this.iTimeo`

These errors do not impact the execution of the automation in any manner.

To resolve them, go to [chrome://extensions](#) [Errors](#) [Clear all](#).

The UI Automation Technology

UI Automation is a standard protocol developed by Microsoft. It enables applications developed with the following technologies to be controlled:

- Microsoft technologies (such as Win32, WinForm, WPF, or Silverlight).
- Any technologies that support MSAA (such as Qt)

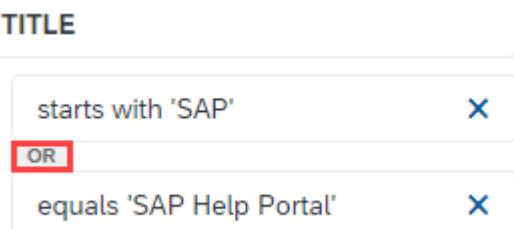
Declaring an Application Using the UI Automation Technology

For general information on this topic, see [Declare an Application](#).

Set Application Criteria

Using the UI Automation technology:

- You can use operators to define the value of properties.
- You must declare **at least one** recognition criterion.
- You can set multiple criteria on the same property, which are then connected by a logical **OR**.



Usually, you set recognition criteria on the **EXE** property (process name of the application). This criterion is automatically set by Cloud Studio when you declare an UI Automation Application.

Advanced Parameters

Application Information

Data and Criteria Advanced

Application Path

C:\Program Files\WindowsApps\Microsoft

Pending delay (in s)

-
+

Polling delay (in ms)

-
+

Refresh on polling

Yes No

Pending delay (in s)

This option is used to add delay in seconds while the driver tries recognition on a newly opening window, waiting for its tree being completely built.

Polling delay (in ms)

This option sets the polling timer frequency in milliseconds (polling mechanism cannot be stopped). By default, the polling delay is 1000 milliseconds.

Refresh on polling

This option allows to suspend pages recognition when polling occurs.

- Click **No** if you do not want to suspend pages recognition when polling occurs.
- Click **Yes** if you want to suspend pages recognition when polling occurs.

Declaring a Screen Using the UI Automation Technology

For general information on this topic, see [Declare a Screen](#).

Set Screen Criteria

Screen criteria are set on the properties of the root component of the targeted screen. They are used by the UI Automation technology as follows:

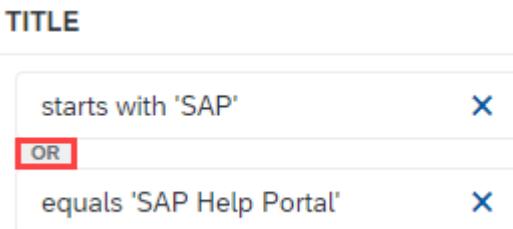
The UI Automation technology searches the DOM tree to find all DOM components matching the screen's criteria. If a component matches, the technology does not search its subtree to find other instances of the screen. The technology creates one instance of the screen for each matching component.

That means that screen criteria must be:

- Precise enough to match the desired root component, and no other.
- Robust enough to still work should the screens change.

When you set recognition criteria on a screen using the UI Automation technology:

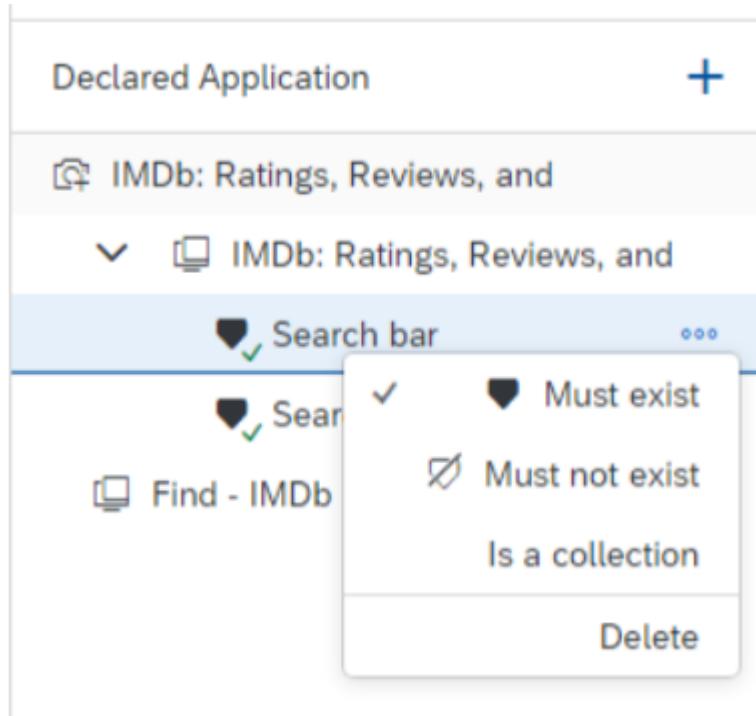
- You can use operators to define the values of properties.
- You must declare **at least one** recognition criterion.
- You can set multiple criteria on the same property, which are then connected by a logical **OR**.



Advanced Recognition Methods for Screens

If the targeted root component has no unique distinguishing properties, you can use one of the following advanced declaration methods:

- **Must exist** means that the screen is not recognized unless the element or elements defined as must exist are detected.
- **Must not exist** means that the screen is not recognized if an element defined as must not exist is detected.
- **Is a collection** means that the element consists of multiple parts - for example, a list with multiple entries.



Advanced Parameters

Screen Information

Data and Criteria **Advanced**

Refresh screen recognition:

At a fixed time interval 

Multi Instance 

Yes No

Automatic Capture on Polling 

Yes No

Multi Instance

Indicates if this screen can have multiple running instances. If **No**, only the first running instance will be recognized. Set this parameter as **No** to optimize the recognition.

Automatic capture on polling

This option allows screen elements to be automatically captured on polling, their latest value being available for every event. By default, this option is set to **No**.

Refresh Mode

Defines when screen recognition should be refreshed. The default value can be computed based on the screen root control type, leaving the setting for advanced users.

This is a screen setting that can take the following values:

- **No:** No refresh is needed.
- **Window:** Refresh is triggered by the **WINDOWOPENED** event.
- **Polling:** Refresh is done at a fixed time interval.

Declaring an Element Using the UI Automation Technology

For general information on this topic, see [Declare an Element](#).

Set Element Criteria

Element criteria are used by the UI Automation technology as follows:

Once the parent screen recognizes a UI component as a declared screen, it can search the component's subtree to find each element's target:

- An element will target the first DOM component that matches its criteria.
- If an element is multiple, it will target all of the DOM components that match its criteria.

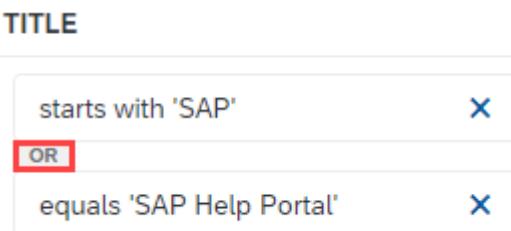
That means that element criteria must be:

- Precise enough to match the desired component.
- Broad enough to still work if the DOM changes.

Elements are usually declared for controlling purposes. You can also use elements to aid page recognition by setting **Must exist** or **Must not exist** parameters.

You set recognition criteria using Cloud Studio.

- You can use operators to define the values of properties.
- You must declare **at least one** recognition criterion.
- You can use **OR** to define a combination of criteria on each property.

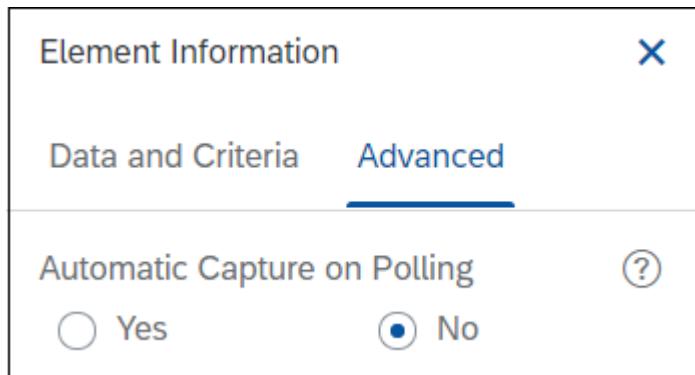


Best Practices

To capture hidden elements, you must expand the elements before capturing the application.

Advanced Parameters

Automatic capture on polling



This option allows elements to be automatically captured on polling, their latest value being available for every event. By default, this option is set to **No**.

The SAP GUI Technology

You can capture SAP applications using the SAP GUI technology. SAP GUI for Windows is a front-end application that you can use to access SAP applications such as SAP ERP, SAP Business Suite, and so on. It is designed for the Windows operating system and provides a Windows-like user experience and integration with other applications based on OLE interfaces or ActiveX controls.

Prerequisites

This is custom documentation. For more information, please visit the [SAP Help Portal](#)

To use recorder with SAP WinGUI, make sure you use only 32-bit SAP GUI for Windows Application.

i Note

In some cases the SAP GUI technology of a SAP GUI popup isn't automatically detected. If so, and if the screen belongs to the SAPLOGON.EXE process, the UI Automation is set as technology, and SAP GUI as optional technology. This way, you can force the capture using SAP GUI.

SAP GUI is also available as optional technology for the main SAPLOGON screen, which is not a SAP GUI screen. Please note that if you manually change the technology to select the SAP GUI technology to capture this screen, it will fail.

i Note

To automate the SAP WinGUI 800 X64 bit, you can use the SAP Intelligent RPA Desktop Agent in 32-bit that is already available. To enable scripting access, you must add the string value D11Surrogate under the registry entry Computer\HKEY_CLASSES_ROOT\WOW6432Node\AppBar\{E2779C61-F87E-4038-98A0-1D9E71334706}. After updating the registry value, make sure to restart your system.

Enabling Scripting

For automation purposes, you need to enable and use the SAP GUI Scripting API. You need to enable both client and server scripting. A JavaScript library (SAPScripting.js) is available to implement specific behaviors.

Declaring an Application Using the SAP GUI Technology

For general information on this topic, see [Declare an Application](#).

Declaring the SAPLogon Application

SAPLogon.exe is a multi-technology application.

It includes:

- Windows technology
- SAP GUI technology
- Web technology

All SAP transactions based on SAP GUI are accessed through the SAPLogon application. This application must be declared using UI Automation technology, by targeting the SAPLogon window (see [Declaring an Application Using the UI Automation Technology](#)). UI Automation technology lets you declare UI Automation screens, SAP GUI screens, and Web screens.

Once you have declared the SAPLogon application, you need to do the same with the SAPLogon screen.

This declaration activity also requires the following tasks to be performed:

- [Declaring a Screen Using the SAP GUI Technology](#)
- [Declaring an Element Using the SAP GUI Technology](#)

Advanced Parameters

Application Information

Data and Criteria Advanced

Application Path

C:\Program Files\WindowsApps\Microsoft

Pending delay (in s)

-
+

Polling delay (in ms)

-
+

Refresh on polling

Yes No

Pending delay (in s)

This option is used to add delay in seconds while the driver tries recognition on a newly opening window, waiting for its tree being completely built.

Polling delay (in ms)

This option sets the polling timer frequency in milliseconds (polling mechanism cannot be stopped). By default, the polling delay is 1000 milliseconds.

Refresh on polling

This option allows to suspend pages recognition when polling occurs.

- Click **No** if you do not want to suspend pages recognition when polling occurs.
- Click **Yes** if you want to suspend pages recognition when polling occurs.

Best Practices while Declaring Multiple Applications within Same Project

You can declare multiple SAP GUI applications within the same project. When you start the SAP GUI application, each of these declared applications will get a LOAD event as shown in the following screenshot:

```

⌚ |18:15:29.467| a_customerCreation[5368].s_sAPEasyAccess[1011]:LOAD
⌚ |18:15:31.960| a_sAPEasyAccess[21472].s_sAPEasyAccess[1012]:LOAD
⌚ |18:15:31.962| a_sAPEasyAccess1[26840].s_sAPEasyAccess[1013]:LOAD
⌚ |18:15:31.964| a_sAPEasyAccess2[32208].s_sAPEasyAccess[1014]:LOAD
⌚ |18:15:31.966| a_sAPEasyAccess3[37576].s_sAPEasyAccess[1015]:LOAD
⌚ |18:15:31.968| a_sAPEasyAccess4[42944].s_sAPEasyAccess[1016]:LOAD
⌚ |18:15:31.970| a_sAPEasyAccess5[48312].s_sAPEasyAccess[1017]:LOAD
⌚ |18:15:31.984| a_sAPEasyAccessQ431[53680].s_sAPEasyAccess[1018]:LOAD

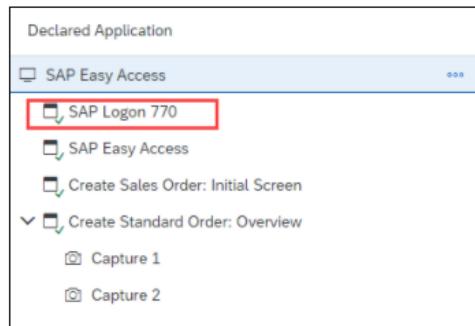
```

Therefore, you can use these screens declared across different application declarations in single automation.

The following is the best practice we recommend while declaring multiple applications within the same project.

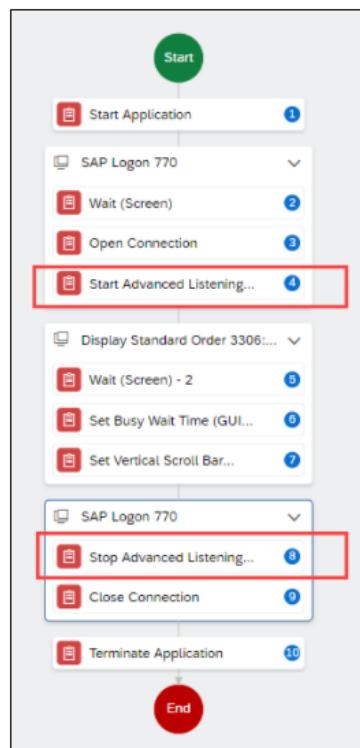
SAP Logon Capture

The **SAP Logon Capture** must exist inside all the declared applications even though they are mutually exclusive sets of screens. If **SAP Logon Capture** does not exist inside any one of the declared applications, there will be issues while generating the UNLOAD event of that application.



Refresh on Polling

- If **Refresh on Polling** is set to "No" in at least one of the declared applications, the polling of SAP GUI will not start when the project gets downloaded into the agent.
- When the automation starts, the polling can be started or stopped by using the **Start Advanced Listening Screens** and **Stop Advanced Listening Screens** activities on an SAP Logon screen of any declared application.
- You must call the **Start Advanced Listening Screens** and **Stop Advanced Listening Screens** activities only in the beginning and the end of the main automation workflow. It is not recommended to call these activities in a sub-automation or a reusable package from the store since calling the **Stop Advanced Listening Screens** activity in the middle of the automation will UNLOAD all the SAP GUI pages and stop the polling thread.



Polling Delay

The minimum value of the polling delay will be considered across all the declared applications.

Application Information	
Data and Criteria	Advanced
Application Path C:\Program Files (x86)\SAP\FrontEnd\gui	
Pending delay (in s) 0	
Polling delay (in ms) 1000	
Refresh on polling <input checked="" type="radio"/> Yes <input type="radio"/> No	

Declaring a Screen Using the SAP GUI Technology

For general information on this topic, see [Declare a Screen](#).

Set Screen Criteria

Screen criteria are set on the properties of the root component of the targeted screen. They are used by the SAP GUI technology as follows:

The SAP GUI technology searches the DOM tree to find all DOM components matching the screen's criteria. If a component matches, the technology does not search its subtree to find other instances of the screen. The technology creates one instance of the screen for each matching component.

That means that screen criteria must be:

- Precise enough to match the desired root component, and no other.
- Robust enough to still work should the screens change.

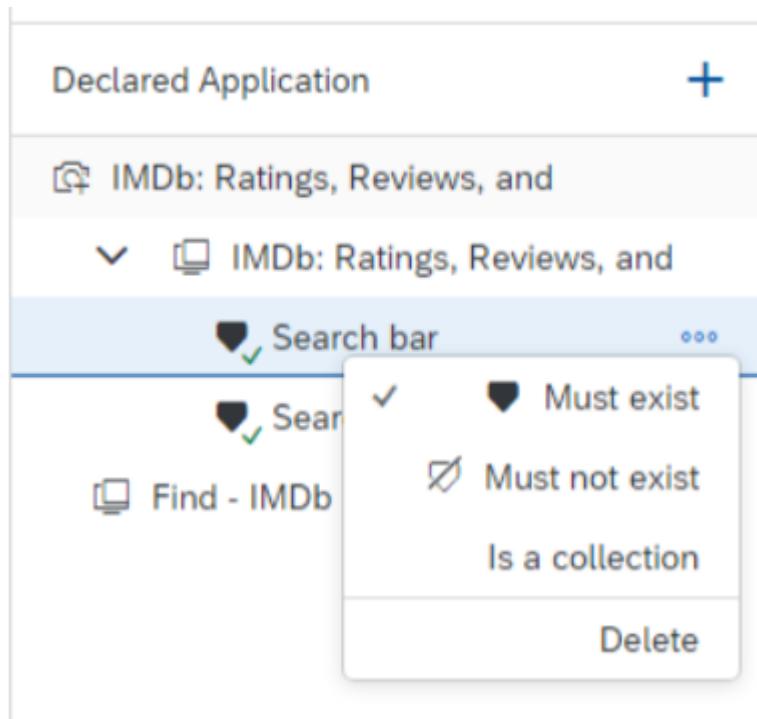
You set recognition criteria using the Cloud Studio:

- You can use operators to define the value of properties.
- You must declare **at least one** recognition criterion.
- You can set multiple criteria on the same property, which are then connected by a logical **OR**.

Advanced Recognition Options

If the targeted root component has no unique distinguishing properties, you can use one of the following advanced declaration options:

- **Must exist** means that the screen is not recognized unless the element or elements defined as must exist are detected.
- **Must not exist** means that the screen is not recognized if an element defined as must not exist is detected.
- **Is a collection** means that the element consists of multiple parts - for example, a list with multiple entries.



Advanced Parameters

Screen Information

Data and Criteria **Advanced**

Refresh screen recognition:

At a fixed time interval 

Multi Instance 

Yes No

Automatic Capture on Polling 

Yes No

Refresh screen recognition

Defines when the screen recognition starts refreshing. **At a fixed time interval** option must be selected for the screens of SAP GUI for Windows Application.

Multi Instance

Indicates if this screen can have multiple running instances. If **No**, only the first running instance will be recognized. Set this parameter as **No** to optimize the recognition.

Automatic Capture on Polling

This option allows screen elements to be automatically captured on polling, their latest value being available for every event. By default, this option is set to **No**.

Declaring an Element Using the SAP GUI Technology

For more general information on this topic, see [Declare an Element](#).

Set Element Criteria

Element criteria are used by the SAP GUI technology as follows:

Once the SAP GUI technology recognizes a UI component as a declared screen, it can search the component's subtree to find each element's target:

- An element will target the first DOM component that matches its criteria.
- If an element is indexed, it will target all of the DOM components that match its criteria.

That means that element criteria must be:

- Accurate enough to match the desired component.
- Broad enough to still work if the DOM changes.

Elements are usually declared for controlling purposes. You can also use items to aid page recognition by setting **Must exist** or **Must not exist** parameters.

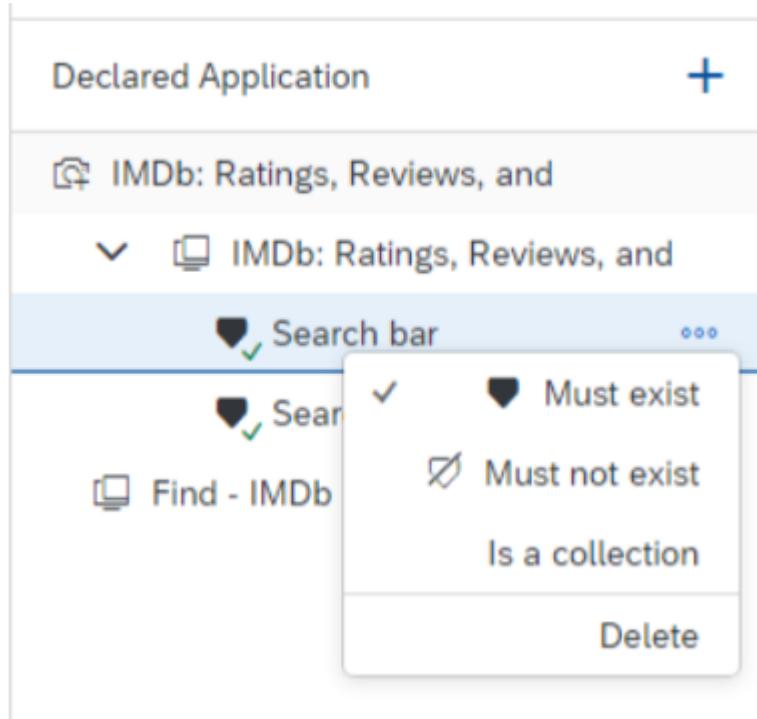
You set recognition criteria using Cloud Studio:

- You can use operators to define the value of properties.
- You must declare **at least one** recognition criterion.
- You can use **OR** to define a combination of criteria on each property.

Advanced Recognition Options

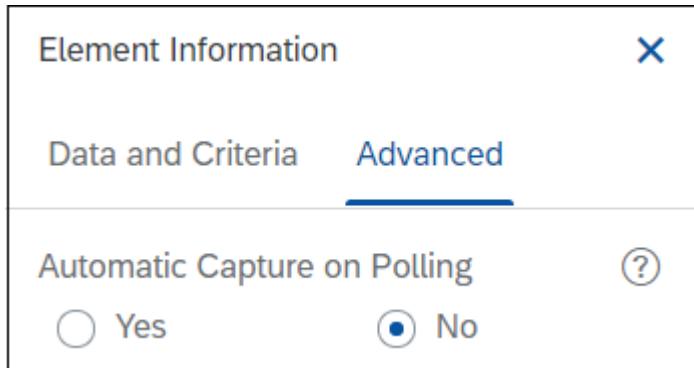
In some cases, it is not possible to recognize a targeted component merely by setting criteria on its properties. Another component is recognized by mistake. To ensure your declaration is correct, you can use advanced recognition options.

- **Must exist** means that the screen is not recognized unless the element or elements defined as must exist are detected.
- **Must not exist** means that the screen is not recognized if an element defined as must not exist is detected.
- **Is a collection** means that the element consists of multiple parts - for example, a list with multiple entries.



Advanced Parameters

Automatic capture on polling



This option allows elements to be automatically captured on polling, their latest value being available for every event. By default, this option is set to **No**.

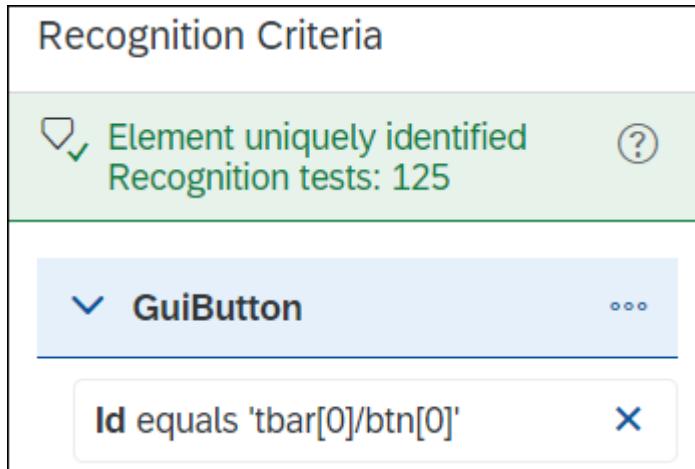
Best Practices

This section provides the best practices for a faster execution of captured SAP applications using the SAP GUI technology.

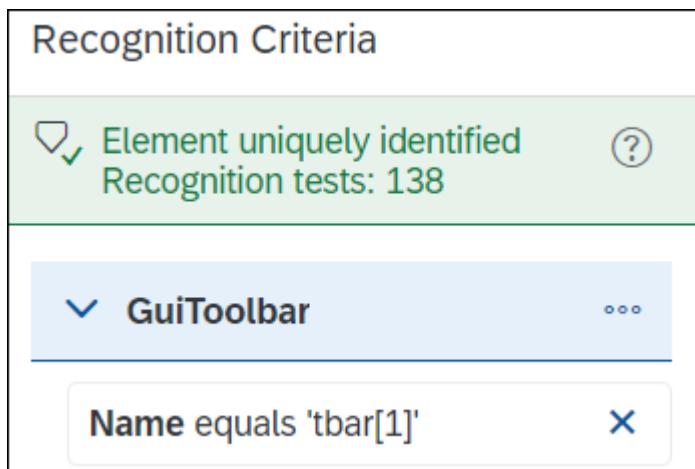
Better Recognition of an Element

When declaring an element, the criteria must be unique to each specific element.

- Only add **Id** as a criteria for an element if Id is stable (not applicable for a screen).



- Only add **Name** as a criteria for an element if Name is stable and if Id is not available for the element (not applicable for a screen).



- Only use **Name** as a criteria for a Collection element if the Name property is available. Commonly used for **GuiTableControl** cells.

Recognition Criteria

List uniquely identified
Total recognition tests: 125

GuiButton (Collection)

Name equals 'btn[0]' X

Better Recognition of a Screen

- Do not declare **subscreens** as they will slow the loading of the screen.
- The **Wait** element is embedded within each activity. Therefore, you don't need to add it to the automation.
- Only use one **Must exist** element on each screen for recognition if two or more screens have the same criteria. Avoid using **Must not exist** element because it can slow the loading of the screen. Declaring either of these elements unnecessarily can also slow the loading of the screen.

Example:

Here are two screens **SAP Easy Access** and **SAP Easy Access 1** that have the same criteria.

Screen Information

Data and Criteria Advanced

*Name:
SAP Easy Access

*Identifier:
sAPEasyAccess

Technology: SAP GUI
Recognition Order: 1

Recognition Criteria

Screen uniquely identified
Recognition tests: 128

Name equals 'wnd[0]' X

Screen Information

Data and Criteria Advanced

*Name:
SAP Easy Access 1

*Identifier:
sAPEasyAccess1

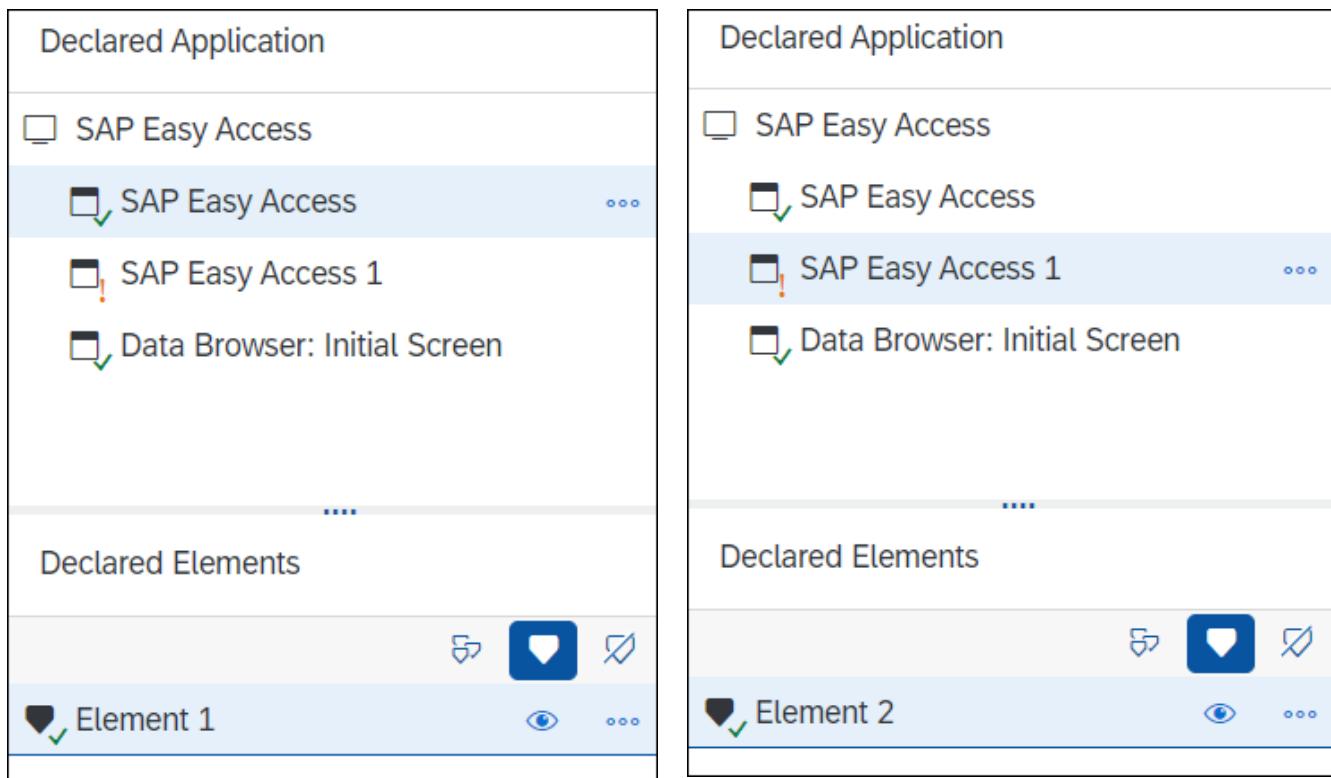
Technology: SAP GUI
Recognition Order: 2

Recognition Criteria

Another screen matches the criteria beforehand

Name equals 'wnd[0]' X

To uniquely identify the two screens, declare a **Must exist** element that is unique to each screen.

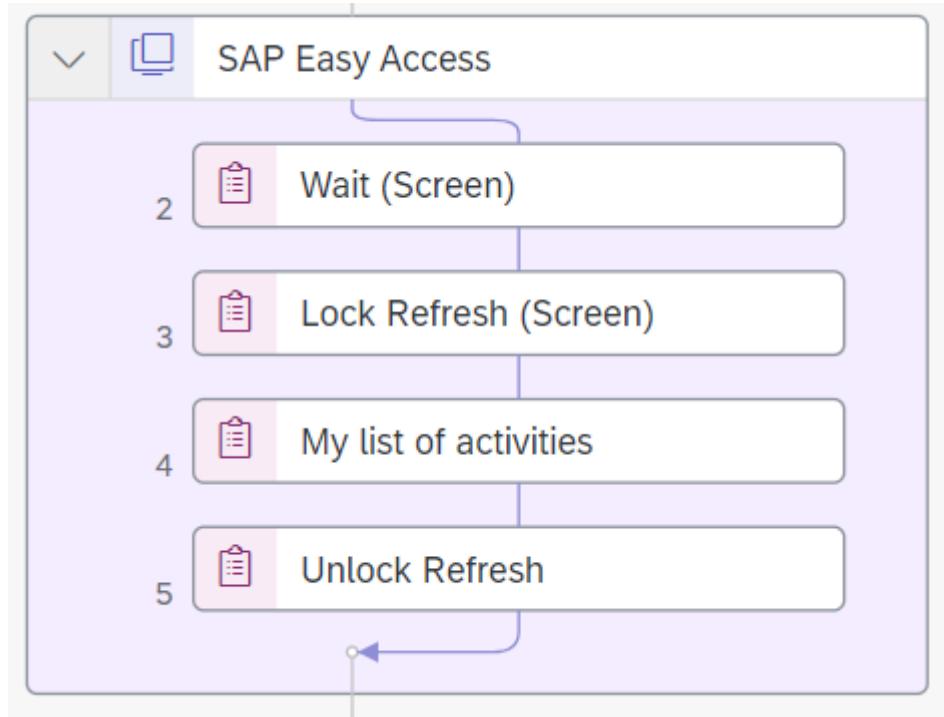


Here **Element 1** is unique to the screen **SAP Easy Access** and **Element 2** is unique to the screen **SAP Easy Access 1**.

Workflow

Set up your workflow by adding the following activities in this specific order for each page:

1. **Wait (screen)**.
2. **Lock Refresh (screen)**.
3. The list of activities on your screen.
4. **Unlock Refresh**



i Note

Ensure that you enclose your steps within **Lock Refresh** and **Unlock Refresh** calls for the following scenarios:

- While performing **Set** and **Get** operations on a **Collection** element
- When executing a set of steps on a screen where elements remain constant

Troubleshooting

If you are still receiving an error message despite implementing the best practices, then try the solutions provided below.

i Note

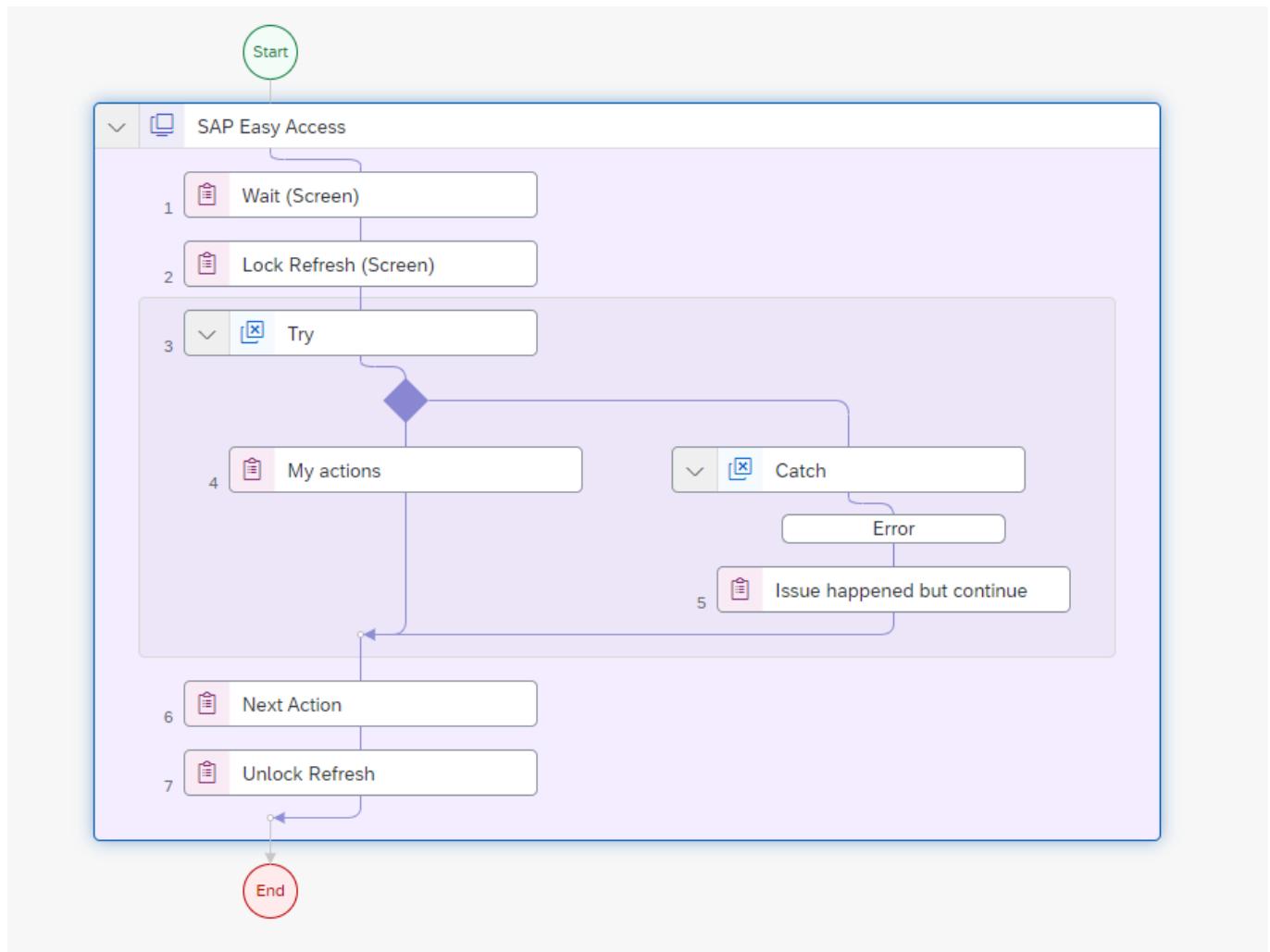
If there are multiple solutions, try the first one before moving onto the next.

Timeout for Text

This error occurs when an action is taking more than 7 seconds, even though the SAP session is not busy. For example, a slow system on a virtual machine.

Solutions

- If the action has been correctly completed but you're still receiving an error, add a **Try Catch** control around the activity, add the next activity after the **Catch** element.



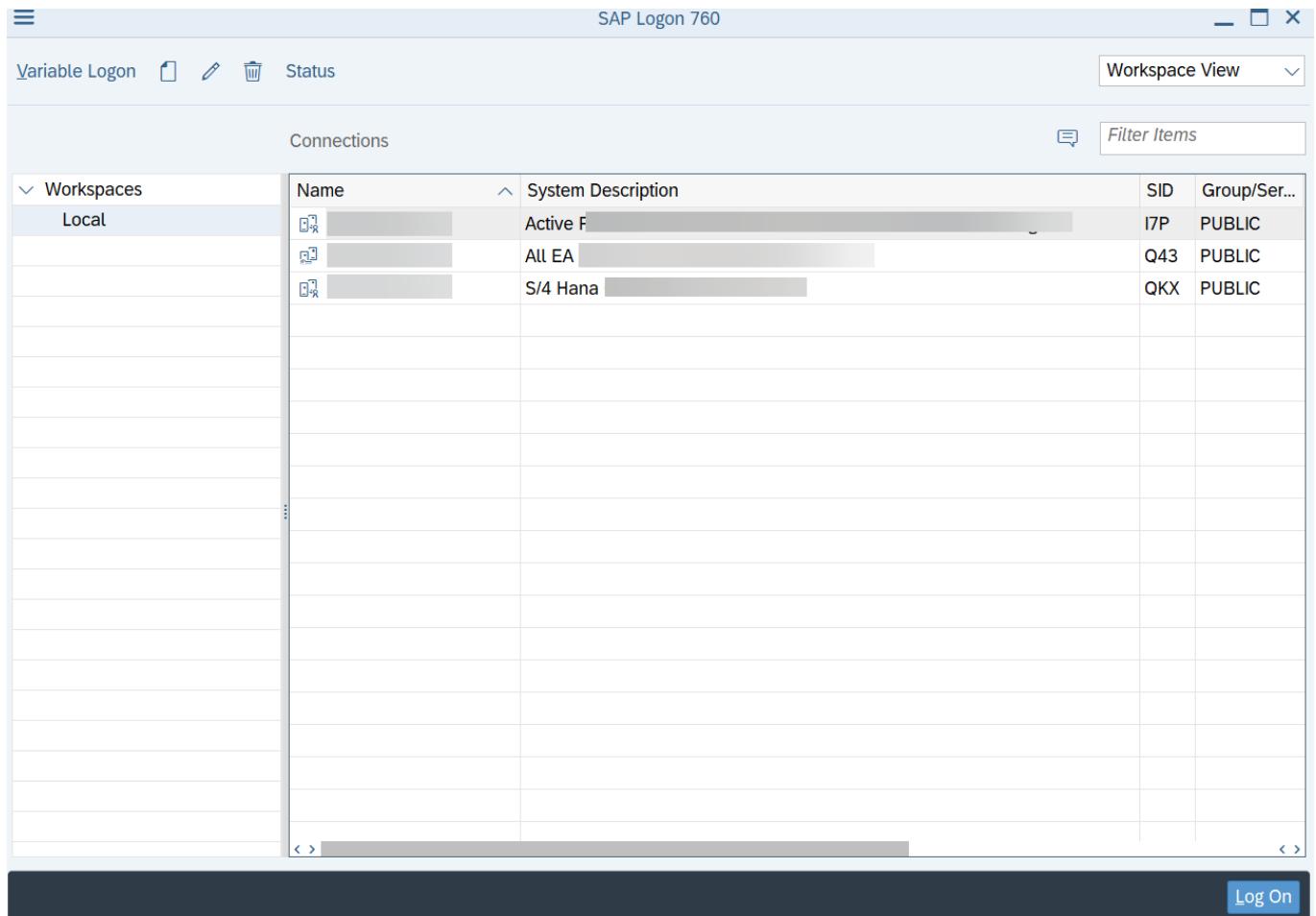
- If the Agent is running on a virtual machine, you might need to increase the disk capacity.

No Item Controller

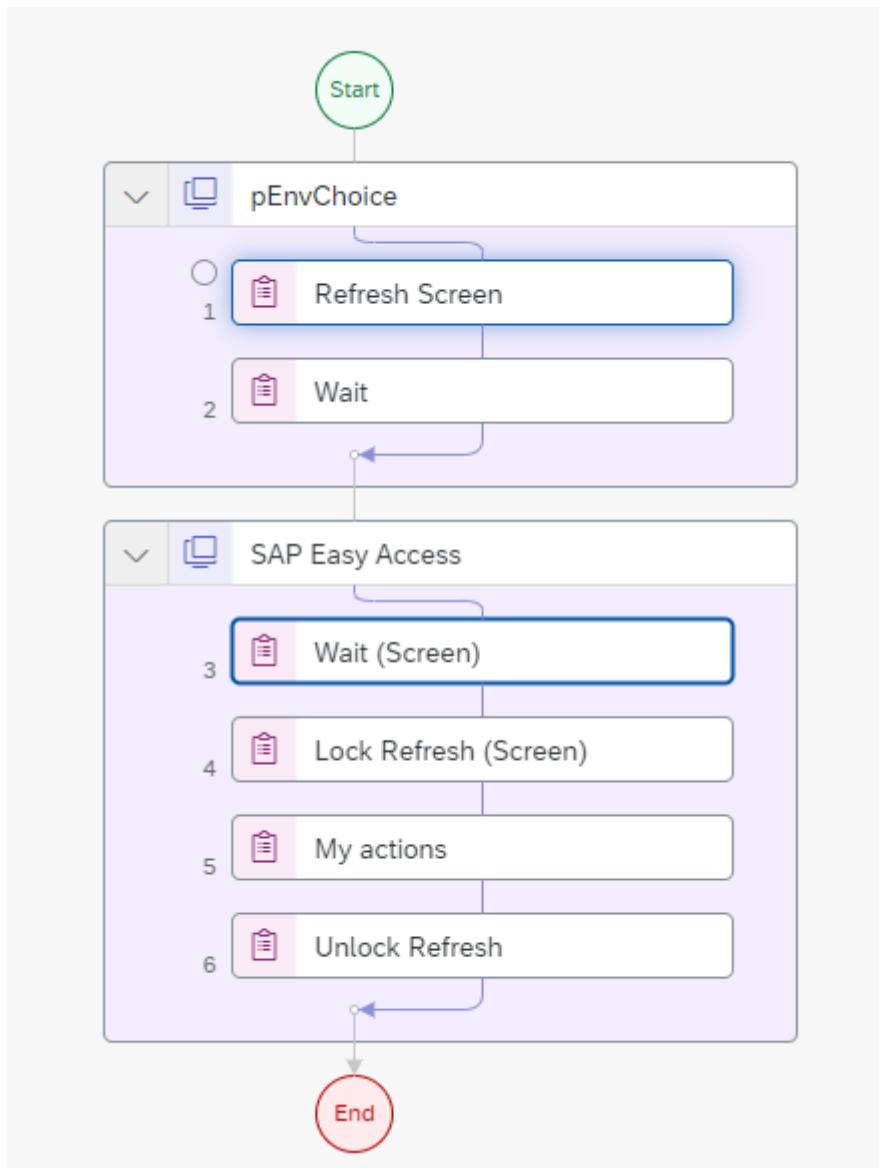
This error occurs when a page does not load correctly when the action is performed.

Solutions

- Refresh your system on Sap LOGON.



- Add a **Wait** element to your workflow.



Busy

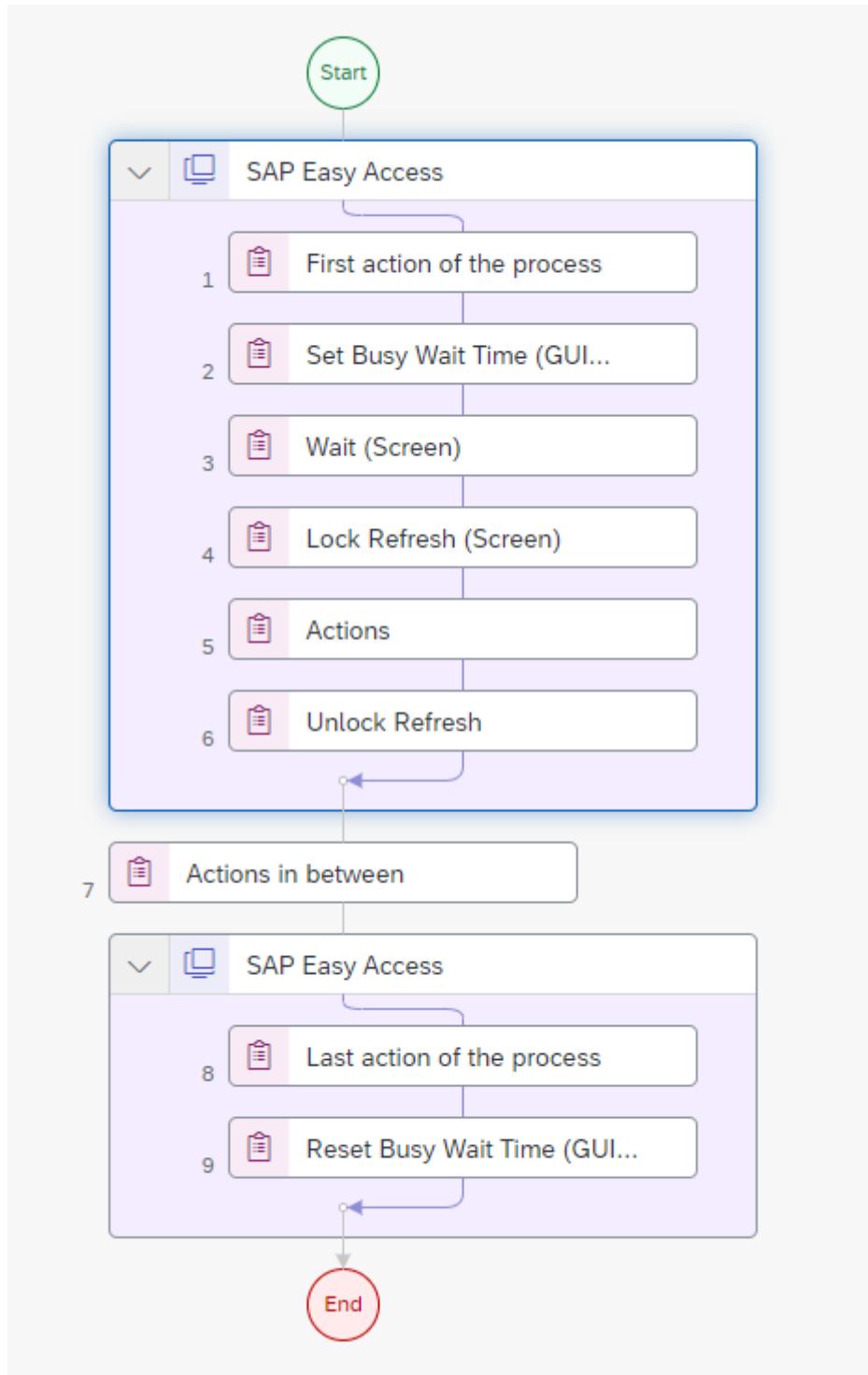
This error occurs when an action is sent to SAP GUI, but the transaction is busy loading pages or doing specific tasks. The agent will reattempt to perform the action when the transaction is not busy.

Solution

Add the **Set Busy Wait Time** activity before the process, and the **Reset Busy Wait Time** at the end to extend the wait period.

i Note

By default, the wait time parameter is 1 second, but you can modify it. All SAP GUI sessions opened during your process will then use that time to handle every action.



Wait Screen or Wait Element Timeout

This error occurs when the system is running slowly, and SAP GUI is taking longer than normal to recognize a page or element.

Solution

Use the **delay** or **timeout** Input Parameter to override the default timeout.

Wait (Element)

Step name*
Wait (Element)

Parameters Tester Advanced

Target*
SAPLogon > pEasyAcc... > btnCheck

Input Parameters

timeout:
40000

Wait (Screen)

Step name*
Wait (Screen)

Parameters Tester Advanced

Target*
SAPLogon > pEnvChoice

Input Parameters

waitReload:
B

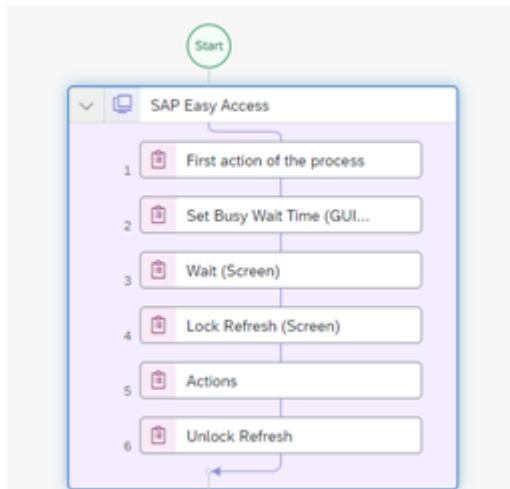
delay:
30000

API Timeout

This error occurs when an action generated by an activity takes longer than 12 seconds to complete.

Solutions

- Put the activity between a **Lock Refresh** and a **Unlock Refresh** element.



- If the action has been correctly completed but you're still receiving an error, add a **Try Catch** control around the activity, add the next activity after the **Catch** element.

Exception has been Thrown by the Target of an Invocation

This error occurs when one of the input parameter's activities is not set with the expected data.

Solution

Verify in the SDK documentation what value must be set for this parameter and change it accordingly.

The Java Technology

You can capture Java applications using the Java technology automation. It supports Java applications that are built using different frameworks such as Java Swing and JavaFx. You can automate Java applications running on Windows operating system.

Prerequisites

Refer to the following prerequisites to configure the system for the Java technology automation.

- The Desktop Agent 3 with V3 extension must be installed to ensure compatibility with the Java technology automation.
- You must install both the following versions of the Java technology.

- **To setup 32-bit Java JDK 11**

1. Download and install 32-bit Java JDK 11. You can refer to the <https://adoptium.net/temurin/releases/> link and choose the following parameters on the website: "Operating System: Windows", "Architecture: x86", "Package Type: JDK", "Version: 11-LTS".
2. Press "windows key (winkey)" and search for "Edit the system environment variable".
3. Press the "Environment variables..." button in the bottom right corner.
4. Under "User variables" create a new variable named as "JAVA_HOME_32" and paste the JDK bin path as value. For example, "C:\Program Files (x86)\Eclipse Adoptium\jdk-11.0.19.7-hotspot\bin".

- **To setup 64-bit Java JDK 11**

1. Download and install 64-bit Java JDK 11. You can refer to the [Sapmachine](#) website and choose the following parameters on the website: "SapMachine 11 (Long Term Support)", "JDK", "Windows x64 installer" and "sapmachine-11.0.19".
2. Press "windows key (winkey)" and search for "Edit the system environment variable".
3. Press the "Environment variables..." button in the bottom right corner.
4. Under "User variables", create a new variable named as "JAVA_HOME_64" and paste the Java JDK bin path as value. For example, "C:\Program Files\SapMachine\JDK\11\bin".

- The target machine must be rebooted for the environment variables to take effect.

i Note

Make sure that the JDK bin folder has a "client"\ "server" folder and "jvm.dll" in it.

Best Practices

The following are the best practices for the Java technology automation.

- Restart the Java Application, if the screen is not listed in the Cloud Studio after multiple refreshes.
- Use the **Terminate Application** activity from the Core SDK for the Java Automations instead of **Close Application** and **Close Screen** activities whenever required.
- Use the **UI Automation** or **OCR** technology for non-Java screens in the Java application. For example, Windows Dialogue Boxes (Open, Save As, and so on).

- In the Java automation, use the **Start Application** activity to start a Java application with exe file. You must provide the **exe** path into the **path** input field of the activity. If required for an application, you can provide parameter to launch the application in the **parameter** input field of the activity.

Target*	<input type="text" value="TestApp"/> X
Input Parameters	
path:	<input type="text" value="C:\AppPath\Launcher.exe"/> (i)
parameter:	<input type="text" value="9b2ddc5a-d05d2dbc9402"/> (i)

- In the Java automation, use the **Start Application** activity to start a Java application with jar file. You must provide the **javaw.exe** path into the **path** input field of the activity. The **javaw.exe** must be part of the JDK which is required to run the application. You can pass classpath and jar file as one string parameter in the **parameter** input field of the activity.

Target*	<input type="text" value="XmlControl"/> X
Input Parameters	
path:	<input type="text" value="C:\Users\JavaPath\bin\javaw.exe"/> (i)
parameter:	<input -classpath="" -duser.language="en" -jar="" \"c:\rpa\softwares="" and="" c:\java\="" javafx.web="" sdk\lib\"="" tools\somejavaapp.jar\""="" type="text" value="-cp \"/> (i)

Multiple parameter values can be passed as space separated single string in the **parameter** input field of the activity. For example, "param1 param2 . . .".

Limitations

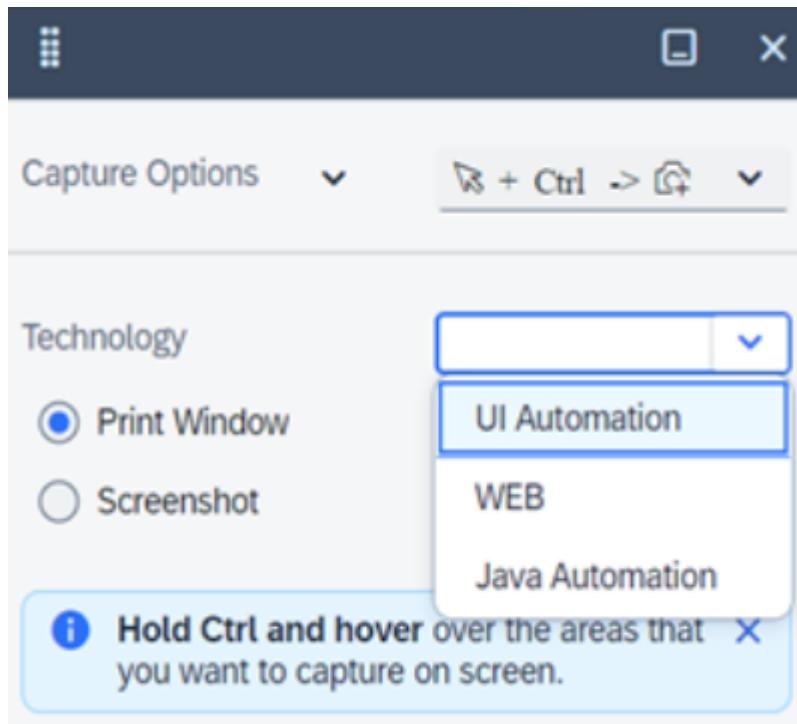
The following are the limitations for the Java technology automation.

- The Java technology automation capability is only available in the SBPA tenants.
- JavaFX Apps that use JavaFX versions earlier than 11 do not support the Mouse Click, Double Click and Keystroke activities for Java Element.
- "Labeled by" criteria is not supported.
- Polling delay configuration is not supported.
- Recorder is not available for Java application.
- The Java technology automation supports 100% DPI (dots per inch) resolution.
- If you use the **Start Application** activity and the application is already launched, the automation doesn't bring the application into focus.

Support Capture on Hover

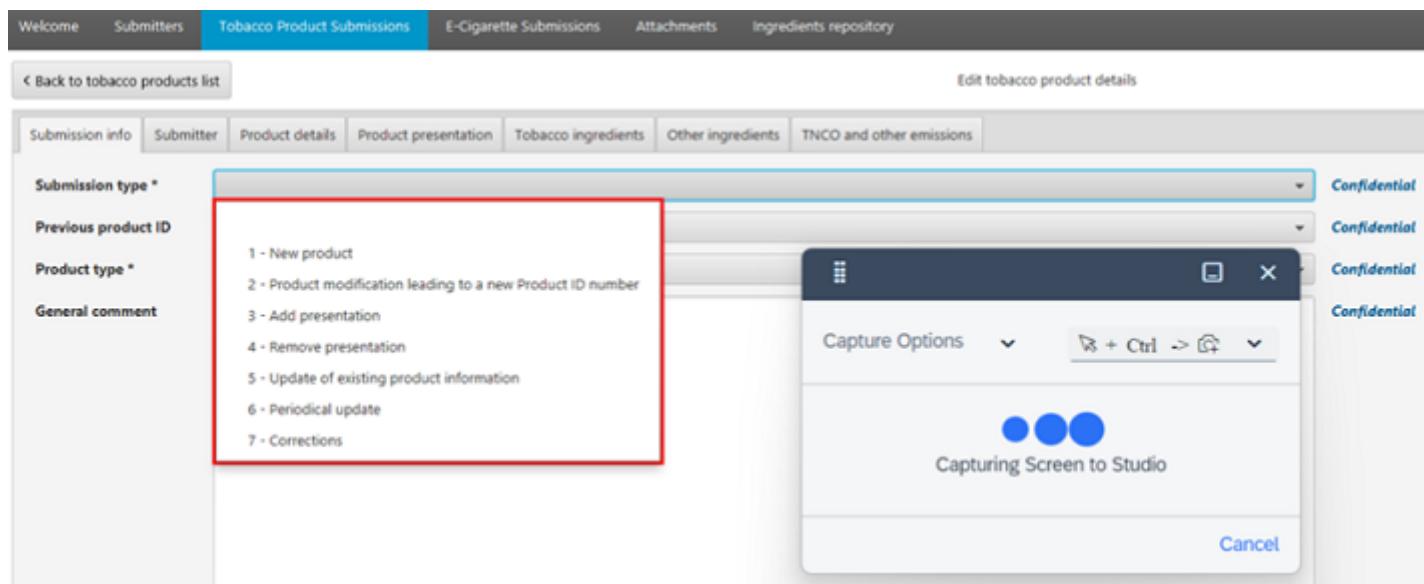
The Capture on Hover feature enhances the Java automation, making it easier to capture disappearing controls such as pop-ups and menu-type controls. This feature is available for the Java applications that are built using Java Swing and JavaFX frameworks.

After launching the capturing tool in the Cloud Studio, you can find the **Java Automation** option in the **Technology** dropdown.



The **Technology** of the window that the cursor is hovering over (the topmost application) is automatically detected. Alternatively, you can choose the option from the dropdown menu.

To ensure proper capture, we recommended that you place the cursor on the control that needs to be captured, and then press either **Ctrl** or **Shift**. When the capture is activated, the window that is being hovered over is highlighted by a red bounding rectangle.



Once the capture is completed, the same window is exported to the Cloud Studio, where you can declare the required elements from that control.

Design Automations

Automations are composed of a succession of steps you build in the Cloud Studio with the automation designer. An automation can orchestrate multiple activities on different applications and screens available on a specific computer.

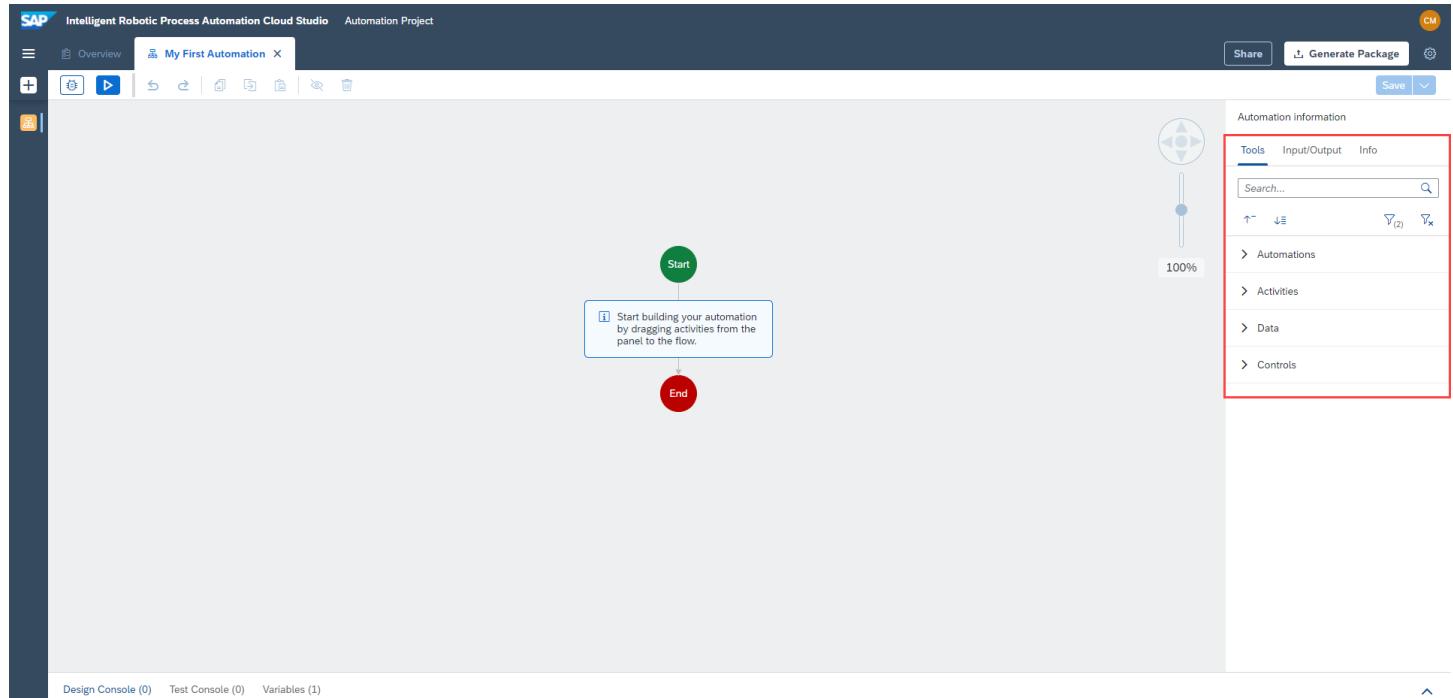
An automation is executed on the Desktop Agent of your local machine. Creating an automation is similar to designing a scenario in the Desktop Studio.

i Note

Automations are not available in projects with a Desktop Package.

Tools

To build an automation, you have access to a list of tools from the side panel.



- Automations

You can use a previously created automation from this cloud project, as a tool to include in the flow of another automation.

- Screens

An Application Screen is a capture of any application. For more information, see the section about [Capture and Declare Applications](#).

- Activities

Activities are used to build the workflow of your automation. They come from the SDK packages imported in the Cloud Studio the first time you create an automation. For more information, see the section about [Automation Activities provided by SDK Packages](#)

- Data Types

A data type is a complex data used to describe a data structure. It can be defined as an input or an output and used throughout your automation. For more information, see [Data Types](#).

- Controls

Controls allow you to add tools such as conditions, loops, and scripts to your automation. For more information, see [Add a Control to an Automation](#).

	Condition	Inserts multiple situations defined by an expression, that determines the following step.
	End	Stops the automation.
	For Each	Inserts a sequence of actions to perform on a list of objects such as users.
	Forever	Repeats the step in a loop until it meets the required conditions.
	Repeat	Repeats the step for a defined number of times.
	Loop End	Stops a step set in a loop.
	Screen switch	Inserts multiple situations defined by the screen the user is working on, that determine the following step.
	Custom script	Inserts a step defined by custom Javascript mode.
	Try	Checks if errors occur in a sequence of actions and defines a behavior depending on the error type.
	Stop automation in error	Stops the automation in an error state.

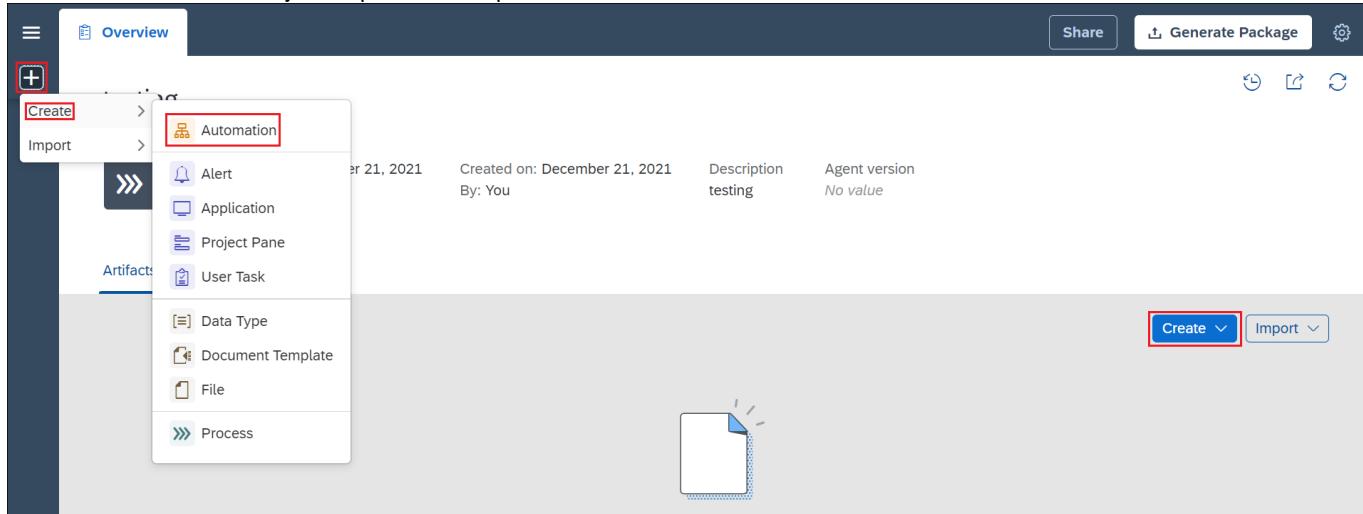
Create an Automation

Context

Perform the following steps to create an automation:

Procedure

1. In the Project Explorer, click the icon, select **Create** and then, **Automation** in the left-hand panel. You can also click the **Create** button in the Project Explorer main panel and select **Automation**.



2. **Optional:** When adding an automation in the project for the first time, you must define the target agent version. For more information, see [Configure an Agent Version](#).

3. Enter a name in the **Name** field.

Create Automation

*Name:

e.g. My first automation

*Identifier:

e.g. MyFirstAutomation

Description:

e.g. An automation that books flights

Can only be started from another automation:



Create

Cancel

i Note

By default, the identifier is the same as the name. An identifier is mandatory for every artifact in the Cloud Studio, so you need to manually enter an identifier if it's not already pre-filled.

4. **Optional:** Edit the **Identifier** field (without space).

i Note

The identifier is the name of the automation that can be referenced in custom scripting.

5. **Optional:** Enter a short description about the automation in the **Description** field.

6. **Optional:** Enable the **Can only be started from another automation** option. When active, this option prevents you from starting this automation independently. Attended and unattended triggers will not be available. For more information about triggers, see the dedicated section in the [Factory User Guide](#).

If you want the automation to run only through another automation inside your project, the **Can only be started from another automation** option must be enabled.

Create Automation

*Name:

e.g. My first automation

*Identifier:

e.g. MyFirstAutomation

Description:

e.g. An automation that books flights

Can only be started from another automation:



Create

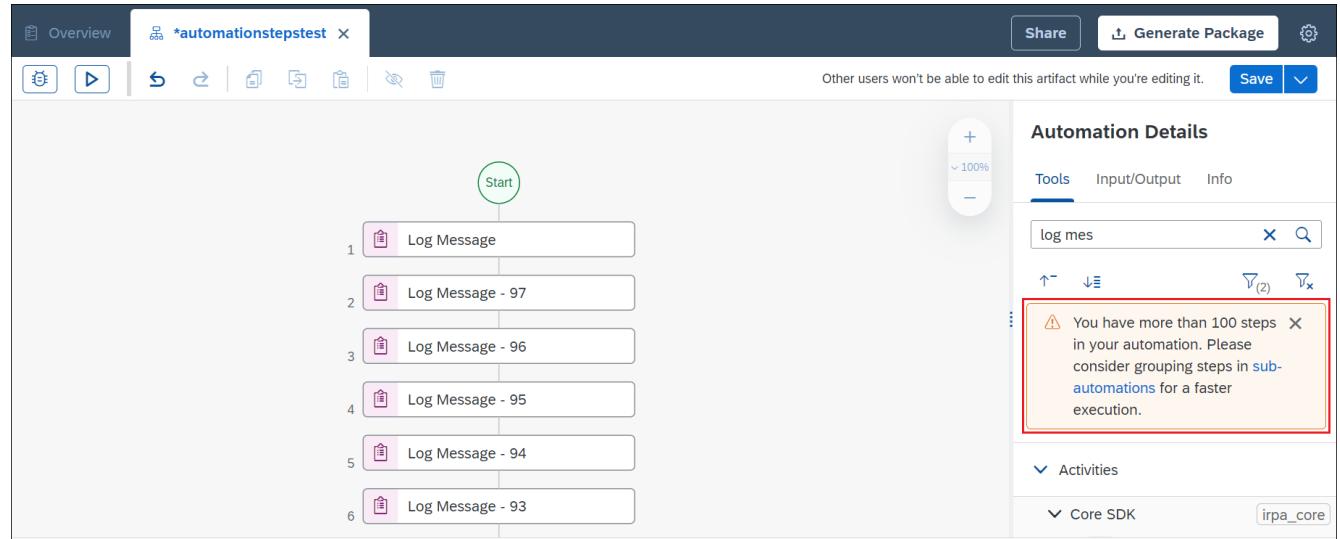
Cancel

7. Click **Create**. A new tab opens in the main panel of the Cloud Studio.

i Note

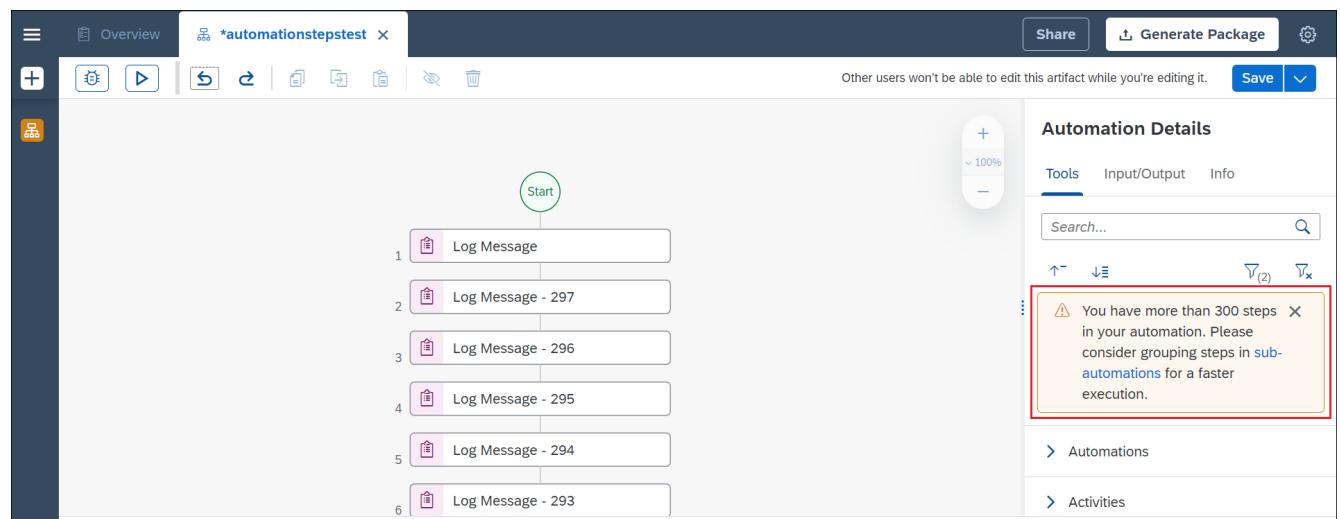
Once the automation is created, you can add steps to it. Currently, up to 99 steps can be added without any change in the step bubble.

If you add the 100th step, a warning message is displayed at the right panel of the Cloud Studio.

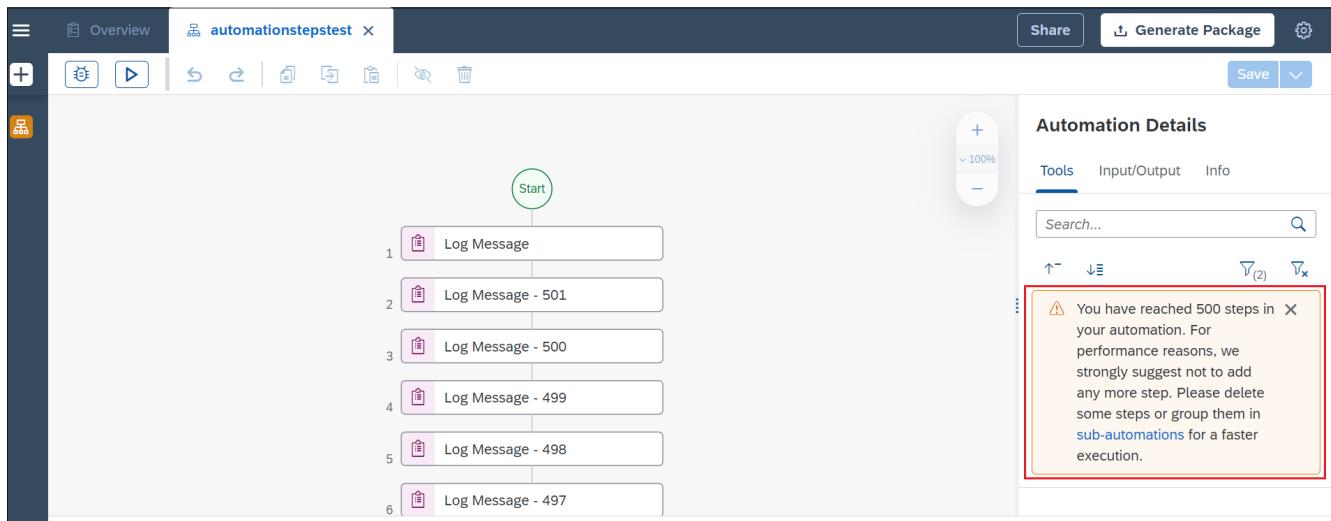


You can close or ignore the warning message and continue adding the steps. It is recommended to create a sub-automation. For more details on creating a sub-automation, see [Generate a Sub-Automation](#).

If you reach the 300th step, a warning message is displayed at the right panel of the Cloud Studio.

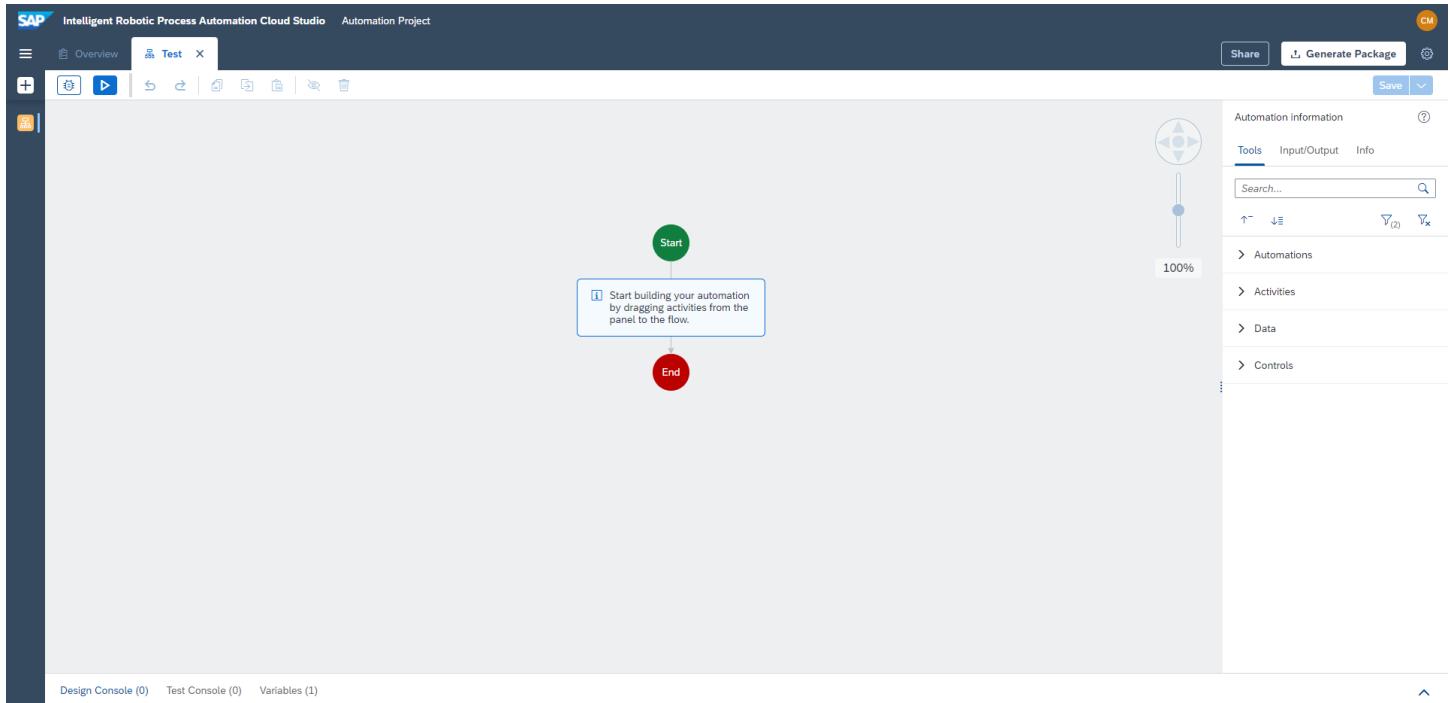


If you reach the 500th step, a warning message is displayed at the right panel of the Cloud Studio. You cannot add more than 500 steps. Instead, you can create a sub-automation.



Results

The automation designer opens and you can now drag and drop [activities](#), [data types](#), [controls](#), [applications](#), and even other automations from the side panel into the workflow to build your automation in the Cloud Studio.



Automation Tools Panel Filters

Automation tools panel filters allow you to find required items like **Automations**, **Screens**, **Activities**, **Data**, and **Controls**.

Automation Tools Filters

Automation information ?

Tools Input/Output Info

🔍

↑-
↓≡
✖₍₂₎
✖_x

> Automations

> Activities

> Data

> Controls

Symbol	Name	Description
□	Collapse all	This is used to collapse all items.
□	Expand	This is used to expand all items. i Note Consider you keep some items open and some close. When you do Collapse all and then Expand , only the items you kept open earlier are expanded. All items will not be expanded. This option expands all items only at the first instance.
□	Filter result	By default, the filter result is applied. This includes two options: <ul style="list-style-type: none"> • Contextual filtering - Filters the result depending on the selection. • Include dependencies - This option includes the dependencies.
□	Clear filter	This is used to clear Filter result filter.

Search Feature

When you start typing in the search bar of the tool panel, all the items matching with your input are displayed in bold.

Automation Details

Tools Input/Output Info

The screenshot shows the 'Automation Details' interface. At the top, there is a search bar with the text 'mo' and a magnifying glass icon. Below the search bar are navigation icons: up and down arrows, a refresh symbol, and two blue Y-shaped icons labeled '(2)' and 'x'. A horizontal line separates the top header from the main content area. The main content area has a light gray background and contains a list of activities. On the left, there is a vertical sidebar with a 'Activities' section header and a small downward arrow icon. The list of activities is organized into two main categories: 'List Management' and 'Mouse'. Each category has a header with a downward arrow icon and a 'irpa_core' button. Under 'List Management', there are two items: 'Remove by Index' and 'Remove Item', each with a small info icon. Under 'Mouse', there are seven items: 'Mouse Left Click', 'Mouse Middle Click', 'Mouse Right Click', 'Drag & Drop', 'Drag & Drop Right', 'Mouse Move', and 'Scroll Wheel', each also with a small info icon.

- mo ✖️ 🔎
- ↑ ⏵ Y₍₂₎ Y_x
- Activities**
- List Management** irpa_core
 - Remove by Index ⓘ
 - Remove Item ⓘ
- Mouse** irpa_core
 - Mouse Left Click ⓘ
 - Mouse Middle Click ⓘ
 - Mouse Right Click ⓘ
 - Drag & Drop ⓘ
 - Drag & Drop Right ⓘ
 - Mouse Move ⓘ
 - Scroll Wheel ⓘ

Favorites Section

You can place activities and controls that you frequently use as **Favorites**.

To do so, hover over the activity or control that you want to add to your **Favorites** and click the □ icon.

Automation Details

Tools Input/Output Info

Search... 

> Automations

> Screens

Activities

Automation Alerting 

 Raise Alert  

Converters (JS, JSON, ... 

 JS to XML 

 JSON to XML 

 XML to JS 

 XML to JSON 

The activity or control will be pinned at the top of the Tools panel in the **Favorites** section.

Automation Details

The screenshot shows the SAP Cloud Studio interface for managing automations. The left sidebar is visible, featuring a search bar at the top, followed by navigation icons (up, down, filter, search), and a red-bordered section for 'Favorites'. Below 'Favorites' are sections for 'Activities' (containing 'Raise Alert', 'Condition', and 'For Each') and 'Controls'. At the bottom of the sidebar are sections for 'Automations' and 'Screens'. The main content area is partially visible.

- Tools Input/Output Info
- 🔍
- ↑↓ Filter (2) X
- ▼ Favorites
 - ▼ Activities
 - 📋 Raise Alert ★ ⓘ
 - ▼ Controls
 - ◇ Condition ★ ⓘ
 - ⬇️ For Each ★ ⓘ
 - Automations
 - Screens

To remove activities or controls from the **Favorites** section, click the ☒ icon.

Manage Data within an Automation

1. [Input/Output Parameters](#)

In the Cloud Studio, input and output parameters allow you to exchange data in the workflow of your automation between activities, screens, and scripts.

2. [Define Complex Input Parameters](#)

You can use input complex parameters by entering the data that defines an object directly in the **Input Parameters** of an automation.

3. [Data Types](#)

A *data type* is an artifact describing a data structure that can be used as an input and/or output parameter in automations or processes.

4. [Create and Update Variables](#)

The variables you use in the Cloud Studio to build your automation are data storage that has a name, a type (example: string, list of string or data type), and a value. A variable in the automation is also associated to a step represented by its number.

5. [Different Types of Data Fields](#)

6. [Expression Editor](#)

Input/Output Parameters

In the Cloud Studio, input and output parameters allow you to exchange data in the workflow of your automation between activities, screens, and scripts.

What are Input/Output Parameters?

An input or output parameter is a variable that is passed, received, or sent from one automation, SDK activity or control (for example the condition) to another. This variable allows you to manipulate data that you can use in your workflow. Input or output parameters have a name (optionally a description) and data that complies to a type. Its type can be defined as:

- A simple type variable such as a string, a boolean, a number.
- A more complex type variable such as a [Data Type](#).

For more information on [Data Types](#), see [Data Types](#).

- A variable of type "any".

Input and output parameters can be defined for later use in your project, or for direct use to set and get data. In the Cloud Studio, you can:

- Define the input or output parameters of the automation or custom script your are building.

i Note

When you define an input or output parameter, you create an empty variable with a type and no value.

- Set the value of the input parameter, or get the value of the output parameter of an activity, an automation or a control.

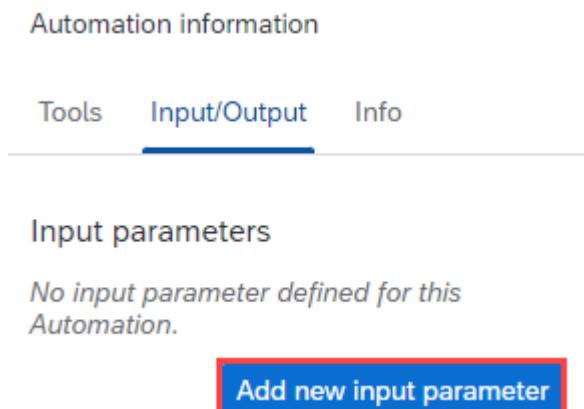
i Note

Defining input and output parameters in your automation is optional.

Creating Input/Output Parameters in an Automation

Create an Input Parameter

1. In your automation, on the right-hand side panel, go the [Input/Output](#) section and click [Add new input parameter](#).



2. Enter a name.

⚠ Caution

You can enter alphabetical and numerical characters:

- the first character must be alphabetical

- o except for the underscore, special characters are not allowed (including blank space characters)

3. **Optional:** Enter a description.

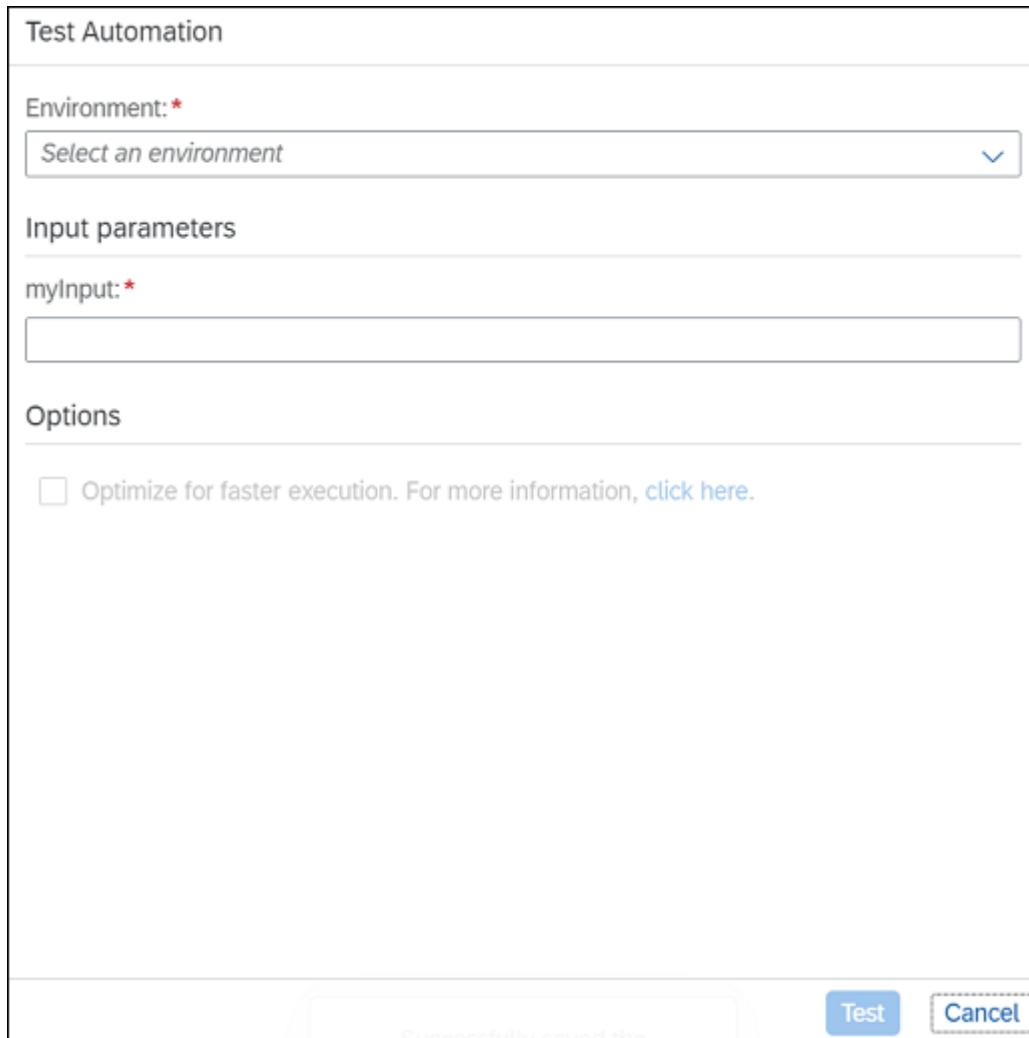
4. Click on the dropdown menu to select a type (simple type or data type).

5. **Optional:** Select the checkbox **List** to have the type presented as an array (for example a list).

6. Click **Save**.

i Note

If you click  , the newly created input parameter is displayed as a mandatory input parameter field under the **Input parameters** section of the **Test Automation** popup window.



The screenshot shows the SAP Test Automation dialog box. At the top, it says "Test Automation". Below that, there's a section labeled "Environment:" with a dropdown menu set to "Select an environment". Under "Input parameters", there is a field labeled "myInput:" which is currently empty. In the "Options" section at the bottom, there is a checkbox labeled "Optimize for faster execution. For more information, [click here](#)". At the very bottom right of the dialog are two buttons: "Test" and "Cancel".

You can also view a help () icon under the **Input parameters** section of the **Test Automation** popup window. If you click , the description of the newly created input parameter is displayed. Therefore,  icon is displayed if you add description to the input parameter.

Test Automation

Environment: *

Select an environment

Input parameters

myInput: *

this is an input

Options

Optimize for faster execution. For more information, [click here](#).

Test Cancel

Create an Output Parameter

1. In your automation, on the right-hand side panel, go the **Input/Output** section and click **Add new output parameter**.

Automation information

Tools **Input/Output** Info

Input parameters

No input parameter defined for this Automation.

Add new input parameter

Output parameters

No output parameter defined for this Automation.

Add new output parameter

2. Enter a name.

⚠ Caution

This is custom documentation. For more information, please visit the [SAP Help Portal](#)

You can enter alphabetical and numerical characters:

- the first character must be alphabetical
- except for the underscore, special characters are not allowed (including blank space characters)

3. Optional: Enter a description.

4. Click on the dropdown menu to select a type (simple type or data type).

5. **Optional:** Select the checkbox **List**, to have the type presented as an array (for example a list).

6. Click **Save**.

Use Case Example

The following use case demonstrates how the data in your automation is passed from one step to another with the input and output parameters.

Prerequisite

You have created the automation Test01.

Procedure

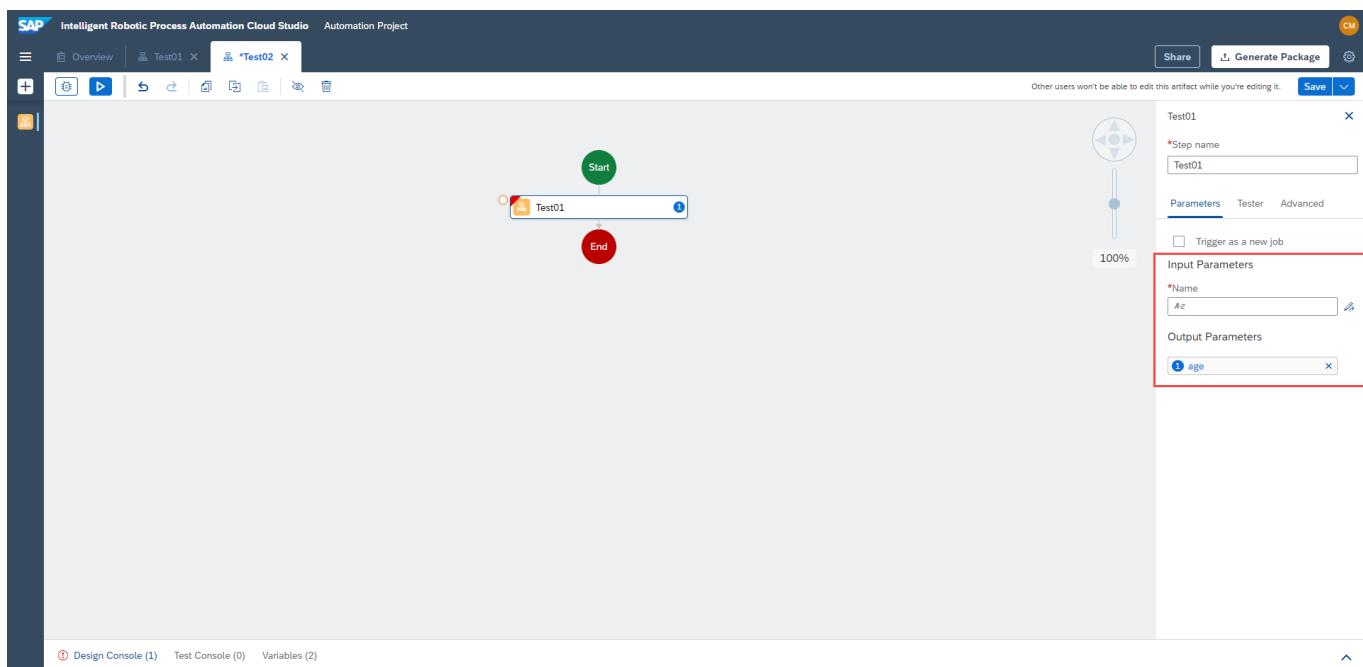
1. In your automation, on the right-hand side panel, go to the **Input/Output** section.

2. Define the input and output parameters of the automation.

a. Add the input parameter **name** of type string.

b. Add the output parameter **age** of type number.

3. Create a second automation Test02 and drag and drop Test01 in the workflow of the automation.



As a result:

- the input parameter of the first step is a name and calls for an input of type string
- the first step returns the output parameter **age**

i Note

You can only set a value of a defined type. For example, in the case of a type string, you cannot set a numeric value.

To set your input parameter to the correct type, you can edit it in the expression editor. For more details, see [Expression Editor](#).

- Set a constant value in the input parameters and click on the second value.

Input Parameters

*Name

"Foo"

+ Foo

1 age

Note that the first text suggestion next to the blue mark creates the input parameter for Step0, which is the input parameter of the automation.

i Note

In the workflow of the automation, each step has a number. This number indicates from which step an input or output parameter comes from.



- Drag and drop the activity Log.

- Set a constant value in the input parameters.

Parent topic: [Manage Data within an Automation](#)

Next: [Define Complex Input Parameters](#)

Reserved Words

You cannot use reserved words as input or output parameter in the Cloud Studio (such as in automation, user task, process, or custom script).

For example, if you use “function” as an input parameter in an automation, the following message is displayed as it is a JavaScript reserved word:

⚠ Caution

Reserved words are not allowed. You need to use a prefix to make the name valid.

Reserved words are categorized into three parts:

- JavaScript Reserved Words
- Old JavaScript Reserved Words

3. Other Reserved Words

JavaScript Reserved Words

In IRPA you cannot use the following list of JavaScript reserved words:

abstract	arguments	await	boolean
break	byte	case	catch
char	class	const	continue
debugger	default	delete	do
double	else	enum	eval
export	extends	false	final
finally	float	for	function
goto	if	implements	import
in	instanceof	int	interface
let	long	native	new
null	package	private	protected
public	return	short	static
super	switch	synchronized	this
throw	throws	transient	true
try	typeof	var	void
volatile	while	with	yield

Old JavaScript Reserved Words

In IRPA you cannot use the following list of old JavaScript reserved words:

abstract	boolean	byte	char
double	final	float	goto
int	long	native	short
synchronized	throws	transient	volatile

Other Reserved Words

You must avoid using the following list of reserved words that are comprised of JavaScript and SDK built-in objects, properties, and methods:

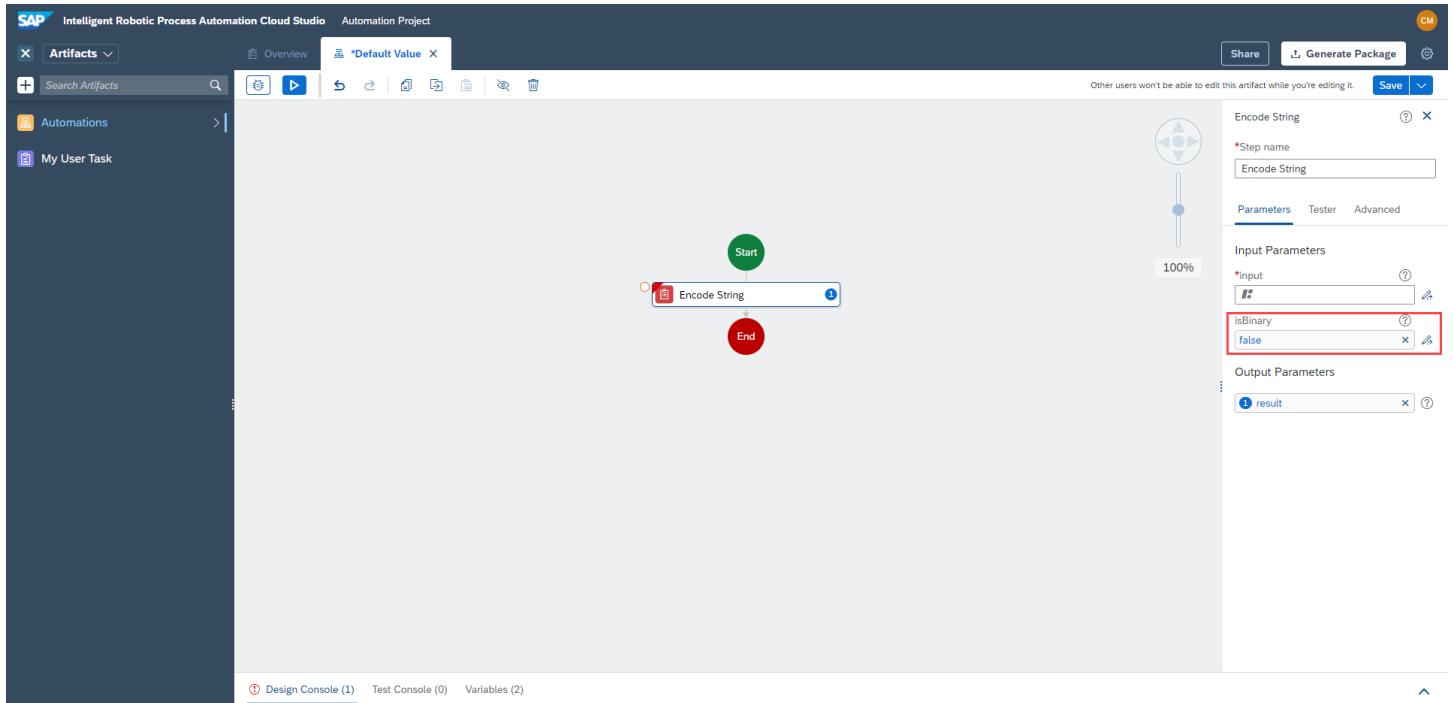
Array	Date	eval	function
hasOwnProperty	Infinity	isFinite	isNaN
isPrototypeOf	length	Math	NaN

name	Number	Object	prototype
String	toString	undefined	valueOf
irpa	require		

Activity Default Value

For some activities in the Cloud Studio, input comes with a default value. It means that the default value will be used if the user does not provide any other value.

You can see a default value of an activity at design level. Refer to the below screenshot.



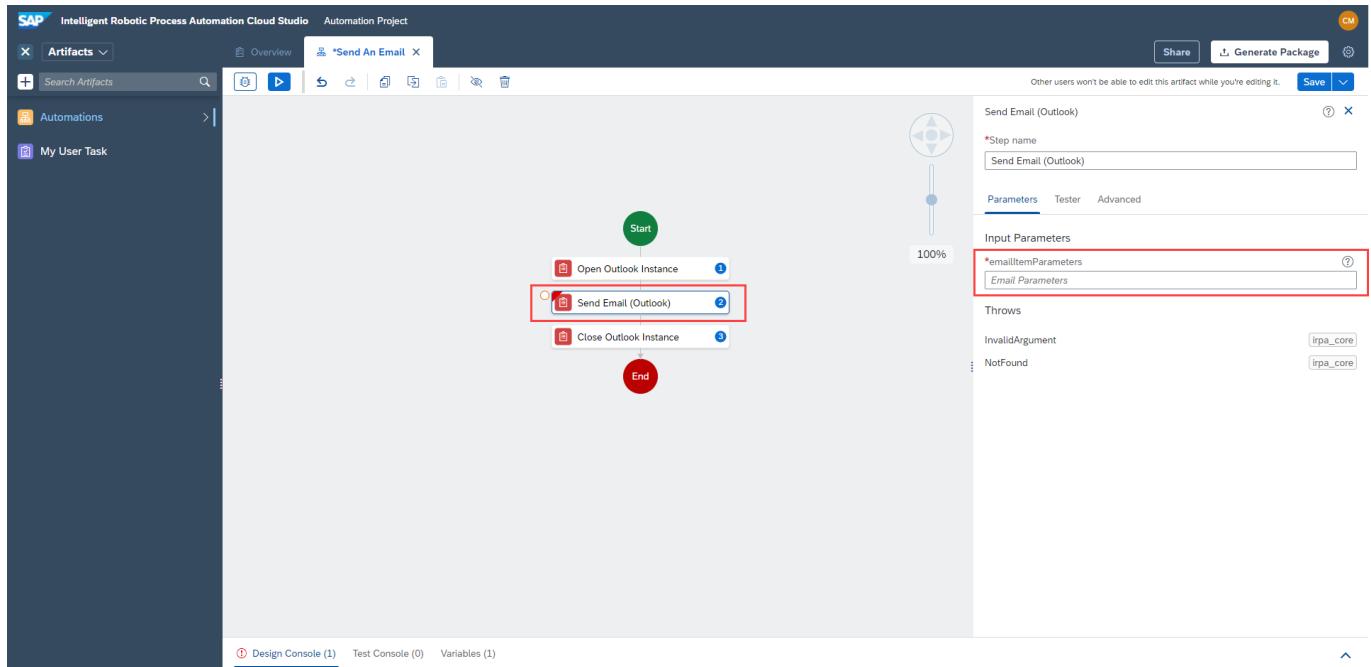
At any point in time, you can update the default value with an appropriate value.

Define Complex Input Parameters

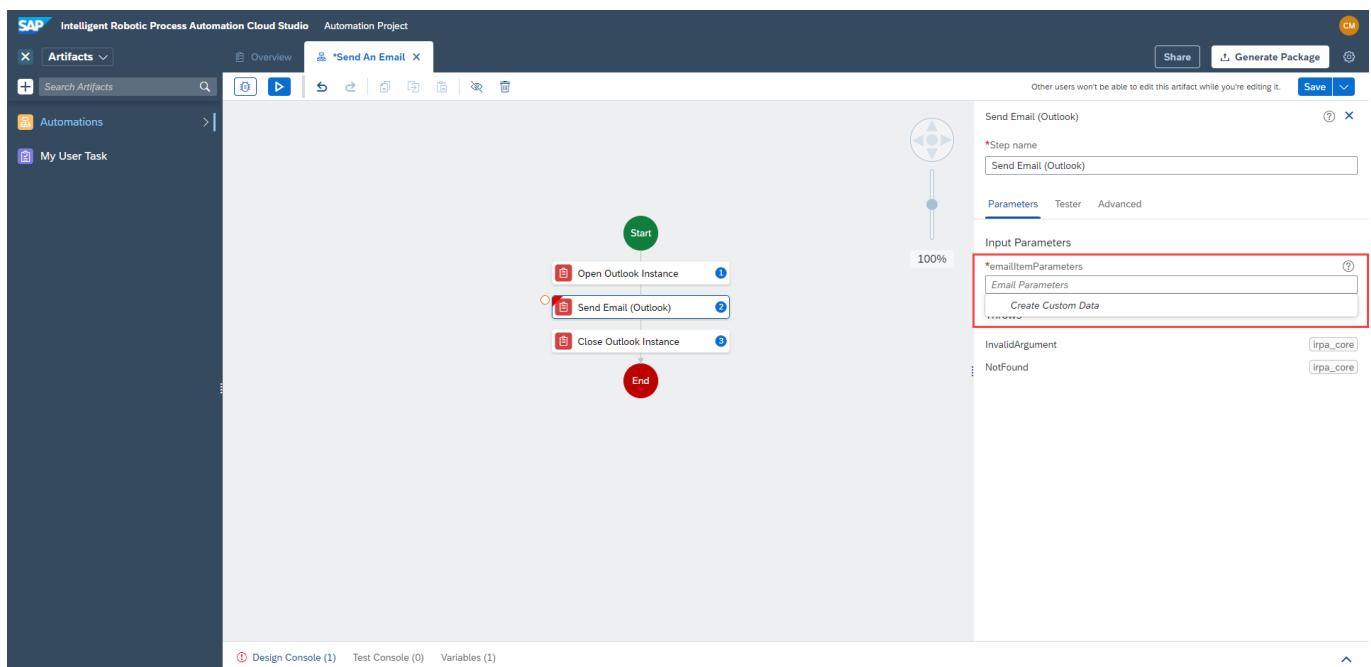
You can use input complex parameters by entering the data that defines an object directly in the **Input Parameters** of an automation.

Procedure

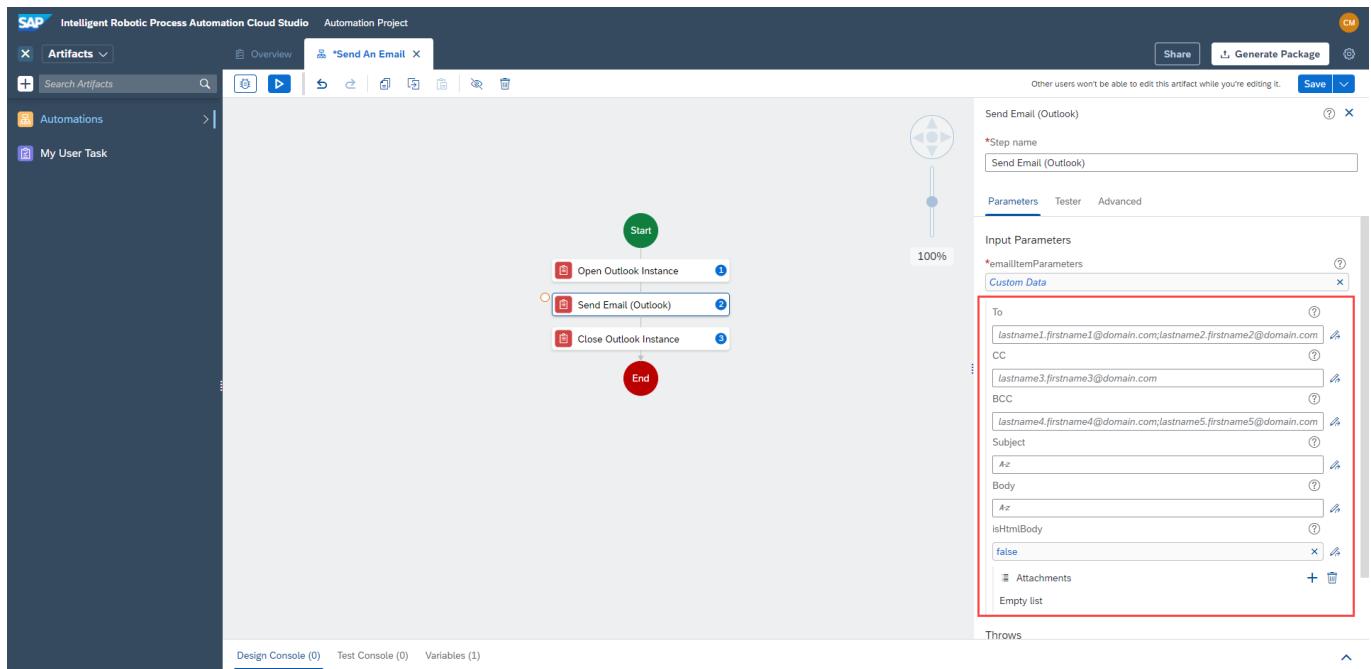
1. In your Automation, select the activity which requires complex input parameters. The activity input parameter field is displayed.



2. Click in the activity **Input Parameters** field and then select **Create Custom Data**.



The Activity Data Panel is displayed.



3. Provide the required data directly in Activity Data Panel.

i Note

You can define a complex input parameters using the expression editor. For more details about the expression editor refer to the [Expression Editor](#) section. Referring to this section helps you to define complex input parameters easily.

If you want to provide the required data or edit the provided data using expression editor, follow the below procedure:

- Click the at the right side of the required field.

The [Edit Expression](#) dialog opens.

You can manually type your expression.

- Click **Test** to check your expression is valid.

- Click **Save Expression**. The data type is now defined and the Activity Data Panel is displayed with the provided or updated data.

Parent topic: [Manage Data within an Automation](#)

Previous: [Input/Output Parameters](#)

Next: [Data Types](#)

Data Types

A *data type* is an artifact describing a data structure that can be used as an input and/or output parameter in automations or processes.

By default, three types of data are available in the Cloud Studio: string, number, and boolean. For more complex data, you need to use *data types*.

Data types enable you to better formalize and describe the data used as input/output parameters for steps, activities, skills, processes, scenarios, triggers, or notifiers. Data types facilitate the manipulation and validation of data.

You can create a data type manually in the Cloud Studio, but some data types can also be created automatically when SDK packages or Desktop Studio scenarios are imported.

Parent topic: [Manage Data within an Automation](#)

Previous: [Define Complex Input Parameters](#)

Next: [Create and Update Variables](#)

Related Information

[Manage Data within an Automation](#)

Create a Data Type

Context

You can manually create a data type in the Cloud Studio.

Procedure

1. In the Home view or in the Project Explorer, click the **Create** button and select **Data Type**.
2. Enter a name and description, and click **Create**.

i Note

A unique identifier is automatically created from the name you entered. If not, you must manually create an identifier.

3. A new tab opens in the main panel of the Cloud Studio. You can now add the fields that will be included in your data type: Click the **New Field** button.
4. In the **Field Properties** on the right, enter the name of the field you just added, and select its type.

i Note

You can insert several types of field in a data type:

- o Simple fields of type string, number or boolean.
- o More complex fields of type object, or other existing data types (imported from SDK packages or from Desktop Studio scenarios).

5. **Optional:** Click the **New Child** button to add children to the field.

i Note

Once you add children to a simple field it automatically becomes a field of type object.

6. **Optional:** If you want to manage a field as an array containing multiple values, check the **List** box in the **Field Properties**.
7. Save your changes.

Create a Strict Data Type

At any point in time you can manually create a **Strict** data type in the Cloud Studio.

If an object is defined using a strict data type, you cannot add other properties to this object (using the [Custom Scripting](#) for instance) other than the ones defined in the data type. If you provide any additional input properties, at runtime, the automation will fail.

In the [Create a Data Type](#) topic, you have learned how to create a data type.

To create a strict data type, you must select the **Strict** option.

The screenshot shows the SAP Intelligent Robotic Process Automation Cloud Studio interface. On the left, there's a sidebar with 'Content' and 'Dependencies' tabs, and a 'Data Types' section containing 'StrictDatatype'. A message box says 'Drop a file here to add it to your project.' In the main area, a table lists a single row for 'StrictDatatype' with columns for Name, Type, Sample, List, Required, and Data Type information. The 'Data Type information' section includes fields for Name, Identifier, Description, and a 'Last Update' timestamp. At the bottom right of the table, there's a 'Strict' checkbox which is checked and highlighted with a red box. Below the table, there are tabs for 'Design Console', 'All', 'Errors (1)', 'Warnings (0)', and 'Info (0)'.

Save your changes. The **Strict** data type is created.

Data Types from SDK Packages

By default, some pre-defined data types exist in the SDK packages and the data types are listed when you import the SDK packages in the Cloud Studio. Every data type coming from SDK are strict and you cannot change the strict data type. You can create variables using the pre-defined data types from the SDK package.

The screenshot shows the SAP Intelligent Robotic Process Automation Cloud Studio interface. On the left, there's a sidebar with 'Content' and 'Dependencies' tabs, and a 'Data Types' section containing 'Add Job Result', 'Data Object Validation', 'Job Parameters', 'Quality for tif or jpg', 'Simple Info Dialog Parameters', 'Simple Input Dialog Parameters', and 'Window Position'. These items are highlighted with a yellow box. A message box says 'Drop a file here to add it to your project.' In the main area, a table lists these data types with columns for Name, Type, Sample, List, Required, and Data Type information. The 'Data Type information' section includes fields for Name, Identifier, Description, and a 'Last Update' timestamp. At the bottom right of the table, there's a 'Strict' checkbox which is checked and highlighted with a red box. Below the table, there are tabs for 'Design Console', 'All', 'Errors (0)', 'Warnings (0)', and 'Info (0)'.

Strict Data Type Property

If a data type is strict, then you must not provide any additional input properties.

If you are working with a strict data type, then you must define the data exactly as the data type.

In the following screenshot shows the information message, not to provide any additional input properties for a strict data type.

Test Automation

Input parameters

The data is of type strict, this means that you cannot provide additional properties.

```
{
  "object1": {}
}
```

[...] nosdt

```
{}
```

Show JSON editor

Show JSON editor

Test Cancel

i Note

If you still provide the additional input properties, then the automation will fail.

Also, you can observe that there is no information message is displayed for a data type which is not a strict data type. So, you can provide the appropriate additional input properties.

Custom Script

If a data type is strict and you try to provide the additional input properties through [Custom Scripting](#) then the system shows the information message, not to provide any additional input properties.

The screenshot shows a SAP Studio interface for editing an artifact named "strictMapipulati...". The main area displays a workflow diagram with a "Start" node, a "Custom script" step (id 1), and an "End" node. A tooltip is open over the "Custom script" step, highlighting a problem with the variable declaration: "myObject.strictData = any". The tooltip message states: "The data is strictly typed, this means you should not provide additional properties." Below the tooltip, there is a "Peek Problem" link and a message: "No quick fixes available". To the right of the diagram, the "Step Details" panel is visible, showing the output parameters: "strictData" (Description: "strictDataType") and "notStrictData" (Description: "notStrictDataType"). The "Output parameters" section is empty. At the bottom of the screen, the status bar shows "Test Console" and "All Errors (1) Warnings (1) Info (0)".

The following example shows the automation failed due to an additional input property provided.

The screenshot shows a SAP Studio interface for editing an artifact named "strictMapipulati...". The left sidebar shows a project structure with "Content", "Dependencies", and "Automations". Under "Automations", there is a folder named "strictMapipulation" containing one item. The main workspace shows a workflow diagram with a "Start" node, a "Create strictDataType data..." step (id 1), a "Custom script" step (id 2), a "Set generatedObject field..." step (id 3), and an "End" node. A tooltip is open over the "Custom script" step, highlighting a problem with the assignment: "strictData.age = 14;". The tooltip message states: "No output parameter defined for this Script.". Below the tooltip, there is a "Peek Problem" link and a message: "No quick fixes available". To the right of the diagram, the "Step Details" panel is visible, showing the input parameters: "strictData" (Description: "strictDataType") and "notStrictData" (Description: "notStrictDataType"). The "Output parameters" section is empty. At the bottom of the screen, the status bar shows "Design Console" and "All Errors (0) Warnings (0) Info (0)".

The screenshot shows the SAP Cloud Studio interface. On the left, the project navigation pane includes 'Content', 'Dependencies', 'Data Types' (with 'strictDataType' selected), and 'Automations'. Under 'Automations', 'strictMapipulation' is selected. The main workspace displays an automation flow with a 'Start' node, a 'Create strictDataType data object' step, and a 'Custom script' step. A red box highlights a validation error message: 'The datatype object is not valid, there is 1 validation error. The age property is unknown. Additional properties are not authorized.' Below the message is a JSON snippet of the datatype object.

```

{
  "uid": "2e5cdd01-87d3-4058-ad3c-8725e246aa86",
  "name": "strictMapipulation",
  "type": "automation",
  "instanceUid": "5930a6d9-bab7-4d2e-8ad9-672d974b5529",
  "parentUid": "369c2ec0-4a68-48df-8fce-f299856b2ec7"
}
  
```

Constraints in Data Types

You can set two types of **Constraints** when you create a data type in the Cloud Studio. The **Required** constraint and the **Enumeration** constraint.

Required

The **Required** constraint allows you to set a data field that must be provided later on in the **Input Parameter** of a trigger.

For example, if you add the **New Field** *mailAddress* with the **Required** constraint (1), you must provide this data to be able to create the trigger (2).

The screenshot shows the SAP Intelligent Robotic Process Automation Cloud Studio interface. The left sidebar shows 'Project Explorer' with 'Content' expanded, showing 'Data Types' (selected), 'Eligibility_equipment', 'Person', 'User Tasks', 'EligibilityAlert', 'Desktop Packages', and 'Eligibility'. The main workspace shows a table for creating a new field. A red box highlights the 'Required' checkbox in the 'Field Properties' column. The table rows are:

Name	Type	Sample	List	Required	Field Properties
name	String		No	Yes	<input type="button" value="New Child"/> *Name: mailAddress
firstName	String		No	No	<input type="button" value="New Child"/> *Type: String
role	String		No	No	<input type="button" value="New Child"/> Sample value: Sample value
equipments	Eligibility_equipment		Yes	No	<input type="checkbox"/> List
name	String		No	No	<input type="checkbox"/> Constraints
type	String		No	No	<input checked="" type="checkbox"/> Required
age	String		No	No	<input type="checkbox"/> Enumeration
eligibility	String		No	No	

-  Add Scheduled Trigger

2

*Name: Scheduled_01

Description: e.g. A scheduled trigger

*Execute: Compute Eligibility (PROCESS) ▾

*Priority: Low ▾

Person	Schedule	Input Parameter
Missing property "mailAddress". Peek Problem No quick fixes available		
<pre> "person": { "name": "", "firstName": "", "role": "Dev", "equipments": [{ "name": "", "type": "", "age": "", "eligibility": "" }] } </pre>		

Enumeration

The **Enumeration** constraint allows you to define a set of several values for a data field. Meaning that you can limit the choice of values to provide for a data field, in the **Input Parameter** of a trigger.

For example, if you add the **New Field role** with the **Enumeration** constraint and set several values (1), you must provide one of the values you previously set, to be able to create the trigger (2).

The screenshot shows the SAP Intelligent Robotic Process Automation Cloud Studio interface. On the left, the Project Explorer lists 'Data Types' under 'Eligibility_equipments' and 'Person'. The main area is titled 'Eligibility *Person X' and shows a table of fields:

Name	Type	Sample	List	Required	
name	String		No	Yes	<button>New Child</button>
firstName	String		No	No	<button>New Child</button>
role	String		No	No	<button>New Child</button>

On the right, 'Field Properties' are defined: Type is String, Sample value is 'Sample value', and Constraints include Required (unchecked) and Enumeration (checked) with values 'Dev,Tester,Manager,PO'. A red box labeled '1' highlights the 'Save' button at the top right.

• Add Scheduled Trigger

2

*Name:

Description:

*Execute:

*Priority:

[Schedule](#)[Input Parameter](#)

```

{
  "person": {
    "name": "Value is not accepted. Valid values: "Dev", "Tester", "Manager", "PO". (1)
    "firstName": "Peek Problem No quick fixes available"
    "role": "[",
    "equipments": [
      {
        "name": "",
        "type": "",
        "age": "",
        "eligibility": ""
      }
    ],
    "mailAddress": ""
  }
}

```

i Note

The **Enumeration** constraint is case sensitive: the values must be spelled identically in both the **Field Properties** of the Cloud Studio and the **Input Parameter** of the Cloud Factory.

Data Types from SDK Packages

Some data types can be automatically created when SDK packages are imported.

When you first create an automation or an application, you select SDK packages that will be imported as dependencies to your project. You can also manually import SDK packages from the store.

The SDK packages contain activities, and some of the activities are displayed as data types. If you go to the **Dependencies** tab in the **Project Explorer** panel, you can see all the dependencies of your project, including the data types that are part of imported SDK packages.

i Note

To maintain compatibility between versions, data types coming from SDK packages are **strict** data types. If an object is defined using an SDK data type, you cannot add other properties to this object (using the [Custom Scripting](#) for instance) other than the ones defined in the data type.

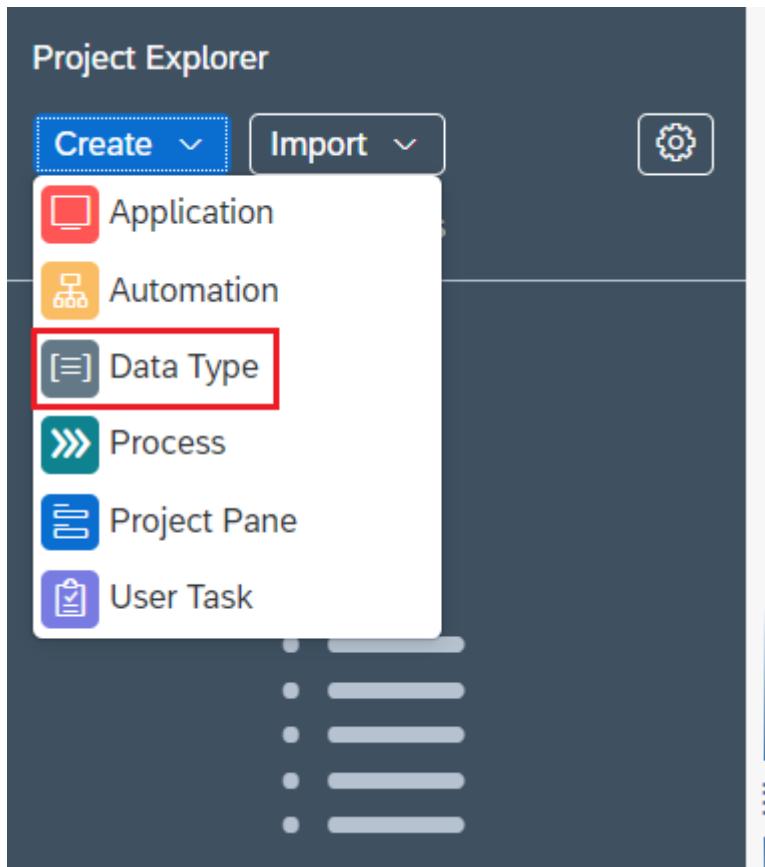
Reuse Data Type from another Package

The data type can use data types coming from an SDK package. This helps in reusing existing data type inside the data type and this can be used to automate many processes.

To achieve this, you must create a data type and then reuse existing data type inside the data type. For more information about data type and creating a data type refer to the [Data Types](#) section.

To reuse the data type inside the data type, follow the below procedure:

1. In the SAP IRPA Cloud Studio, Project Explorer, under **Content** click **Create** and then click **Data Type**.



The **Create Data Type** pop-up screen is displayed.

Create Data Type

***Name:**

e.g. *My first Data Type*

***Identifier:**

e.g. *MyFirstDataType*

Description:

e.g. *A Data Type describing a person*

Create

Cancel

2. Enter a name and description, and click **Create**.

3. A new tab opens in the main panel of the Cloud Studio. You can now add the fields that will be included in your data type:
Click the **New Field** button.

4. In the **Field Properties** on the right hand side, in the **Name** field enter the name of the field.

5. Click ***Type**. The drop-down list is displayed.

Field Properties

***Name:**

FieldSimpleParams

***Type:**

Any X ▼

- Number
- Object
- Password
- String
- Simple Input Dialog P... irpa_core
- Data Object Validation irpa_core
- Add Job Result irpa_core
- Window Position irpa_core
- Simple Info Dialog Par... irpa_core
- Job Parameters irpa_core

i Note

You can define your field type by:

- Choosing predefined simple types (String, Number, Boolean etc).
- Choosing a Data Type defined inside the same project.
- Choosing a Data Type coming from SDK packages (core or excel etc).

6. Select the required data type coming from an SDK package.

The Data Type details are displayed.

The screenshot shows the SAP Intelligent Robotic Process Automation Cloud Studio interface. A new data type named 'FieldStats' is being created under the 'TestDataType' category. The 'Field Properties' panel on the right shows various configuration options for the field, including its name, type, sample value, and constraints. A red box highlights the list of fields in the central table.

7. Click **Save** to save your changes.

i Note

Data types coming from SDK packages are **strict** data types: to maintain compatibility between versions, you cannot manually modify the data type properties using the [Custom Scripting](#).

Related Information

[Data Types](#)

Import Excel File as a Data Type

You can create a data type from an existing Excel spreadsheet by importing the data from the spreadsheet using the [Import Excel File](#) feature.

Prerequisites

Be sure that your Excel spreadsheet is organized in columns so that each column corresponds to a type of data (number, string, Boolean). Such as:

	A	B	C	D	E	F
1	Test	Number	bool			
2		1	1	TRUE		
3	a		2	FALSE		
4	s		3	FALSE		
5						
6						
7						
8						
9						
10						
11						
12						
13						

Procedure

1. Choose **Import Excel File**.

Name	Type	Sample	List	Required	
test	Any		No	No	<button>New Child</button>
number	Number		No	No	<button>New Child</button>
boolean	Boolean		No	No	<button>New Child</button>

Data Type Details

General Information

Name: *

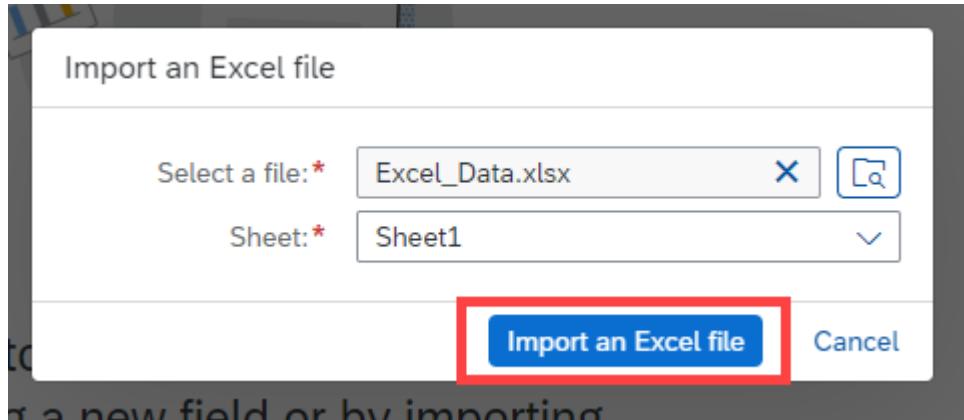
Identifier: *

Description:

Data type is active

Strict

2. Open the file browser to find your Excel book.
3. Select the corresponding spreadsheet within the book.
4. Choose **Import an Excel File**.



Results

You now see the imported data organized as a data type and split into corresponding fields.

Name	Type	Sample	List	Required	
 Start to create your data type by creating a new field or by importing an Excel file. New fields will show up here.					

Data Type Details

General Information

Name: *

Identifier: *

Description:

Data type is active

Strict

Information

Last Update: 8/15/2022, 08:59:02 AM
By: brienne.bennett@sap....

Created: 8/15/2022, 08:59:02 AM
By: brienne.bennett@sap....

i Note

You can import multiple excel spreadsheets into a single data type. In this case, repeat columns are added to the existing field. For example, if your spreadsheets each have a column for numerical data, they are all combined into one data type field.

Create and Update Variables

The variables you use in the Cloud Studio to build your automation are data storage that has a name, a type (example: string, list of string or data type), and a value. A variable in the automation is also associated to a step represented by its number.

Variables allow you to:

- store content that can be reused inside the automation flow.
- set or modify the content of a variable that contains value in the automation flow.

To view **Data Management** and **Data Types** activities, in the **Automation information** panel, go to the **Tools** section and click **Data**. The dropdown list displays:

- The dropdown **Data Management** list where you can find all activities related to data management.
- The dropdown **Data Types** list where you can find all data types.

The screenshot shows the SAP Fiori interface for automation. At the top, there are tabs for 'Tools', 'I/O', and 'General Data'. Below the tabs is a search bar with a magnifying glass icon and a three-dot menu icon. The main content area has a tree navigation on the left with nodes for 'Automations', 'Activities', and 'Data'. The 'Data' node is expanded, revealing two sections: 'Data Management' and 'Data Types'. A red box highlights both of these sections. Under 'Data Management', there are two items: 'Add Item' and 'Validate data object', each with an info icon. Under 'Data Types', there are ten items: Boolean, Number, String, Any, Add Job Result, Data Object Validation, Job Parameters, Quality for tiff or jpg, Screenshot Window Position, Simple Info Dialog Parameters, Simple Input Dialog Parameters, and Window Position. Each item has an info icon. To the right of the 'Data Management' section, there is a small box labeled 'irpa_core'. At the bottom of the main content area, there is a link to 'Controls'.

Visual Representation of Data Type Variables

There are two types of data type variables:

- Primitive data types
- Complex data types

These primitive and complex data type variables allow you to model your data to define how the automation intends to use the data.

The different types of variables are described as follows:

Primitive data type: It can be a simple data such as a string, a number, and a boolean.

A variable of type "any" can provide any type of data such as a string, a number, a boolean, a complex subject and so on. A variable of type "any" allows you to get rid of the data validation.

Complex data type: You need to use complex data types for more complex data. A data type allows you to compose primitive types into a compound type representing your data structure. It contains several property fields which can be either primitive types defined that could be arranged in a hierarchy form, or other project's data types. A data type enables you to organize and manage your data which can impact the access and modification efficiency. For more information on data types, see [Data Types](#).

You can define a primitive or a complex data type as a list that represents a collection of those elements.



Parent topic: [Manage Data within an Automation](#)

Previous: [Data Types](#)

Next: [Different Types of Data Fields](#)

Using Primitive Data Type Variables

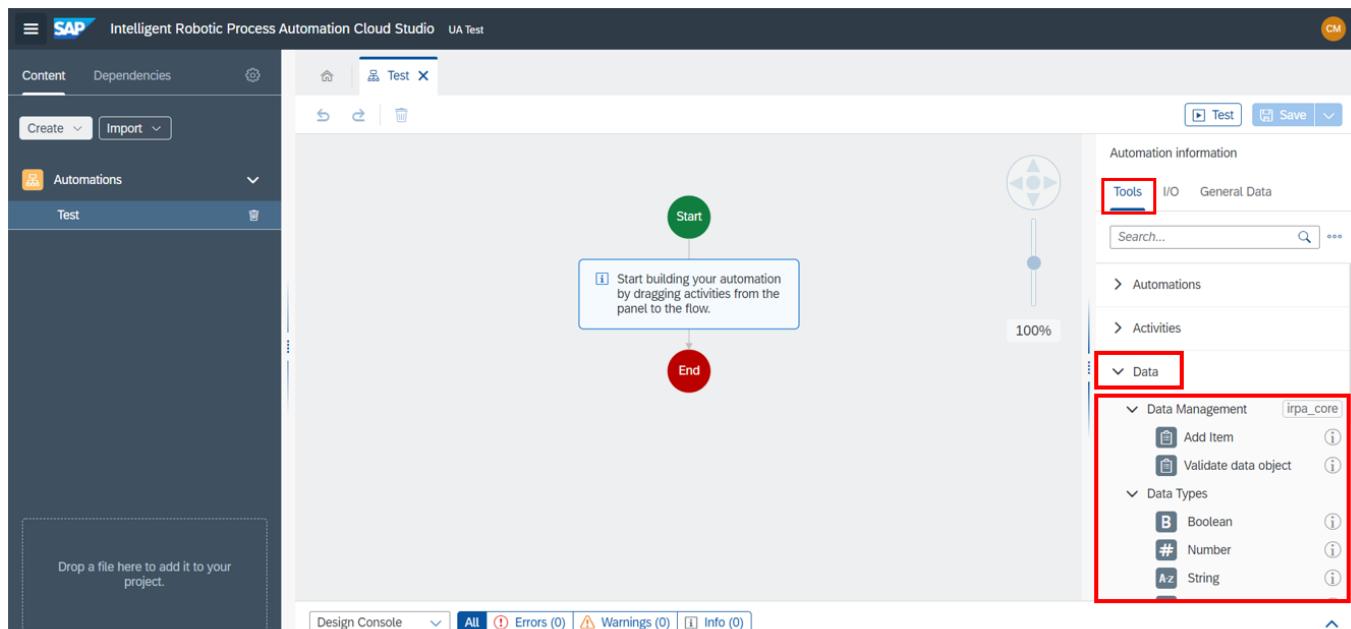
Prerequisites

You have created one or several automations in the Cloud Studio as described in [Create an Automation](#).

Example

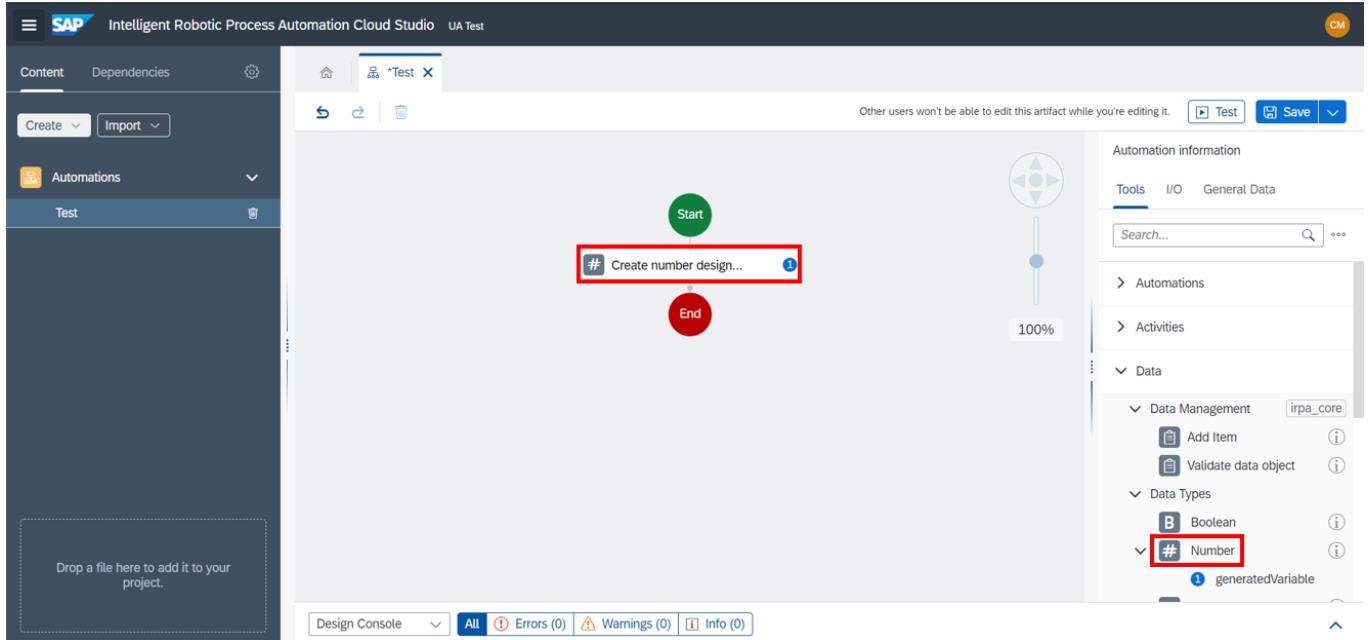
The following use case demonstrates how to create a variable of type "number". You can follow a similar procedure for "boolean", "string" and "any" types of variables.

1. In the automation, on the right-hand side panel, go to the **Tools** section and click **Data**. The dropdown list of different kinds of data is displayed.

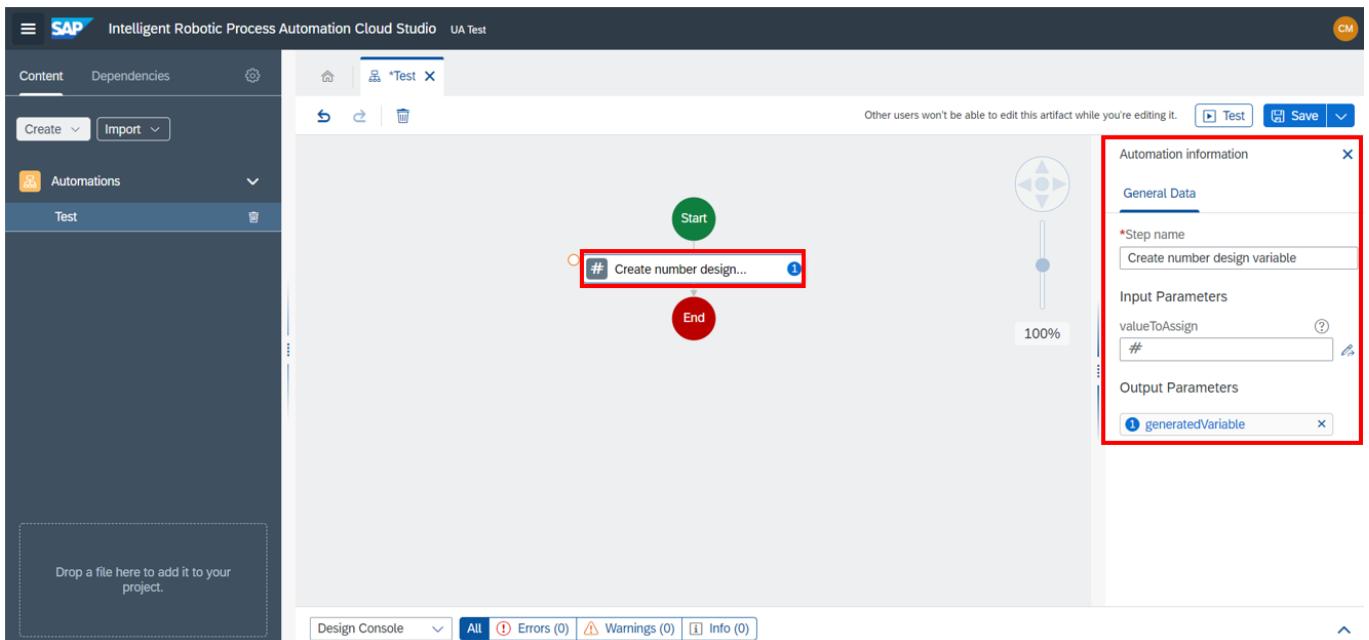


2/26/2024

2. Drag and drop **Number** data into the workflow of the automation.



3. Select the step **Create number design variable**. The **Automation information** panel appears.



4. Modify the **Step name** as per your requirement. Set an input parameter in the **valueToAssign** field. The value must be a number for a variable of type number. You can also modify the value of output parameter as per your requirement.

This screenshot shows the 'Automation information' panel with the following modifications:

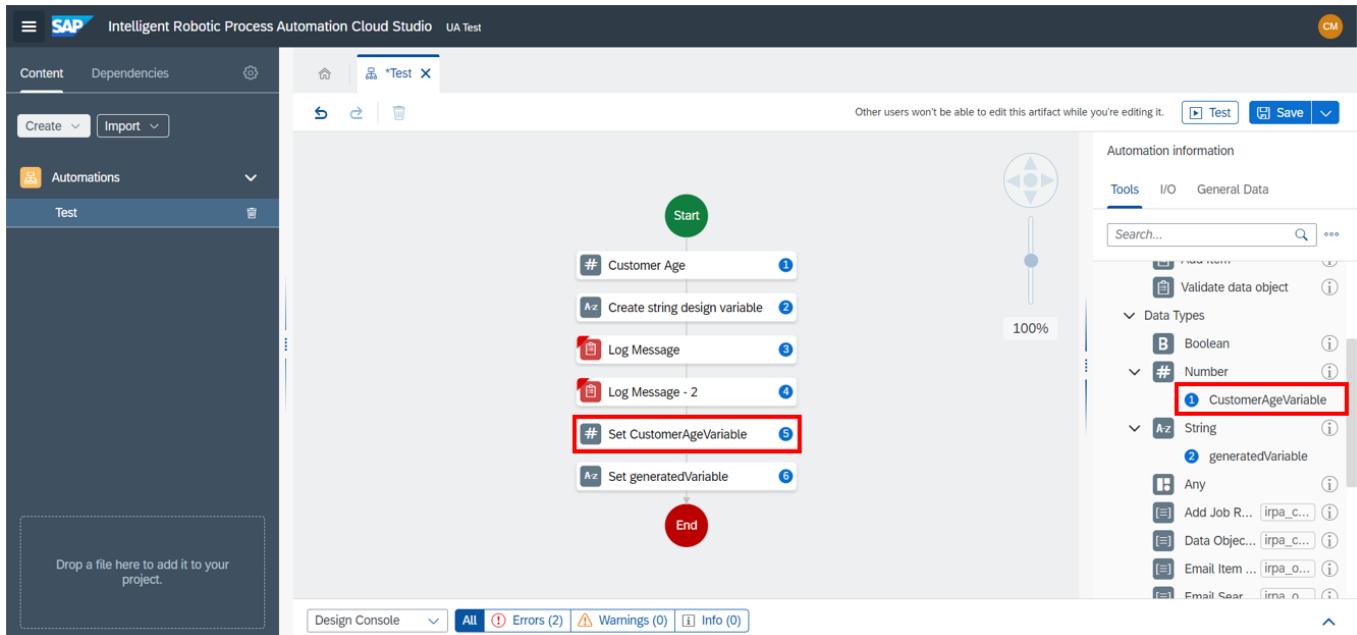
- Step name:** Customer Age
- Input Parameters:** valueToAssign: 23
- Output Parameters:** CustomerAgeVariable

The 'Customer Age' entry in the Step name field, the '23' in the valueToAssign field, and the 'CustomerAgeVariable' entry in the Output Parameters field are all highlighted with red boxes.

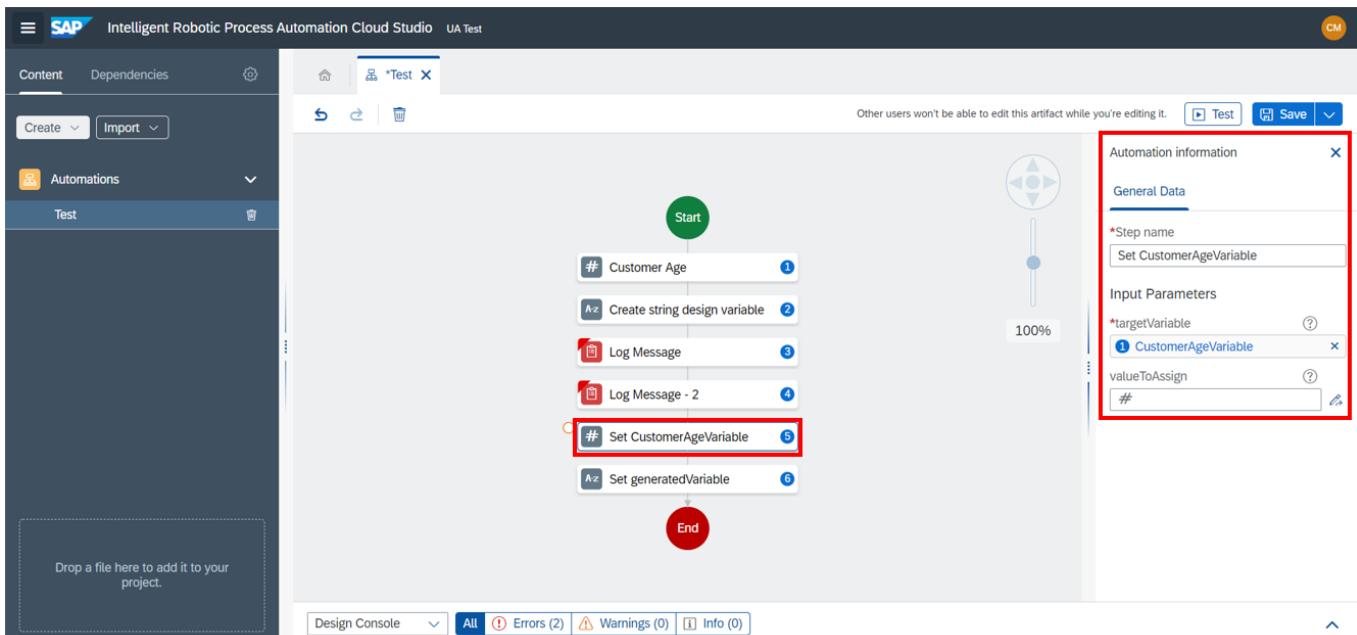
i Note

The value must be the same type as the generated variable for any variable such as number for number, string for string and so on.

5. Drag and drop **CustomerAgeVariable** under **Number** data into the workflow to update its value.



6. Select the step **Set CustomerAgeVariable**. The **Automation information** panel appears.



7. Modify the **Step name** as per your requirement. Set an input parameter in the **valueToAssign** field. You can modify the value of **targetVariable** field.

*Step name
Set Customer Age

Input Parameters

*targetVariable
① CustomerAgeVariable

valueToAssign
0

i Note

The value of **valueToAssign** field can be a formula or another previously computed variable.

8. You can use the **Log** activity if you want to display the variable value at runtime.

Using Complex Data Type Variables

Complex Type Data Variables

Complex type data variables allow you to:

- initialize data depending on the type of object. It allows you to parameterize data and store data in different data field.
- modify the content of an object.
- validate the type of data variable. It allows you to know whether the data is valid or not.

User Interface (UI) of Complex Data Variables

Create Variable: You must drag and drop the variable to create a complex variable.

Set Variable: You must drag and drop the variable for the set variable.

i Note

Create and set variables both have almost similar UI. The representation of the fields of the variables is similar. The only difference is that you can map only one data field in set variable.

Validation Variable Activity: Validation variable activity is accessible from right hand side panel of the **Activities**. It allows you to validate the data explicitly. This activity is no longer required as data validation is now done automatically at each step of automation. For more details on data validation, refer to the following note.

i Note

When a data type is used as input of automation or parameter in an activity or a custom script, it gets validated automatically. An exception will be thrown if:

- any field does not have the correct data type. For example, string field is expected however the value is a number.
- some mandatory fields are missing.

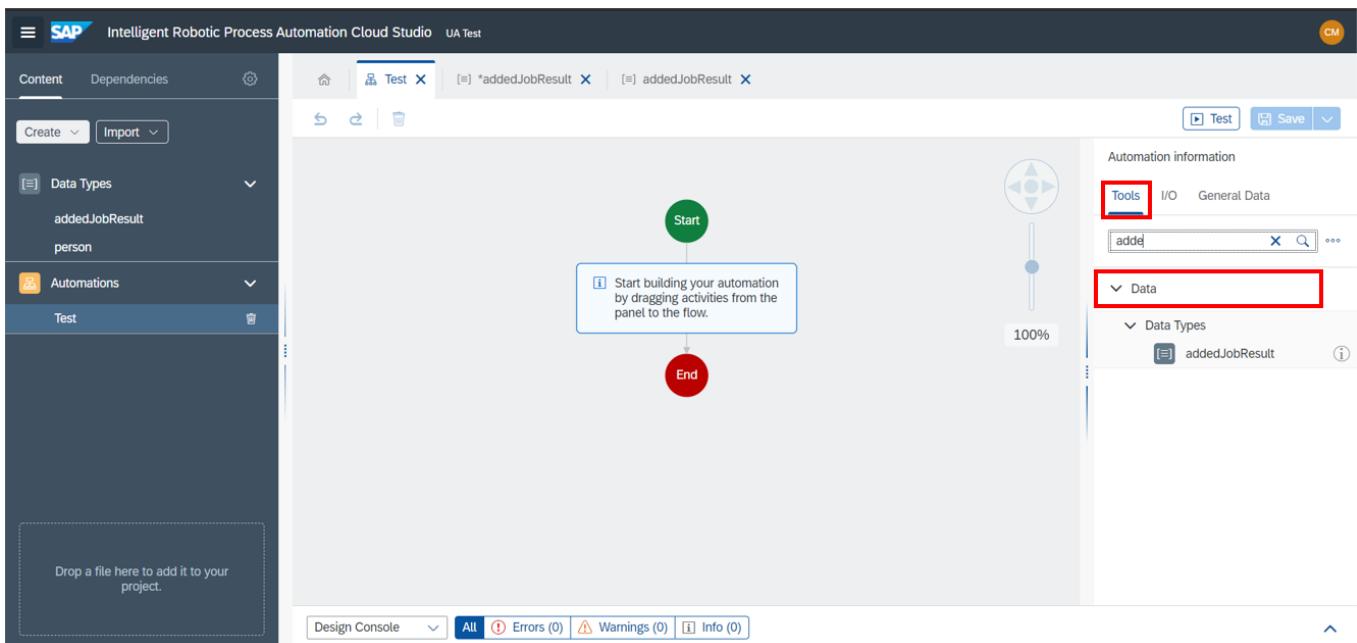
Using Primitive Data Type Variables

As a prerequisite, you must have created one or several automations in the Cloud Studio as described in [Create an Automation](#).

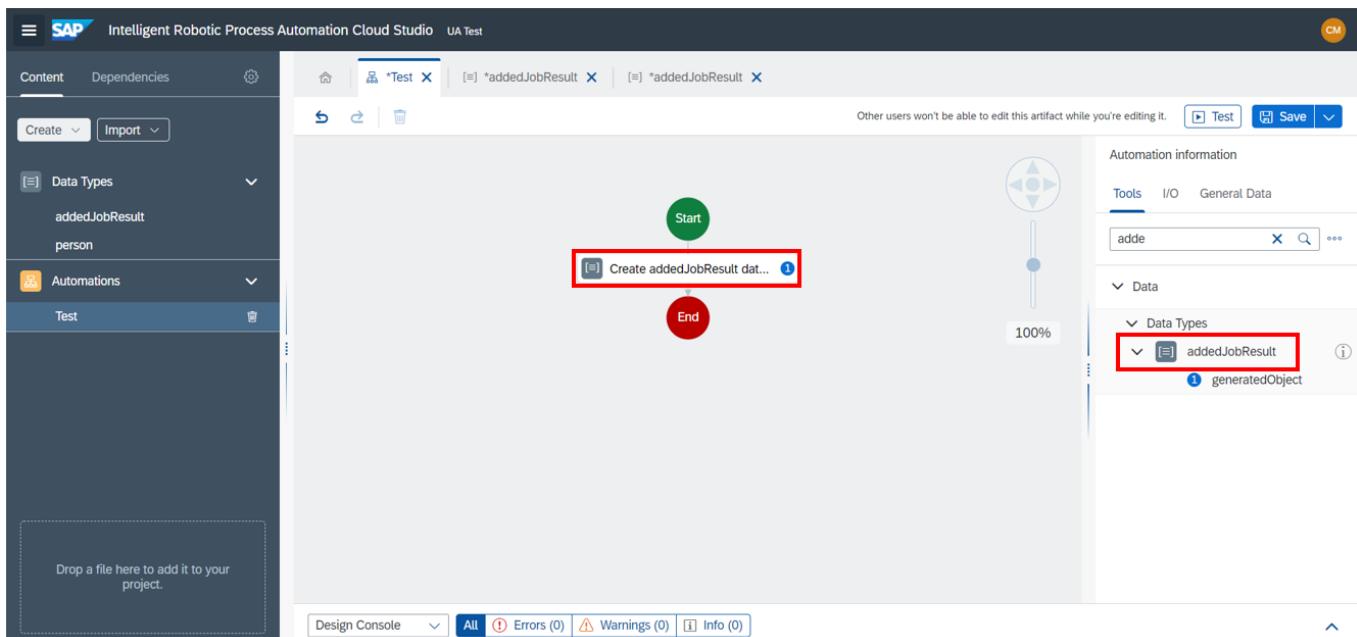
Example

The following use case demonstrates how to create, set, and validate complex type data variables.

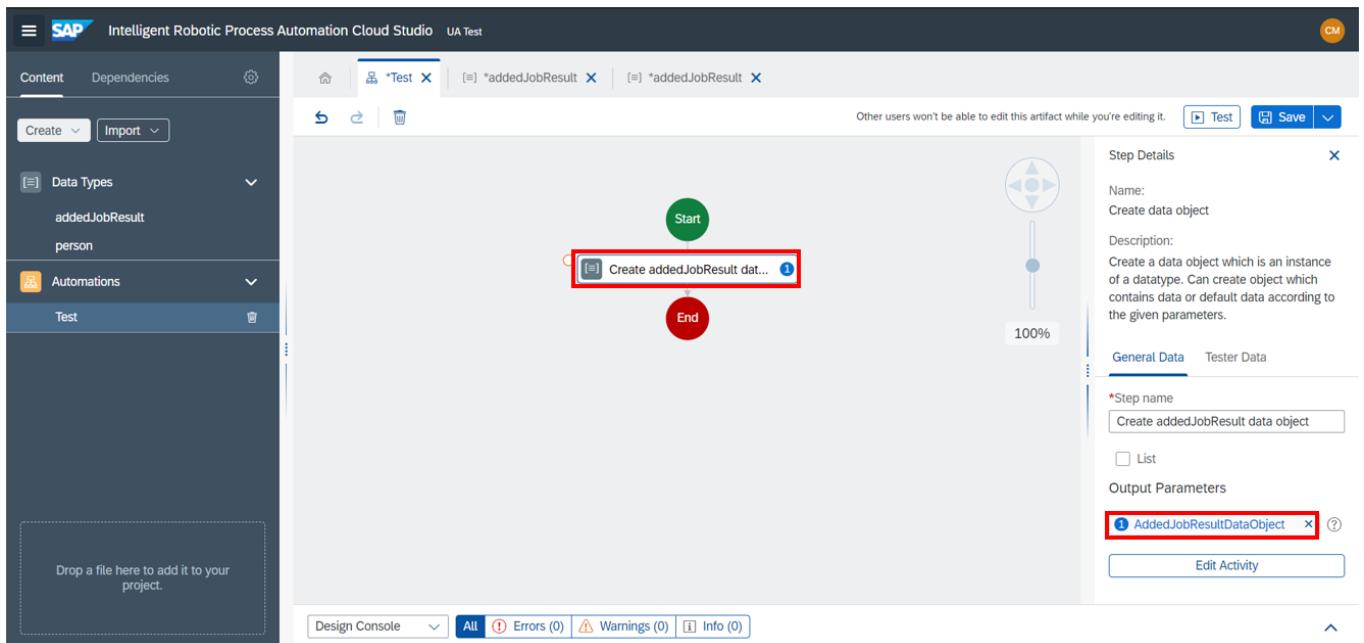
1. In the automation, on the right-hand side panel, go to the **Tools** section and click **Data**. You must know the type of the object that need to create in the automation.



2. Drag and drop **addedJobResult** data into the workflow of the automation.

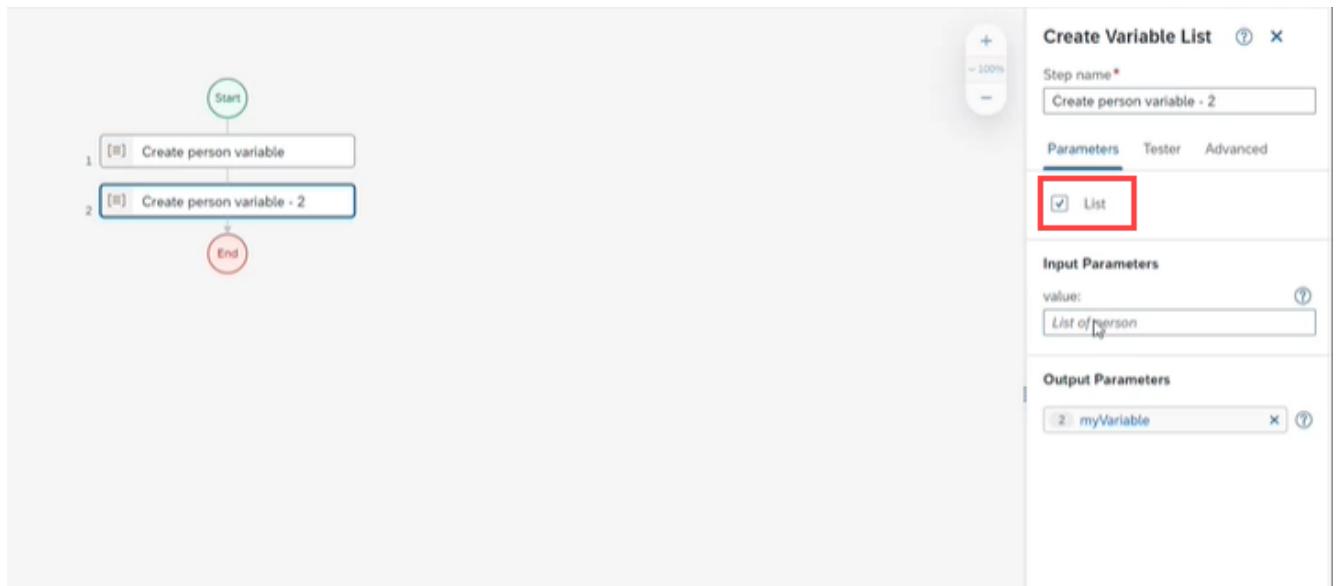


3. Select the step **Create addedJobResult data object**. The **Step details** panel appears. You can modify the value of the **Output Parameters** field as per your requirement. The **Output Parameters** field has the correct structure with empty fields.



i Note

To create a list data type, tick the **List** checkbox in the parameters side panel.



A list data type allows you to add multiple variables to your data type and can be imported from an API or an input from another automation.

For more information on list data types, see [List Management](#).

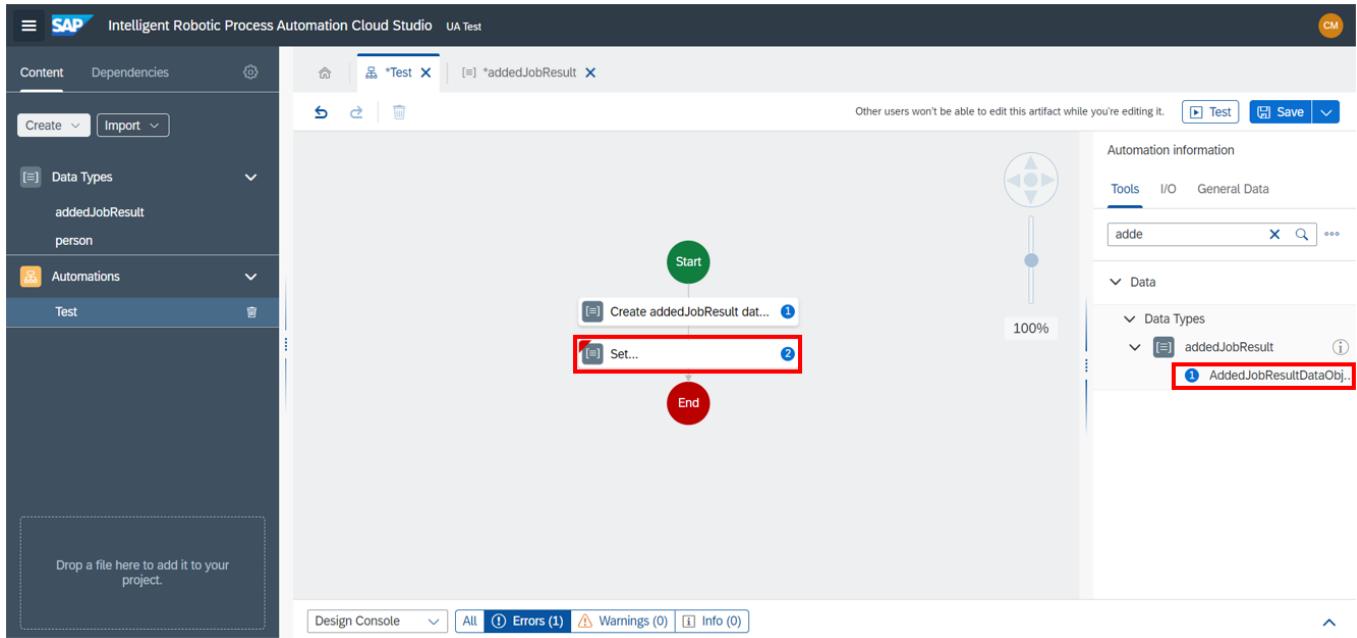
4. Double click on the step **Create addedJobResult data object** or click **Edit Activity** in the right pane. Different data fields appear.

The screenshot shows the SAP Intelligent Robotic Process Automation Cloud Studio interface. On the left, there's a sidebar with 'Content' and 'Dependencies' tabs, and sections for 'Data Types' (selected) and 'Automations'. A central workspace displays a workflow diagram with a green 'Start' node, a blue rounded rectangle labeled 'Create addedJobResult data...', and a red 'End' node. A circular progress bar indicates the workflow is at 100%. To the right, a 'Step Details' panel is open for the selected activity. It contains a message about mapping contextual data to fields, followed by a form for creating an 'addedJobResult' object with fields: iud (containing 'incomingUid'), code, and label (containing 'information'). Below this, the 'General Data' tab is selected, showing the step name 'Create addedJobResult data object'. The 'Edit Activity' button is also highlighted with a red box.

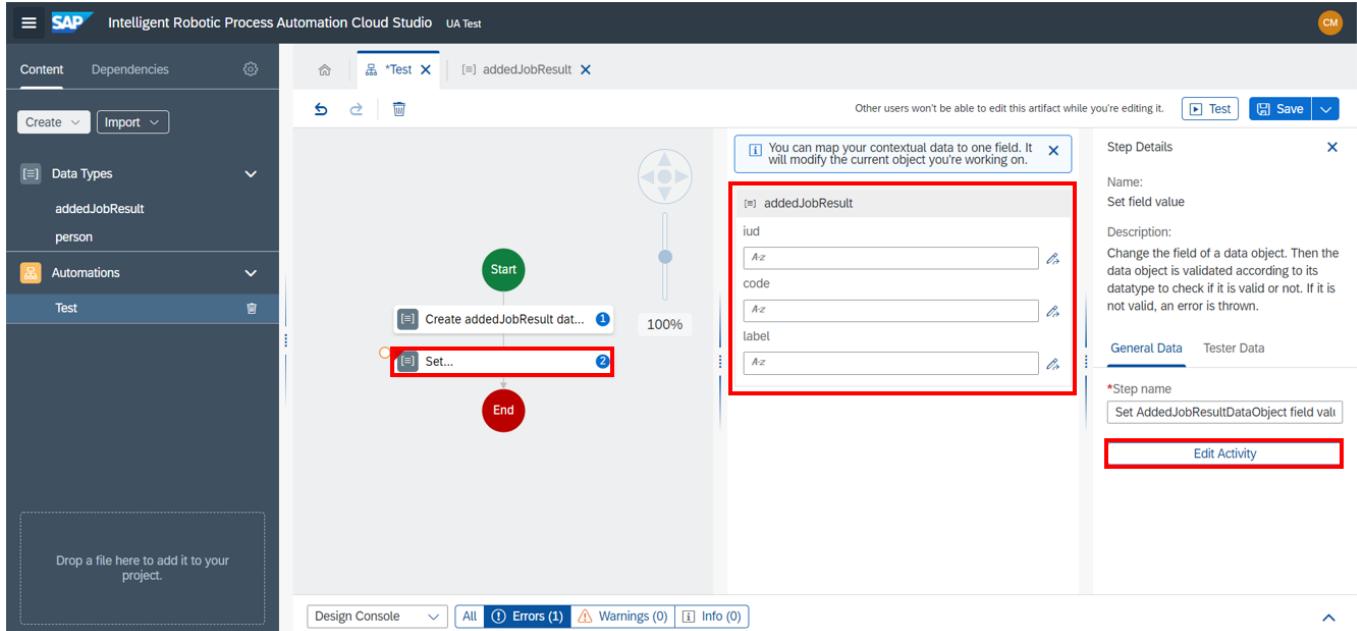
5. Provide the value into the complex fields. You can also edit formula with expression editor.

This screenshot shows the same SAP studio interface after step 5 has been completed. The 'Step Details' panel now displays the provided values for the 'iud' and 'label' fields: 'incomingUid' and 'information' respectively. The 'Edit Activity' button is again highlighted with a red box.

6. Drag and drop **AddedJobResultDataObject** under **addedJobResult** data into the workflow of the automation to update the value of complex type data variable.



7. Double click on the step **Set AddedJobResultDataObject field value** or click **Edit Activity** in the right pane. A similar view of different data fields like the create variable activity appears. However, the only difference is that you can provide only one data field value in the set variable activity.

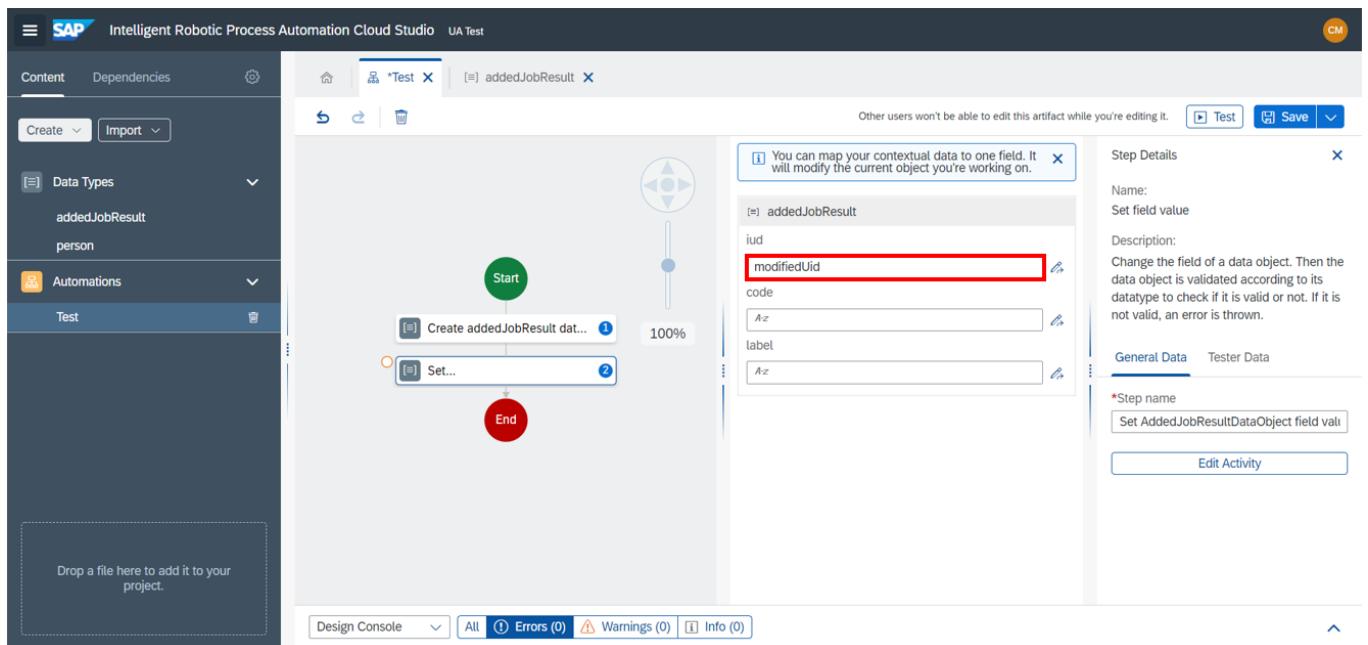


8. Modify the field value of a data variable.

i Note

You need to use:

- o text to enter static data.
- o suggest capability to bind previously defined variables.
- o expression editor to enter formula with [Expression Editor](#).



9. You can use the **Log** activity if you want to display the data variable value at runtime.

i Note

You can not use the expression editor to set or modify complex data variable since you can do it through the complex data variable.

List Management

In the Cloud Studio, a collection of activities provide enhanced support to manage the lists of your automation.

i Note

To learn how to use lists, you can download the [How to use list of objects](#) sample on the SAP Intelligent RPA Store.

You can now have a powerful way of handling list items using data management activities.

The screenshot shows the SAP Fiori Launchpad interface. At the top, there are three tabs: 'Tools' (which is selected), 'I/O', and 'General Data'. Below the tabs is a search bar with a magnifying glass icon and a '...' button. The main area is titled 'Automation information' and contains a 'Data' section. Under 'Data', there is a 'Data Management' category. This category is highlighted with a red box and contains six items: 'Add All', 'Add Item', 'Remove by Index', 'Remove Item', 'Set', and 'Validate data object'. Each item has a small icon to its left and an 'info' icon (a blue circle with a white 'i') to its right. To the right of the 'Data Management' title, the text 'irpa_core' is displayed in a box.

Using Data Management Activities to Manage Lists

Add Item

Use this activity to add an item to the existing list.

In the **Input Parameters**, you must provide the name of a list in the **list** field and to be added value of an item in the **itemToAdd** field.

For example,

The screenshot shows the 'General Data' tab selected. Under 'Step name', the value 'Add Item' is entered. In the 'Input Parameters' section, there is one parameter: '*list' with value '0 stringlist1'. There is also a placeholder for '*itemToAdd'.

Add All

Use this activity to concatenate a list of items to another list.

In the **Input Parameters**, you must provide the name of an existing list in the **list** field and the items to be added or the name of another list in the **listToConcatenate** field.

For example,

The screenshot shows the 'General Data' tab selected. Under 'Step name', the value 'Add All' is entered. In the 'Input Parameters' section, there are two parameters: '*list' with value '4 numberList1' and '*listToConcatenate' with value '[1, 2, 3]'. Both fields have a placeholder '(?)' and a pencil icon.

Remove by Index

Use this activity to remove an item from a list based on its index in the list.

In the **Input Parameters**, you must provide the name of an existing list in the **list** field and the index value to be removed from the list in the **indexToRemove** field.

For example,

[General Data](#) [Tester Data](#)

*Step name

Remove by Index

Input Parameters

*list

4 numberList1

*indexToRemove

1

i Note

If the index is higher than the length of the list, no item is removed.

Remove Item

Use this activity remove an item from a list based.

In the **Input Parameters**, you must provide the name of an existing list in the **list** field and the item or name of the item to be removed from the list in the **itemToRemove** field.

For example,

General Data	Tester Data
*Step name	Remove Item
Input Parameters	
*list	(?) 4 numberList1 x
*itemToRemove	(?) 3 x edit
allOccurrences	(?) false x edit

i Note

allOccurrences

If false, only the first item will be removed. In case of **dataObject**, the remove is performed by reference but not by the value, that means only the repetition of the same instance corresponding to **itemToRemove** will be removed but not the other instances with the same property values.

If true, removes all occurrences of the item if it is found several times in the list.

Set

Use this activity to update the new value or reference of an object.

In the **Input Parameters**, you must provide the variable in the **variable** field and the new value or reference an object in the **value** field.

General Data		Tester Data
*Step name		
Set		
Input Parameters		
*variable	(?)	
H		
*value	(?)	
H		edit

Validate data object

Use this activity to validate an object which is datatype instance..

In the **Input Parameters**, you must provide the data type object to be validated in the **objectToValidate** field.

The screenshot shows a configuration interface for a step named "Validate data object".

- General Data:** Step name is "Validate data object".
- Input Parameters:**
 - *objectToValidate: Value is "B", with a help icon (?) and a copy icon (copy).
 - *throwError: Value is "B", with a help icon (?) and a copy icon (copy).
- Output Parameters:**
 - validationResult: Value is "1", with a delete icon (X) and a help icon (?).

i Note

throwError

Throws an error if the object is not validated. If this error is not caught later, the automation will stop.

Date and Time Data Type Variables

You can now handle dates and times using data types.

Date and Time Data Types Format

You can now handle date and time using three data types:

- [Date](#)
- [Time](#)
- [DateTime](#)

The representations of these data types are based on the [ISO 8601](#) standard. According to this standard,

- A date is written in the **2018-11-13** format.
- A time is written in the **20:20:39+00:00** format.
- A date and a time are written in the **2018-11-13T20:39+00:00** format.

If you want to use a date or a time, you must follow the [ISO 8601](#) format.

Set Input Parameters in Picker Panels

When you define an input parameter which is expecting a date or a time, a picker panel appears and you can select dates and/or times. This picker panel appears in three cases:

- When you want to test your automation.
- When you want to provide environment variables.
- When you want to input parameters from triggers.

i Note

You cannot open a picker panel for dates and times in the Automation Editor. However, you can still enter dates using the keyboard or by binding a variable.

Test Automation

*Environment:
Date and Time Data Types Env

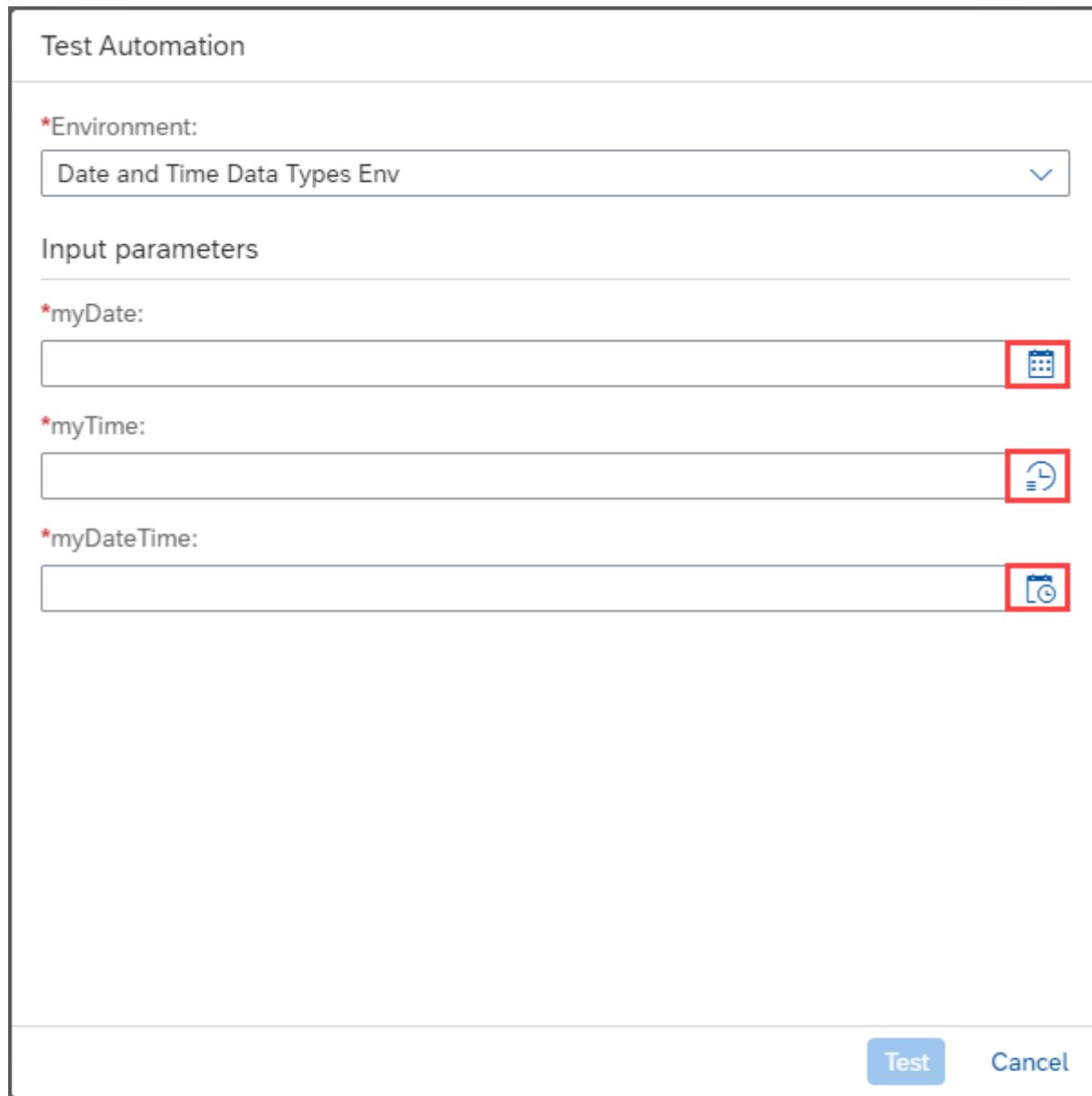
Input parameters

*myDate:
 

*myTime:
 

*myDateTime:
 

Test **Cancel**



Click the icons to open the corresponding picker panels:

Test Automation

*Environment:

Date and Time Data Types Env

Input parameters

*myDate:

	Sun	Mon	Tue	Wed	Thu	Fri	Sat
35	29	30	31	1	2	3	4
36	5	6	7	8	9	10	11
37	12	13	14	15	16	17	18
38	19	20	21	22	23	24	25
39	26	27	28	29	30	1	2

Test

Cancel

Manipulate a Date in JavaScript

If you want to manipulate a date with all the APIs available in JavaScript, you must convert this date into a JavaScript one:

- Using a [Custom Script](#).

You can do the conversion with: `new Date(myDate)`

- Using the [Expression Editor](#) and entering `new Date(myDate)`.

For more information, see [Dates](#).

Different Types of Data Fields

When you create an automation in the Cloud Studio, the data fields that you set can be one of the following types:

- Mandatory Data Field
- Optional Data Field
- Default Data Field
- Sample Data Field

Mandatory Data Field

You must set values to the data fields that are marked with a red asterisk (*). These fields are called the mandatory data fields. If you forget to fill the mandatory data fields, you will see the errors displayed in the **Design Console**.

For example, when you use the **Search Email (Outlook)** activity in your automation, if you add a search criterion and keep the mandatory fields empty, errors are displayed in the **Design Console** and you won't be able to launch the test session.

The screenshot shows the SAP Cloud Studio interface with a workflow diagram. The diagram starts with a green 'Start' node, followed by a blue 'Search Email (Outlook)' step, and ends with a red 'End' node. The 'Search Email (Outlook)' step has a red border around its configuration area, indicating an error. In the configuration pane on the right, under 'Input Parameters', there is a section for 'searchCriteriaList' with three fields: 'element', 'operand', and 'value'. The 'value' field contains the value 'YYYY-MM-DD'. A red box highlights this entire configuration area. At the bottom left, the 'Design Console' tab is selected, showing a list of errors:

- In step "Search Email (Outlook)", the field "searchCriterionList[0].element" is mandatory and must be set with a real value.
- In step "Search Email (Outlook)", the field "searchCriterionList[0].operand" is mandatory and must be set with a real value.
- In step "Search Email (Outlook)", the field "searchCriterionList[0].value" is mandatory and must be set with a real value.

Optional Data Field

You can either set values to the data fields or leave the data fields blank if the data fields are not marked with a red asterisk (*). These data fields are called the optional data fields. If you keep the optional fields empty, you won't see any error displayed in the **Design Console**.

The screenshot shows the SAP Cloud Studio interface with a similar workflow diagram. The 'Search Email (Outlook)' step now has a green border around its configuration area, indicating it is valid. In the configuration pane on the right, under 'Input Parameters', there is a section for 'searchCriteriaList' with three fields: 'element', 'operand', and 'value'. The 'value' field contains the value 'YYYY-MM-DD'. A red box highlights this entire configuration area. At the bottom left, the 'Design Console' tab is selected, showing a list of errors (all are now zero):

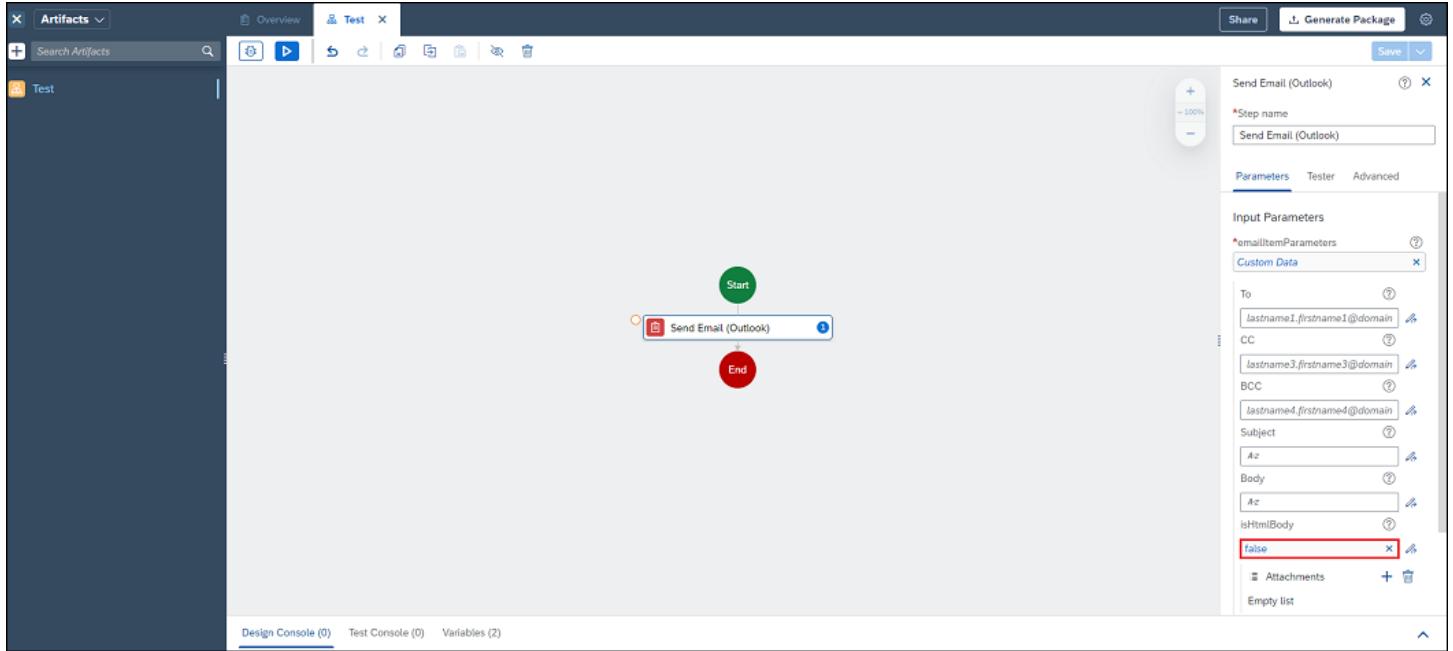
- All Errors (0)
- Warnings (0)
- Info (0)

Default Data Field

In the Cloud Studio, the default value is displayed with priority on the default data fields.

For example, when you use the **Send Email (Outlook)** activity in your automation, you can see that **false** is selected as default value for **isHtmlBody** default data field.

You can change the default value by clicking the **X** button and selecting **true**. You can return to the default value by clicking the  button.

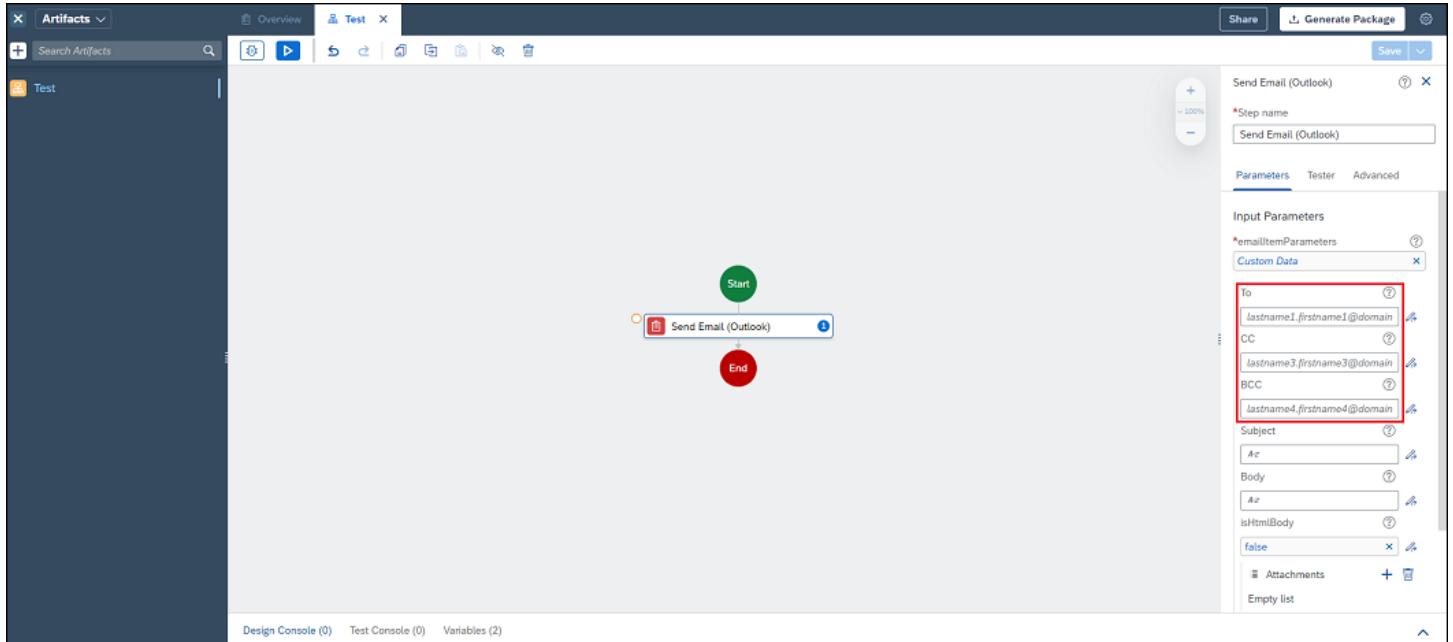


Sample Data Field

In the Cloud Studio, the sample values are the displayed values that help you while setting values to the sample data fields. These values are not considered as the real values because these values only give you the hint of the correct data values.

For example, when you use the **Send Email (Outlook)** activity in your automation, you can see that sample values are displayed on the sample data fields (**To**, **CC** and **BCC**). By seeing the sample values, you can easily understand what values you must set to the sample data fields.

If you enter email address into the sample data fields, the sample values will disappear from the sample data fields. If you remove the defined email address from the sample data fields, the sample values will appear again.



Parent topic: [Manage Data within an Automation](#)

Previous: [Create and Update Variables](#)

Next: [Expression Editor](#)

Expression Editor

What is the Expression Editor?

The expression editor allows you to create and edit expressions in your automation.

i Note

An expression is a single code line returning a typed value.

In the interface of the Cloud Studio, expressions can be:

- A constant
 - as a string
 - as a number
 - as a boolean
- A variable: coming from the input of the current automation you are editing, or from the output of a specific step used in the automation

❖ Example

`Step0.name`

- A complex expression (for example to express a condition). Unlike constants or variables, you can define a complex expression only by using the expression editor.

❖ Example

`Step0.name.trim() == "jane.smith"`

When building an automation in the Cloud Studio, you use expressions in:

- Input and output parameters of steps
For more information on input and output parameters, see [Input/Output Parameters](#).
- Conditions: to express the different cases, or in the outputs of a condition

❖ Example

`Step0.name.trim() == "jane.smith"`

- Loops (for iterators)

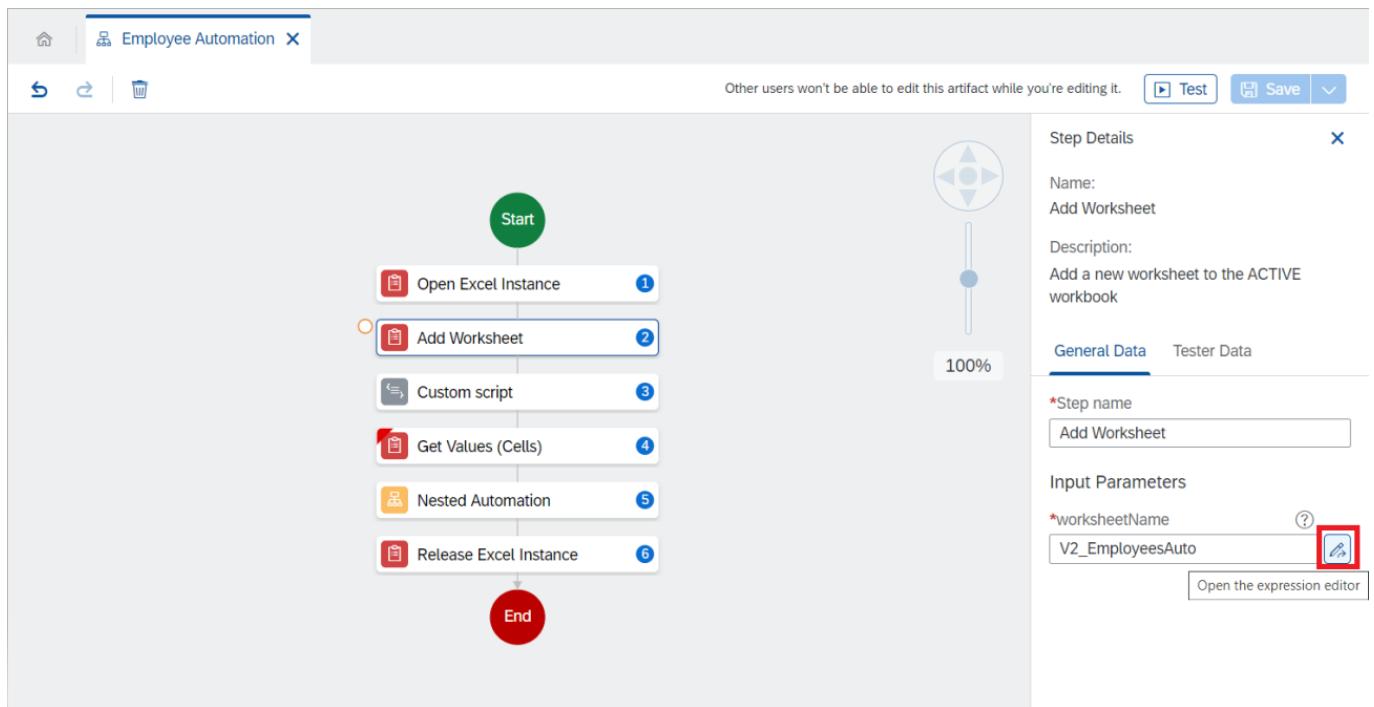
❖ Example

Step1.nameList

- Output parameters of the automation

Using the Expression Editor

1. In your automation, select a step and click the button on the side panel. The following example shows the expression editor button that you can use to define the input parameter of an activity.



2. The expression editor dialog opens. It's an editor of type JavaScript in which you can manually type your expression, or use the predefined references (variables, operators and functions).

Edit Expression

1 Step0.V2_EmployeesAuto

- > **Variables**
- > **Operators**
- > **Functions**

For more information, [click here](#).

Save Expression

Test

Cancel

i Note

The variables you use to create an expression with the expression editor must be prefixed with the name of the step they come from. In the example above, Step0 represents the input of the current automation.

i Note

Auto-completion is available when typing variables and functions (except for functions of type **any**).

3. Click **Test** to check that your expression is valid.

The test of your expression checks if:

- the syntax is correct
- the variables used in the expression exist
- the functions and operators used in the expression are allowed in the expression editor grammar.
- the type of the expression is compatible: an expression of type **any** can be of any type, but the other types of expressions must respect the expected type.

4. Click **Save**.

i Note

Don't forget to save your automation after you saved your expression in the formula editor.

References in the Expression Editor

Variables

A variable represents a data and is used to store information. Once you name a variable and define its value and type, you can use and reference it throughout your project.

When you edit the expression of a step, the expression editor shows you the variables and their type, that are available for this step.

Variables

- 4 returnedValues
any

In the example above, the variable `returnedValues` comes from the step 4 of the automation and is of type `any`.

Operators

An operator performs a specific operation on a set of expressions. You can use three types of operators:

- arithmetic
- logic
- comparison

In the expression editor, when you click on the dropdown menu next to **Operators**, you can see a list of all the operators available, with a definition. You can click on any operator to add it to your expression.

Operators

Arithmetic Operators

- + Add
- Subtract
- * Multiply
- / Divide

Logic Operators

- == Equal to

Functions

A function performs a specific operation on the expression (for example remove the space at the beginning and end of a string of characters). Four types of functions are available:

- string
- number
- array
- math

In the expression editor, when you click on the dropdown menu next to **Functions**, you can see a list of all the functions available, with a definition. You can click on any function to add it to your expression.

 **1x1 Functions**

- > **String**
All methods available on String interface and objects
- > **Number**
All methods available on Number interface and objects
- > **Array**
All methods available on Array interface and objects
- > **Math**
All methods available on Math interface

Parent topic: [Manage Data within an Automation](#)

Previous: [Different Types of Data Fields](#)

Enumerations

Combination of Enumerations

You can now have a powerful way of using enumerations to handle expressions of your automation.

You can use the combination of enumerations inside the expression of an [Automation Activities provided by SDK Packages](#) to get the desired result from an automation.

The enumerations values are considered as a string and you can use the enumerations on every string input parameter. The enumerations must be handled in the [Expression Editor](#).

Using Combination of Enumerations

You can use the enumerations coming from [SDK Packages](#). However, you must ensure that the enumeration values from various sources and models must be included in the Project.

Edit Expression

```
i
  {} irpa_core
  {} irpa_excel
>  {} irpa_outlook
>  {} irpa_sapui5
>  {} irpa_sapwebgui
  {} irpa_word
```

namespace irpa_core

For more information, [click here](#).

Save Expression

Test

Cancel

For more details about SDK packages and their enumerations, refer to the [Expression Editor](#) section.

The following example shows the combination of enumerations using `irpa_core`.

Edit Expression

```
irpa_core.enums.key.Ctrl + irpa_core.enums.SAPScripting.key._F10_
```

- > **Variables**
- > **Operators**
- > **Functions**

For more information, [click here](#).

Save Expression

Test

Cancel

Edit Expression

`irpa_core.enums.key.Ctrl + 'F'`

- > Variables
- > Operators
- > Functions

For more information, [click here](#).

Save Expression

Test

Cancel

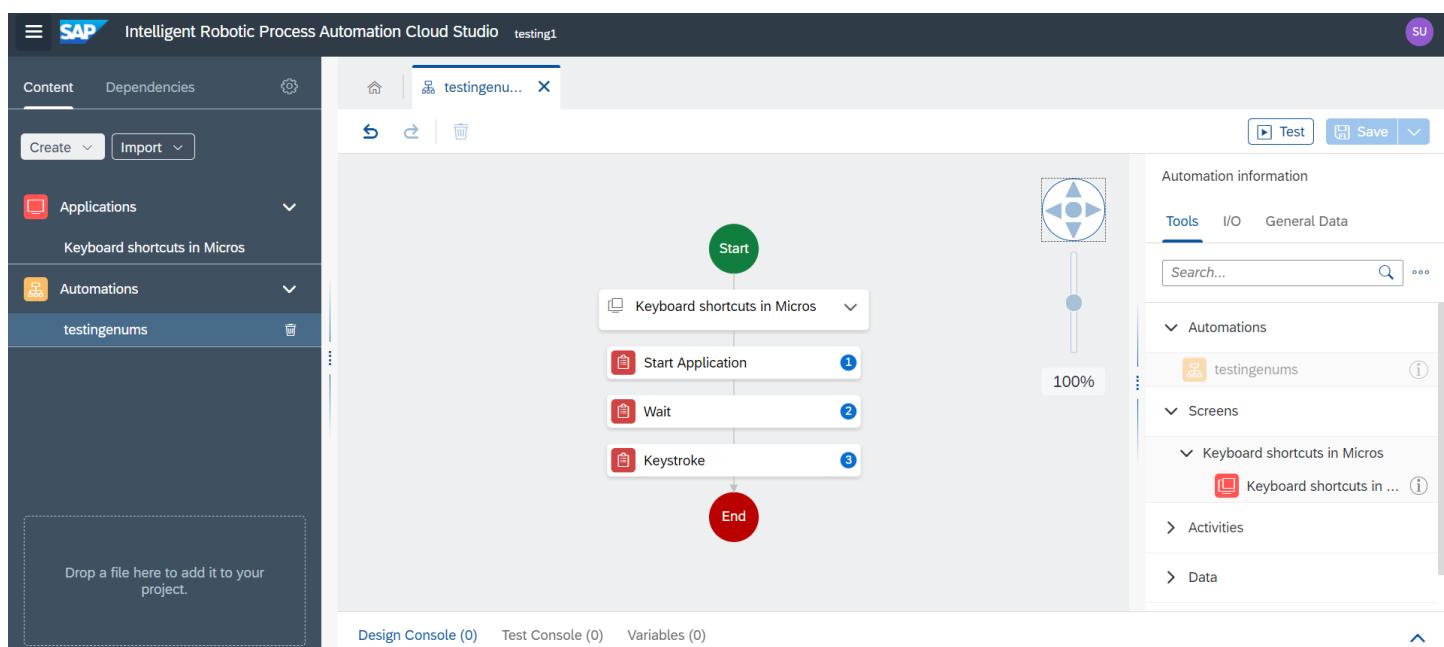
Use Case Example

The following example shows the keyboard shortcuts used to get the expected result.

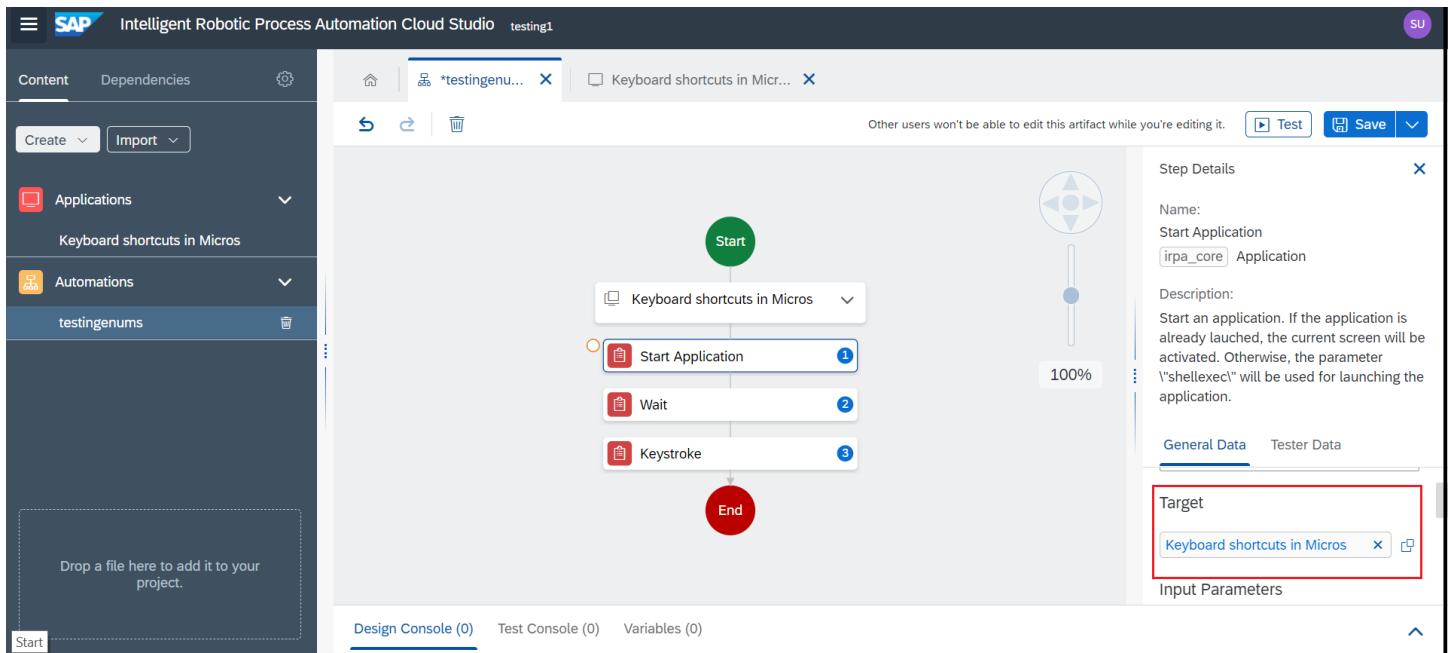
Prerequisites

- You have captured the Microsoft Edge browser application and declared the required elements.

The following screenshot shows the activities used to execute the automation.



In the **Start Application** activity, you must define the [Activity Target Inputs](#).



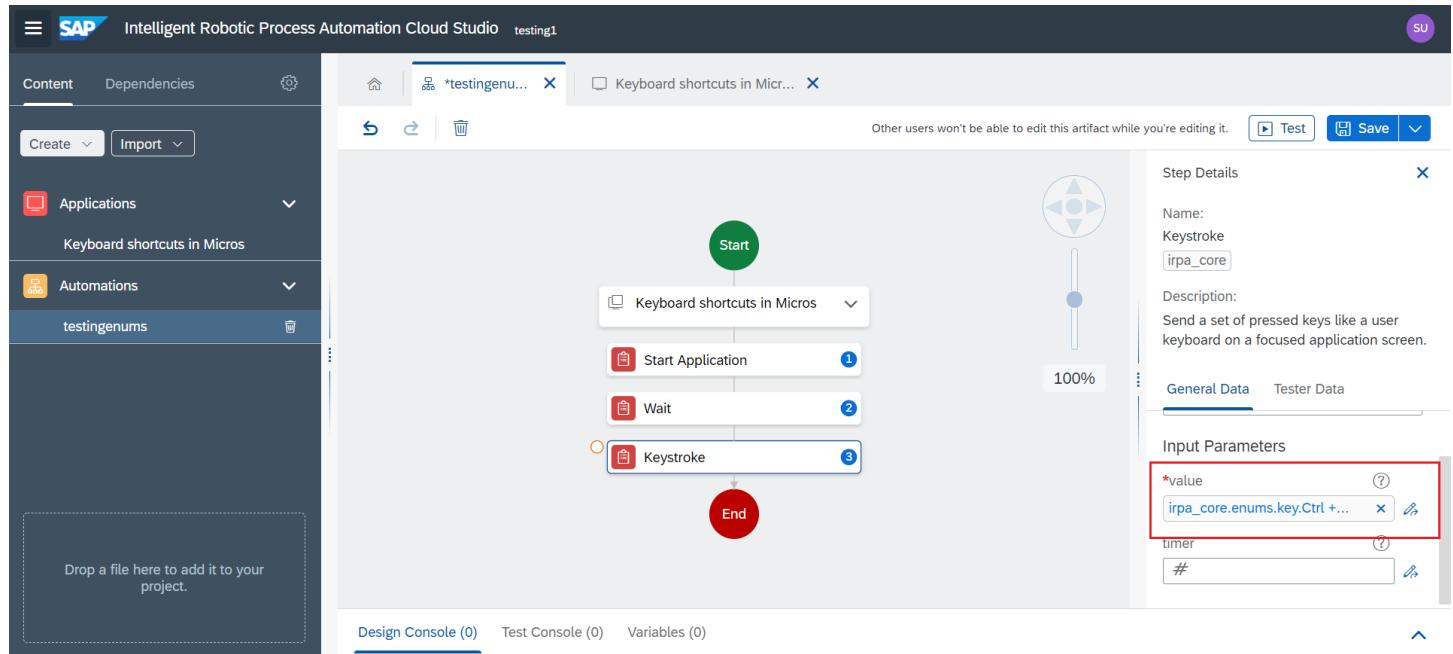
and in the **Keystroke** activity [Expression Editor](#), you must define the required combination of enumerations value.

Edit Expression

```
irpa_core.enums.key.Ctrl + 'F'
```

- > **Variables**
- > **Operators**
- > **Functions**

For more information, [click here.](#)



Result

Opens Microsoft Edge and uses the keyboard shortcut, Ctrl + F which enables the *[Find on page](#)* in Microsoft Edge.

Dates

i Note

Dates are expressed according to the [ISO 8601](#) standard. For more information, see [Date and Time Data Type Variables](#).

You can use dedicated data types to set dates in the Cloud Studio. However, if you want to manipulate dates with all the APIs available in JavaScript, you must convert them into JavaScript dates. To perform this conversion, you can use the [Expression Editor](#).

To convert dates into JavaScript ones, use the following syntax: new Date(yourDate)

Edit Expression

```
new Date(Step0.myDate),|
```

- > Variables
- > Operators
- > Functions

- getDate (method) Date.getDate...
- getDay
- getFullYear
- getHours
- getMilliseconds
- getMinutes
- getMonth
- getSeconds
- getTime
- getTimezoneOffset
- getUTCDate
- getUTCDay

For more information, [click here.](#)

Save Expression [Test](#) [Cancel](#)

Thus, you will be able to access the APIs available in JavaScript, as shown in the screenshot above.

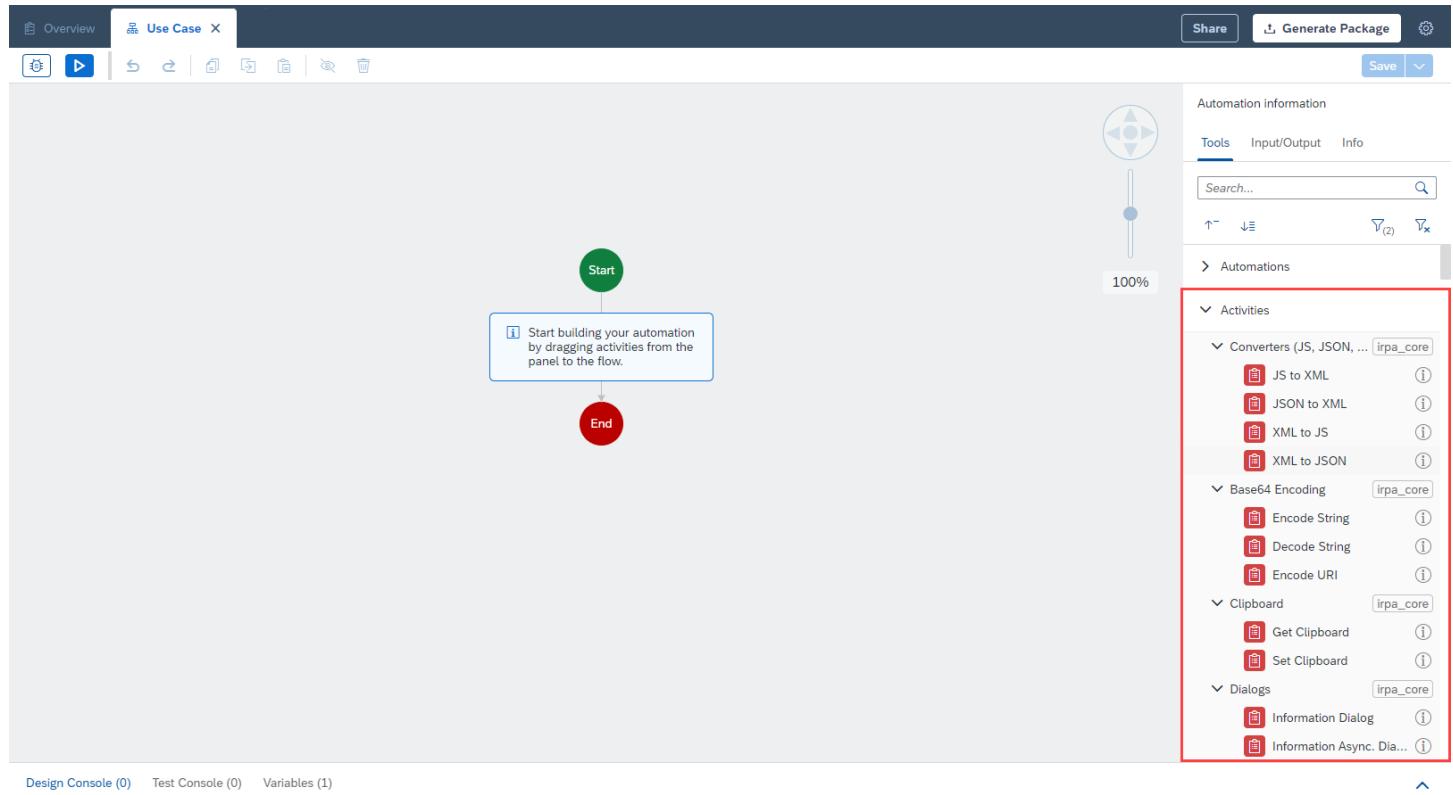
Related Information

[Expression Editor](#)

Add an Activity to an Automation

In the Cloud Studio, activities are provided with the SDK packages you import the first time you create an automation, to build the workflow of your automation.

Activities are organized in classes and subclasses in the tools section on the side panel of the automation.



Each activity executes an action according to the class it belongs to. The following example displays the activities in the `irpa-core` subclass activity **Clipboard**.

The screenshot shows the Activity Catalog with a search bar containing 'Clipboard'. Below the search bar, a tree view shows 'Clipboard' expanded, and 'irpa_core' selected. Under 'Clipboard', two activities are listed: 'Get Clipboard' and 'Set Clipboard', each with an info icon.

The activities in this subclass perform actions linked to the clipboard:

- **Get Clipboard:** Retrieve the textual content of the clipboard.
- **Set Clipboard:** Clear the clipboard and then set text on clipboard.

To build your automation you can drag and drop the activities in your workflow where they become the steps of your automation. Each activity has different parameters that you can define such as the name of the step, and the input and output parameters.

Filtering Feature

To help you look for an activity, the Cloud Studio provides a filtering feature.

The screenshot shows the 'Automation information' panel with tabs for 'Tools' (selected), 'Input/Output', and 'Info'. Below the tabs is a search bar with the placeholder 'Search...' and a magnifying glass icon. The search bar is highlighted with a red border.

The filtering feature displays the tools from the **Tools** side panel that match your search text. The following example displays the filtering feature when you search for a workbook activity.

Automation information

The screenshot shows the SAP Cloud Studio interface with the 'Tools' tab selected. A search bar at the top contains the text 'workbook'. Below the search bar, there are two filter icons: a funnel with '(2)' and a funnel with an 'x'. A section titled 'Activities' is expanded, showing two categories: 'Application' and 'Workbook'. Both categories have a placeholder 'irpa_excel' next to their names. Under 'Application', there are two items: 'Get Active Workbook Na...' and 'Get Workbook Names'. Under 'Workbook', there are nine items: 'Open Workbook', 'Close Workbook', 'Save Workbook', 'Save As Workbook', 'Export As (Workbook)', 'Add Workbook', 'Activate Workbook', 'Get Worksheets Name', and 'Get Worksheets Count'. Each item has a small icon and a help button (a blue circle with an 'i'). A red box highlights the search bar and the list of activities.

- Tools Input/Output Info
- X 🔍
- ↑ ↓ Filter (2) Filter x
- Activities
 - Application irpa_excel
 - 📋 Get Active Workbook Na... i
 - 📋 Get Workbook Names i
 - Workbook irpa_excel
 - 📋 Open Workbook i
 - 📋 Close Workbook i
 - 📋 Save Workbook i
 - 📋 Save As Workbook i
 - 📋 Export As (Workbook) i
 - 📋 Add Workbook i
 - 📋 Activate Workbook i
 - 📋 Get Worksheets Name i
 - 📋 Get Worksheets Count i

More Activities

When you search for activities, controls, or data in the Search bar of **Tools** and if you find no result for your search criteria, then the Cloud Studio suggests more options to find appropriate items.

The screenshot shows the SAP Intelligent Robotic Process Automation Cloud Studio interface. On the left, there's a toolbar with various icons like Overview, Use Case, Share, Generate Package, and Save. The main workspace displays a simple workflow diagram with a green Start node at the top, a blue End node at the bottom, and a yellow rounded rectangle in the middle containing the text "Start building your automation by dragging activities from the panel to the flow." To the right of the workspace is a sidebar titled "Automation information" with tabs for Tools, Input/Output, and Info. Under the Info tab, a search bar shows "outlook" with a magnifying glass icon. Below the search bar, it says "No result for those criteria" and "More features and activities are available here: Manage dependencies, Visit the Store". A red box highlights the search bar and the results area.

Manage Dependencies

Use this option to add appropriate dependencies.

To add appropriate dependencies, click **Manage dependencies** and you will be redirected to the **Add Dependency Screen**.

The screenshot shows the "Project Properties" screen. On the left is a sidebar with options: General, Share, Environment Variables, Dependencies (which is selected and highlighted in blue), Configure Agent Version, Attributes, and Data Protection. The main area is titled "Add Dependency" and contains fields for "Package" (SAP Intelligent RPA Outlook SDK), "Version" (1.15.66, status: Released), and "Alias" (irpa_outlook). At the bottom is a blue "Add" button. A red box highlights the "Dependencies" option in the sidebar and the "Add" button.

To know more details about the dependencies and adding appropriate dependencies, refer to the [Manage Project Dependencies](#) section.

Visit the Store

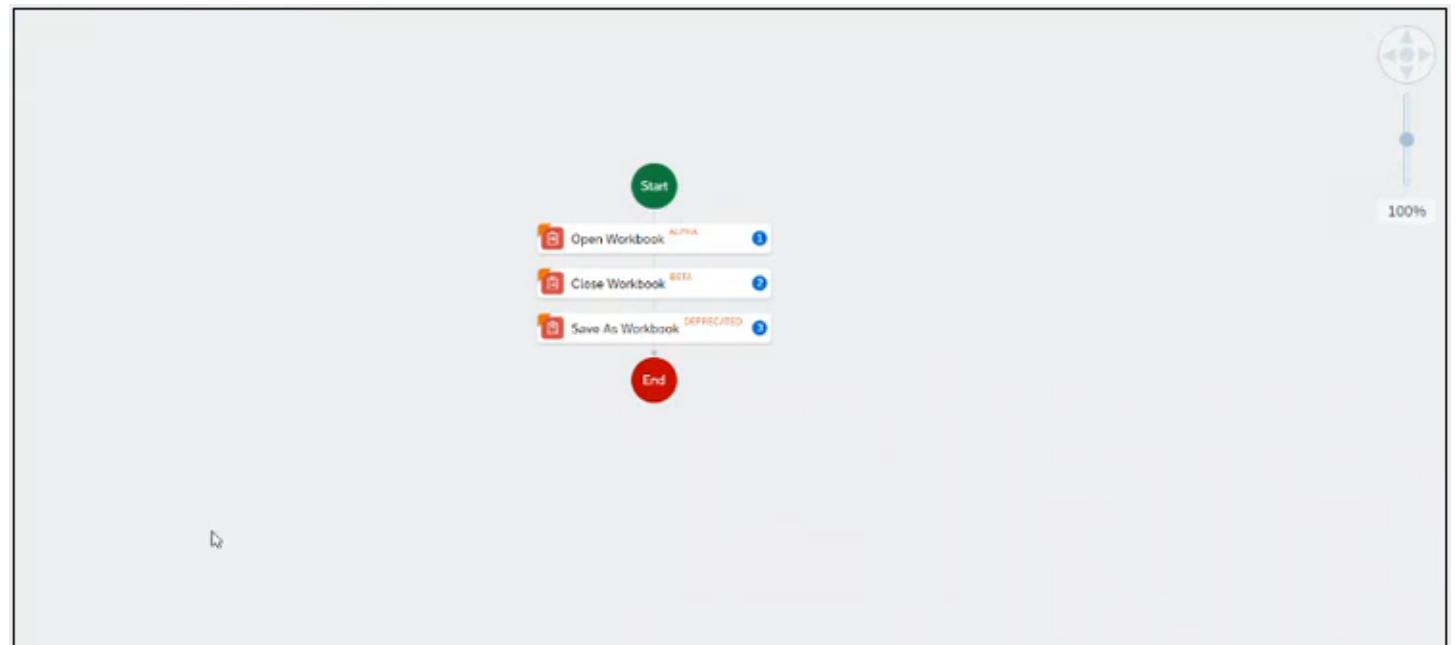
Use this option to search for appropriate predefined content for your automation in Store.

To visit the Store, click [Visit the Store](#). You will be redirected to Store, where you can access predefined content for your automation.

To know more details about the Store, refer to the [Store](#) section.

Display an Activity Status

You can view the status (Alpha, Beta, Active, Deprecated, and Decommissioned) of an activity in an automation workflow.



If an activity status is Alpha, Beta, or Deprecated then you cannot use that activity in the production.

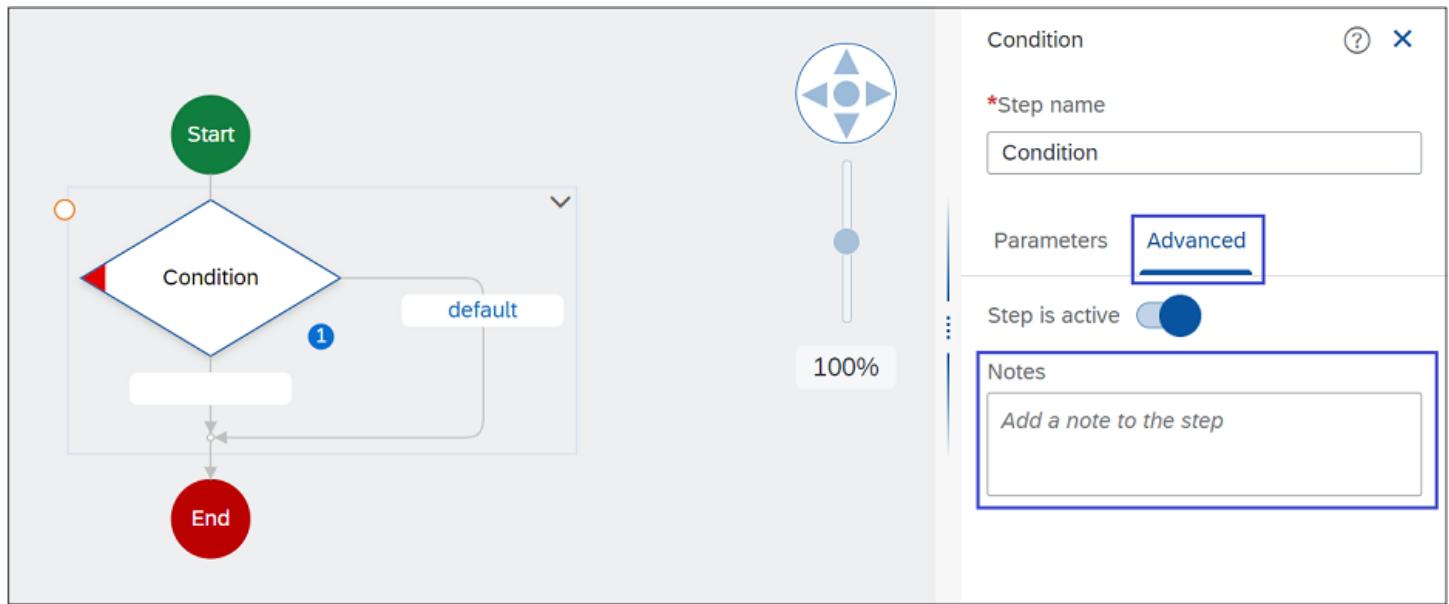
In an automation workflow, right-click on a deprecated activity to view the substitute activity suggested by the system. If the suggested activity has the same output and input as the deprecated activity, then you can keep the value of input and output parameters.

For more details about the activity status, refer to the [Lifecycle of SDK Activities](#) section.

Annotation

In the Cloud Studio, you can add a note for each step (activity, node, user task, and many more) in an automation workflow. This helps you to add a brief description about an automation step.

The **Notes** option is available under **Advanced** tab of a selected step or node on the right panel of the Cloud Studio. For example, refer to the following screenshot.

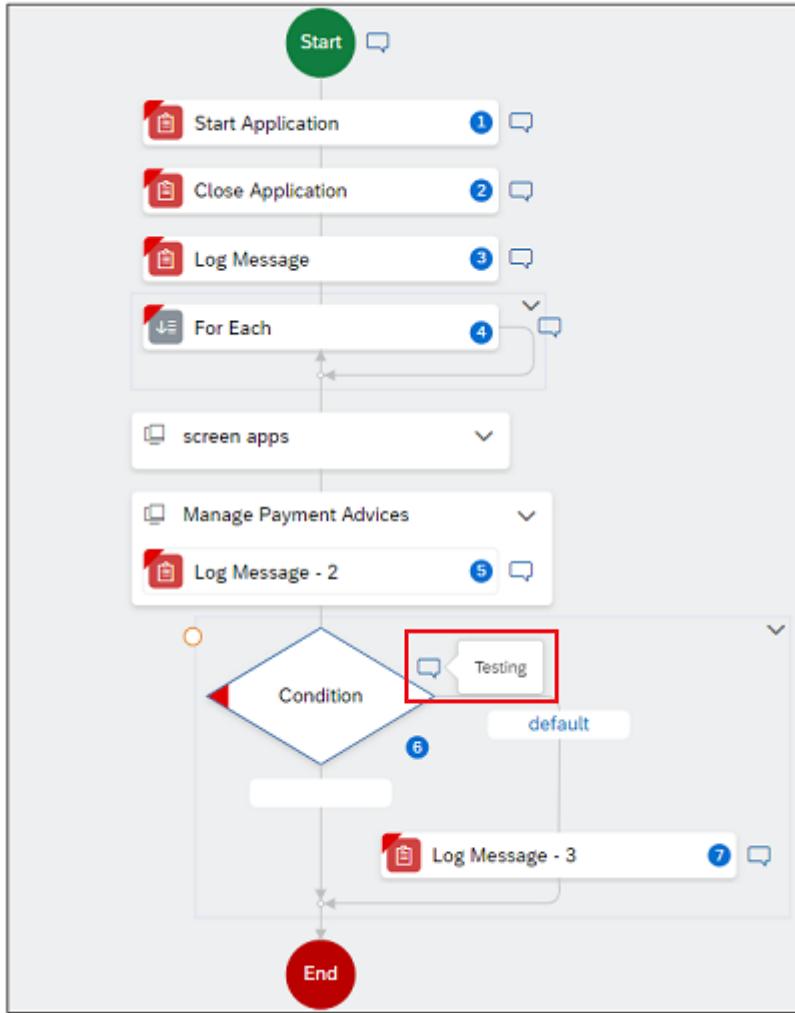


You can enter an appropriate information in the [Add a note to the step](#) field.

i Note

You can enter a maximum of 1024 characters.

After adding the information for a step, the **Note** icon is displayed on the right side of the step. Click the **Note** icon to view the content of the note.



i Note

You cannot add a note for a screen.

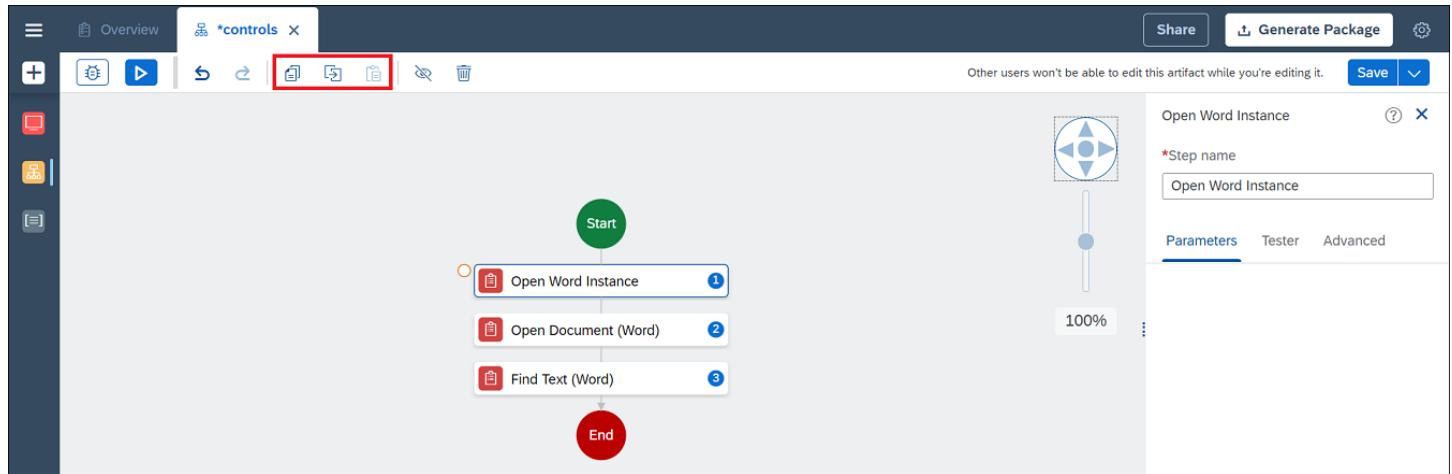
Copy, Paste, and Duplicate Items

You can copy and paste or duplicate the [Activities](#), [Data](#) (Data Management and Data Types), and [Controls](#) with their value in the workflow of your automation.

→ Tip

The Copy, Paste and Duplicate options can be used from one automation to another across your different projects.

The following screenshot shows the Copy, Paste, and Duplicate options in the Cloud Studio.



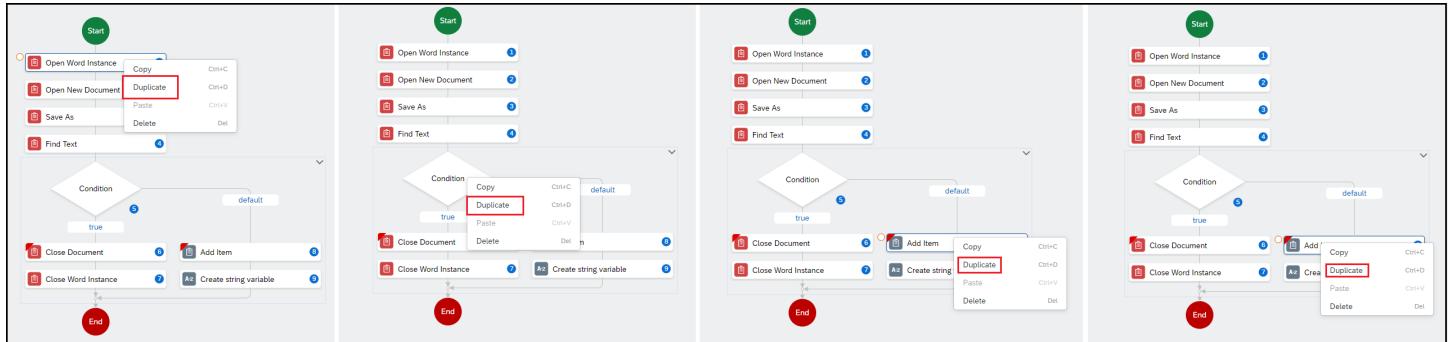
Icon	Icon Name	Description	Procedure
	Copy	The copy function is handled by the Cloud Studio. Use this option to copy an item (Activity or Data (Data Management and Data Types), and Controls).	Select an item to be copied (Activity or Data (Data Management and Data Types) and Controls) in a workflow of your automation and then click the Copy icon. The item is now copied.
	Paste	The Paste function is handled by the Cloud Studio. Use this option to paste a copied item (Activity or Data (Data Management and Data Types), and Controls) at the desired location.	Select an item (Activity or Data (Data Management and Data Types) and Controls) in a workflow and then click the Paste icon. The item is now pasted after the selected item. i Note If do not select an item and you paste, the copied item will be pasted outside the workflow of your automation.
	Duplicate	The Duplicate function is handled by the Cloud Studio. Use this option to duplicate an item (Activity or Data (Data Management and Data Types), and Controls).	Select an item (Activity or Data (Data Management and Data Types) and Controls) in a workflow and then click the Duplicate icon. The item is now duplicated and is available after the selected item.

i Note

When you copy an item from one automation to another, click on the green start node of the second automation to paste the items directly in the workflow of the automation.

You can also use the Duplicate option at item (Activity or Data (Data Management and Data Types), and Controls) level.

The following screenshot shows the availability of the Duplicate option of items.



To create a duplicate item, follow the procedure below:

1. Select an item and right-click.

The more options are displayed.

2. Click **Duplicate**.

The duplicate item is created after the selected item.

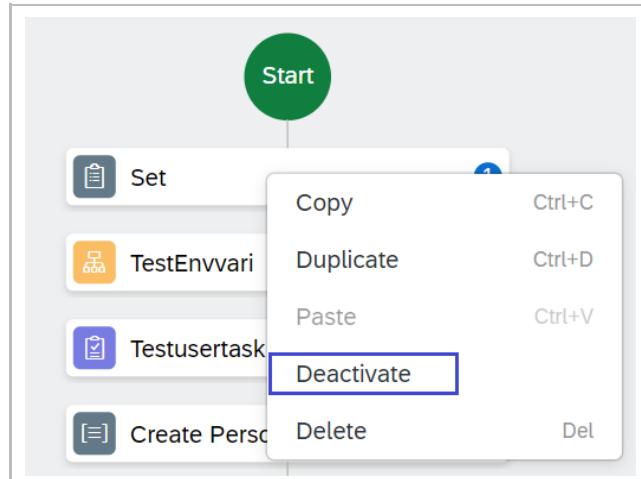
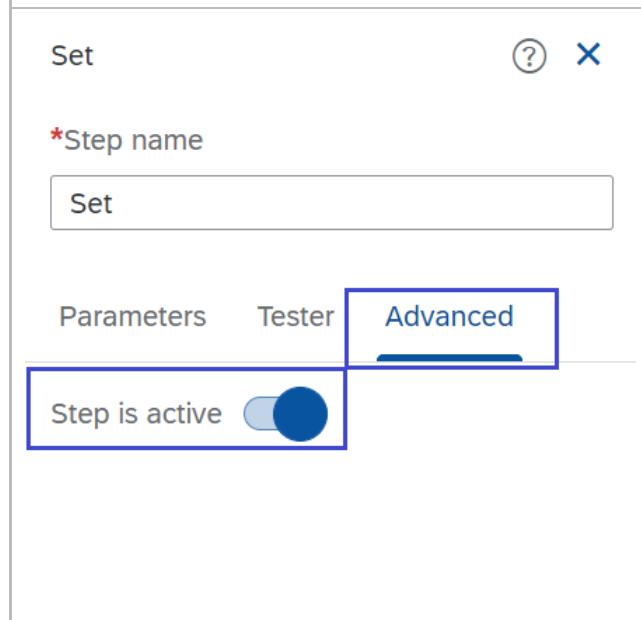
Deactivate an Automation Step

You can deactivate certain steps in a workflow of an automation.

To deactivate a step in a workflow, you must select the step or node to be deactivated and then deactivate. The step or node is deactivated. This means the deactivated step or node will not be executed during automation.

In the Cloud Studio, the Deactivate option is available in various locations.

Deactivate option	Description
	<p>By default, this icon is disabled.</p>  <p>Once you select the step, the  icon is enabled.</p> <p>Select the step or node to be deactivated and click the  icon to deactivate a step.</p> <p>i Note</p> <p>At any point in time, you can activate the step by clicking the  icon.</p>

	<p>In a workflow of your automation, right click on any step or node. The Deactivate option is displayed.</p> <p>Select the step or node to be deactivated, right click, and then click Deactivate to deactivate the step or node.</p> <p>After successful deactivation, the step or node will be greyed.</p> <p>i Note</p> <p>At any point in time, you can activate the step by clicking Activate.</p>
	<p>The activate or deactivate option is available under Advanced of a selected step or node on the right panel of the Cloud Studio.</p> <p>By default, the step status is active.</p> <p>Select the step to be deactivated and then click the  icon to deactivate the step.</p> <p>After successful deactivation, the step or node will be greyed.</p> <p>i Note</p> <p>At any point in time, you can activate the step by clicking the  icon.</p>

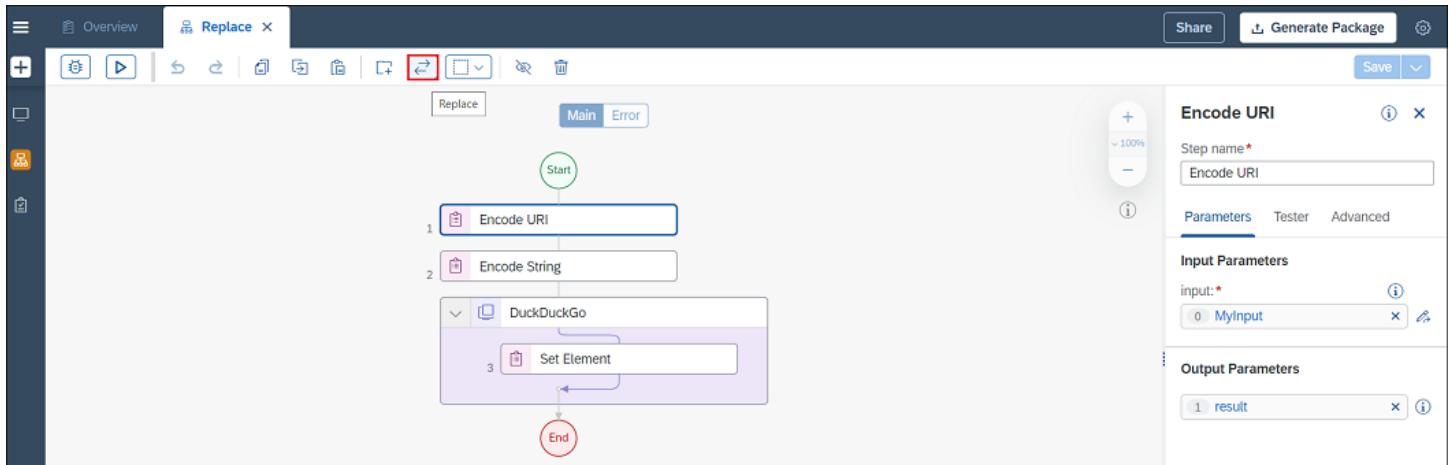
Replace an Activity with Another Activity

You can replace an activity and its value with another activity in a workflow of your automation.

i Note

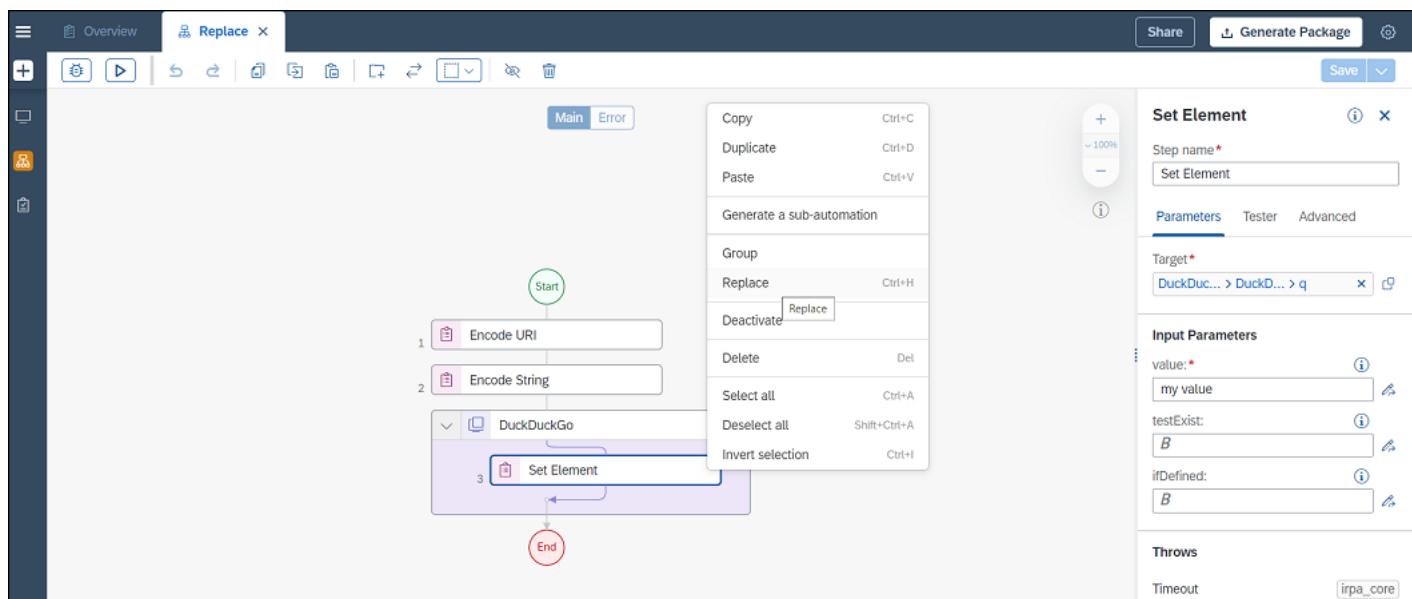
- It is recommended to use the replace activity functionality when both the activities have similar type of input or output parameters.
- It is possible to replace multiple activities at once if you select multiple activities in the workflow of your automation.

The following screenshot shows the Replace option in the Cloud Studio.



i Note

It is also possible to replace an activity that has specific screen element of an application.



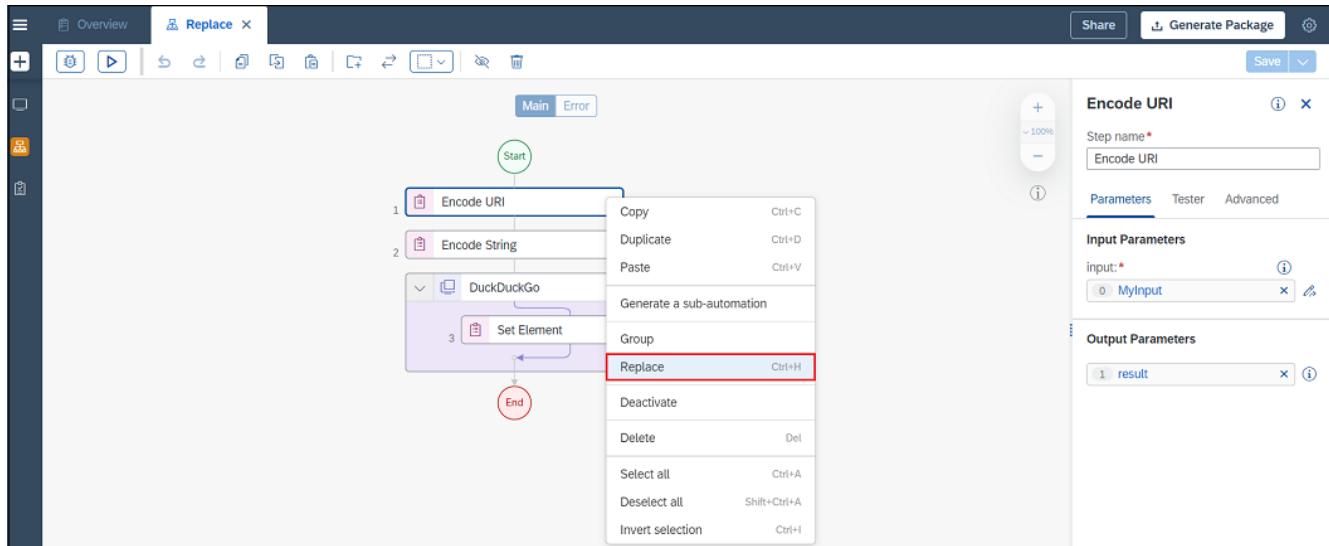
Icon	Icon Name	Description	Procedure
➡	Replace	<p>The Replace function is handled by the Cloud Studio.</p> <p>Use this option to replace an activity with another activity in the workflow of your automation.</p>	<p>Select an activity to be replaced in a workflow of your automation and then click the Replace icon. The Replace Activity pop-up is displayed.</p> <p>From the Replace Activity pop-up window, choose the activity which will replace the selected activity in the workflow of your automation. The activity is now replaced.</p>

To replace an activity, you can also refer to the following procedure:

1. Select an activity, then right-click. The more options are displayed.
2. Click **Replace**. The **Replace Activity** pop-up is displayed.

i Note

You can also use the keyboard shortcut, Ctrl + H to open the **Replace Activity** pop-up window.



3. From the **Replace Activity** pop-up window, choose the activity which will replace the selected activity in the workflow of your automation.
4. Click **Replace**. The activity is now replaced.
5. Type an activity name in the search bar of the **Replace Activity** pop-up window to quickly find an activity. Only activities matching the name you type are displayed.
6. Check the **Keep original name** checkbox if you want to keep same name of the replaced activity.
7. Click **Close** to close the **Replace Activity** pop-up window.

Replace Activity

🔍

⬆️ ⬇️ ✖️ (2)

▼ Converters (JS, JSON, XML...) irpa_core

📋	JS to XML	ⓘ
📋	JSON to XML	ⓘ
📋	XML to JS	ⓘ
📋	XML to JSON	ⓘ
>	Mouse	irpa_core
>	Base64 Encoding	irpa_core
>	Dialogs	irpa_core
>	Clipboard	irpa_core
>	WMI Client	irpa_core
>	Local Agent Variables	irpa core
<input type="checkbox"/>	Keep original name	Replace Close

More Information

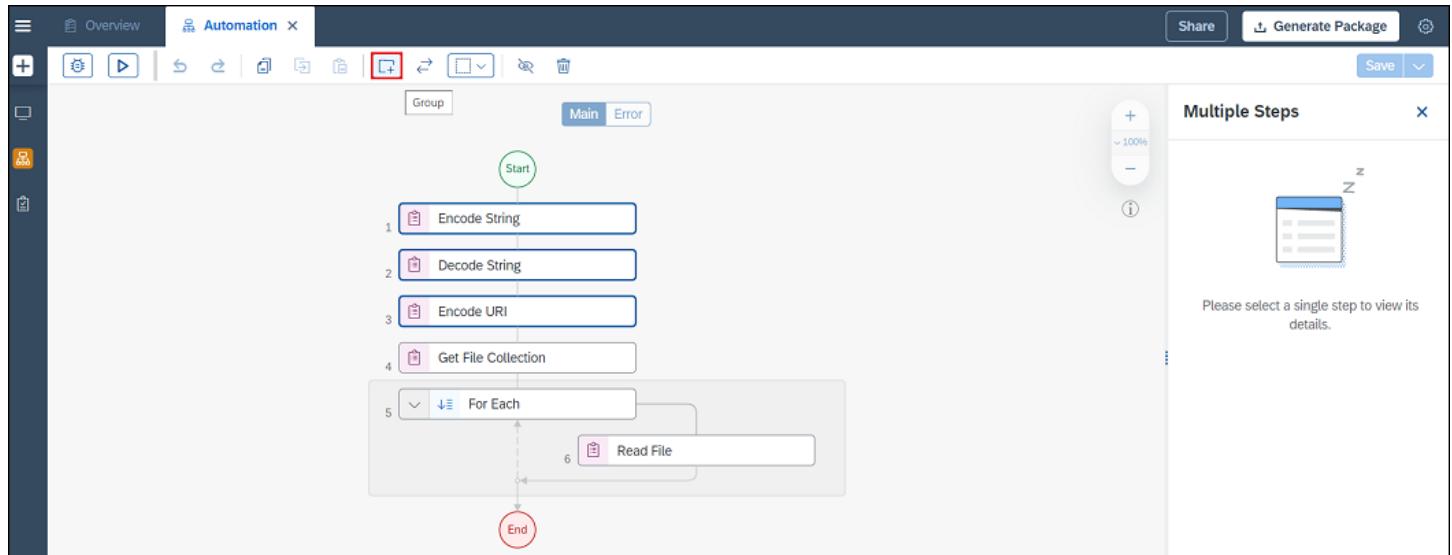
Once the activity is replaced, only the target properties will be copied to the replaced activity. The unmatched properties will be deleted automatically.

Group and Ungroup Activities

Group Activities

You can group a set of activities in the workflow of your automation.

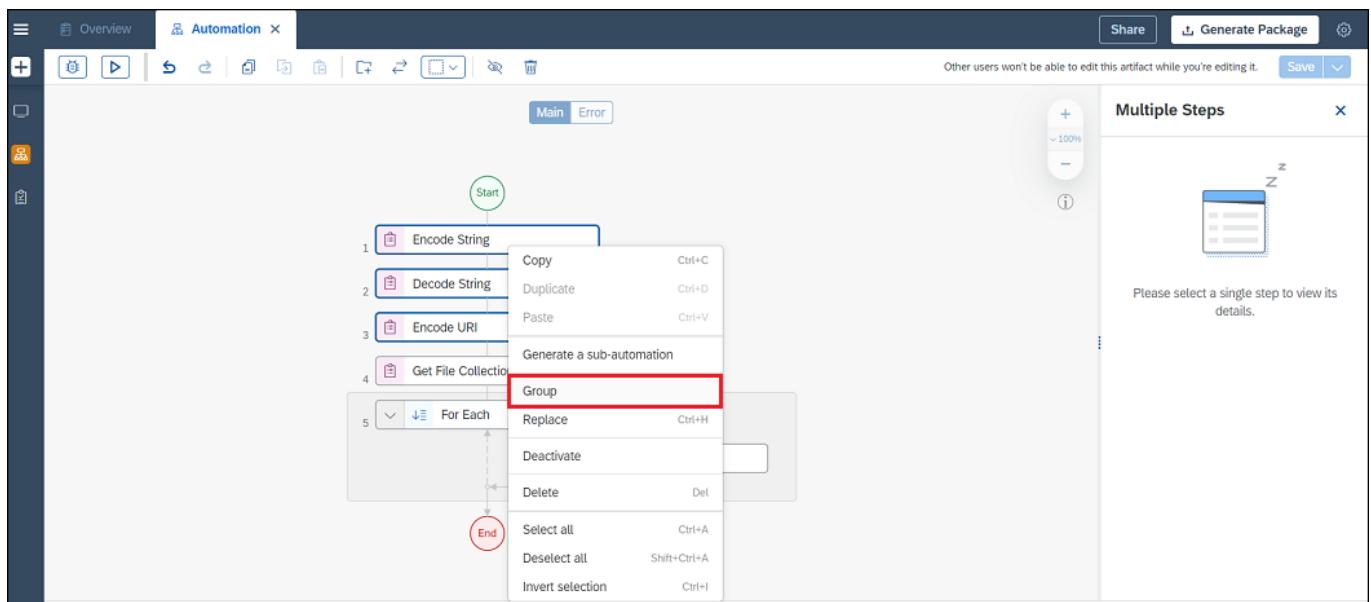
The following screenshot shows the **Group** option in the Cloud Studio.



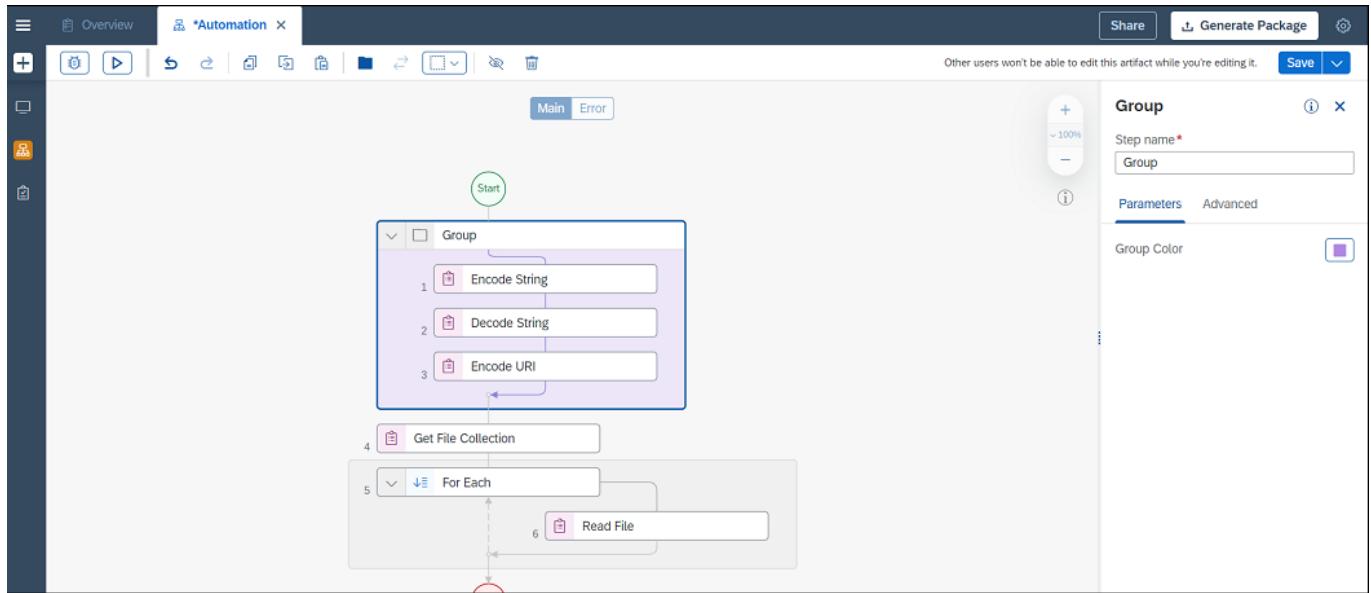
Icon	Icon Name	Description	Procedure
	Group	<p>The Group function is handled by the Cloud Studio.</p> <p>Use this option to group a set of activities in the workflow of your automation.</p>	<p>Select multiple activities to be grouped in a workflow of your automation and then click the Group icon.</p> <p>The set of activities is now grouped together and will be displayed as a single step in the workflow of your automation.</p>

To group a set of activities, you can also refer to the following procedure:

1. Select multiple activities, then right-click. The more options are displayed.
2. Click **Group**.

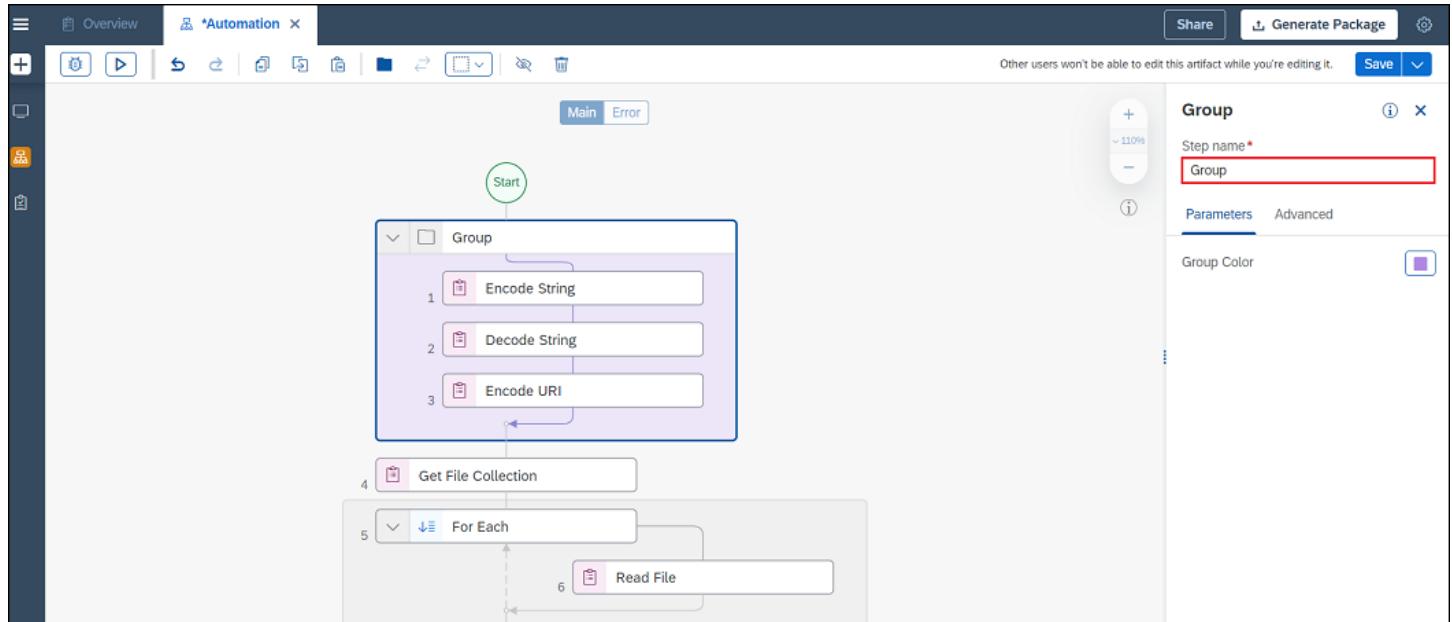


The set of activities is grouped together and is displayed as a **Group** or single step in the workflow of your automation.

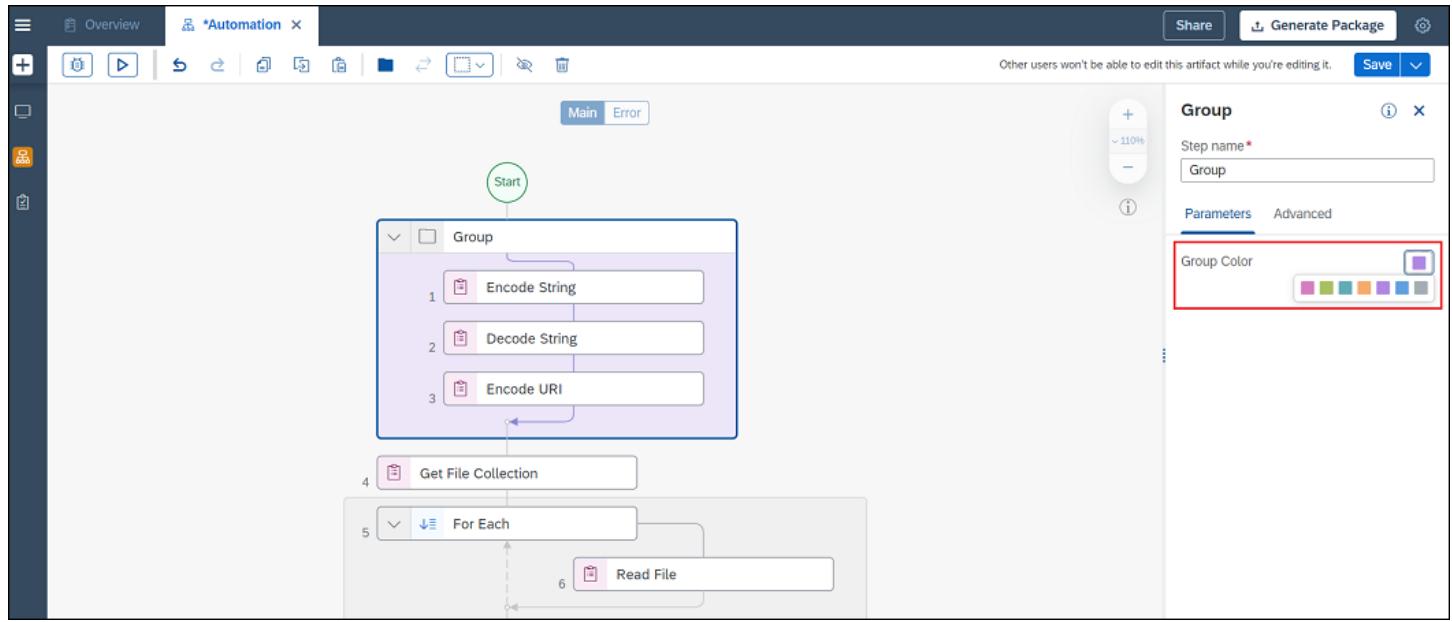


More Information

To change a **Group** name, select the group and then modify the group name on the **Step name** field.

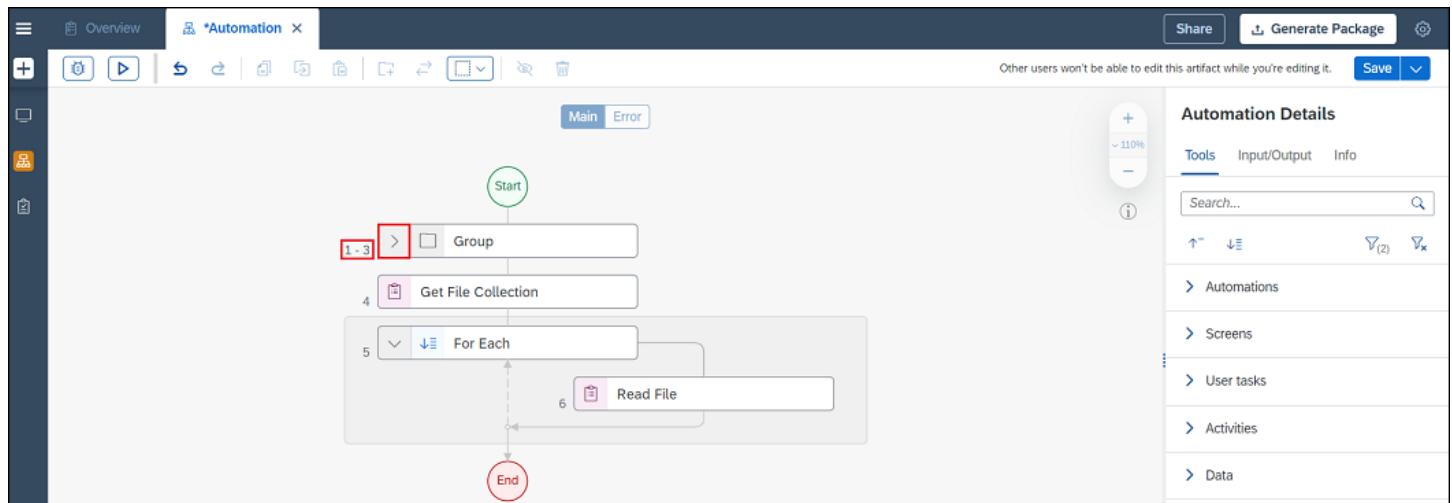


To change the background color of a **Group**, select the group and then click the **Group Color** button and select the color from the various color options.

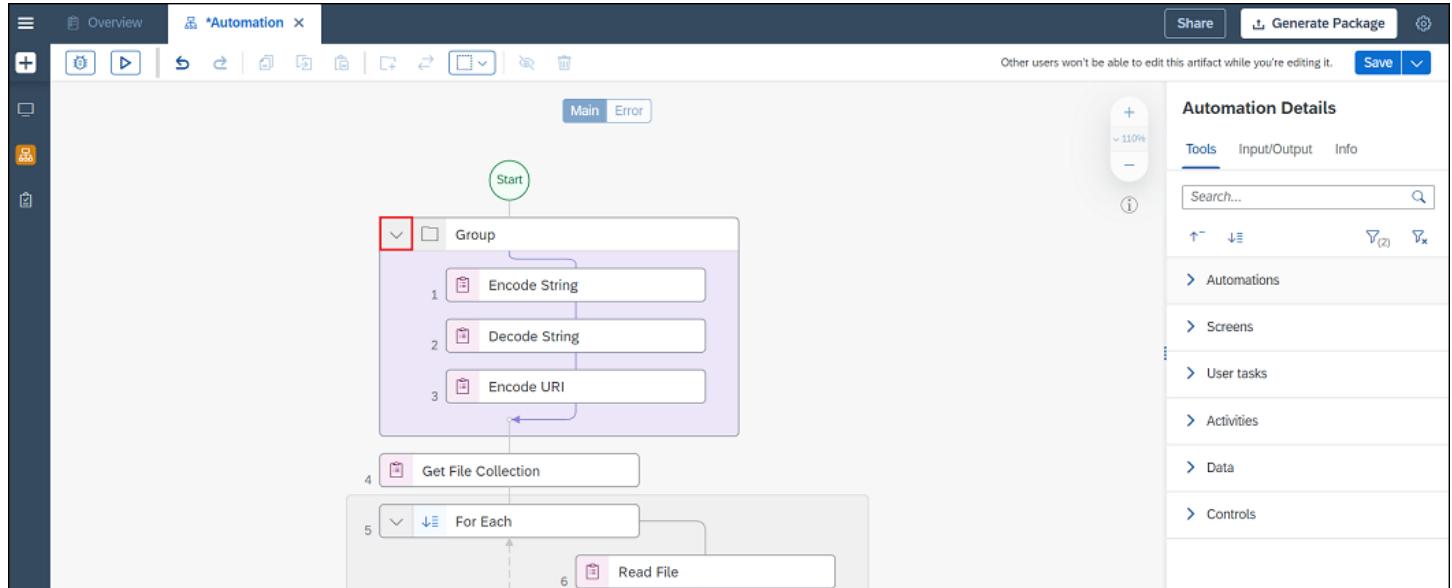


To expand set of activities inside a **Group**, click

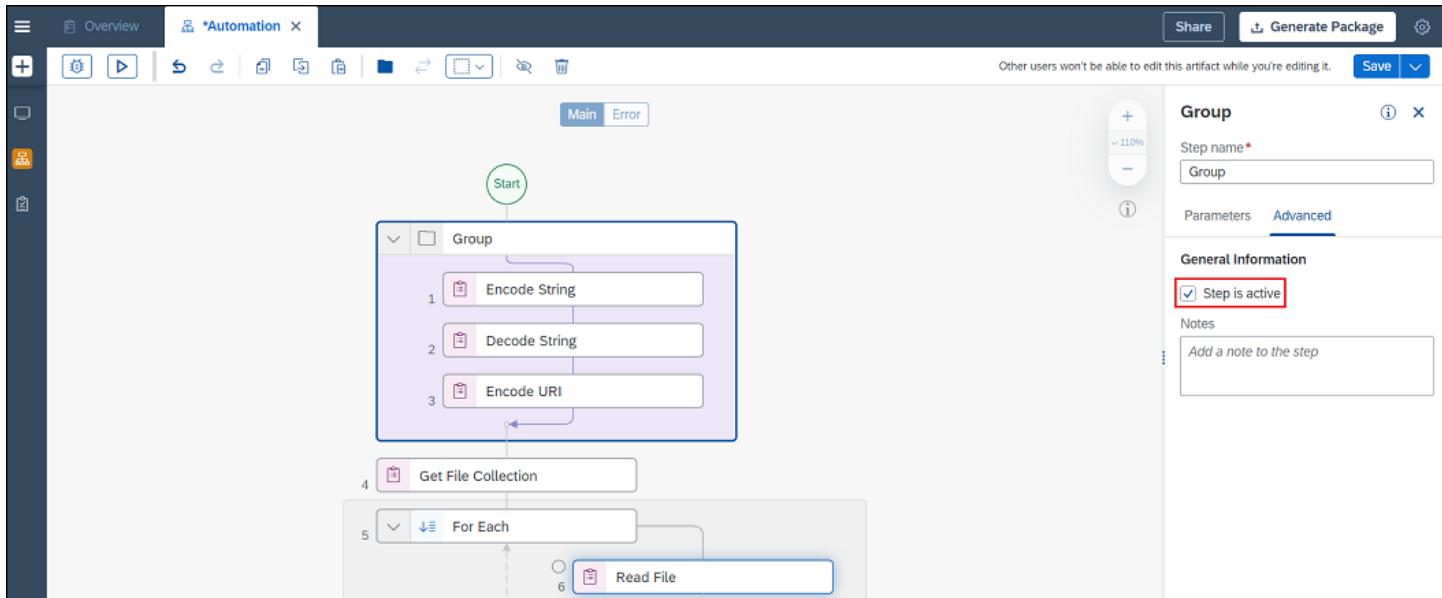
The number beside a **Group** step indicates the number of activities are grouped together. In the following example, **1 – 3** indicates that the **Group** consists of three activities.



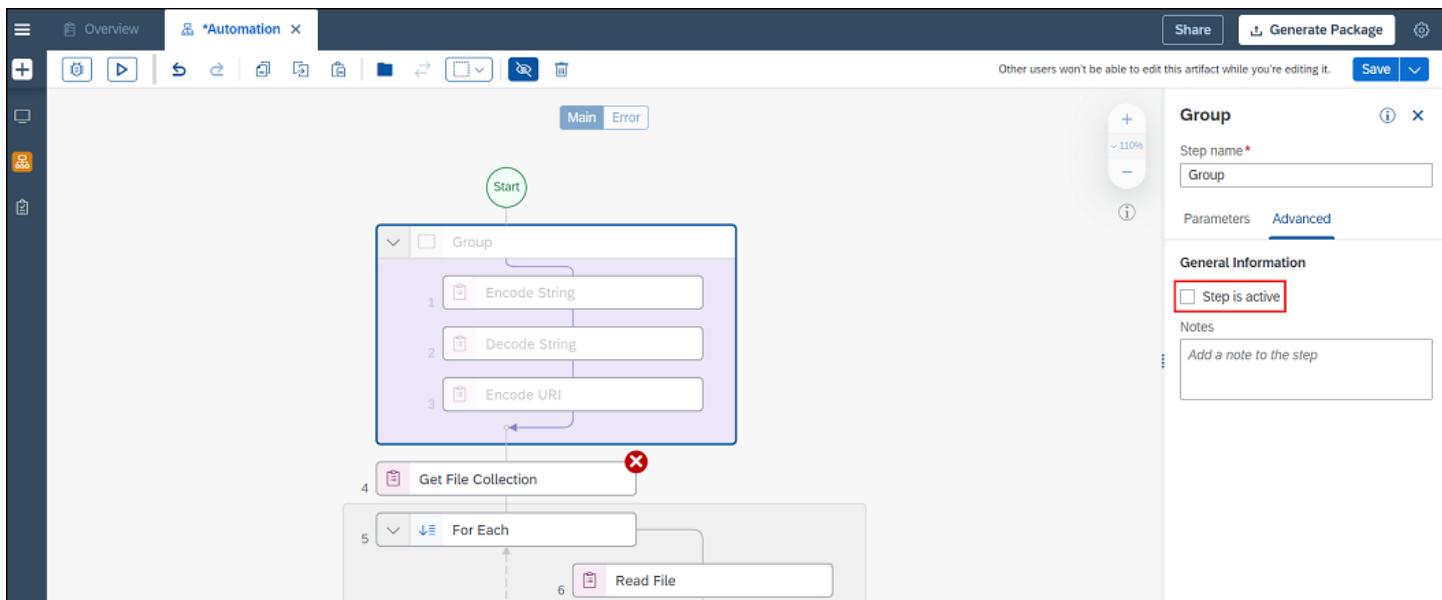
To collapse set of activities inside a Group, click



By default, a **Group** step is always active.



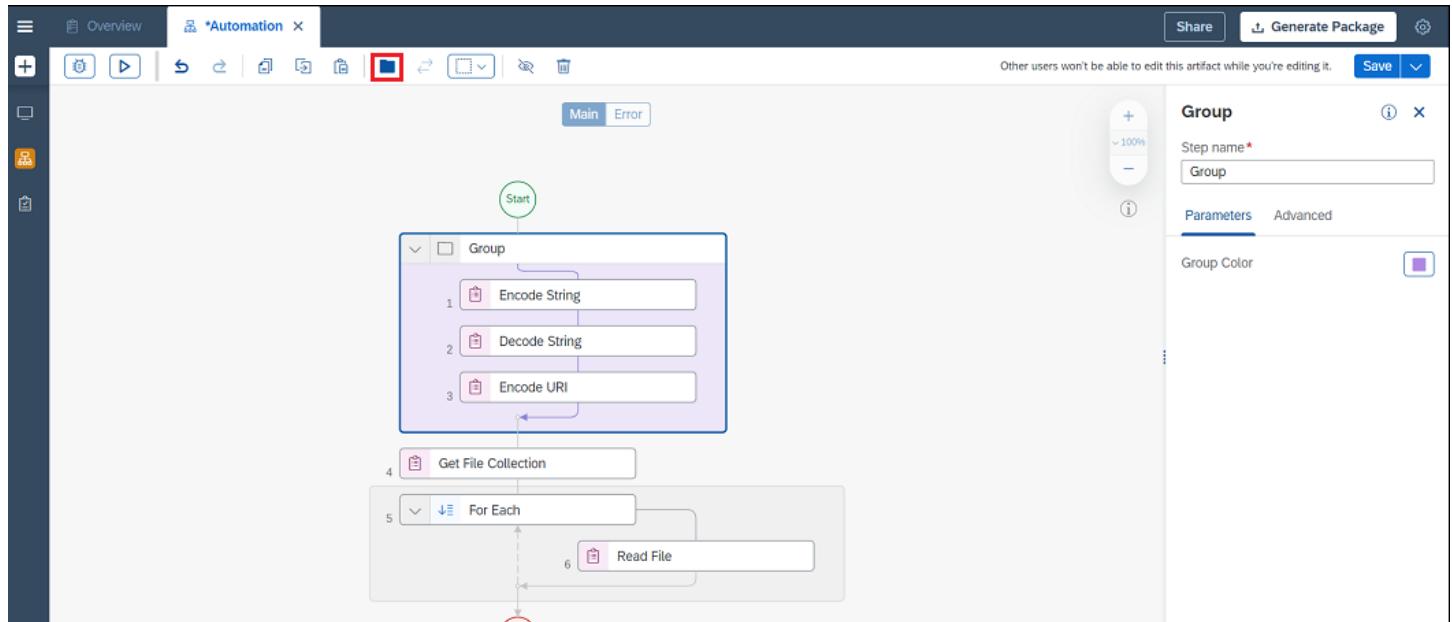
If you uncheck the **Step is active** checkbox, a **Group** step becomes inactive.



Ungroup Activities

You can ungroup a group of activities in the workflow of your automation.

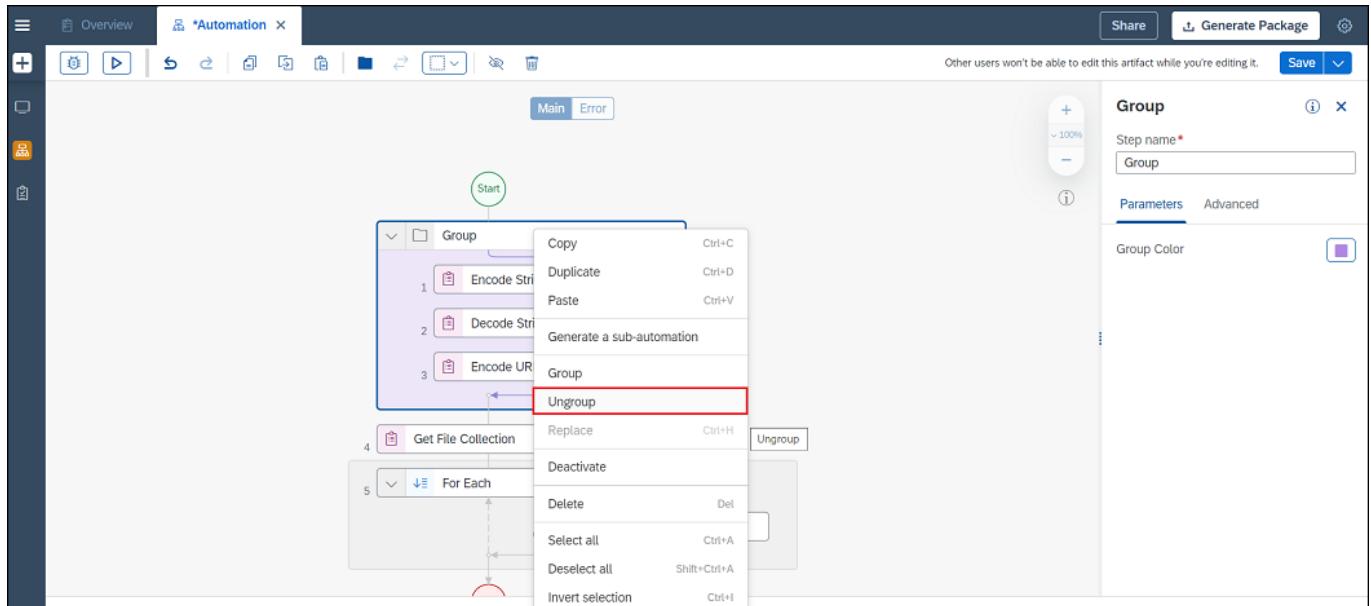
The following screenshot shows the **Ungroup** option in the Cloud Studio.



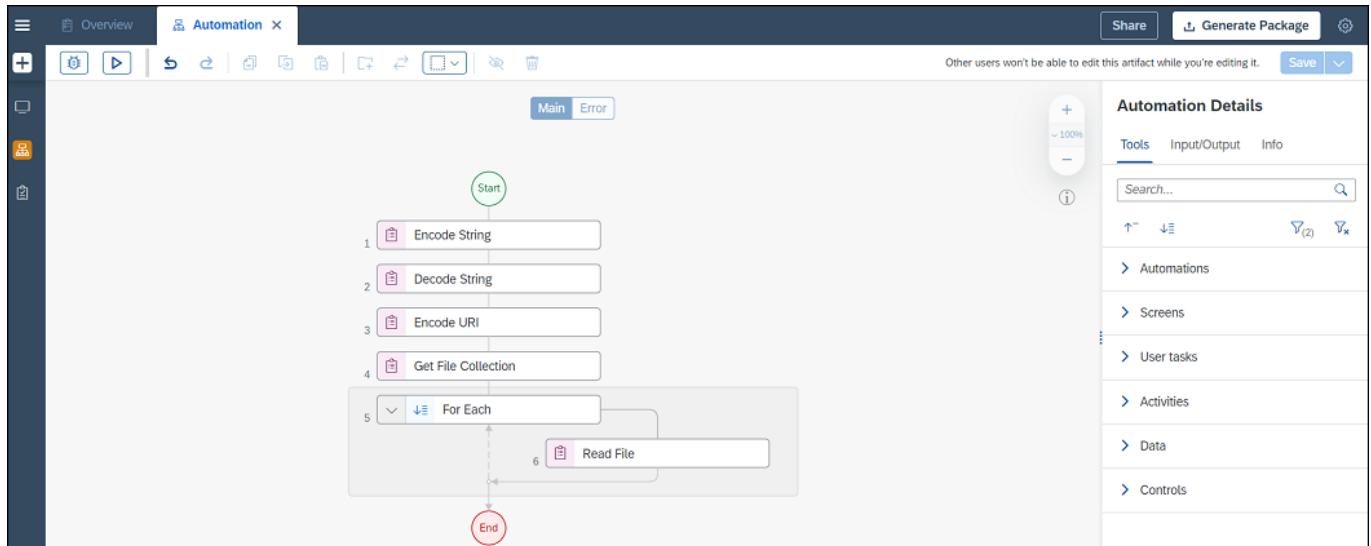
Icon	Icon Name	Description	Procedure
	Ungroup	The Ungroup function is handled by the Cloud Studio. Use this option to ungroup a group of activities in the workflow of your automation.	Select a group to be ungrouped in a workflow of your automation and then click the Ungroup icon. The group of activities is now ungrouped.

To ungroup a group of activities, you can also refer to the following procedure:

1. Select a group, then right-click. The more options are displayed.
2. Click **Ungroup**.



The group of activities becomes ungrouped and is displayed as multiple steps in the workflow of your automation.

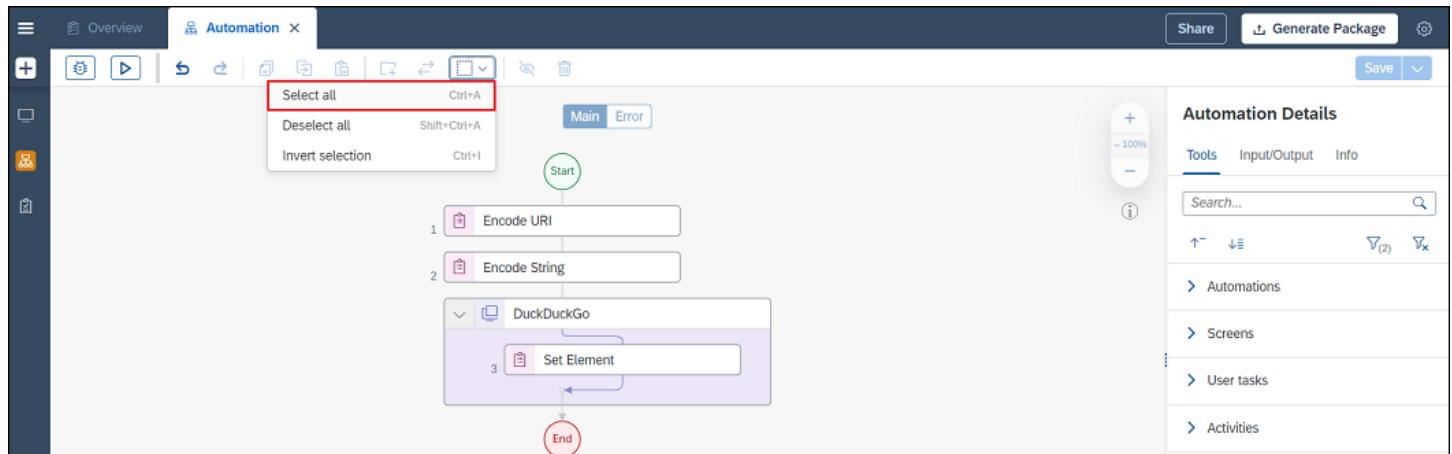


Select or Deselect All Activities and Invert Selection of Activities

Select All Activities

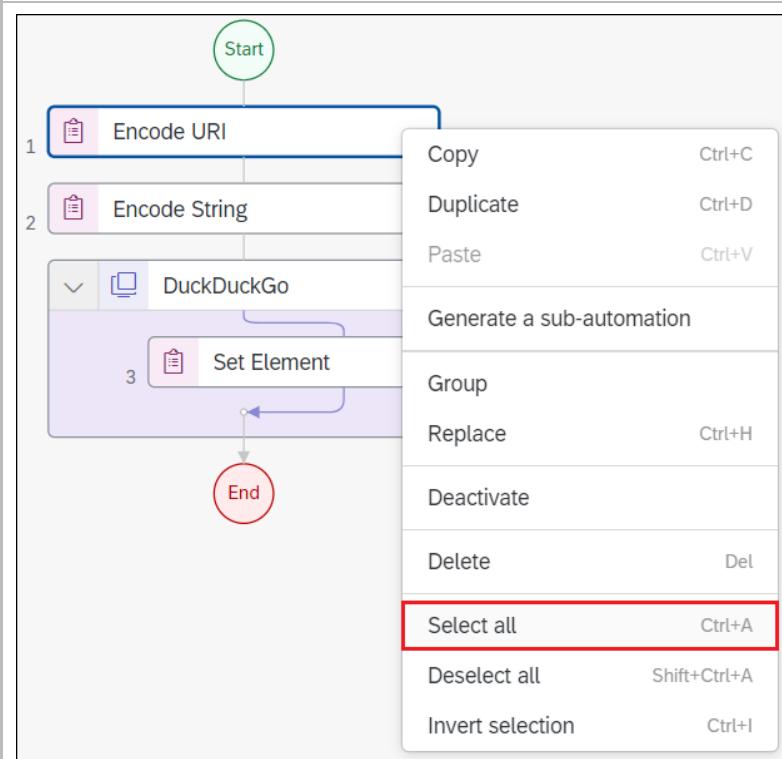
You can select all activities at once in a workflow of automation.

The following screenshot shows the **Select all** option in the Cloud Studio.



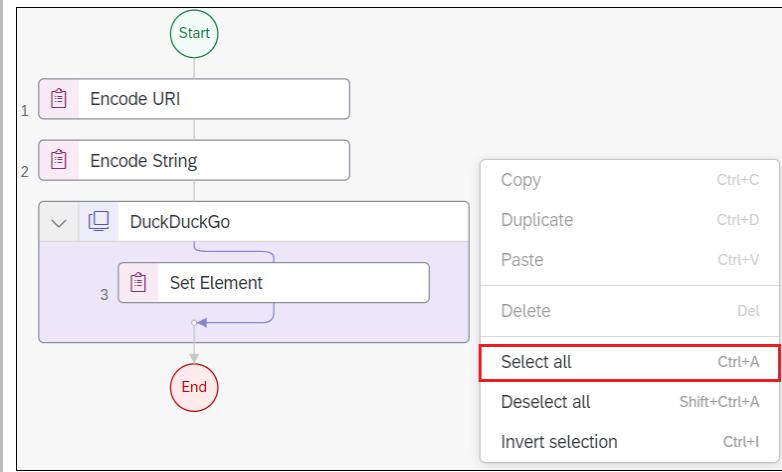
In the Cloud Studio, the **Select all** option is available in various locations.

Select all Option	Description
 Select all Ctrl+A Deselect all Shift+Ctrl+A Invert selection Ctrl+I	<p>Click button and then select the Select all option to select all the steps at once in a workflow of automation.</p>



In a workflow of an automation, select any step and then right-click. The more options are displayed.

Click **Select all** to select all the steps in a workflow of automation.



Right-click on anywhere in a workflow of automation. The more options are displayed.

Click **Select all** to select all the steps at once in a workflow of automation.

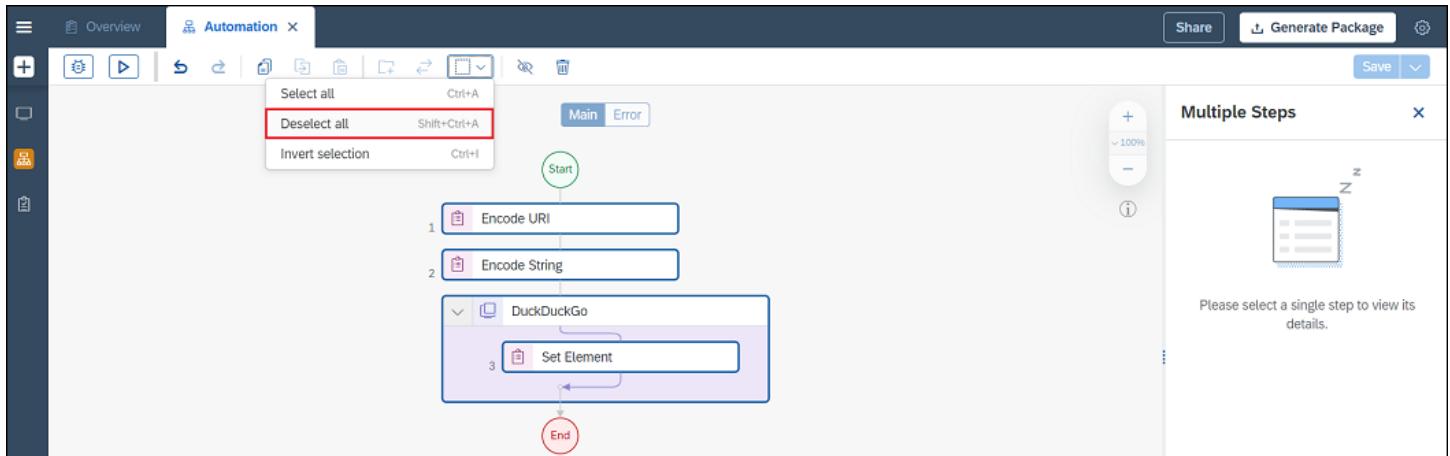
i Note

You can also use the keyboard shortcut, Ctrl + A to select all the steps at once in a workflow of automation.

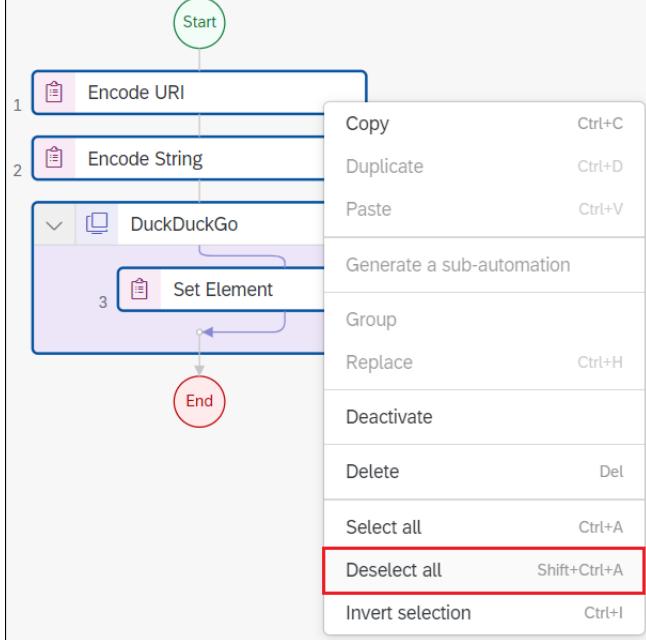
Deselect All Activities

You can deselect all selected activities at once in a workflow of automation.

The following screenshot shows the **Deselect all** option in the Cloud Studio.



In the Cloud Studio, the **Deselect all** option is available in various locations.

Deselect all Option	Description
 Select all Ctrl+A Deselect all Shift+Ctrl+A Invert selection Ctrl+I	Click  button and then select the Deselect all option to deselect all the selected steps at once in a workflow of automation.
	<p>In a workflow of an automation, right-click on any selected step. The more options are displayed.</p> <p>Click Deselect all to deselect all the selected steps at once in a workflow of automation.</p>

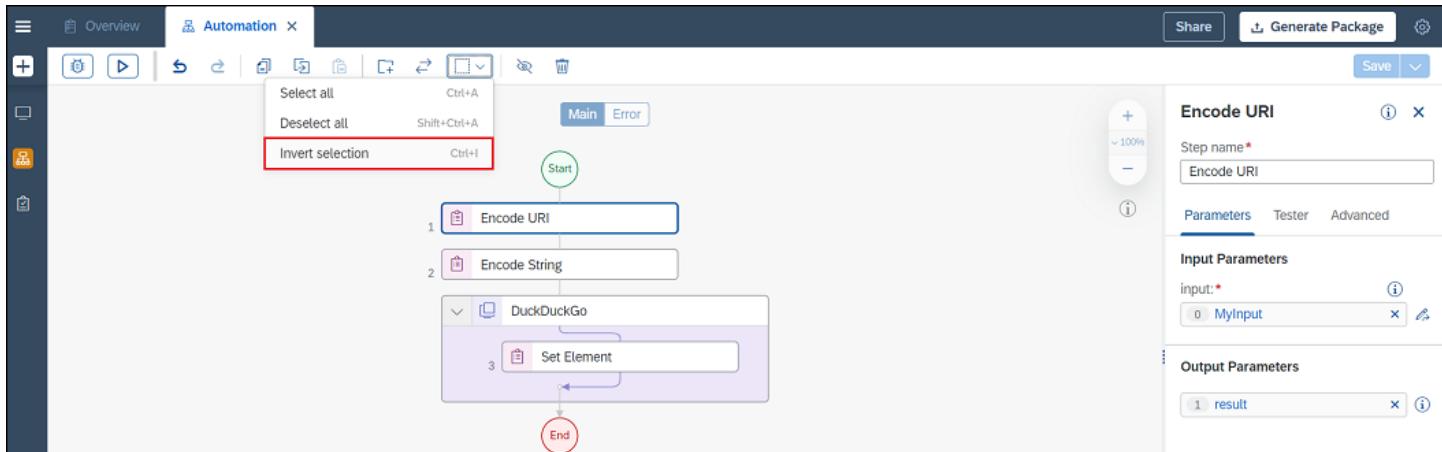
i Note

You can also use the keyboard shortcut, Shift + Ctrl + A to deselect all the selected steps at once in a workflow of automation.

Invert Selection of Activities

You can invert the selection of activities at once in a workflow of automation. Inverting a selection changes the unselected activities into selected activities.

The following screenshot shows the **Invert selection** option in the Cloud Studio.



In the Cloud Studio, the **Invert selection** option is available in various locations.

Invert selection Option	Description
Select all Ctrl+A Deselect all Shift+Ctrl+A Invert selection Ctrl+I	Click button and then select the Invert selection option to invert the selection of steps at once in a workflow of automation.
	In a workflow of automation, select any step(s) to be inverted and then right-click on it. The more options are displayed. Click Invert selection to invert the selection of steps at once in a workflow of automation.

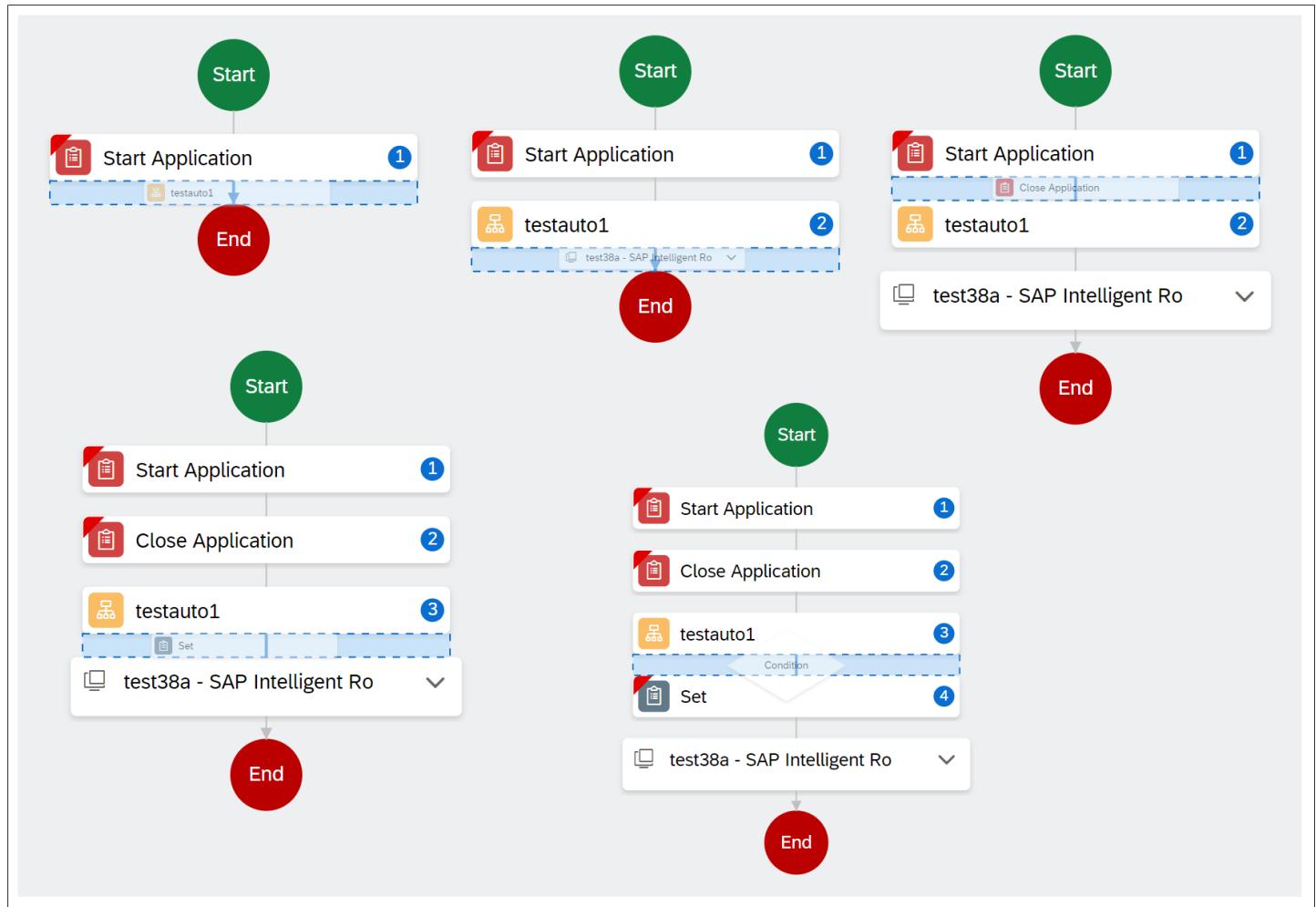
i Note

You can also use the keyboard shortcut, Ctrl + I to invert the selection of steps at once in a workflow of automation.

Drag and Drop Items

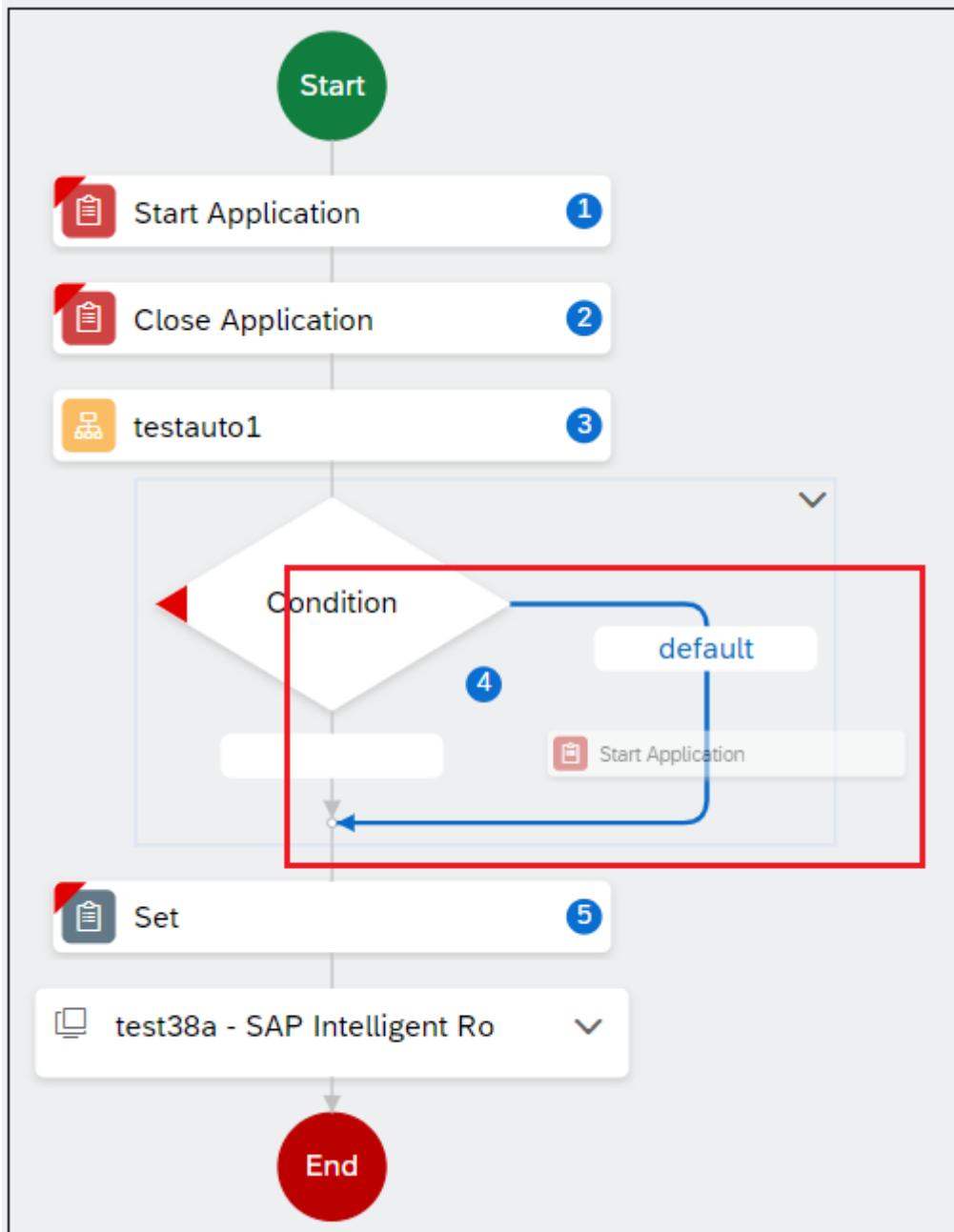
You can easily drag and drop items (Automations, Screens, Activities, Data, and Controls) in an automation.

In an automation, when you mouse hover with an item on the straight line, the straight line is highlighted with a dotted box. This helps you to drop an item at an appropriate place.



i Note

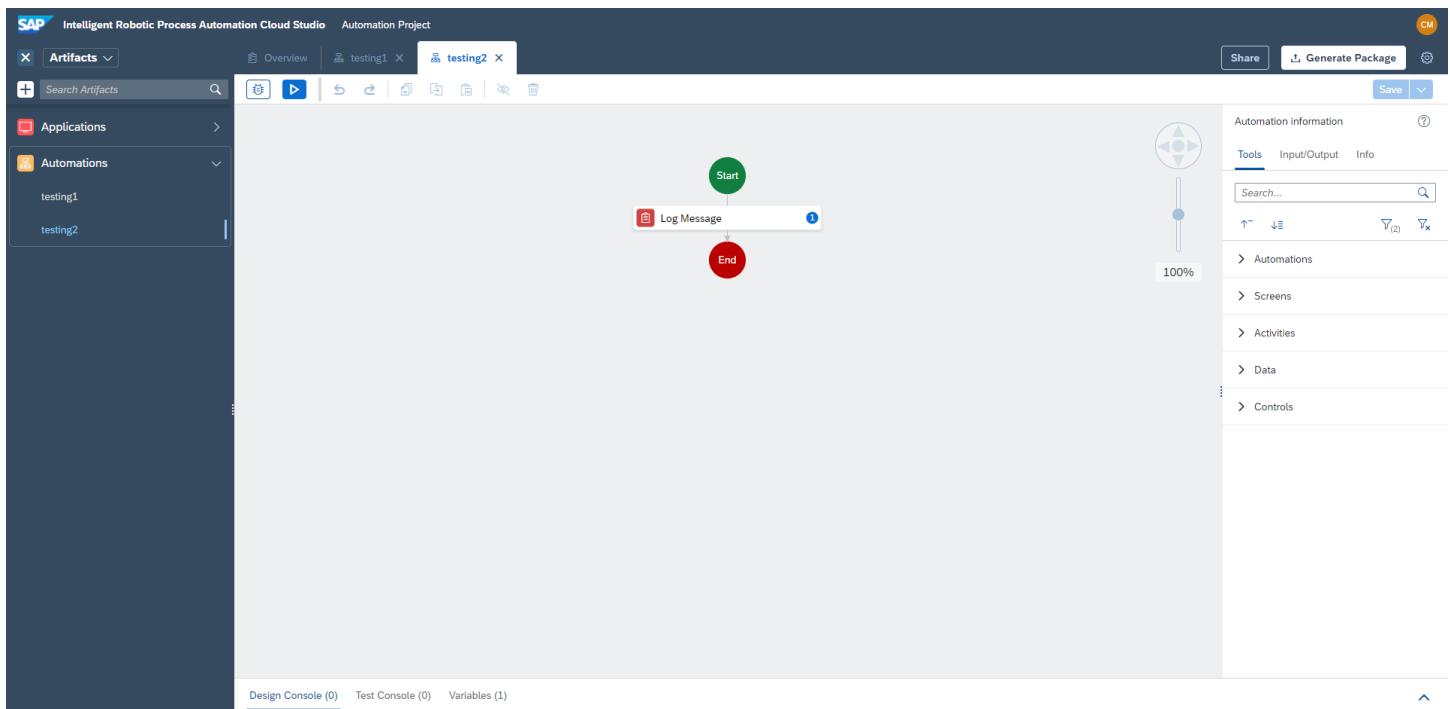
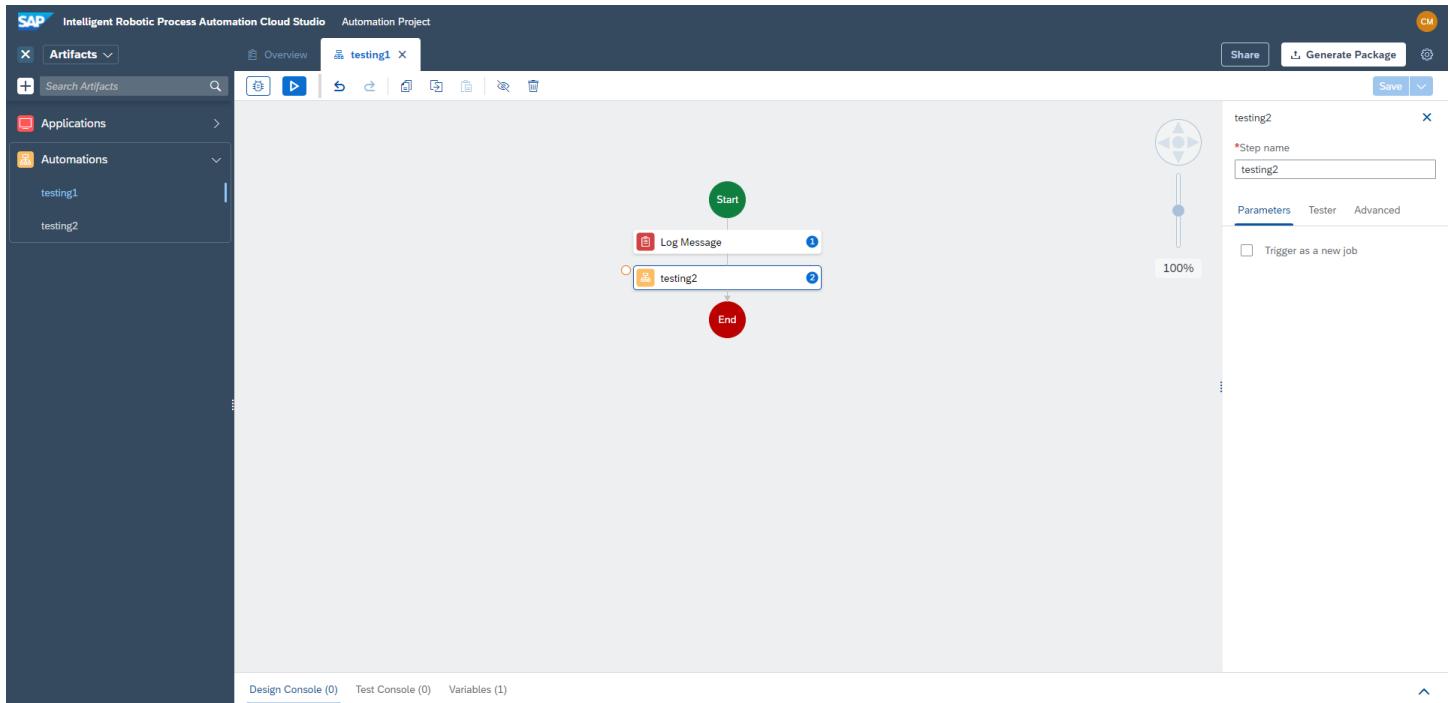
No dotted box is displayed for curved lines.



Open Automation Node in Separate Automation Editor

You can open an automation node in a separate automation editor by double-clicking the automation node in a currently active workflow. The automation or sub-automation opens in a separate automation editor.

In the following example, the **testing2** automation is opened in a separate automation editor by double-clicking the **testing2 automation node** in the **testing1** automation.



Canvas Controls

In the Cloud Studio, you can use different canvas controls and hotkeys while designing your automations.

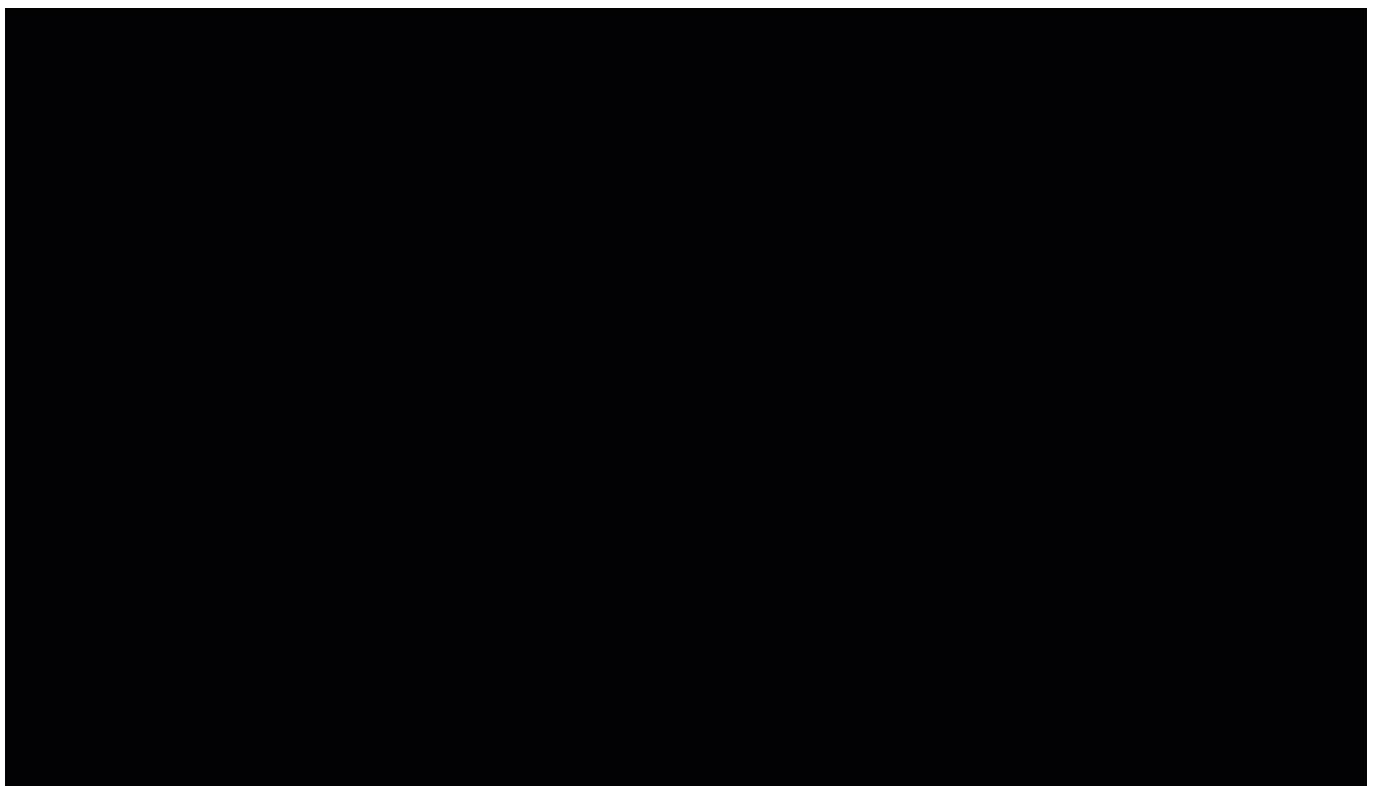
Canvas

- To move the canvas, hold the **space bar** + hold the left/right **mouse key** + move **mouse**, or just hold the **scroll wheel** + move **mouse**. The cursor changes into a hand cursor to show that you can move the canvas.



Simple Selection

- To select the whole group of steps under a control and the control, click the control.
- To select every step in a screen group, click the header of the screen group.
- To select multiple steps or groups of steps, hold down the **Ctrl** key and click each step or group of steps one by one. You can also drag and drop them.



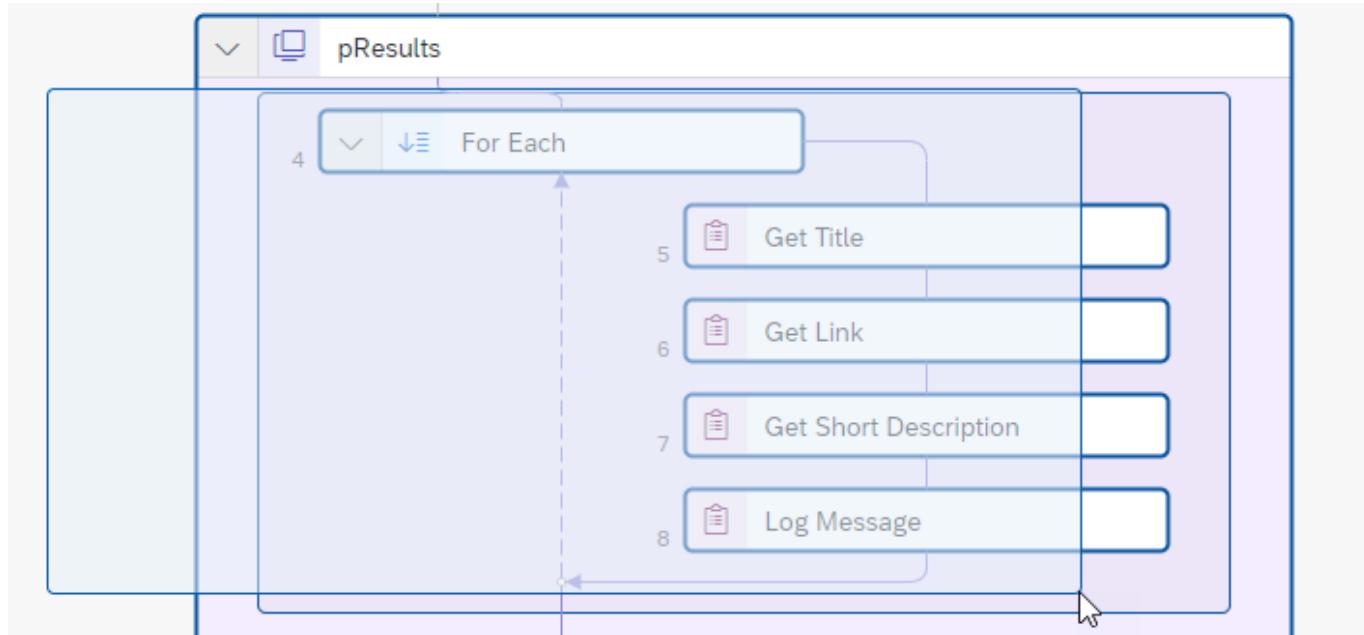
i Note

The step details will not be displayed on the left-hand side panel if you select multiple elements. To view the step details, you must select a single element.

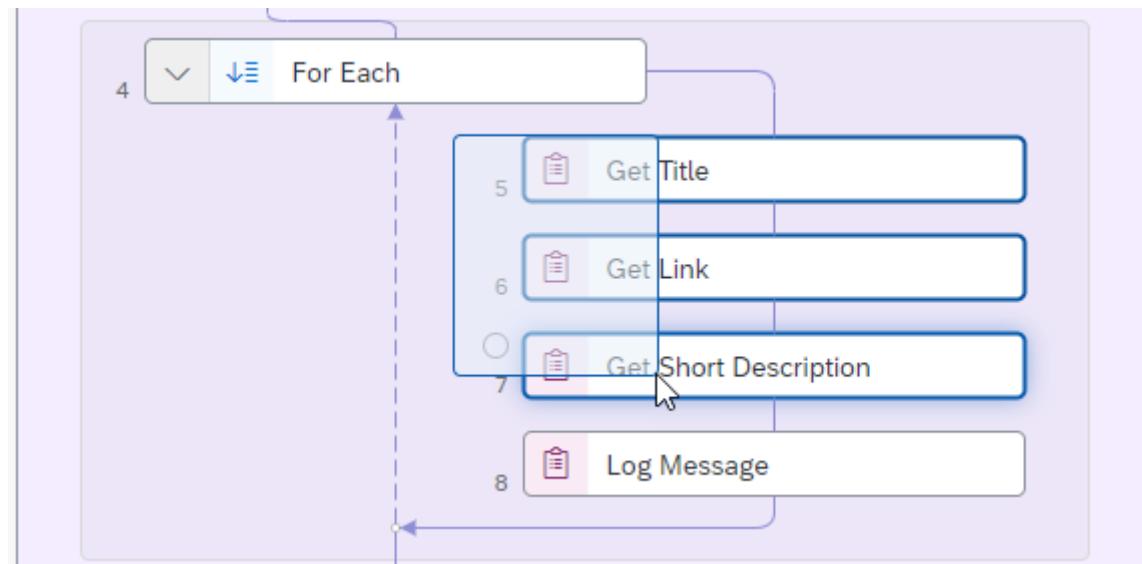
Box Selection

- To select several steps or groups of steps, click an empty area, hold down the **left mouse button** and drag the cursor across the steps.

If you drag the cursor across the control that contains a group of steps, you will select the control but also the whole group of steps.



- To select several steps inside a previously selected group of steps, hold down the **Ctrl** key. Then hold down the **left mouse button** and drag the cursor across the steps.



- To select steps or groups of steps one by one, hold down the **Ctrl** and **Shift** keys simultaneously. Then hold down the **left mouse button** and drag the cursor across the steps. Repeat for each step or group of steps you want to select.
- To drag a screen group, click its header and hold down the **left mouse button**.

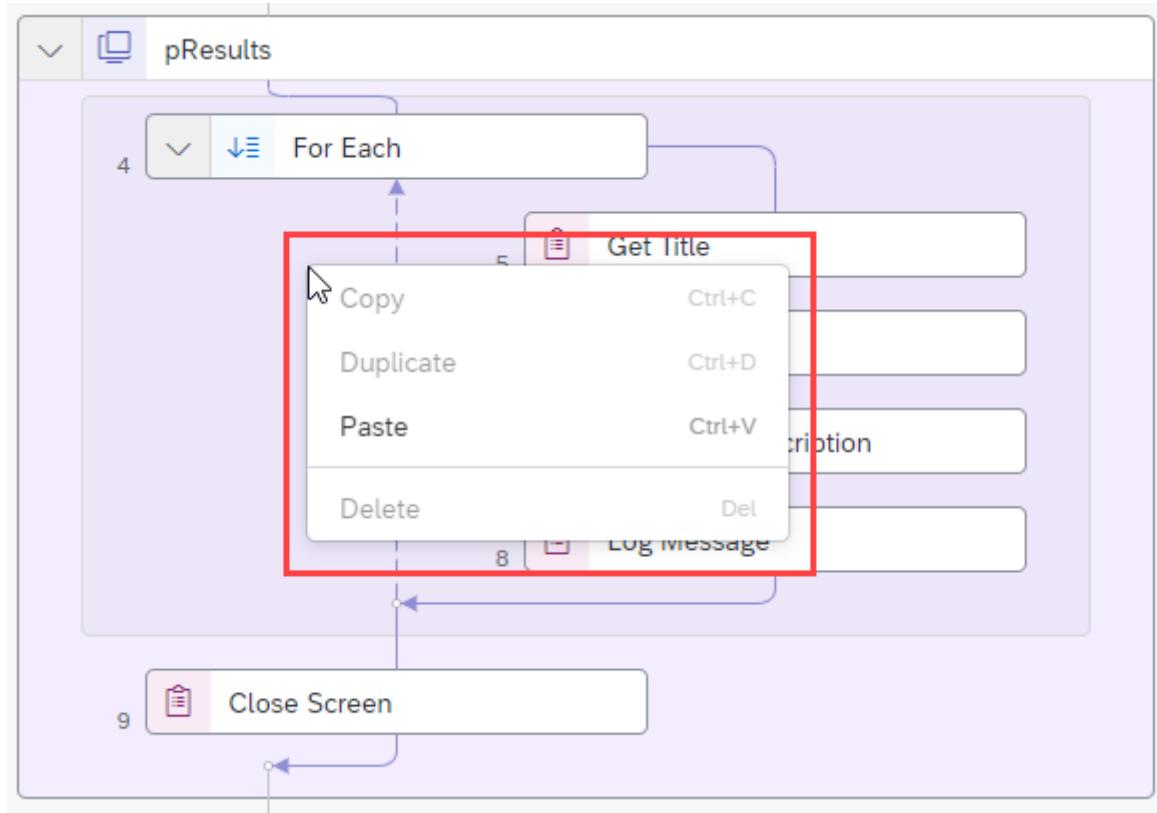
i Note

You cannot duplicate multiple elements that you have selected together. However, you can copy and paste or delete them.

Context Menu Options

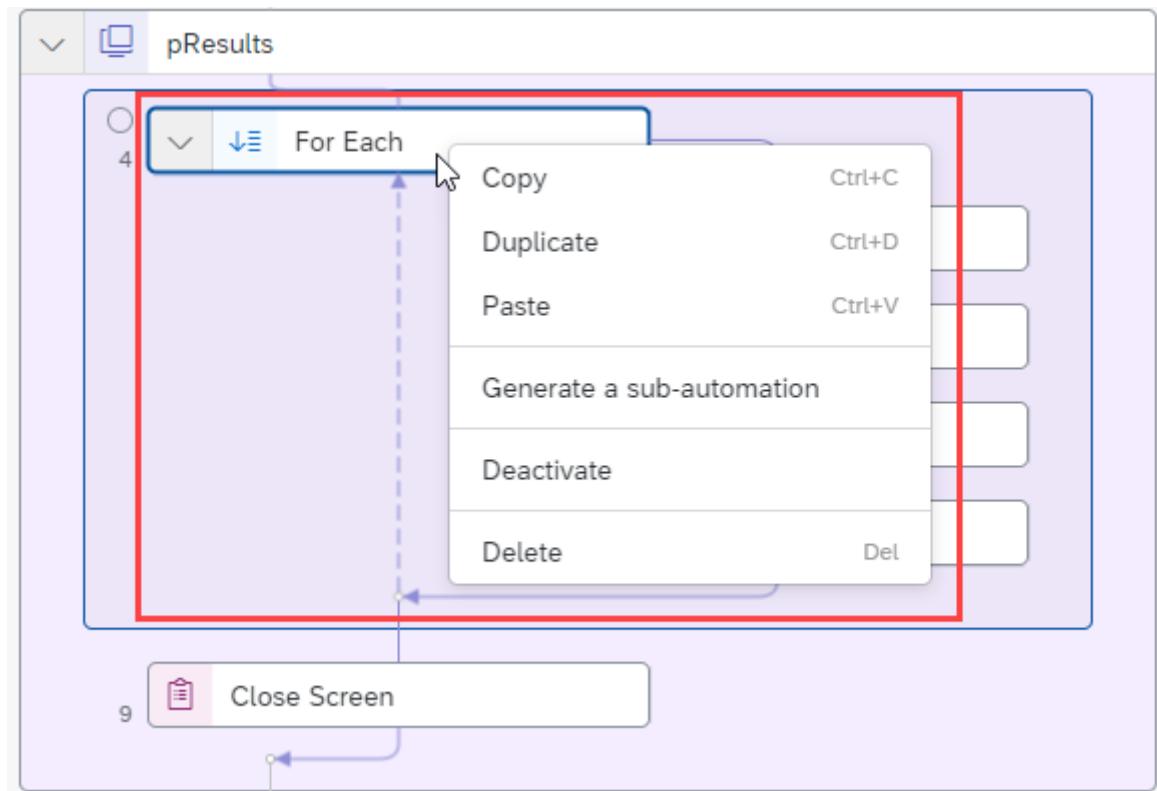
- To open the context menu, right-click the canvas. You can also right-click the background of a group of steps under a control or the background of a screen group.

You will get access to four options: **Copy**, **Duplicate**, **Paste** and **Delete**.



- To get access to the **Generate a sub-automation** and **Deactivate** additional options, right-click the control that contains the group of steps.

To get access to similar options for a screen group contained in your automation, right-click the header of the screen group.



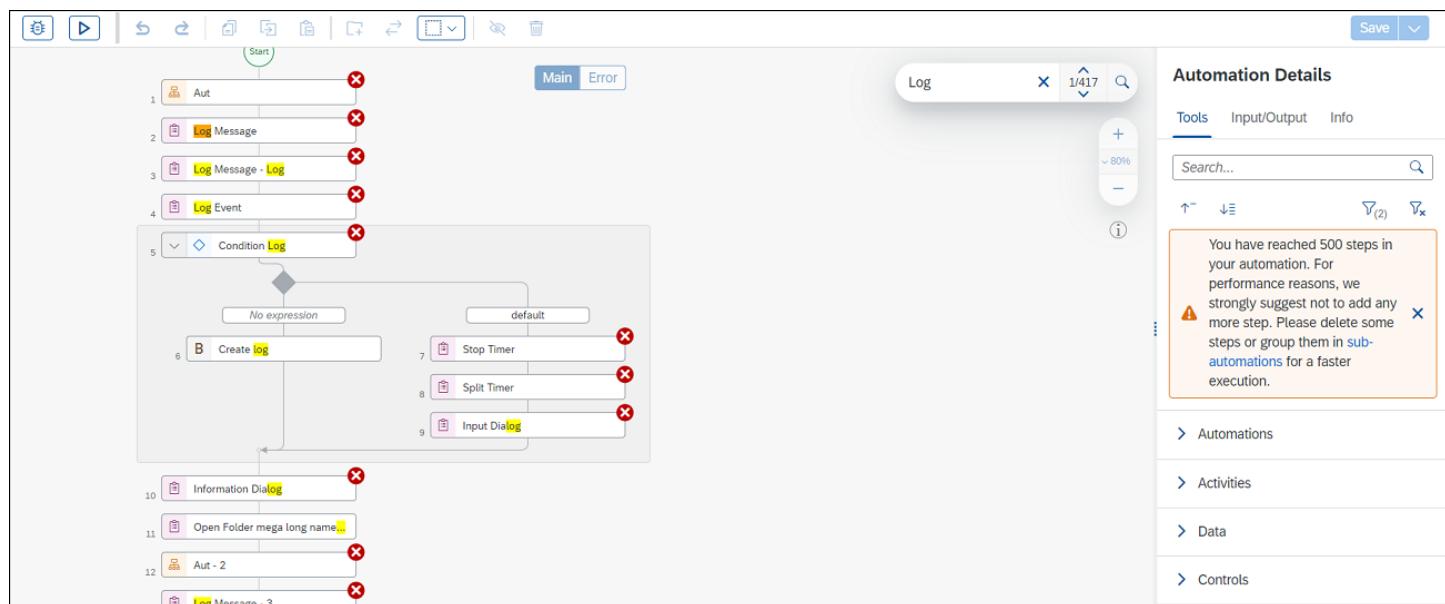
Zoom Factor

- To set the zoom level to 100%, press the **Ctrl** and **0** keys simultaneously.
- To automatically resize the canvas area to fit the window, press the **Shift** and **1** keys simultaneously.
- Press **+** to zoom in and **-** to zoom out.

Search Keywords Inside Step Name

To search for any keyword inside the name of the steps, click  or press the **ctrl** and **F** keys simultaneously within the automation canvas, then type the keyword in the search field and click  or press the **Enter** key.

In the following screenshot, the Log keyword is searched and highlighted inside the name of the steps of the automation. There are 417 matches for the Log keyword inside the automation.



Related Topics

- [Copy, Paste, and Duplicate Items](#)
- [Deactivate an Automation Step](#)

Manage Files

In Cloud Studio, you can create, import, and edit a file.

- [Create a File](#)
- [Integrate Files in Automation](#)
- [Import Files](#)

Business Activity Monitoring (BAM)

Business activity monitoring (BAM) lets you measure the activities performed in your automation by sending data like counters and timers that you insert at chosen points in your automation. In the Cloud Factory, you can download the BAM data in a CSV file.

Use BAM in Your Automation

On the [Automation information](#) side panel, type monitoring in the search bar.

In the [Monitoring](#) section, you can find the different Business Activity Monitoring timer, counter, and notify activities .

Timer activities

The different types of timer activities:

- **Start timer:** the start timer allows you to start a timer.

In the input parameters, you must enter an ID.

- **Split timer:** the split timer allows the Desktop Agent to send a message counting how much time passed between the execution of the start timer and the execution of the split timer.

In the input parameters, you must enter an ID matching the ID of a previously defined start timer.

- **Stop timer:** the stop timer sends the duration from the start timer to the Factory as data and then it resets the timer to zero.

In the input parameters, you must enter an ID matching the ID of a previously defined start timer.

- **Cancel timer:** the cancel timer stops the timer without sending any data to the Factory.

In the input parameter, you must enter an ID matching the ID of a previously defined start timer.

Counter activities

The different types of counter activities:

- **Initialize counter:** the initialize counter starts a counter with an initial value.

In the input parameters of the counter, you must enter:

- an ID
- an initial value for the counter.

i Note

By default, the initial value is set to 0.

- **Increment counter:** the increment counter allows the Desktop Agent to count how many times a specific activity occurs.

In the input parameters of the counter, you must enter:

- an ID matching the ID of a previously defined initialize counter
- an increment value

i Note

By default, the increment value is set to 1.

Notify activity

The notify activity sends a notification message to the Factory. In the inputparameters, you must enter:

- a message
- the severity of the message ("info" , "warning" , "error")

Drag and drop the **Start timer** in the workflow of the automation. In the input parameters of the statement, enter an ID. Add any activity following the timer in the workflow of the automation.

Drag and drop a **Split timer** in the workflow. In the input parameters of the activity, enter the same ID you entered for the start timer.

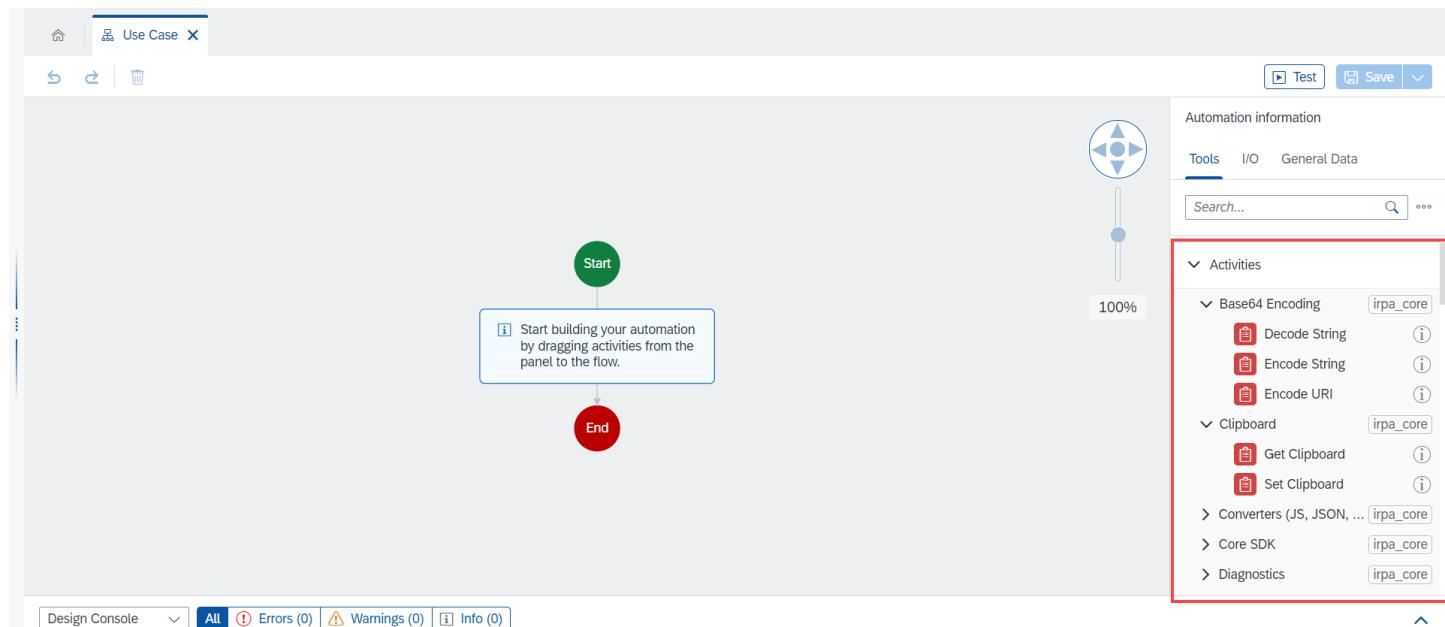
Once you have tested your automation, choose **Monitor** and **Automation Overview** and **Data Export**. Select **Business Activity**.

Choose **Download CSV** to download a file with the measurement data of your automation activities.

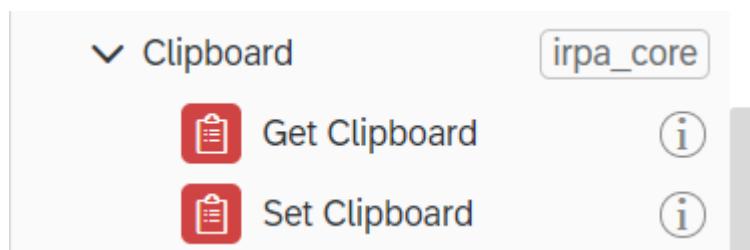
Automation Activities provided by SDK Packages

In the Cloud Studio, activities are provided with the SDK packages you import the first time you create an automation, to build the workflow of your automation.

Activities are organized in classes and subclasses in the tools section on the side panel of the automation.



Each activity executes an action according to the class it belongs to. The following example displays the activities in the **irpa_core** subclass activity **Clipboard**.



The activities in this subclass perform actions linked to the clipboard:

- **Get Clipboard:** Retrieve the textual content of the clipboard.
- **Set Clipboard:** Clear the clipboard and then set text on clipboard.

To build your automation you can drag and drop the activities in your workflow where they become the steps of your automation. Each activity has different parameters that you can define such as the name of the step, and the input and output parameters.

Filtering Feature

To help you look for an activity, the Cloud Studio provides a filtering feature.

Automation information

Tools I/O General Data

Search... ...

The filtering feature displays the tools from the **Tools** side panel that match your search text. The following example displays the filtering feature when you search for a workbook activity.

Automation information

Tools I/O General Data

workbook ...

▼ Activities

▼ Workbook

Open Workbook	(i)
Close Workbook	(i)
Save Workbook	(i)
Save As Workbook	(i)
Add Workbook	(i)
Activate Workbook	(i)

Dependencies

In the Cloud Studio, the characteristics and content of a package can be reused by another project through the structure of a dependency.

How artifacts are integrated through a package dependency

You can import packages as dependencies in a project. The packages contain artifacts such as automations, applications, data types and activities that you can use to build your automation. For more information on how to add a dependency to your project, see [Manage Project Dependencies](#).

When a package is imported as a dependency in a project, part of the artifacts present in the imported package are available to be integrated to your project (linked in black in the interactive image below). These artifacts can be reused like the other artifacts directly created in your project.

i Note

A package needs to have sharing authorizations to be used as a dependency. For more information, see [Share a Package](#).

i Note

When you import packages in the Cloud Studio even if the package dependencies do not exist on the tenant (not imported or acquired from the store). These packages show validation errors.



You must validate the package dependencies in the [Dependencies](#) tab by clicking the **Update** button. After successful update, the **Dependency successfully updated** message is displayed.

Project Properties

Edit Dependency

Alias: * irpa_core

Version: * 1.23.6 Invalid Released

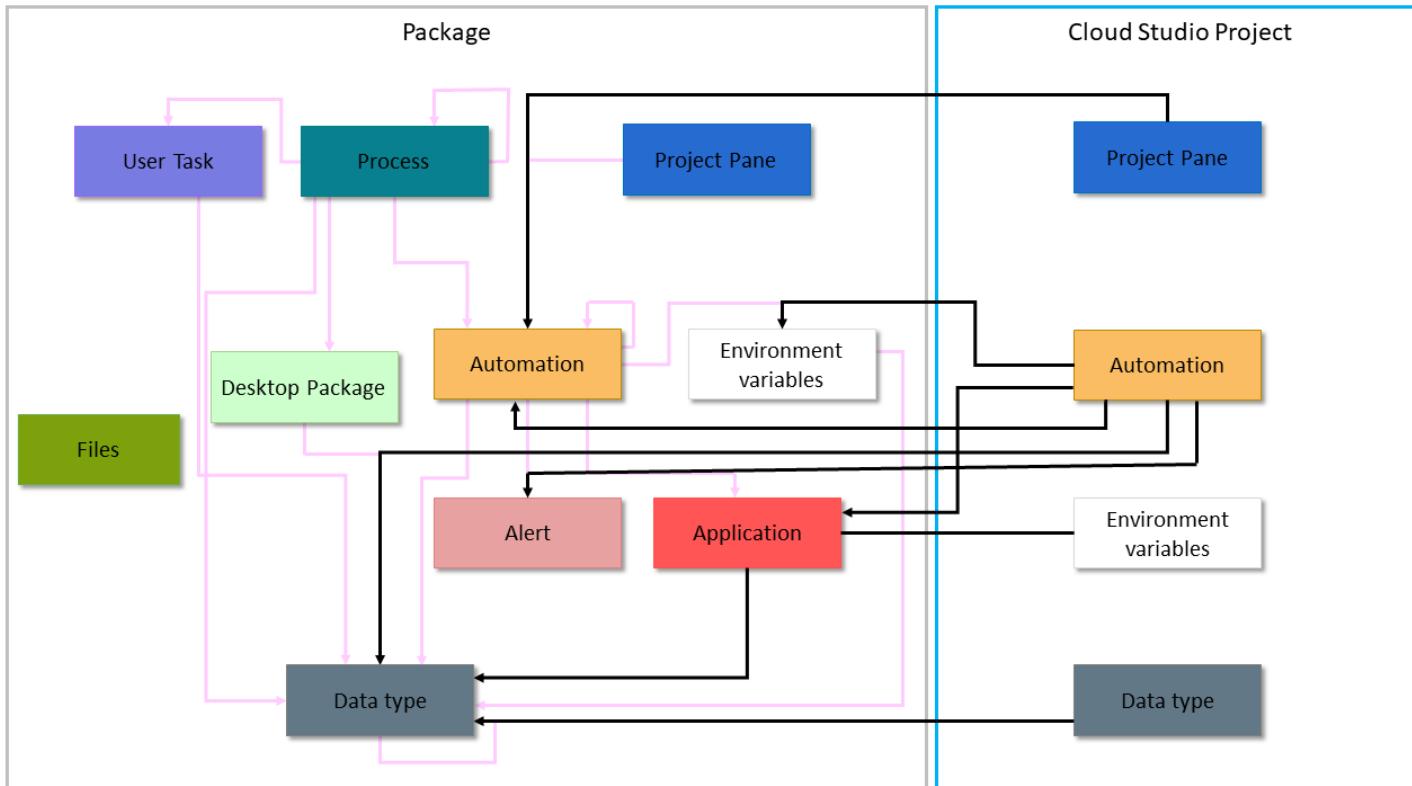
The current dependency version must be updated, the dependent object was added after importing this project. Please confirm by clicking on Update.

Delete Update

Update this dependency

Close

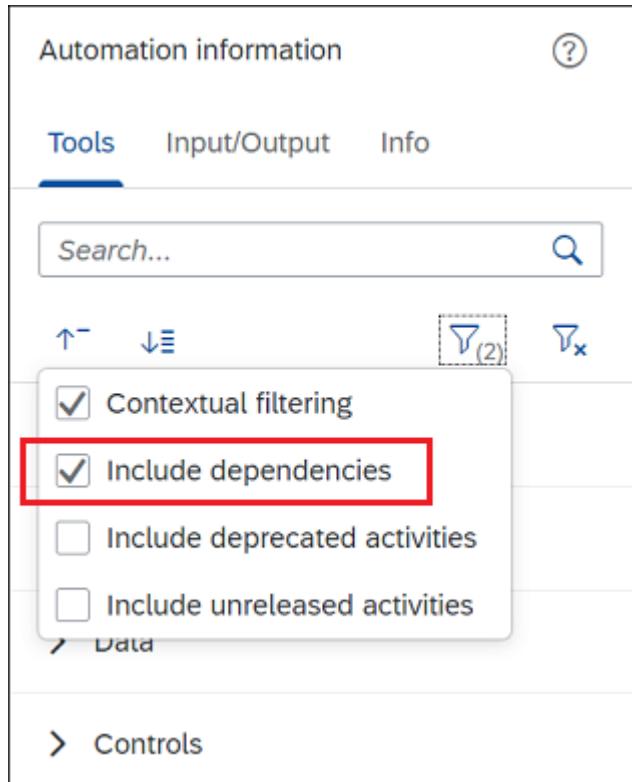
This image is interactive. Hover over each area for a description. Click highlighted areas for more information.



Please note that image maps are not interactive in PDF output.

Further Details

In the **Tools** section of your project, you can filter the content to include or exclude the specific artifacts coming from dependencies.



i Note

You can only use dependencies directly imported in your project. If the package you import as a dependency has dependencies of its own, you cannot use the artifacts present in those dependencies.

Create a Data Type

A data type is an artifact describing a data structure that can be used as an input and/or output parameter in automations or processes.

Context

By default, three types of data are available: string, number, and boolean. For more complex data, you need to use data types.

Data types enable you to better formalize and describe the data used as input/output parameters for steps, activities, skills, processes, scenarios, triggers, or notifiers. Data types facilitate the manipulation and validation of data.

You can create a data type manually but you can also reuse data types that are contained in SDK packages or other packages that are available in the Store.

Procedure

1. In the **Overview** tab of your project, choose **Create > Data Type**.
2. Provide a **Name**, **Identifier**, and optionally a **Description**.

i Note

By default, the identifier is the same as the name. An identifier is mandatory for every artifact in the SAP Build Process Automation, so you must manually enter an identifier if it's not already pre-filled.

3. Choose **Create**.

4. To add new data types, choose **New Field**.

The screenshot shows the SAP Build Process Automation interface. At the top, there's a header with the SAP logo and the text "Build Process Automation". Below the header, there's a toolbar with several icons: a magnifying glass, a refresh arrow, a left arrow, a right arrow, a "New Field" button (which is highlighted with a red box), and a trash can icon. The main area is titled "My Data Type" and contains a table with two columns: "Name" and "Type". There are no rows in the table.

5. In the **Field Properties** on the right, enter the name of the field you just added, and select its type.

i Note

You can insert several types of field in a data type:

- Simple fields of type string, number or boolean.
- More complex fields of type object, or other existing data types (imported from SDK packages or from scenarios).

6. **Optional:** Choose **New Child** to add children to the field.

i Note

Once you add children to a simple field it automatically becomes a field of type object.

7. **Optional:** If you want to manage a field as an array containing multiple values, check the **List** box in the **Field Properties**.

8. Choose **Save**.

Document Processing and Information Extraction

When you automate business processes you can use document processing features to extract data from business documents, such as invoices, purchase orders, payment advice, or other custom documents.

[Document Information Extraction](#) is an SAP BTP service that is used to extract information from business documents. It can be used to process large volumes of business documents that have content in headers and line items, such as tables. You can use these functions directly within an automation in SAP Build Process Automation.

In general, you can extract structured or semistructured information from documents using machine learning models. These models are available when you process a document using a pretrained model. For other cases, you can process data using a template.

You have the following options for processing documents and extracting data:

- [Process Documents Using Online OCR](#)
- [Process Documents Using Pretrained Models](#)
- [Process Documents Using Templates](#)

i Note

The document information extraction activities are available in the [Document Information Extraction SDK](#).

The document information extraction activities that were previously available in the PDF SDK have been **deprecated**. You need to replace these activities with similar activities from the Document Information Extraction SDK.

Create an Environment Variable

Use variables to reuse information across different projects.

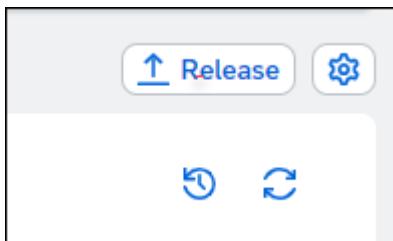
Context

Environment variables are placeholders for the values required when you deploy and run an automation project in an environment. You can reuse environment variables across different projects within an environment. Examples of environment variables are dates, times, destinations, and credentials such as users and passwords.

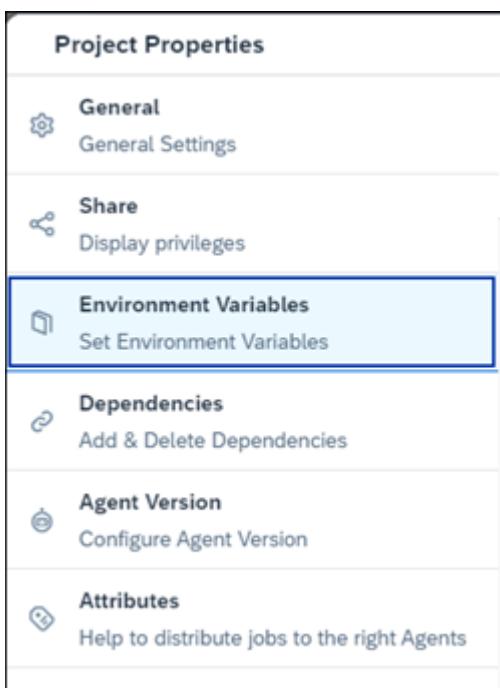
You create multiple variables directly in your project. You define the values of the variables later when you deploy the project in an environment.

Procedure

1. To create a variable, in the [Project Overview](#), choose *Manage the project properties*.



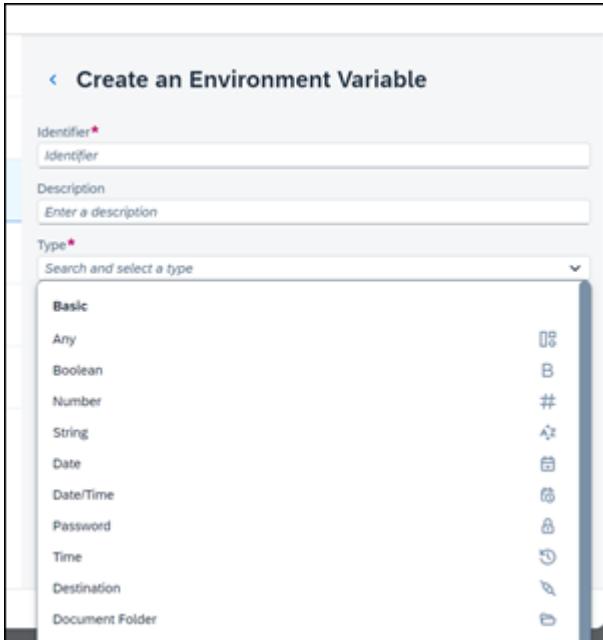
2. On the [Project Properties](#) screen, choose [Environment Variables](#).



3. Choose [Create](#).

4. Enter an [Identifier](#), a [Description](#), and a [Type](#).

- Enter a **Type** or select a variable type from the dropdown list.



i Note

To select a variable you can:

- Choose predefined simple types. For example **String**, **Number**, **Boolean**.
- Choose a **Data Type** that has been defined inside the same project.
- Choose a **Data Type** from **Dependencies**.

The list of variable types that is automatically displayed is truncated. To search for more results, enter keywords in the **Type** field.

5. Choose **Create**.

Next Steps

When you deploy your project, set values to the variables. This way the values of the variables are filled automatically next time you run your project. For more information, see [Set Values for Environment Variables](#).

Best Practices

See some best practices and recommendations for working with automations.

Create a Decision

Create decisions to add them to your process.

Prerequisites

You have created the data types required to configure the input and output of the decision. See [Create a Data Type](#).

Context

A decision consists of a collection of rules, which are used to digitize and automate the business decisions within a business process. After you create a decision, you can define your business logic by adding rules to it. There are two types of rules:

- Decision table: A decision table is a collection of input and output rule expressions in a tabular representation. Multiple rule conditions and results can be configured using a decision table.
- Text Rule: A text rule is the collection of rule expressions in a simple if-then format.

Decision Diagram is a graphical representation of the decision logic from the input to the output. Using a decision diagram, you can view the inputs/output of a decision and the rules of the decision. You can also click on each component of the decision diagram, for example, a text rule or a decision table, to view the configured details on the right panel.

Vocabulary includes input/output parameters and decision variables. This vocabulary can be used to configure the conditions and results when modelling a rule.

Reusable Rules: Reusable rules, as the name implies, are versatile rules which can be incorporated into other rules. These rules exhibit the following features:

- Both text rules and decision tables can serve as reusable rules.
- The result returned by the reusable rule, which can only be executed within another rule, can be consumed in a rule expression.
- When configuring conditions and results in the wizard, reusable rules will be displayed under **Vocabulary**, allowing you to incorporate them into rule expressions.
- A text rule with execute operations cannot be used as reusable rule.

To designate a rule as a reusable rule while modelling a decision, simply mark the rule as **Reusable Rule** in the **Create Rule** screen of the wizard. The rule will then be listed under **Reusable Rules** in the **Rules** tab.

Procedure

1. In the **Overview** tab of your business process project, choose  **Create** **> Decision**.
2. Provide a **Name**, **Identifier**, and optionally a **Description**.
3. Choose **Create**.
4. To configure the inputs and output of the decision, do the following:
 - a. Choose the decision from the **Decision Diagram** and then choose the **Inputs/Output** tab to configure the inputs and the output
 - b. Under **Input Parameters**, choose **Add Input Parameter** and configure the inputs by providing a **Name**, **Description** and choosing the **Type** of the input parameters.

You can either select any of the **Current Project** data types that you created earlier or one of the following **Basic** data types:

- Boolean
- Date
- Number
- String
- Date/Time

Select **List** if the input parameter is a collection of a specific data type.

i Note

- Once you select **List**, this list input parameter will be displayed under **Vocabulary** when you configure your conditions and results in the wizard.
- However, you will not be able to use these list parameters to configure conditions in condition headers and these will appear greyed out in the **Configure Conditions** screen. See [Model a Rule in a Decision Table](#).
- Once a rule is added, you can use these list parameters in the cells of the decision table while modelling rule expressions.

❖ Example

- If you have the following input in a tabular form with the IDs of employees and their cities as list input parameters, you can use 'BLR' EXISTSIN (Employees, CityCode) to determine whether it contains the input 'BLR'. When you configure your inputs, you can use 'Employees' and select the **List** checkbox.

Employees		
Employee_ID	CityCode	Age
L321	BLR	30
L322	MUM	40

- If 'Age' is a list input parameter, you can use $\text{AVG}(\text{Employees}, \text{Age})$ to get the average of the age of all the employees.
- c. Similarly, under **Output Parameter**, choose **Add Output Parameter** and configure the output for the decision by providing a **Name**, **Description** and choosing the **Type** of the output parameter.

Type can be either any of the **Current Project** data types that you created earlier or one of the **Basic** data types.

→ Tip

Select **List** if you want to model the decision table with **All Match** hit policy. This allows you to fetch all instances that meet the specified conditions and return them as an array of multiple values in the result.

i Note

- If you want to change this decision table with **First Match** hit policy later, deselect **List**. Also, edit the rule to confirm that the hit policy is displayed as **First Match**. The rule engine then fetches the first occurrence that matches the condition and returns it as result.
- When you select a basic data type for your output parameter in the **Type** dropdown, you will be unable to select the **List** checkbox.

Add Rules to a Decision

Add decision tables or text rules to a decision based on your business scenario.

Prerequisites

You're familiar with the operators and functions used to model rule expressions. See

- [Operators](#)
- [Functions](#)

Procedure

1. Choose the **Rules** tab and then choose **Add Rule**.
2. To add a decision table, see [Model a Rule in a Decision Table](#).
3. To add a text rule, see [Model a Text Rule](#).
4. To create a duplicate of a rule, see [Duplicate a Rule](#)

i Note

Once you add multiple rules, they will be executed in the sequence listed in the **Rules** tab under **Execution Order**. You can modify this order by moving the rules up or down.

Also, you can perform the following actions in the **Rules** tab:

- Delete any rule by selecting the rule and choosing **Delete** (⊖)
- Drag and drop a rule to change the order.
- Move a rule up or down to change the order by choosing **Move Up** (↑) and **Move Down** (↓)
- Use the **More** (⋮) option beside each rule, to do the following:
 - Add a rule
 - Duplicate a rule. See [Duplicate a Rule](#).
 - Delete a rule
 - Move a rule up
 - Move a rule down

Add Variables to a Decision(Optional)

Add variables to a decision to store information of one rule execution and reuse it in another rule.

Context

In the **Variables** tab, you can add intermediate variables to your decision, that can be used to temporarily store the output of one rule execution and reused as an input for another rule execution.

i Note

The scope of these variables are only within the particular decision to which these are being added and are not available elsewhere in the process.

Procedure

1. Choose the **Variables** tab and then choose **Add Variable**. The **Decision Variable** screen opens up.

2. Provide a **Name**, **Description** and choose a data type for the variable from the **Type** dropdown.

In the **Type** dropdown, you can either select any of the **Current Project** data types that you created earlier or one of the following **Basic** data types:

- Boolean
- Date
- Number
- String
- Date/Time

3. Choose **Save**. The variable gets added successfully and you can see it in the **Variables** tab.

You can also edit, duplicate or delete a variable by using the **More (□)** option available beside each variable.

4. You can use these decision variables when you configure your conditions or results while modelling a rule. These variables will be displayed in the rule wizard as part of the vocabulary for a rule condition as well as a result.

Deploy a Project

Deploy a project so that it can be run by end users.

Prerequisites

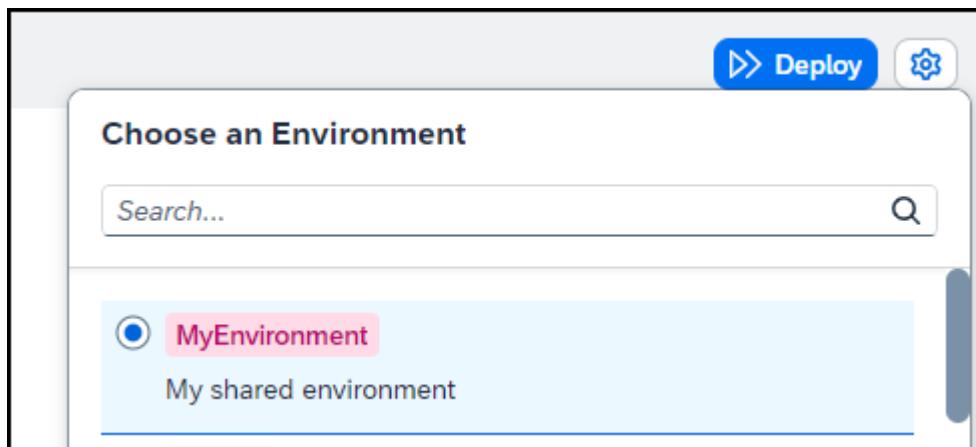
- If you want to deploy your project to a shared environment, you need to have access to this environment. Access is granted by administrators. For more information, see [Create an Environment](#) and [Share an Environment](#).
- Your project has been released.

Context

You deploy a released project so that it can be run by end users.

Procedure

1. In the **Project Overview**, select a released version of the project.
2. Choose □ **Deploy**.
3. In the **Choose an Environment** screen, select the environment in which you want to deploy the project.



4. If you have created variables for your environment, set the values for these in the **Define Variables** screen.

For more information, see [Set Values for Environment Variables](#)

5. Choose **Deploy**.

Results

The version is now deployed. The same project can now be deployed in a different shared environment, regardless of its version.

If you make changes, you can upgrade to a higher product version or redeploy the project.

Related Information

[Upgrade](#)

[Redeploy a Project](#)

[Undeploy a Project](#)

Manage Project Members

The project creator can add or remove members from the Lobby, and also grant authorizations to members and user groups.

Prerequisites

- You've created a project. For more information, see [Create and Manage Projects](#).
- You've configured the `Identity_Authentication_Connectivity_IDS` destination to connect to the identity directory of Identity Authentication and retrieve the required information about the users in the search field. For more information, see [Connect to the Identity Directory](#).
- You have created groups in your Identity Authentication service. For more details on how to create groups, see [Create a New Group](#).

Context

Manage the members of your project to allow the members of your team to collaborate. Each member of your team can be assigned an authorization based on the level of access that member requires. A project can have a variety of members, including business process managers, developers, and IT admins.

Lobby projects support the following types of authorizations:

Authorization	Description
Viewer	Authorization for read-only actions, such as viewing, but not modifying the existing content of the project. Viewers can also export and deploy the project.
Developer	Includes all viewer permissions, plus permission to modify the project, such as modifying existing configurations, and releasing and publishing the project.
Administrator	Authorization to carry out various important administrative tasks. Includes all viewer and developer permissions. For example, administrators can assign authorizations and add or remove members. The project creator has administrator authorization by default.

Projects can be controlled using the following general access permissions:

- **Everyone with access:** Allows your entire team to collaborate and contribute to one project at the same time.
- **Only project members:** Members whose email addresses you've entered are permitted access to the project.

You can manage members directly from the Lobby page. Your project's member's column provides the option to modify the member list when you hover your mouse over the members in the column.

Procedure

1. Navigate to the [Lobby](#) and select a project.

2. Click  *More Options*.

3. Choose [Manage Members](#).

An overview of all the members currently assigned to the project is shown in a list along with their authorizations.

4. In the [Type](#) field, choose one of the following options.

- **Member:** You can share your project with users by entering their email addresses.
- **User Group:** You can share a project with a group of users by entering the name of the group.

5. Enter a valid email address or group name in the [Share with](#) field.

→ Tip

You can choose the email address and user group from the available list using the  *Value help* option.

6. In the [Authorization](#) field, select an authorization level.

7. Choose [Add](#).

8. **Optional:** Use one of the following options if you want to use general access, and then select the appropriate authorization.

- [Everyone with access](#)
- [Only project members](#)

9. Save your changes.

i Note

- As a viewer, if you would like to perform activities on projects such as editing, sharing, and deleting. You can get this access by hovering over the **Everyone** text in the members' column and checking the project's creator since the project's creator is by default an administrator and can grant authorizations. If you need access, you can contact them.
- All members of a tenant will automatically have viewer authorization on the projects as soon as the projects are published to the library.

Project Options

Learn how to publish, export, rename, copy, and delete a project.

Publishing a Project

Once you've created and released a project, you can publish it to the library, so that it can be downloaded and reused by other users.

i Note

You need **Developer** or **Administrator** authorizations to publish a project to the library. For more information, see [Manage Project Members](#).

Procedure

1. Navigate to the **Lobby** and select a project.
2. Click **More Options**
3. Select **Publish to Library**.
4. In the **Publish Content to Library** dialogue, choose a **Version** in the drop-down list.
5. Select **Line of Business** from the drop-down list.
6. Select **Product** from the drop-down list.
7. Click **Publish**.

Exporting a Project

Export a project to your local system. Then you can reuse the project in another tenant by importing it.

The project is exported as an MTAR file. It contains the artifacts and dependencies of the project.

i Note

You can only export Business Process projects. You can't export projects that have been imported from the Store.

Procedure

1. Navigate to the **Lobby** and select a project.
2. Click **More Options**
3. Choose **Export**.
4. In the **Export Project** dialog, select the version you want to export from the drop-down list.
5. Click **Export**.

Renaming a Project

Give your projects a new name or change their description.

i Note

You need **Developer** or **Administrator** authorizations to rename a project. For more information, see [Manage Project Members](#).

Procedure

1. Navigate to the **Lobby** and select a project.
2. Click **More Options**
3. Choose **Rename**.

4. Change the name of the project in the **Name** field.
5. **Optional:** change the description of the project in the **Description** field.
6. Click **Save**.

Deleting a Project

If you no longer need a project, you can delete it from the **Lobby**.

i Note

You need **Developer** or **Administrator** authorizations to delete a project. For more information, see [Manage Project Members](#).

i Note

You can't delete a deployed project, you must undeploy it first. For more information, see [Undeploy a Project](#).

Procedure

1. Navigate to the **Lobby** and select a project.
2. Click  **Delete**
3. Click **OK**.

Save as New Project

You can save the project as a new project. The newly saved project is a copy of the original.

i Note

It's only available for business process projects.

Procedure

1. Navigate to the **Lobby** and select a project.
2. Click  **More options** and select **Save as New Project**.
3. (Optional) Update **Project Name** and **Description**.
4. Click **Save as New**.

Desktop Agent User Guide

Use the Store

The **Store** is available if you've subscribed to the service and are licensed to it.

Disclaimer:

SAP does not warrant the correctness and completeness of the store content. SAP shall not be liable for errors or damages caused by the use of content unless damages have been caused by SAP's gross negligence or willful misconduct. SAP is free to retire store content at any time from the Store. All content is versioned and only the latest version is available for download.

About the Store

The **Store** offers prebuilt content created and curated by SAP and authorized partners. The content available in the **Store** can be both paid and free. The **Store** can be accessed directly from the home page. The **Store** also offers a variety of filters, a search feature, and content descriptions to help you quickly discover the content based on your needs.

In the **Store**, you can choose from a wide range of pre-built content packages for different industries and business lines. It provides a catalog of ready-to-use content and templates that are directly accessible, configurable, and extendable. A content package contains a description, detailed artifact information, and a configuration guide that explains how to set it up. The content provided in the **Store** benefits you because it allows you to quickly use ready content that is tailored to your specific industry and needs. It also reduces the amount of time and effort required to develop content from scratch.

The **Store** contains different types of projects. The contents of the projects are easily customizable, making them an excellent resource for all types of business. Only the ready-to-use content is updated regularly based on the latest product changes and developments to ensure you've access to the updated content.

The Different Types of Projects and Their Descriptions

Project Type	Description
Business Process	Content created using SAP Build Process Automation. It can contain all artifacts and features associated with business process projects.
Actions	Use the Actions project to encapsulate APIs as actions in your business scenarios.
Task Automation	Content created using SAP Intelligent Robotic Process Automation. It can be used with the SAP Build Process Automation application and contains only automation artifacts.
Live Process	Process content created using SAP Workflow Management service. It can be used with SAP Build Process Automation, but it requires some additional configuration.

Tags for Different Project Format Types

To make it easier to recognize and arrange all of the projects, the following **Format Type** tags are used.

Tag Name	Description
Ready to use	These projects are ready to be used, having already been designed and are available for use in your business processes.
Template	A project template is a project that has prebuilt business process artifacts and actions that are commonly used in your industry-specific business processes. You can use the template to create new projects and it can then be customized and extended as needed.

Store Projects: More Information

The projects available in the **Store** consist of information listed in the following table:

Sections	Details

Sections	Details
Header	<ul style="list-style-type: none"> • Name of the project • Type and sub type of the project • Publisher of the project • Version of the project • Catalog category of the project • Last modified details of the project
Body	<ul style="list-style-type: none"> • Description that explains the purpose and prerequisites of the project. It can include the key process steps that are covered. • Media section to view additional material such as images • Lines of business and industries corresponding to the project • Associated artifacts • Associated documentation • What's New information • Version details • Dependent project details <p>i Note</p> <p>The project versioning is a combination of:</p> <ul style="list-style-type: none"> • A first number corresponding to the number of the major version • A second number corresponding to the number of the minor version • A third number corresponding to the patch version <p>⌚ Example</p> <p>Version: 1.19.33</p> <p>Here are a few types of artifacts that are included in the project:</p> <ul style="list-style-type: none"> • Decision • Process • Visibility Scenarios • Automation • Form • Data Type • Files (such as documentation and test scripts) • Document Template • Dependencies, and so on

SAP Build Content Catalog

Explore the publicly available content catalog that consists of various projects under the SAP Build umbrella. With the catalog, you can learn more about line of business and industry-specific content that is designed specifically to enhance SAP applications such as SAP ECC, SAP S/4HANA, SAP Business by Design, SAP Ariba, SAP SuccessFactors. For more information, see [SAP Build Content Catalog](#).

Related Information

[Explore the Store](#)

Monitor Processes

The **Monitoring** page collects events coming from the deployed workflows and automation jobs and presents them to the user. These events represent the state of the agents, the status of the jobs that have been run, and the business activity of monitoring events.

Prerequisites

You've been assigned to the ProcessAutomationAdmin role collection. For more information, see [Authorizations](#).

Context

SAP Build offers several interlinked views for Process and Workflow Instances, Automation Jobs, Automation Overview, Visibility Scenarios, Business Rules, Triggers, and Acquired Events. All can be accessed using dedicated tiles in the SAP Build Work Zone.

The screenshot shows the SAP Build Work Zone interface with the 'Monitoring' section selected in the sidebar. The main area displays four monitoring tiles: 'Process and Workflow Instances' (0 Failed today), 'Automation Jobs' (0 Errors today), 'Acquired Events' (0 Errors today), and 'Automation Overview'. Below these are two sections: 'Automation Concurrent Quota Usage Today' and 'Manage'. The 'Manage' section includes tiles for 'Processes and Workflows' (0 Deployed), 'Triggers' (0 Deployed), 'Visibility Scenarios' (0 Deployed), and 'Business Rules' (represented by a gear icon).

Monitoring Process and Workflow Instances

In the [Process and Workflow Instances](#) view, you can see a list of all workflow instances and act on those instances.

For more information, see [Process and Workflow Instances](#).

Monitoring Automation Jobs

In the [Automation Jobs](#) view, you can view all your automation errors, warnings, variables, and other information on the console.

For more information, see [Monitor an Automation Using the Console](#).

Monitoring Acquired Events

In the [Acquired Events](#) view, you can view the events that have been acquired by the process visibility capability.

For more information, see [Acquired Events](#).

Monitoring Automation Overview

The [Automation Overview](#) page displays the status of your jobs. You can check whether any job has failed today, the last 7 days, or last month.



For more information, see [Automation Overview](#).

Monitoring Visibility Scenarios

In the [Visibility Scenarios](#) view, you can manage your visibility scenarios by scheduling the job for processing the data and clearing the processed data. You can also view the processing log information and the details of a scenario definition in this view.

For more information, see [Visibility Scenarios](#).

Managing Business Rules

Using the [Business Rules](#) tile, you can launch the [Manage Rule Projects](#) (MRP) application and perform advanced configurations on business rules projects that are imported as part of live process packages. Additionally, you can model complex scenarios and consume them using public REST APIs.

For more information, see [Business Rules](#).

Manage Control Tower

You use the Control Tower page to check the details and several settings required for your tenants.

You can find Control Tower as a separate page in SAP Build Process Automation. It includes various details, statuses, and error messages. You can maintain the several settings that are required to run your projects.

Overview of Control Tower

The **Control Tower** page lets you view the registered agents along with their status, versions, and modes. For more information, see [Overview of Agents](#).

Use My Inbox

Process your tasks from the My Inbox application. You can use My Inbox on your desktop or mobile device.

Depending on your role, you can access My Inbox as follows:

- For citizen developers – from the My Inbox icon (✉) in the **Lobby**. For more information, see [My Inbox for Citizen Developers](#).
- For process participants – from the **My Inbox** tile in your SAP Build Work Zone home page. For more information, see [My Inbox for Process Participants](#).