Automation Guides

Version: 2023.1.0 FP1

Copyright AppViewX, Inc.

External Reference Links

This product includes software developed by the CentOS Project (www.centos.org).

This product includes software developed by Red Hat, Inc. (www.redhat.com).

This product includes software developed by VMware, Inc. (www.vmware.com).

Contact Information

AppViewX, Inc.

222 Broadway, FL 19

New York, NY 10038

Email: info@appviewx.com

Web: www.appviewx.com

Contents

Preface

Revision History

Revision	Description	Date
1.2	Updated release of document for release 2023.1.0 FP2	February 2024
1.1	Updated release of document for release 2023.1.0 FP1	November 2023
1.0	Initial release of document for Release 2023.1.0	September 2023

About the Automation Guides

This section includes the following guides that give you an overview of the functionalities offered in the Automation module:

- Automation User Guide
- Automation API Guide
- Automation Troubleshooting Guide
- Service Catalog User Guide

Audience

This guide is useful for experienced users with reasonable programming experience (Python), trying to enhance their skills with automating network tasks. You can use AppViewX pre-built workflow tasks, commands, API, SDK to automate across the L2-L7 layer, such as automate load balancers, DNS, WAF, PKI, switches, routers, and other network elements.

Text Conventions

The following text conventions are used in this document:

Convention	Description
boldface	Boldface type indicates graphical user interface elements associated with an action, or terms defined in the text or the glossary.
italic	Italic type indicates book titles, emphasis, or placeholder variables for which you supply particular values.

Convention	Description
`codeblock`	Indicates commands with a paragraph, URLs, codes in examples, text that appears on the screen, or text that you enter.
	Indicates a feature available only in the on-premise deployments of AppViewX.
	Indicates a feature available only in the SaaS deployments of AppViewX.

Chapter 1: Automation User Guide

Welcome to the complete guide to getting started with Automation. Learn how to use workflows for Network Service Orchestration and Automation with AppViewX modules, including working with inventory, connecting tasks and apps, and designing workflows to automate repetitive tasks and solutions.

Module Overview

AppViewX's Automation module is a hybrid cloud automation and orchestration platform that allows for simple and complex network automation within your infrastructure.

The module supports a wide range of device types, vendors, actions and integrations, in order to help you manage and automate your entire network with a unified automation and orchestration platform. Its primary interface is through both the Web GUI interface and a HTTP Restful API. A developer would use this API for programmatic access.

AppViewX's Automation module offers:

- Hybrid/Multi-cloud network orchestration and automation
- Single service platform with self-servicing Automation and API orchestration
- Workflow Orchestration of system of records (SOR) across diverse application and infrastructure domains
- DevOps, GITOps automation.

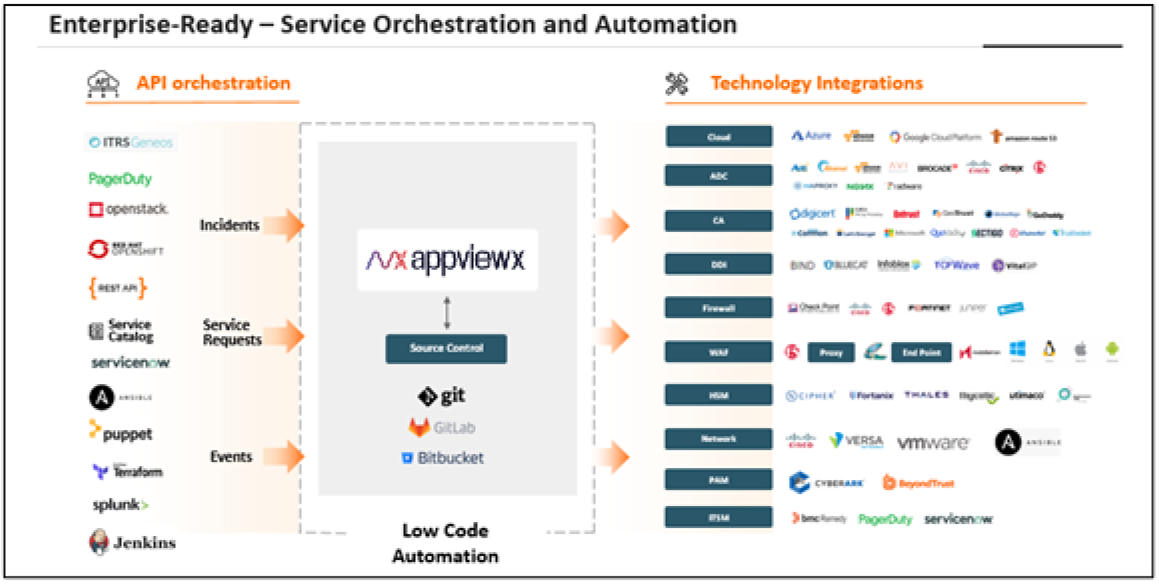
Solutions

- Automate Application delivery and security (LB, DNS, WAF) services
- Automate Certificate Lifecycle Management (CLM)

- Automate Network Firewall changes
- Automate Cloud services
- Routers/switch automation
- Continuous compliance Automation
- Automate Vulnerability Detection (CVE) and Reporting
- Event driven automation, Closed loop remediation.

Salient Features

- **Automate** network configurations and change automation
- **Automate** repetitive tasks for faster network changes and free up your time for more strategic initiatives
- **Automate** repeatable network patterns across operations and development using the same powerful and agentless automation platform
- Separate the workflow task execution from the execution layer (via AppViewX vendor modules)
- **Prebuilt tasks and solutions**: Automate application and security services quickly using out of the box automation tasks and solutions
- **Design** simple and complex workflows
- **Improve** agility, reduce operational costs, and automate repetitive tasks
- **Programmable GUI**: Bring your own automation within a unified GUI-based framework
- **BPM (Business Process Automation)**: Streamline and map the business process, teams, and technology
- **Workflow tasks**: Objectivity of business logic is maintained through separate tasks
- **Reusability**: Offers reusability of workflow(s) and workflow task(s) for quick turnaround
- **Intuitive**: Offers control directly to the user to define a customized automation workflow
- **Rule Engine**: Custom Rule builder to aid event driven automation
- **Visual Representation**: Offers a visual representation of the entire automation and orchestration flow
- **Integration**: Integrate and automate with Multi-vendors (ITSM, ADC, PKI, DDI, DevOps, SDK, Open source tools)



- Accessing the Automation Module

Accessing the Automation Module

1. Hover your mouse over the ⠿ icon.
2. From the menu displayed, select **Automation**.

   The Automation module is displayed with the **Request :: Overview** section open by default. There are three sections here:

   - **Service Requests**: The Service Requests section allows you to execute workflow(s). Workflows can be executed manually or can be scheduled.
   - **Workflow Studio**: The Workflow Studio allows you to design simple and complex workflow(s) from scratch and also use the out of the box (OOB) workflows.
   - **Service Catalog**: The Service Catalog allows you to design and customize self-service catalogs by persona.

Prerequisites

Web Browser Requirement

Browsers	Version
Internet Explorer	v11.0.9600.18817 or later
Firefox	v74.0.1 (64-bit) or later

Browsers	Version
Google Chrome	v85.0.4183.83 (64-bit) or later

Getting Started with Automation

Automation Concepts

- **Workflow Studio**: An inventory to design and manage automation workflows and design self-service catalogs for NetOps, SecOps, App teams and PKI admin.

  For more information, refer to the section on Workflow Studio.

- **Tasks**: Tasks are the units of action (aka automation lego blocks) in workflows. Tasks can be of various types depending on the nature of automation such as User Interface tasks, Scripts, YAML, REST API, or pre-built network automation tasks.

  For more information, refer to the section on Workflow Tasks.

- **Variables**: Workflow variables are properties that help define or describe the workflow activity required to perform per task. Variables provide the ability to store data (within a task) and can be referenced across any stage of the workflow automation process, thus enabling the flow of data seamlessly.

  For more information, refer to the section on Variable Mapping.

- **Nested Workflows**: An ordered list of tasks and workflows bundled together to enable their execution in an order repeatedly. Nested flows can include variables, tasks, and even a complete workflow.

  For more information, refer to the section on Nested Workflows.

- **Pre-built Task Library**: A folder consisting of pre-built automation tasks and solutions with device, vendor information, tags, and standard Input/Output variables defined.

  For more information, refer to the section on Getting Started with Prebuilt Workflow Tasks.

- **Role Based Access Control (RBAC)**: RBAC ensures access control permissions across tasks, workflows, permissions to one or more users, and user groups.

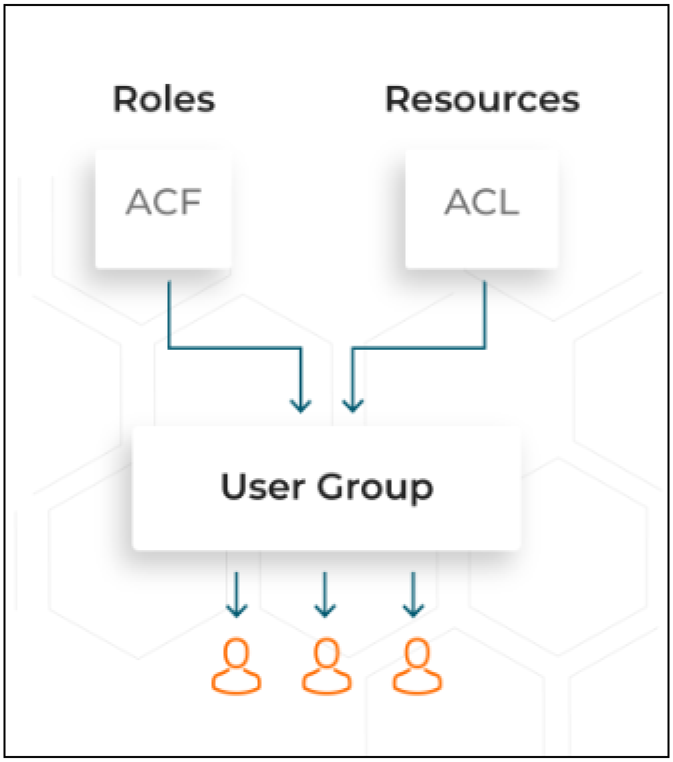
  For more information, refer to the section on RBAC.

- **Service Requests**: The Service Requests section allows you to execute workflow(s). Workflows can be executed manually or can be scheduled.

  For more information, refer to the section on Service Requests.

Configuring Role Based Access Control

AppViewX offers comprehensive support for Role and Resource-Based Access Control (RBAC). RBAC is a method of restricting AppViewX functions, network resources that can be managed and monitored in AppViewX based on the roles of individual users within an enterprise. It allows you to integrate with the existing identity stores such as Active Directory (AD) and Lightweight Directory Access Protocol (LDAP) to enforce authorization policies. Roles and Resources can be customized to suit any organizational structure and user requirements.



RBAC lets employees have access rights only to the AppViewX functions and network resources they need to do their jobs and prevents them from accessing information that doesn't pertain to them.

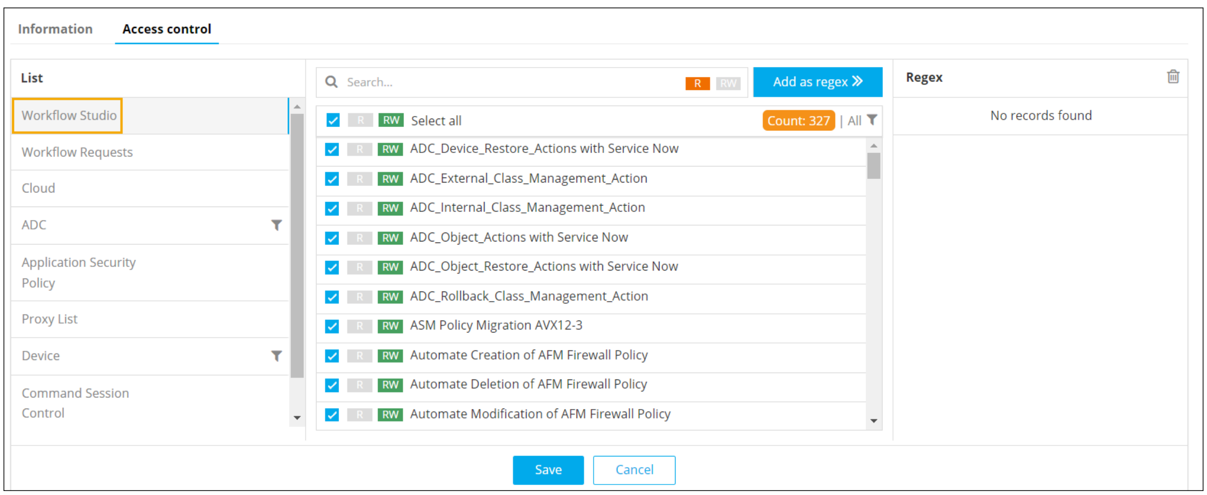
- Resource
- Role
- User Group
- User

Resource

To get an overview of the actions authorized for a particular resource:

New Menu	Old Menu
In the **Platform** module, from the left pane, under **IDENTITY**, select **Resource**. The **Resource** page is displayed, with all the roles listed with their **Name**, **Description**, and **Status**.	From the main menu, select **Account** > **Resource**. The **Resource** page is displayed, with all the roles listed with their **Name**, **Description**, and **Status**.
Note: For more information on how to switch between menus, click here.	

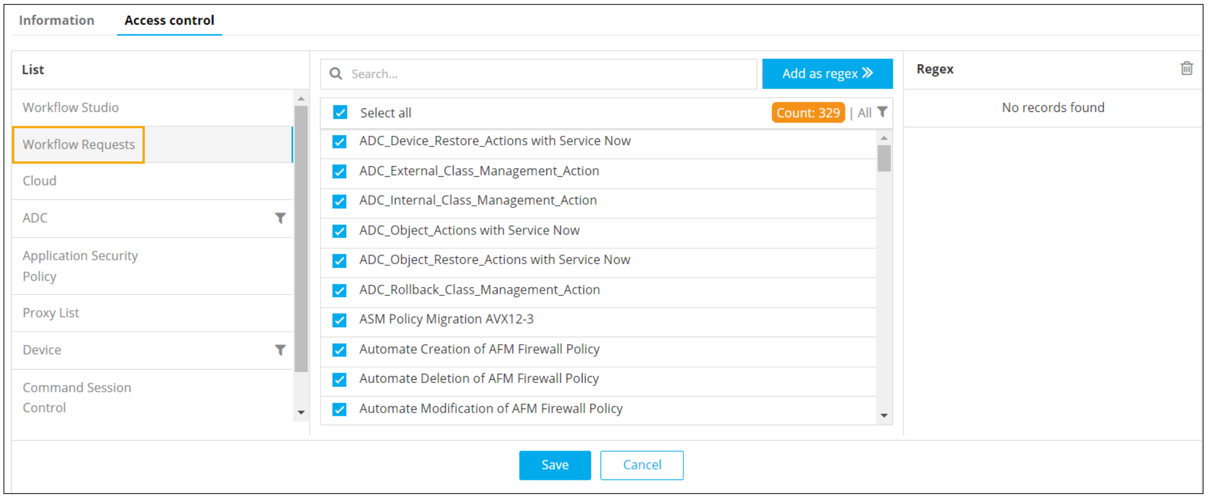
1. On the **Resource** page, click on any resource.
2. To see the permissions that can be defined for the resource in the Workflow Studio, under the **Access Control** tab, from the **List** menu, select **Workflow Studio**.



The following permission(s) can be defined for the resource:

- Read (**R**): To allow only Read access to the resource, click **R** against the Workflow name. The resource will not be able to modify the workflow.
- Read/Write (**RW**): To allow a resource to view and modify a workflow, click **RW** against the workflow name.
- Select/Deselect: If you deselect a particular workflow, that workflow will not be available on the Workflow page for this resource.
- When trying to add the regex "*", a validation error will be triggered, resulting in the disabling of the "Add as regex" option.

3. To see the permissions that can be defined for the resource in the Service Requests console, select **Workflow Requests**.



- When trying to add the regex "*", a validation error will be triggered, resulting in the disabling of the "Add as regex" option.

Note: If you deselect a particular workflow here, it will not be available on the Request :: View/ Run page for this resource.

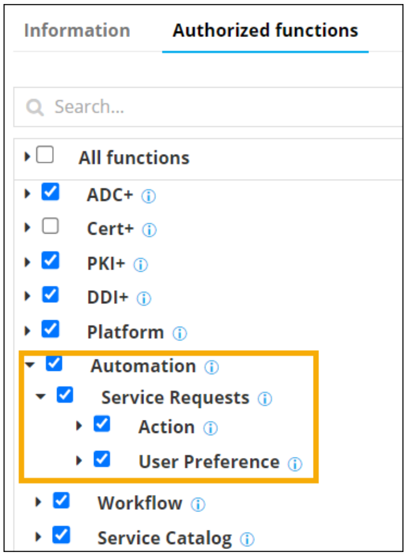
4. Click **Save**.

Role

To get an overview of the actions authorized to a role:

New Menu	Old Menu
In the **Platform** module, from the left pane, under **IDENTITY**, select **Role**. The **Role** page is displayed, with all the roles listed with their **Name**, **Description**, and **Status**.	From the main menu, select **Account** > **Role**. The **Role** page is displayed, with all the roles listed with their **Name**, **Description**, and **Status**.
Note: For more information on how to switch between menus, click here.	

1. On the **Role** page, select any role from the list of roles, for example **admin**.
2. To see the list of permissions that can be defined for this role, under the **Authorized Functions** tab, expand **Automation** > **Service Requests**.



> **Note:** If you are using the old menu, under the **Authorized Functions** tab, expand **Request** > **Workflow**.

3. To see the permissions available under **Action**, click **(Select Right)** icon.

Various actions that can be authorized or restricted for a user within workflows

Action	Description
Abort	Provision to allow or restrict a user to abort a workflow request instantaneously.
Clone	Provision to allow or restrict a user to clone a workflow request with a different name.
Pause / Resume	Provision to allow or restrict a user to pause or resume a workflow request.
Rollback	Provision to allow or restrict a user to trigger an alternative rollback workflow that you have added or imported for a workflow.
Schedule	Provision to allow or restrict a user to schedule a specific time to initiate a workflow request.
Trigger	Provision to allow or restrict a user to trigger or submit a workflow request
View all requests	Provision to allow or restrict a user to view submitted workflow requests of all users.

Action	Description
View my requests	Provision to allow or restrict a user to view workflow requests initiated by the user.
View usergroup requests	Provision to allow or restrict a user to view workflow requests initiated by the usergroup.

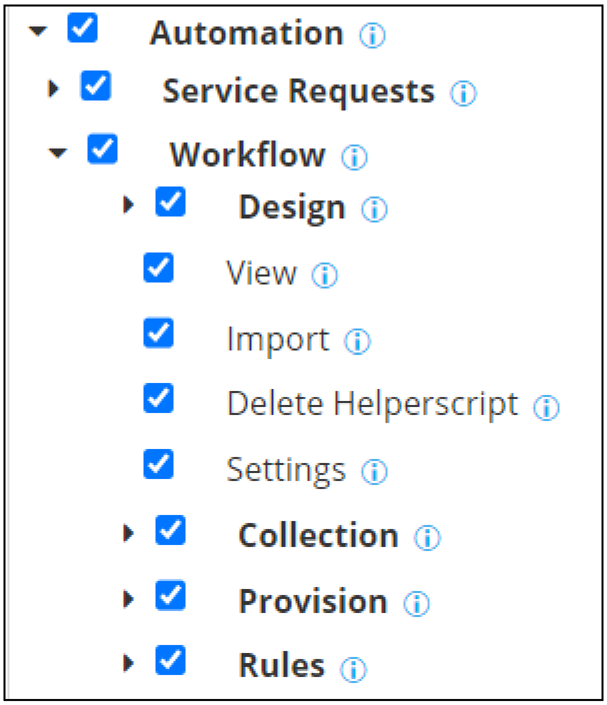
4. Select/Deselect the checkbox to allow/restrict that action for the selected user.
5. To see the permissions available under **User Preference**, click ▸ **(Select Right)** icon.

Various actions that can be authorized or restricted for a user under User Preference

Action	Description
Show Overview	Provision to show or hide the Overview menu for a user.
Show Custom reports	Provision to show or hide Custom reports menu for a user.
Show My workflow	Provision to show or hide My workflow menu for a user.
Show Scheduled jobs	Provision to show or hide Scheduled jobs menu for a user.
Show All requests	Provision to show or hide All requests menu for a user.
Show Open requests	Provision to show or hide Open requests menu for a user.
Show Success requests	Provision to show or hide Success requests menu for a user.
Show Failed requests	Provision to show or hide Failed requests menu for a user.
Show Assigned requests	Provision to show or hide Assigned requests menu for a user.
Show Audits	Provision to show or hide Audit menu for a user.
Show Archival requests	Provision to show or hide Archival requests menu for a user.
Show Restore requests	Provision to show or hide Restore requests menu for a user.
Show Preference	Provision to show or hide Preference menu for a user.
Show Request logs	Provision to show or hide Request logs for a user.
Show Manage catalog	Provision to show or hide the Manage catalog window for a user.

Action	Description
Show Request stage	Provision to show or hide the service request stage view for a user.

6. Select/Deselect the checkbox to allow/restrict that action for the selected user.
7. To see the permissions that can be defined for the user in the Workflow Studio, expand **Automation** > **Workflow**.



> **Note:** If you are using the old menu, under the **Authorized Functions** tab, expand **Request** > **Workflow**.

Various actions that can be authorised or restricted for a user in the Workflow Studio module

Action	Description
Design	Provision to allow or restrict a user to design a workflow. The following permissions are available under Design: • **Create/Modify**: Create a new workflow or modify an existing workflow • **Delete**: Delete existing workflow(s). • **Clone**: Clone existing workflow(s).

Action	Description
	• **Export**: Export workflows (that are in disabled state) to other environments. • **Shared folder**: Share folder(s) containing logically grouped workflows.
View	Provision to allow or restrict a user to view detailed information about a workflow.
Import	Provision to allow or restrict a user to import workflows from other environments.
Delete Helperscript	Provision to allow or restrict a user to delete helper script(s).
Settings	Provision to allow or restrict a user to update workflow settings.
Collection	Provision to allow or restrict a user to access Collection.
Provision	Provision to allow or restrict a user to manage provisioning settings.
Rules	Provision to allow or restrict a user to define action-based rules to trigger workflows for ADC objects and dynamically tag resources.

8. Select/Deselect the checkbox to allow/restrict that action for the selected user.
9. Once the required permissions have been assigned, click **Save**.

User Group

You can see the details of a particular User Group such as the Roles and Resources mapped to it.

New Menu	Old Menu
In the **Platform** module, from the left pane, under **IDENTITY**, select **Usergroup**. The **User Group**page is displayed, with all the user groups listed with their **Name**, **Description**, and other related information.	From the main menu, select **Account** > **Usergroup**. The **User Group**page is displayed, with all the user groups listed with their **Name**, **Description**, and other related information.
Note: For more information on how to switch between menus, click here.	

User

You can see the details of a group that a User is mapped to and also modify user information.

New Menu	Old Menu
In the **Platform** module, from the left pane, under **IDENTITY**, select **User**. The **User** page is displayed, with all the user groups listed with their **Name**, **Full Name**, and other related information.	From the main menu, select **Account** > **User**. The **User** page is displayed, with all the user groups listed with their **Name**, **Description**, and other related information.
Note: For more information on how to switch between menus, click here.	

Getting Started with Prebuilt Workflow Tasks

In order to get you started with automation, the FAQs folder within the studio has some quick references to aid you in using simple tasks, variables and workflows, and basic workflow features.

Note: You can also access FAQs and a few Pre-built workflows to get you started in the Solutions section.

- Accessing FAQs
- Building Auto-generate Forms
- Sample Workflows using Prebuilt Tasks
- Application Delivery Automation
- Automate DNS Services
- Change Automation
- DevOps and Continuous Configuration Automation
- GitOps Integration
- Utilities
- SDK Automation

Accessing FAQs

The FAQs folder contains quick reference workflows that will help you in designing and understanding workflow features. The FAQ workflows are available as part of the AppViewX Installation.

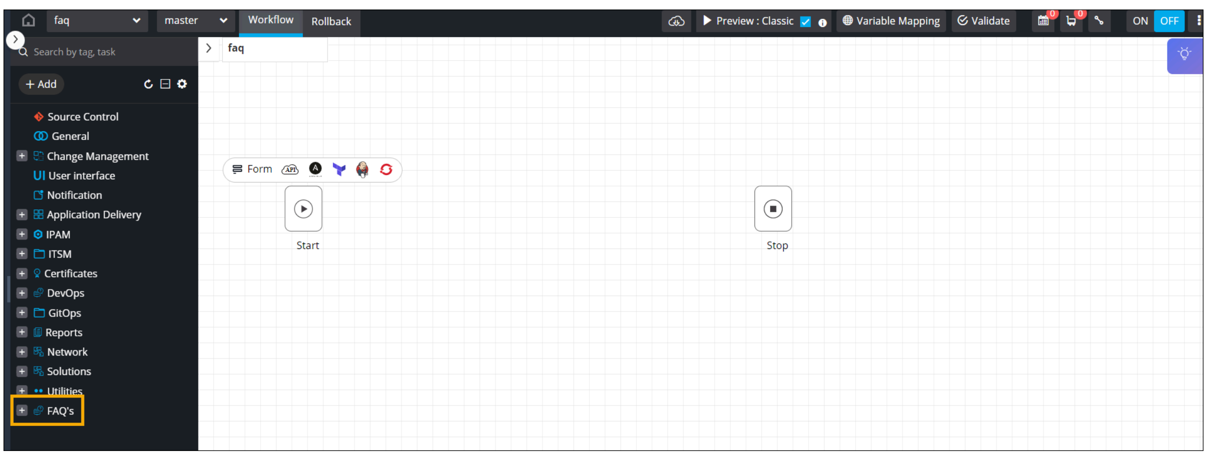
Following are a few sample flows that will help you learn how to connect tasks:

- Script to any task
- Form to any task
- Using Diff Checker
- Using Charts
- Using Loops
- Using Auto Email Approvals
- Using Hooks

To access the FAQs folder from the workflow design space:

1. Go to **Automation > WORKFLOW > Design** the **Workflow** page is displayed.
2. On the **Workflow** page that is displayed, click **Create New**.

   The design space opens, with the FAQs folder visible in the left menu.

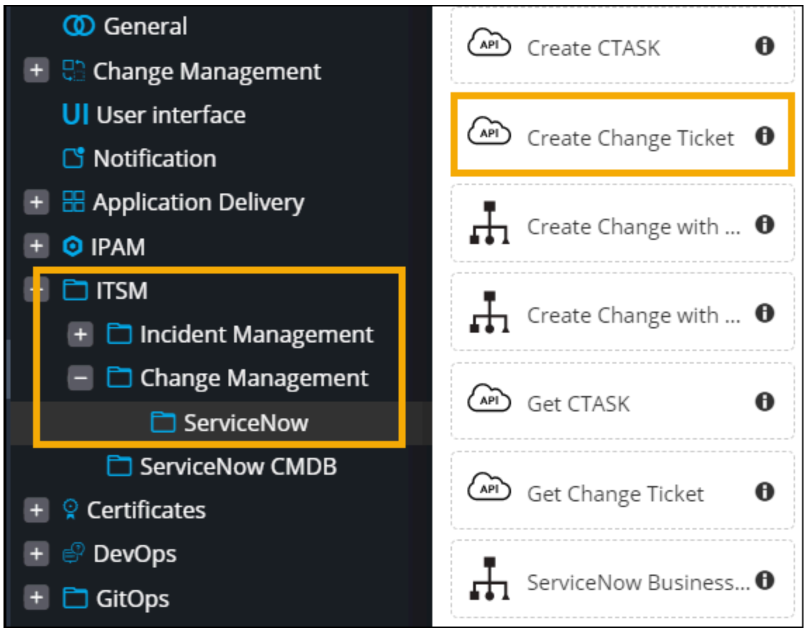


Note: You can also access the FAQs by opening any of the existing workflows on the **Workflow** page.

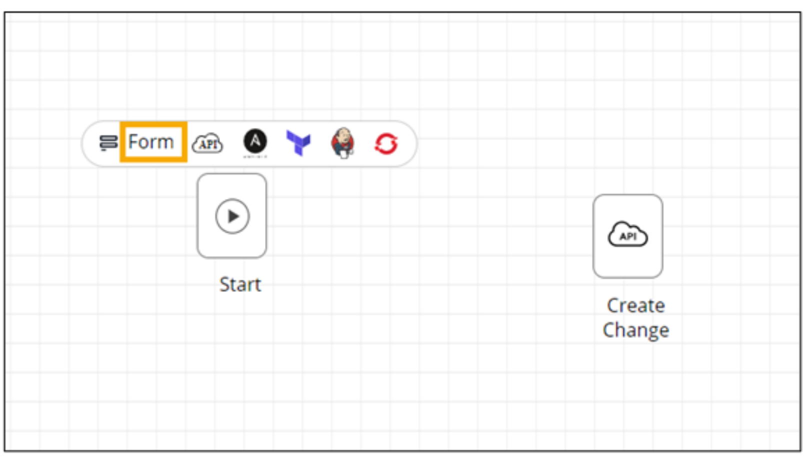
Building Auto-generate Forms

You can auto-generate a self-service form from any workflow task(s). The variables used within a task are rendered into the form automatically.

1. Design a new workflow.
2. Drag and drop a prebuilt workflow task. For example, **Create Change Ticket** task from the **ITSM** > **Change Management** > **ServiceNow** folder.



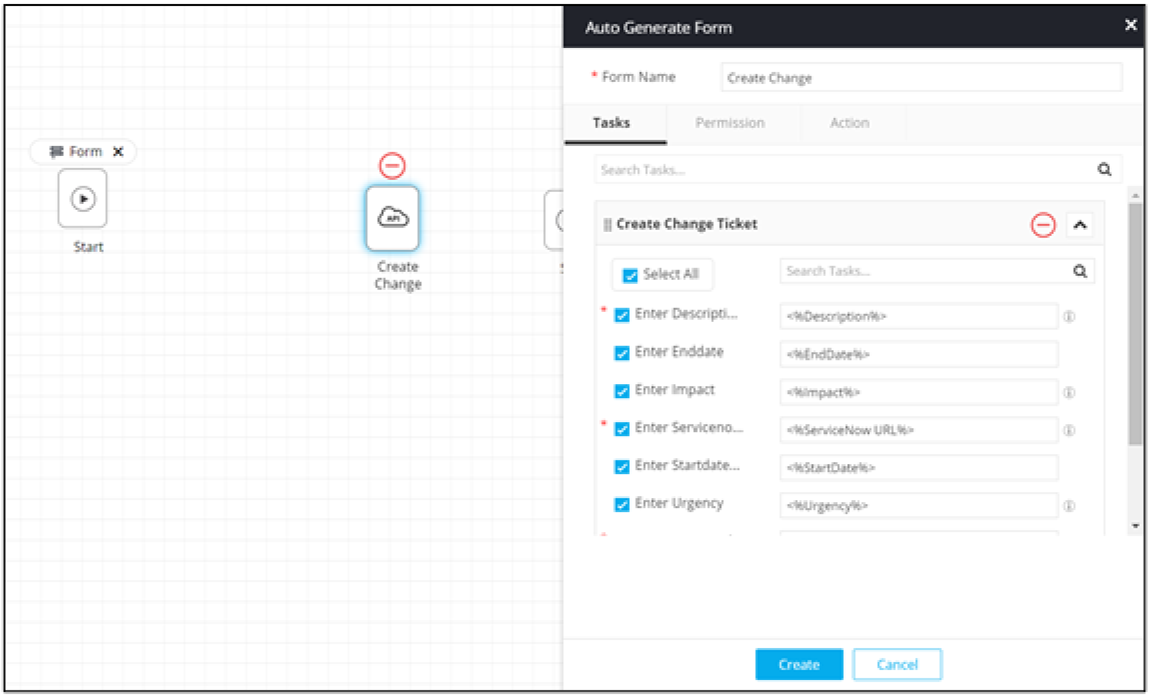
3. To auto-generate a self-service form, click **Form** above the **Start** task.



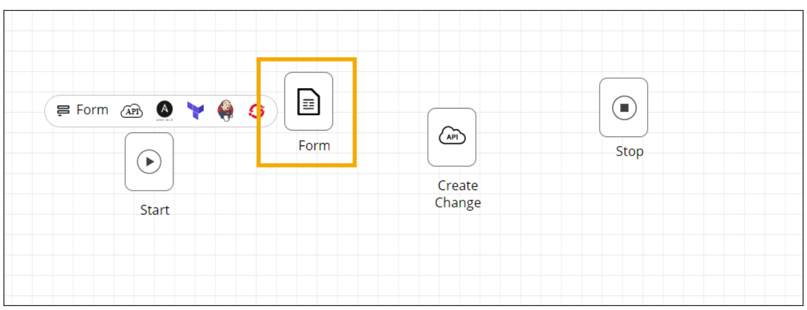
The **Auto Generate Form** window is displayed.

4. Enter the **Form Name**.

5. To auto-fill the variables in the form, click **(Add)** icon above the **Create Change** task.
   The input tasks for the **Create Change** task are auto-populated in the **Auto Generate Form** window, under **Tasks**.



6. Select/deselect the fields to be displayed in the self-service form and click **Create**.
   The **Form** task is created and added to the design space.



Sample Workflows using Prebuilt Tasks

This section will walk you through a few sample workflows that describe the step-by-step process of creating workflows using prebuilt tasks.

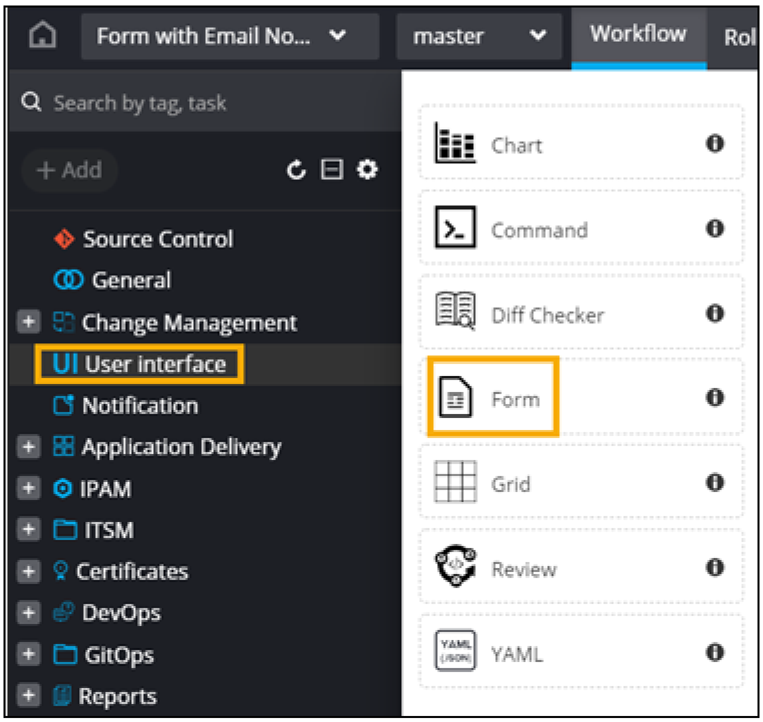
- Form with Email Notification
- Executing a Ping Check using Command Task

- Creating A Record on Infoblox device using prebuilt tasks
- Creating a VIP with Incident Ticket

Form with Email Notification

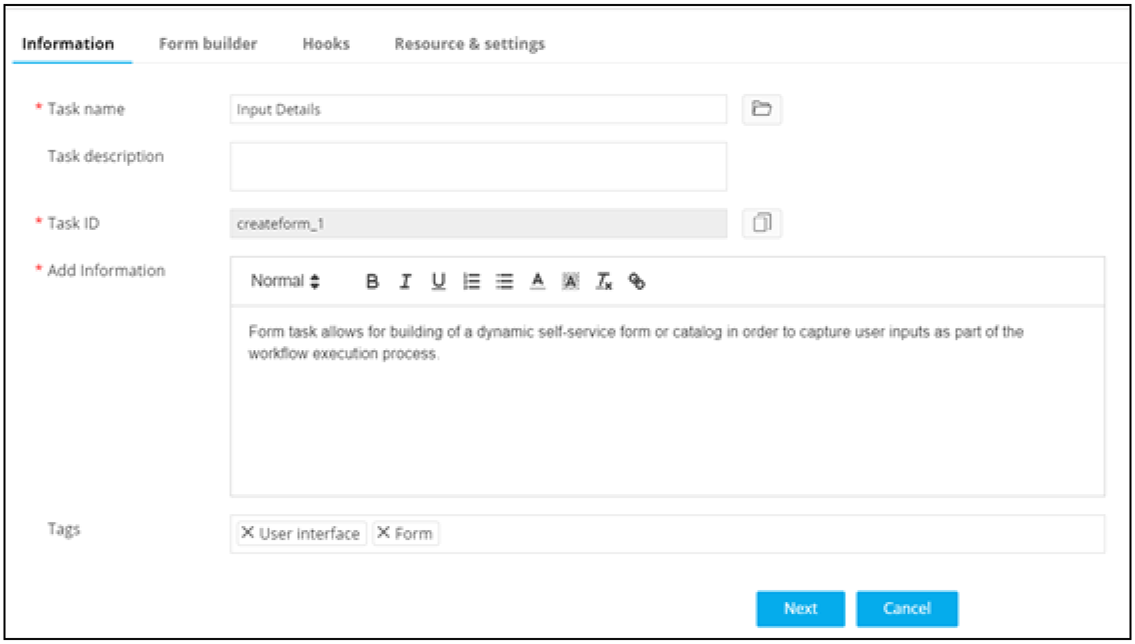
You can design a workflow to send an email notification based on the details provided in a form.

1. Design a new workflow.
2. In the design workspace, from the User Interface section in the left menu, drag and drop a **Form** task.



The **Gallery** window is displayed on the screen.

3. In the **Gallery** window, to design a new form, click **(Create Form)** icon.
4. In the Form task window that is displayed, under the **Information** tab, enter the **Task name**.

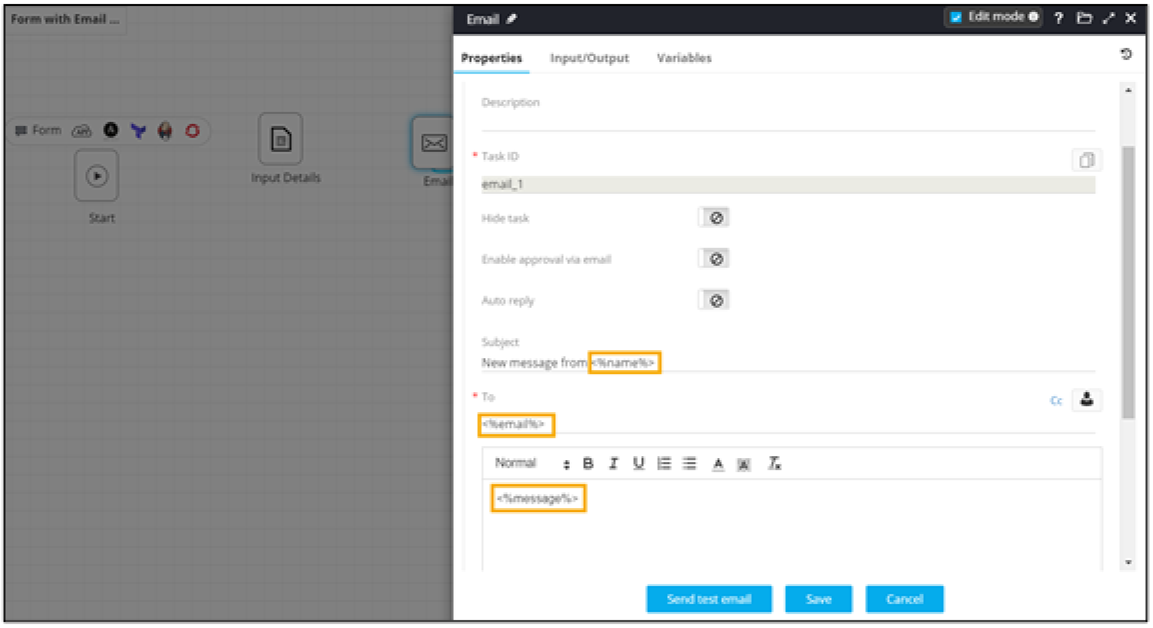


5. Click **Next**.

   The **Form builder** tab opens with the **Field Properties** window displayed at the bottom right corner of the screen.
6. Add the following form fields and turn on the Global variable toggle to enable the field IDs as global variables:

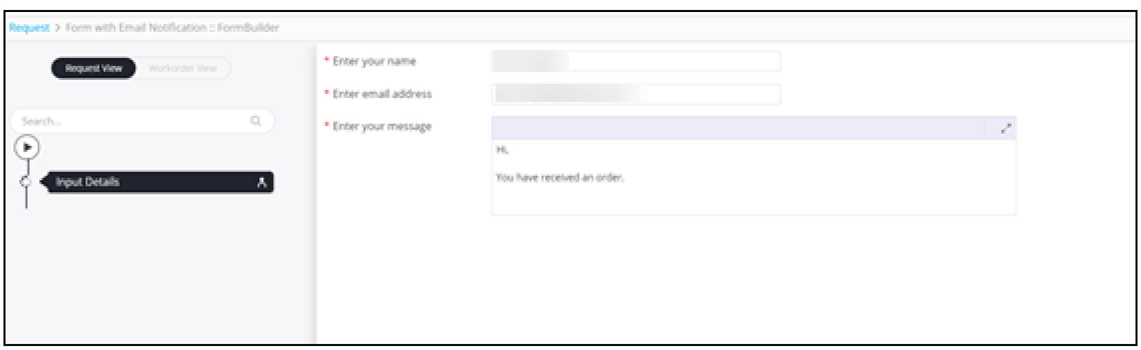
Label name	Field type	Field ID
Enter your name	Text box	name
Enter your email address	Text box	email
Enter your message	Multi-line	message

7. Click **Save**.
8. From the Notification section in the left menu, drag and drop **Email** task.
9. In the **Email** task window that is displayed, under **Properties**, in the **General** section, refer the global variables defined in the Form task.



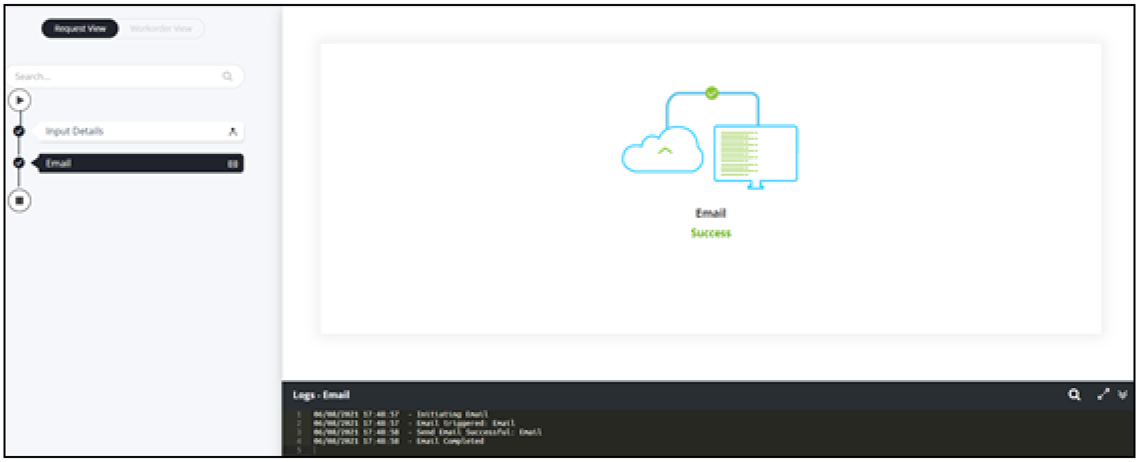
10. Click **Save**.
11. Connect the workflow tasks and enable the workflow.
12. Trigger the workflow.

    The workflow execution page is displayed with the workflow inputs requested at the first stage.
13. Enter the details in the input form.



14. Click **Submit**.

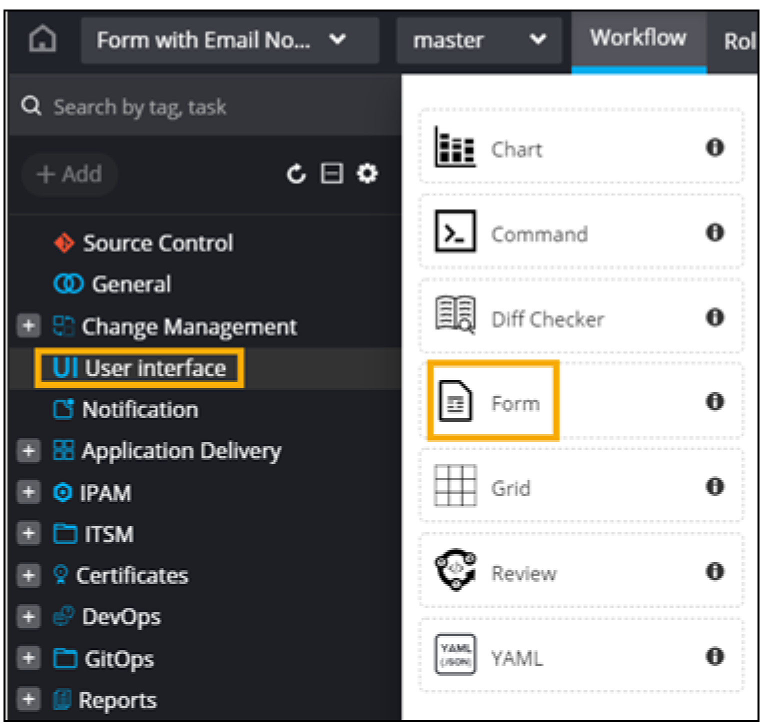
Email notification is successfully sent to the email address provided in the form.



Executing a Ping Check using Command Task

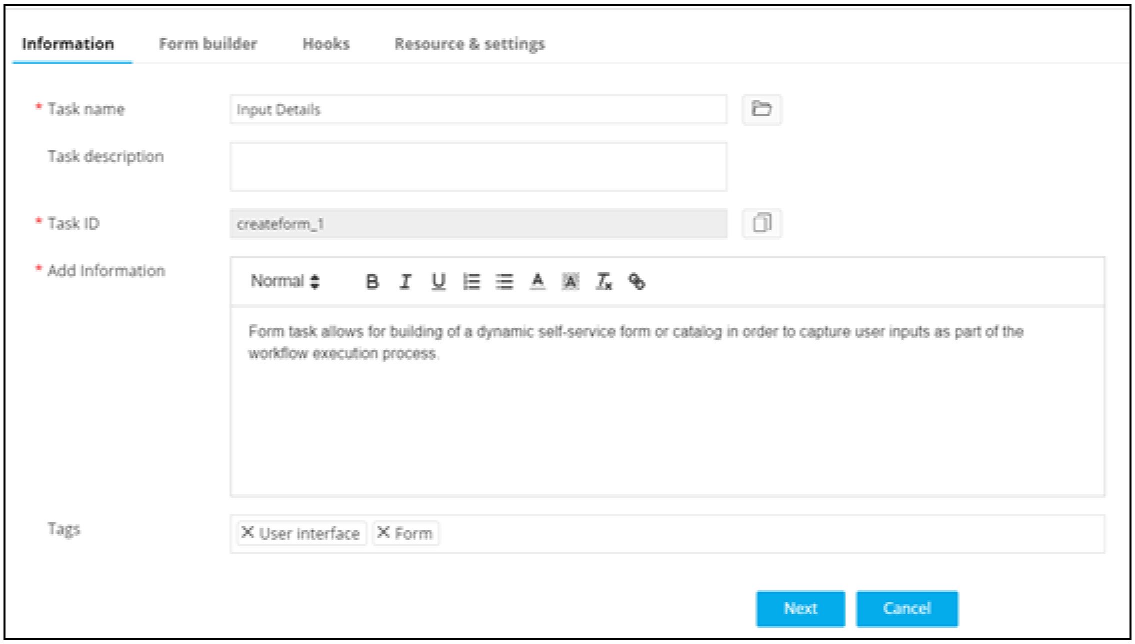
You can design a workflow to use the Command task to execute a ping check on a hostname or server.

1. Design a new workflow.
2. In the design workspace, from the **User Interface** section in the left menu, drag and drop a **Form** task.

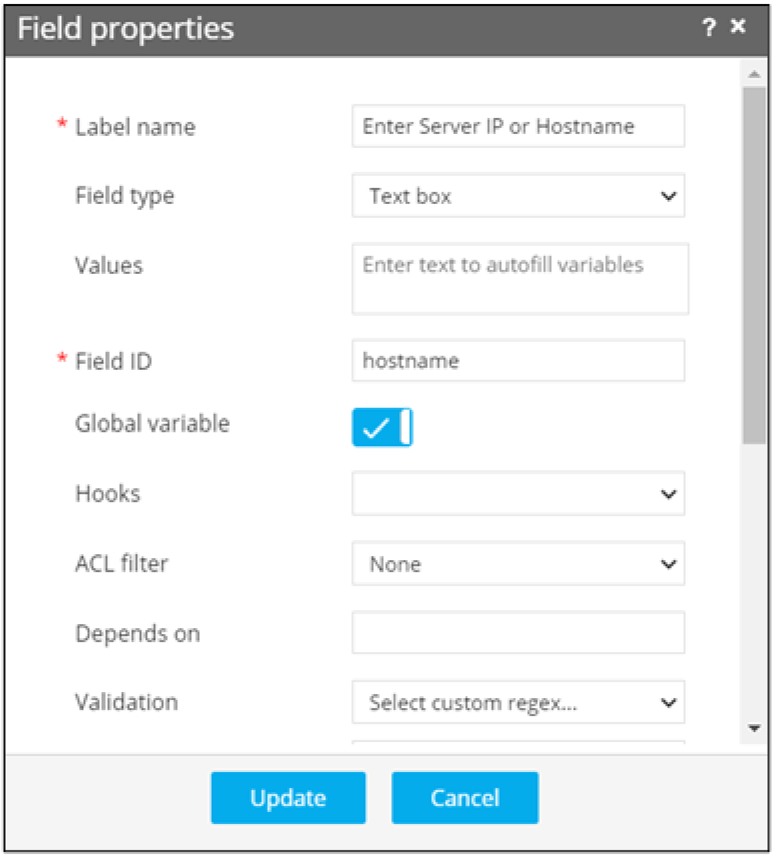


The **Gallery** window is displayed on the screen.

3. In the **Gallery** window, to design a new form, click **(Create Form)** icon.
4. In the **Form** task window that is displayed, under the **Information** tab, enter the **Task name**.



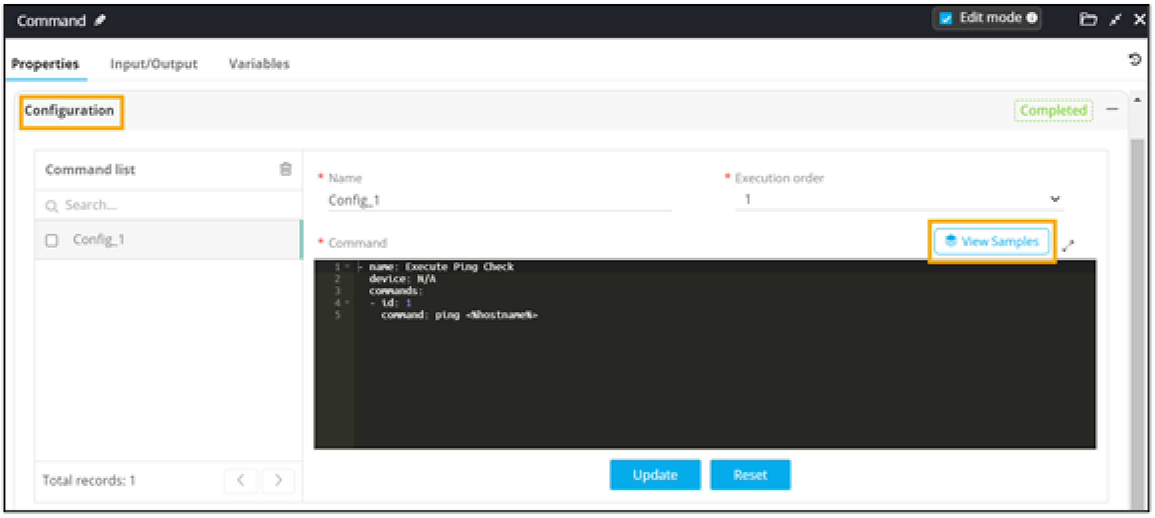
5. Click **Next**.
   The **Form builder** tab opens with the **Field Properties** window displayed at the bottom right corner of the screen.
6. In the **Field properties** window, define the form field to get the Server IP or Hostname as shown.



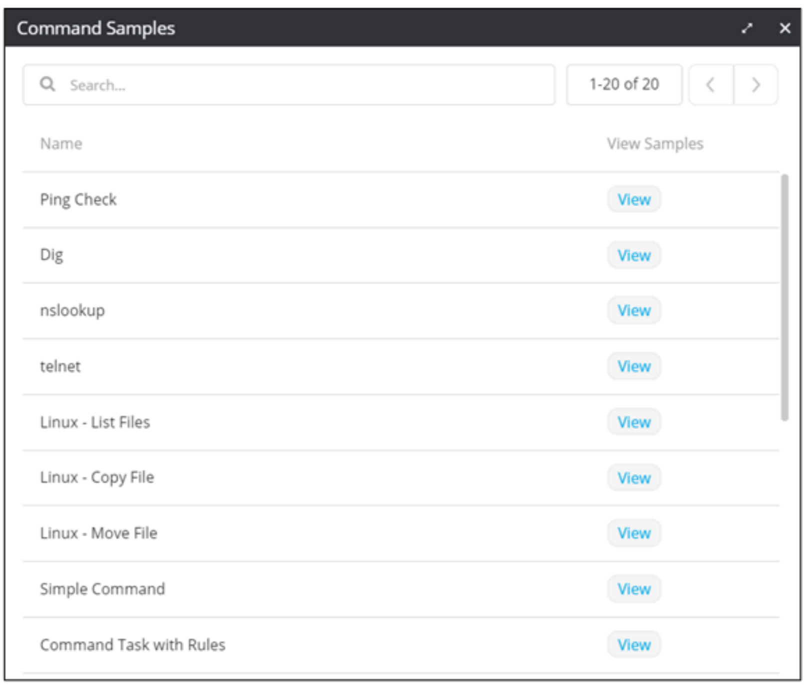
7. To declare the Field ID as a global variable, `<%hostname%>`, and refer it in the next task, turn on the **Global variable** toggle.

> **Note:** For more information on adding form fields, click here.

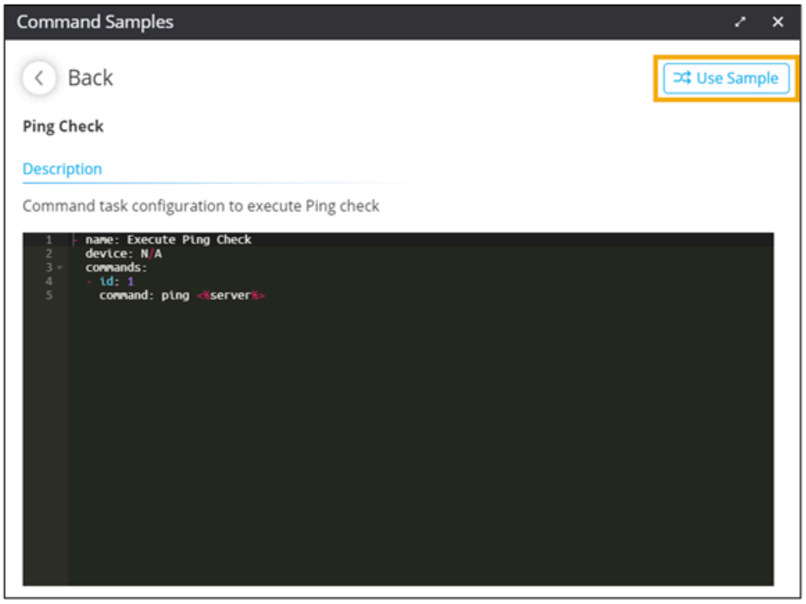
8. Click **Save**.
9. From the User Interface section in the left menu, drag and drop the **Command** task.
10. In the **Command** task window, under **Properties**, in the **Configuration** section, click **View Samples**.



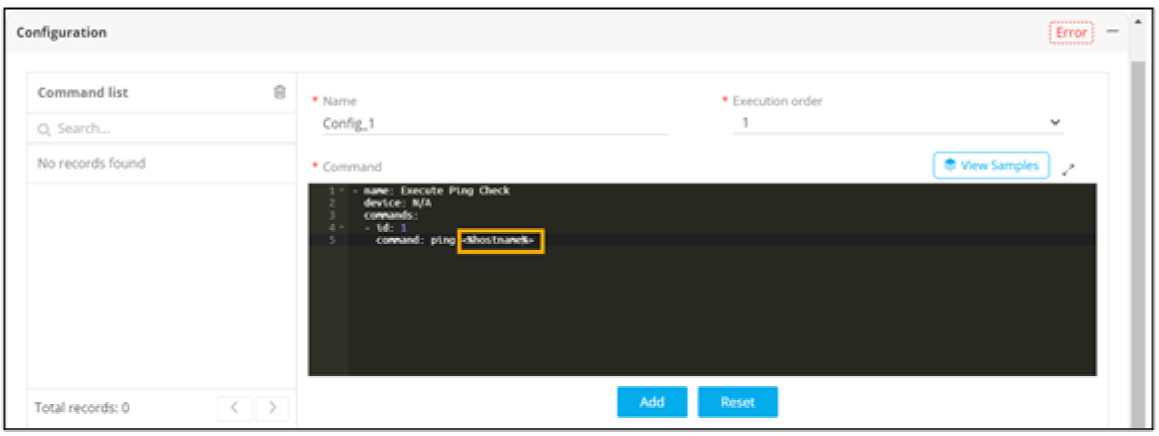
The **Command Samples** window is displayed.



11. Click **View** next to the **Ping Check** command.
12. To use this command, click **Use Sample**.



13. In the **Configuration** section, refer the global variable `<%hostname%>` defined in the form task.



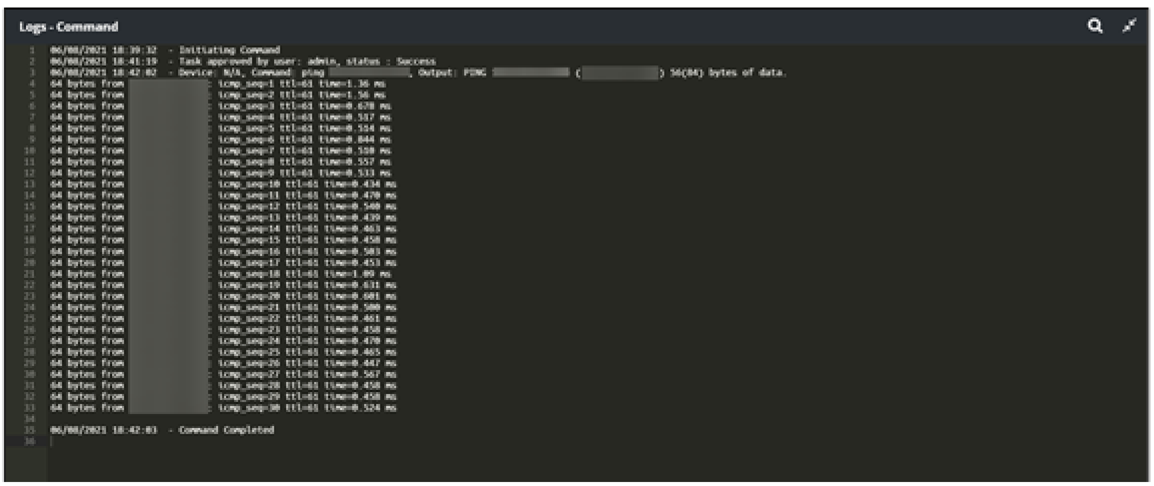
Note: device: N/A implies that the command will be executed on AppViewX server.

14. To add this configuration, click **Add**.
15. Click **Save**.
16. Connect the workflow tasks and enable the workflow.
17. Trigger the workflow.

    The workflow execution page is displayed with the workflow inputs requested at the first stage.
18. Enter the Server IP or Hostname in the form.

19. Click **Submit**.
20. In the next stage of workflow execution, to execute the command for performing ping checks on the server, click **Submit**.

    Command is executed with the Logs showing the server response.

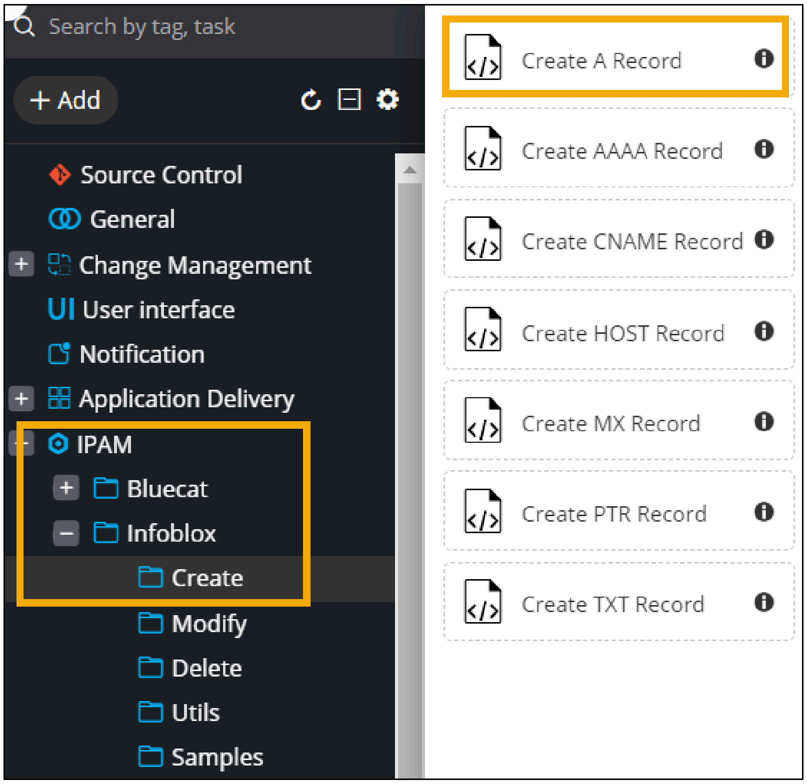


Creating A Record on Infoblox device using prebuilt tasks

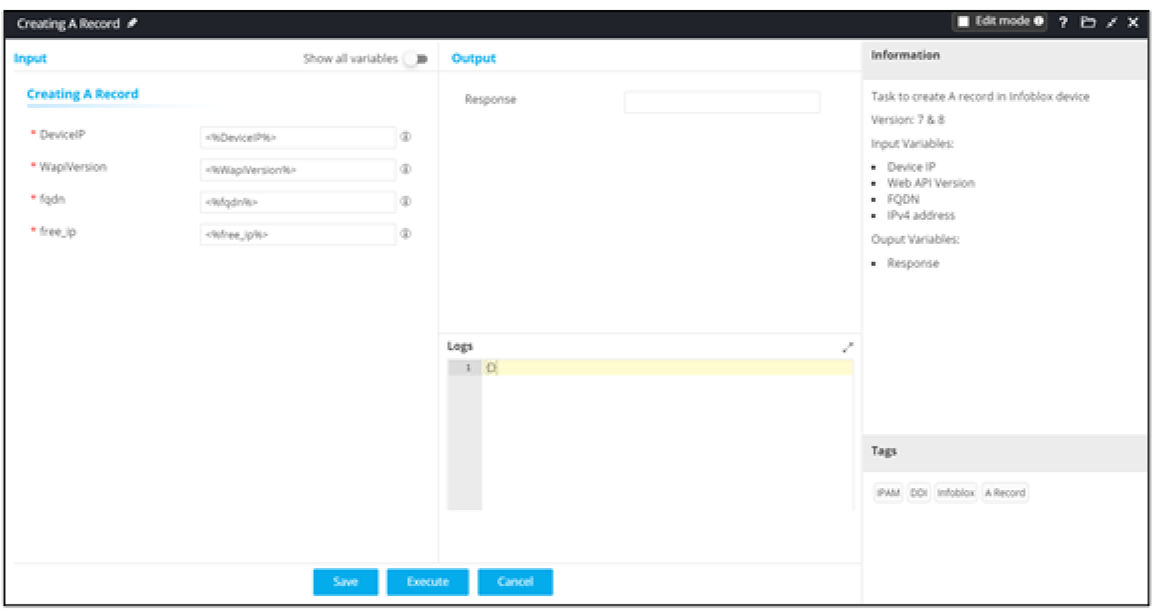
You can design a workflow with an auto-generated form to create A record on an Infoblox device using prebuilt tasks.

> **Note:** Before creating the workflow, ensure that the device(s) are available in the Device Inventory.

1. Design a new workflow.
2. In the design workspace, from the **IPAM** > **Infoblox** > **Create** folder drag and drop the **Create A Record** prebuilt task.

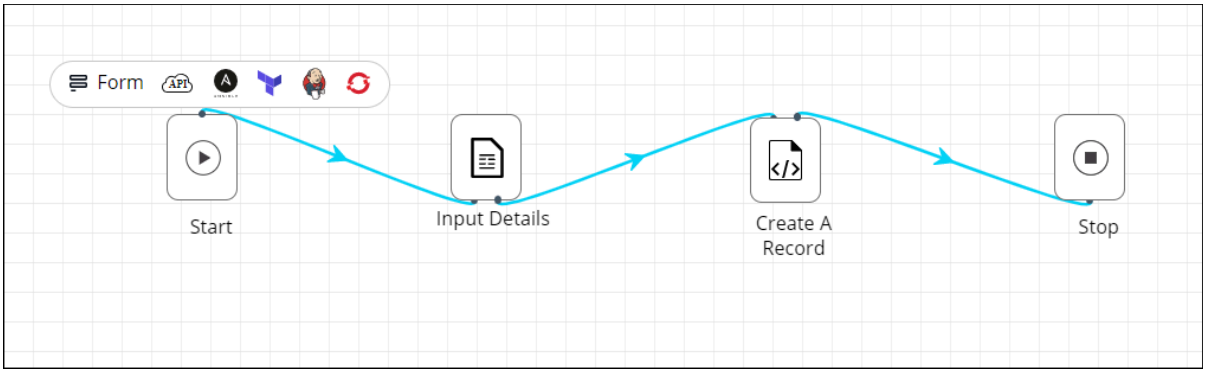


The **Creating A Record** window (Citizen mode) shows the **Input** variables required for this task.



3. To view the API configuration details, in the **Creating A Record** window, select the **Edit mode** checkbox.
4. Click **Save**.
5. Auto generate a self-service form for this workflow.

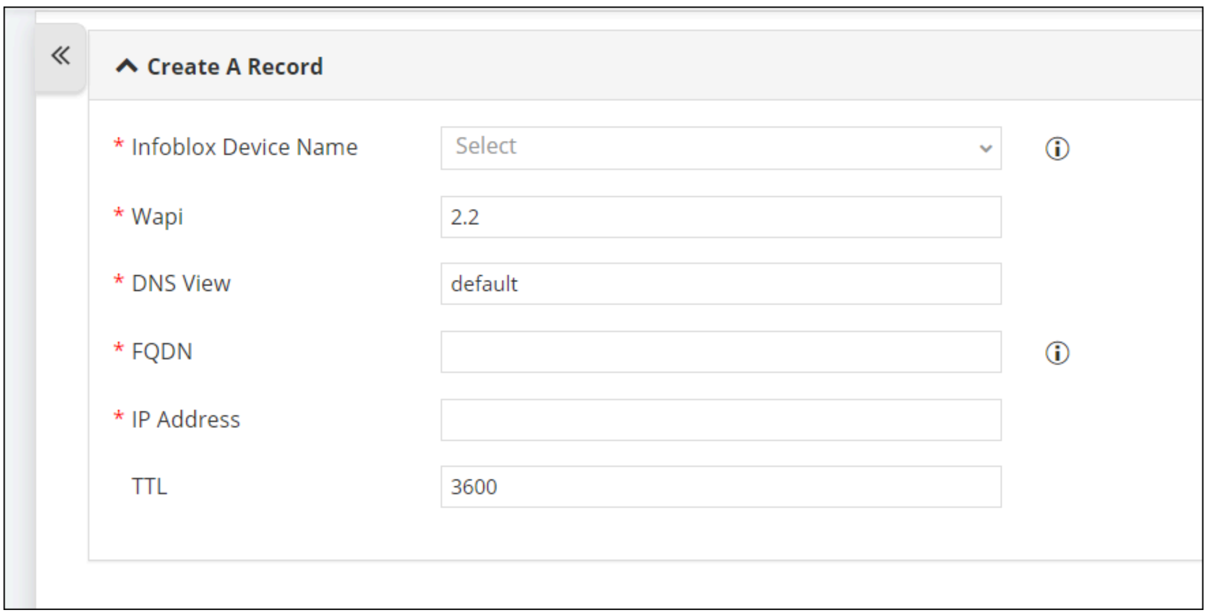
6. Connect the workflow tasks and enable the workflow.



7. Trigger the workflow.

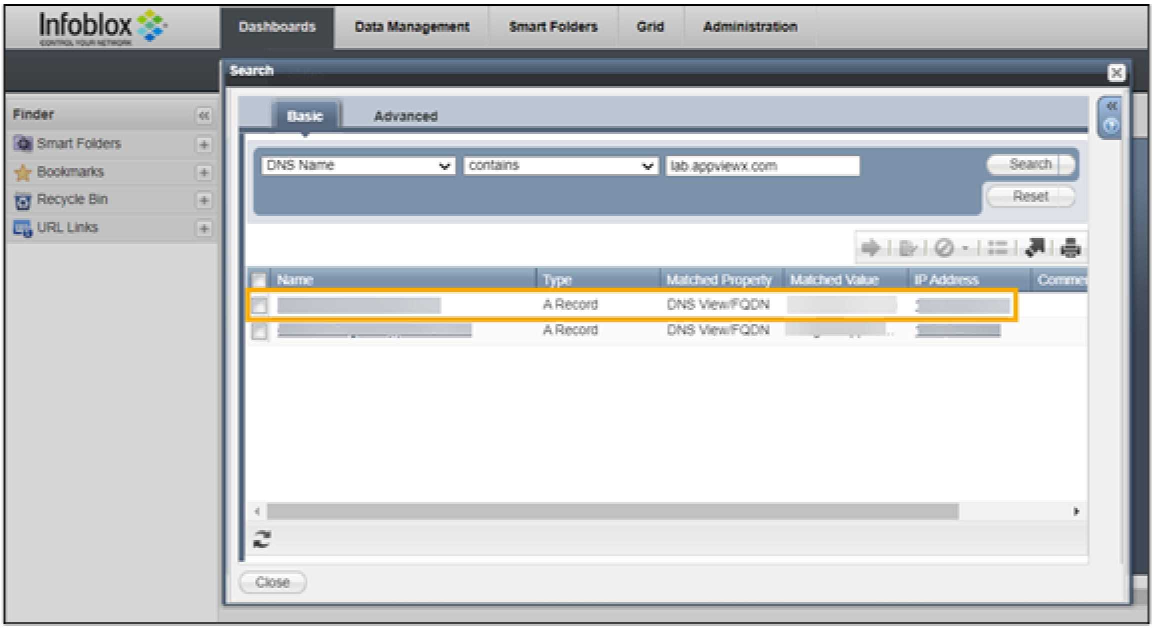
   The workflow execution page is displayed with the workflow inputs requested at the first stage.

8. Enter the input details in the form.



9. Click **Next**.

A record is created on Infoblox.



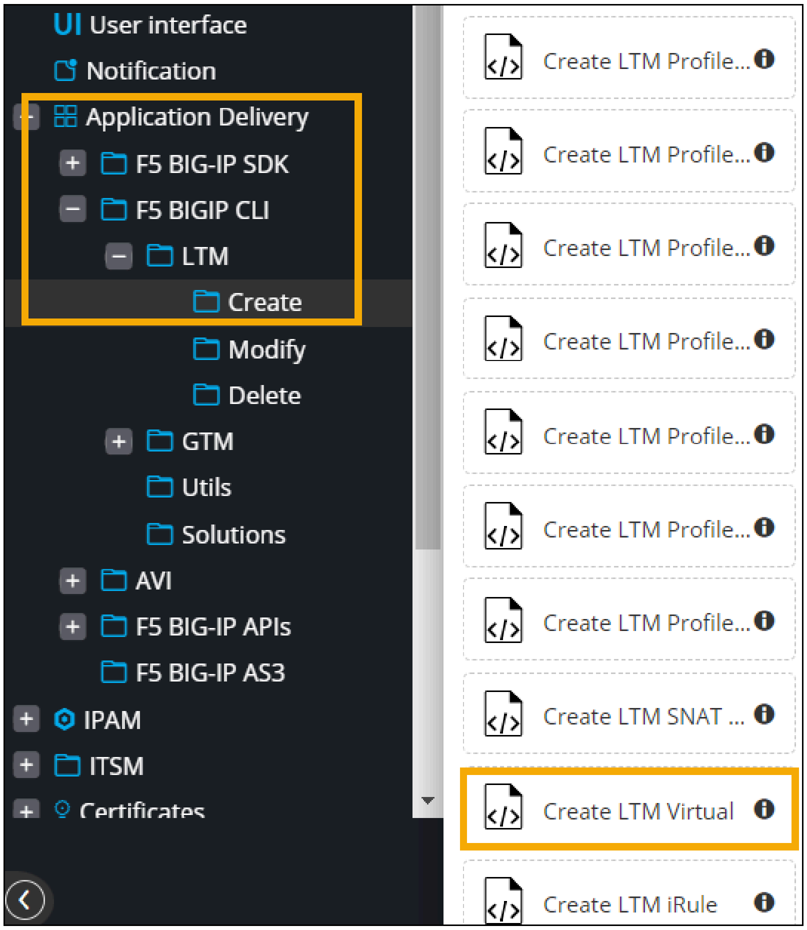
Creating a VIP with Incident Ticket

The ITSM folder allows you to leverage the prebuilt tasks for designing ITSM automation workflows.

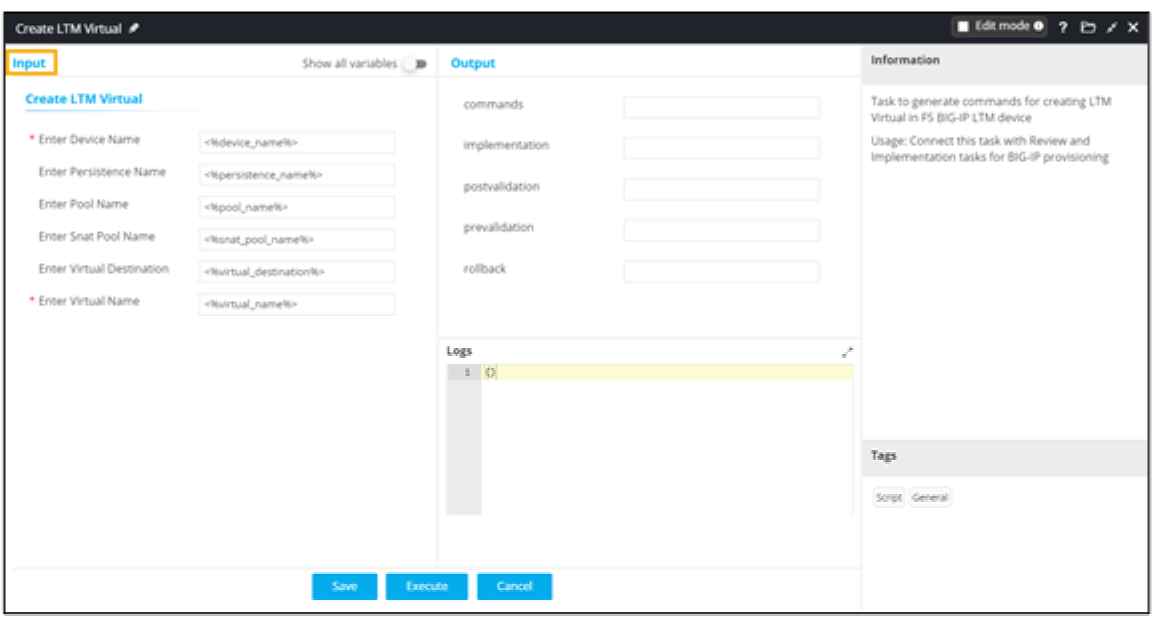
> **Note:** Before designing the workflow, ensure that you have configured an instance to integrate with the ITSM vendor. For more information, refer to the section on ITSM Vendor Configuration

To create a VIP with Incident ticket:

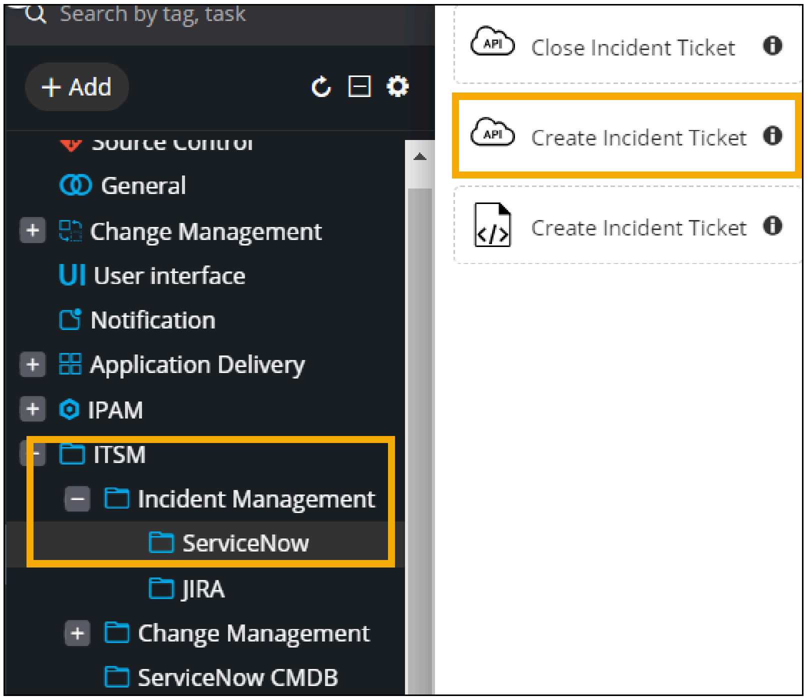
1. Design a new workflow.
2. To create a virtual LTM server, from the **Application Delivery** > **F5 BIG-IP CLI Commands** > **LTM** > **Virtual** folder, drag and drop the prebuilt **Create LTM Virtual task**.



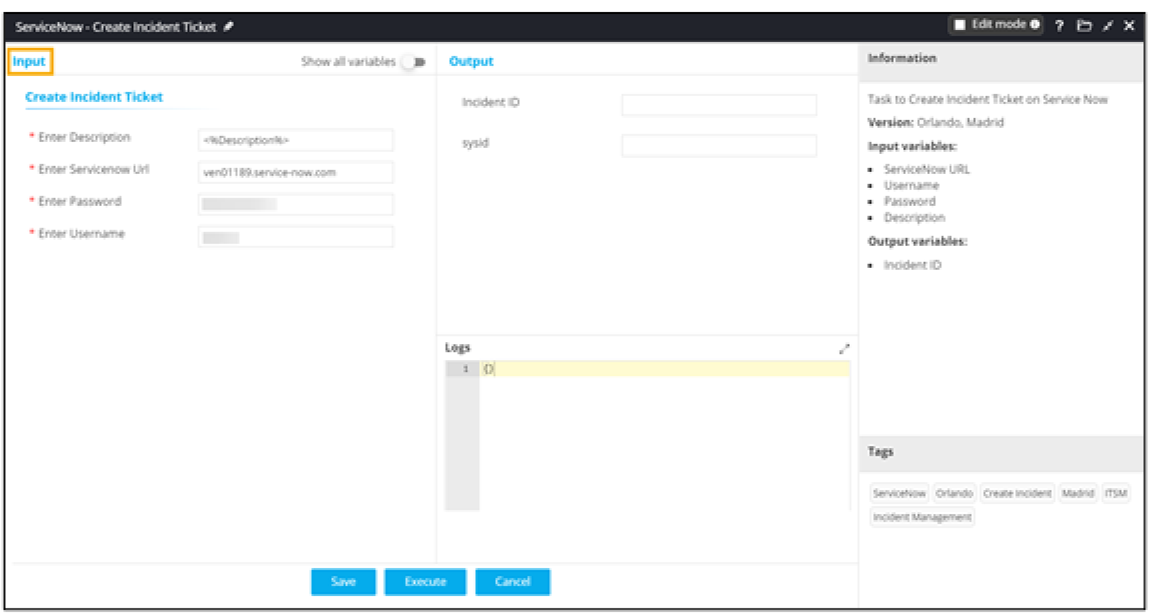
The **Create LTM Virtual** window (Citizen mode) shows the **Input** variables required for this task.



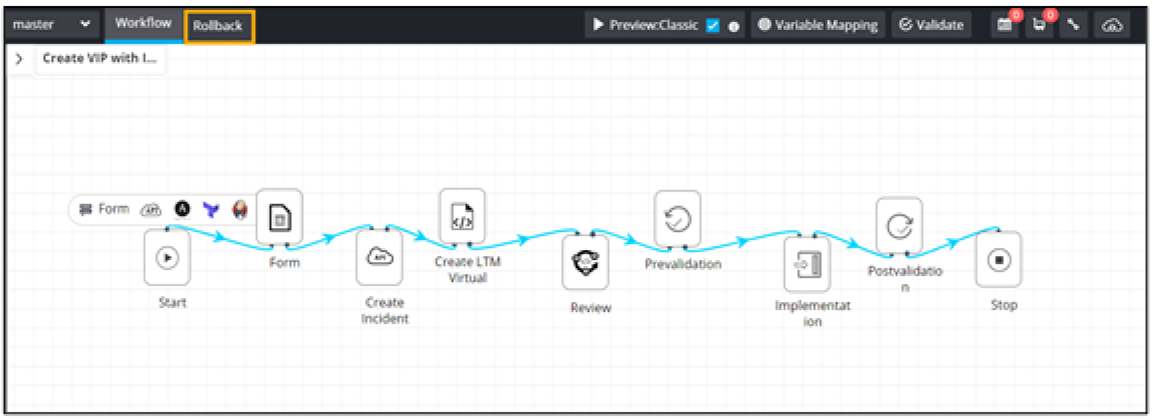
3. To view or edit the task configurations, in the **Create LTM Virtual** window, from the top right corner, select the **Edit mode** checkbox.
4. Click **Save**.
5. From the **ITSM** > **Incident Management** > **ServiceNow** folder, drag and drop the **Create Incident Ticket** task.



6. In the **ServiceNow - Create Incident Ticket** window, enter the **Input** variables required to create the Incident ticket.



7. Click **Save**.
8. From the **User Interface** section in the left menu, drag and drop the **Review** task.
9. Click **Save**.
10. From the **General** section in the left menu, drag and drop the **Prevalidation** task.
11. Click **Save**.
12. From the **General** section in the left menu, drag and drop the **Implementation** task.
13. Click **Save**.
14. From the **General** section in the left menu, drag and drop the **Postvalidation** task.
15. Click **Save**.
16. Auto generate self-service form for this workflow.
17. Connect the workflow tasks.
18. To add a rollback workflow, from the menu on the top, click **Rollback**.

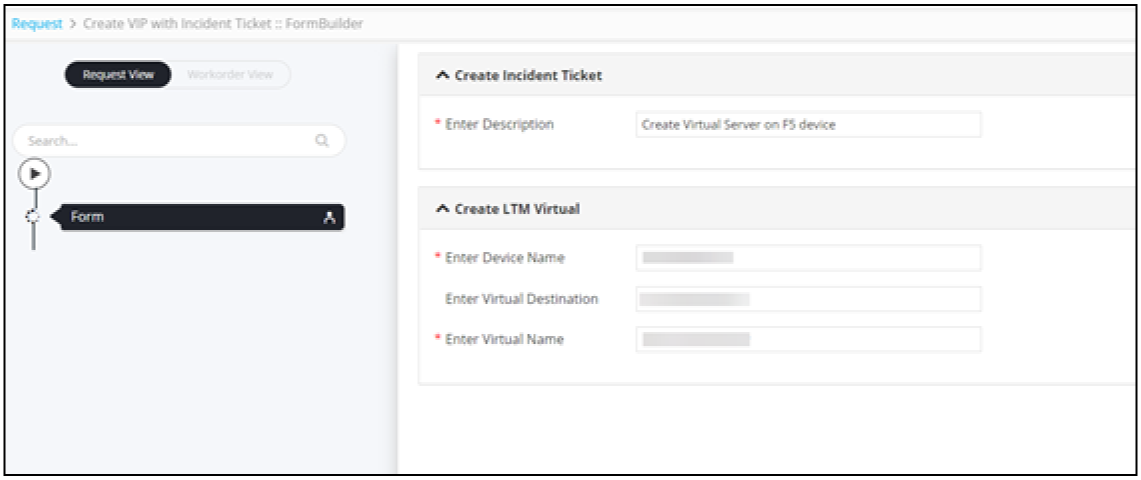


19. From the **General** section in the left menu, drag and drop the **Rollback** task.
20. Click **Save**.
21. Connect the rollback workflow tasks and enable the workflow.

> **Note:** No values are passed in the Review, Prevalidaton, Implementation, Postvalidation, and Rollback tasks, since these tasks are already mapped to the Create LTM Virtual task as output .

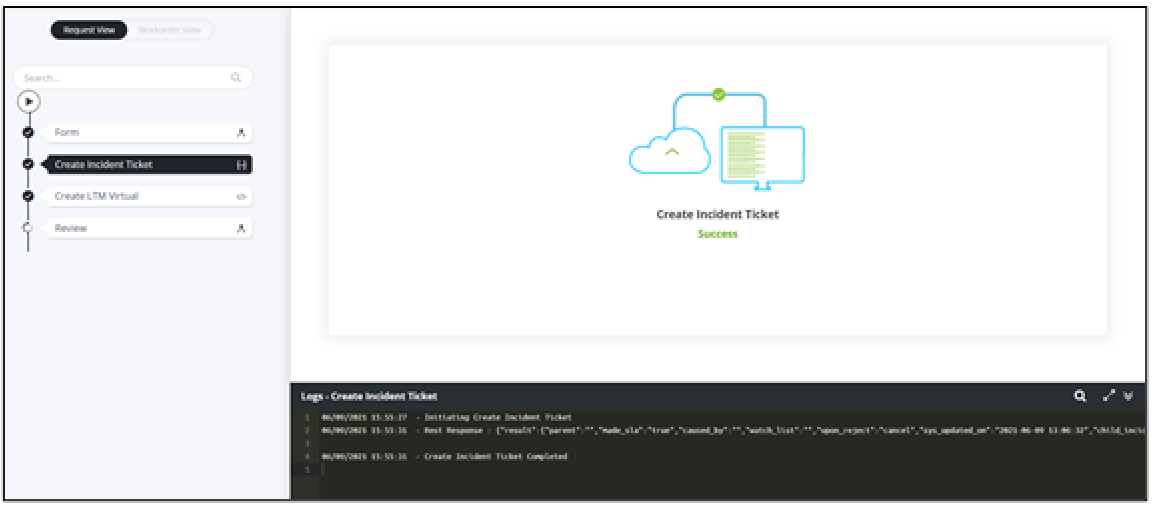
22. Trigger the workflow.

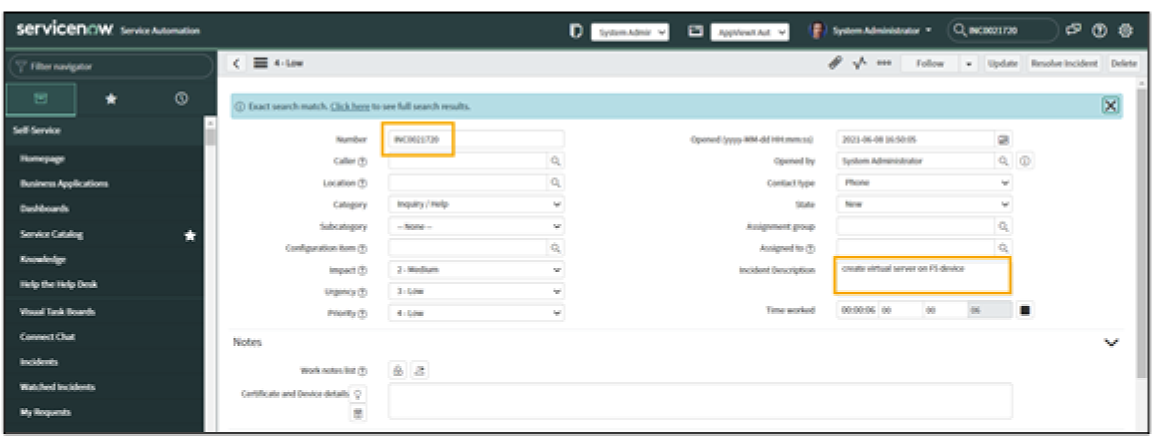
    The workflow execution page is displayed with the workflow inputs requested at the first stage.
23. Enter the input details in the form.



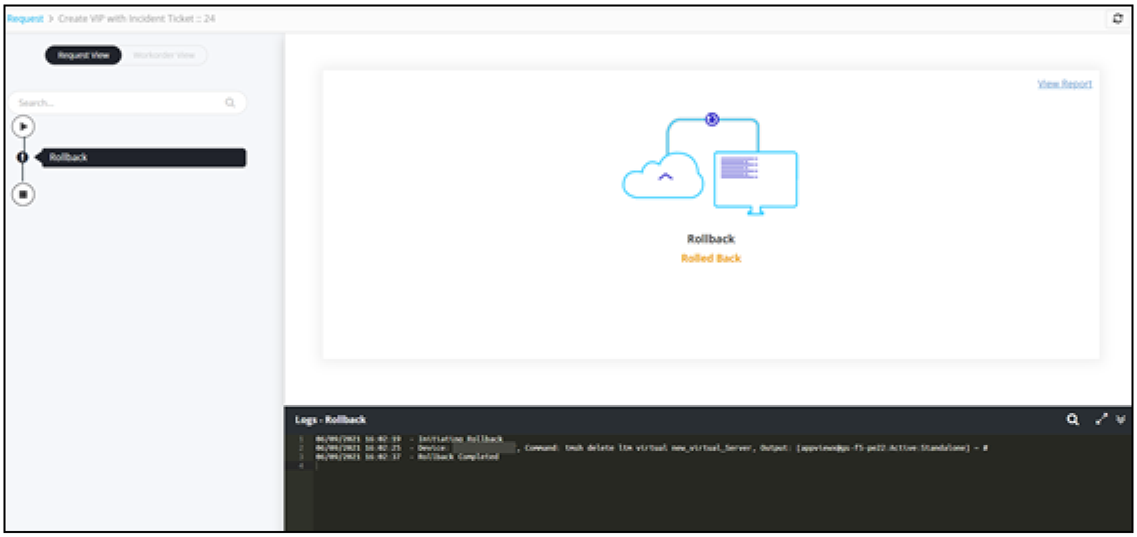
24. Click **Next**.

    Incident Ticket is created on ServiceNow.

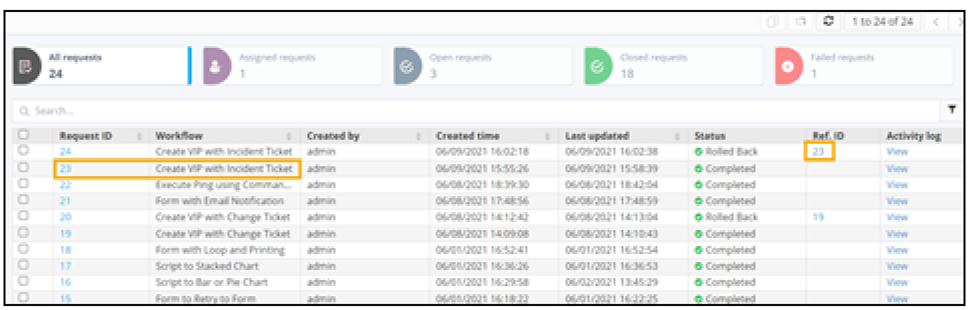




25. At the **Review** stage, to execute the implementation, prevalidation, rollback, and postvalidation commands, click **Submit**.
26. To execute the rollback workflow, on the Request :: All page, select the checkbox next to the workflow **Request ID**.
27. From the upper right corner of the screen, click **(Rollback)** icon.
    Rollback request to delete the virtual server that was created, is executed successfully.



28. To view the request ID of the rollback workflow, navigate to the Request :: All page.
    Rollback workflow request displays the **Ref. ID** of the workflow for which rollback was initiated.



Application Delivery Automation

With AppViewX's Application Delivery Automation, you can seamlessly deploy applications in heterogeneous environments (on-premise, multi-cloud, and hybrid) and simplify and accelerate application delivery.

- F5 BIG-IP Solutions
- AVI Solutions

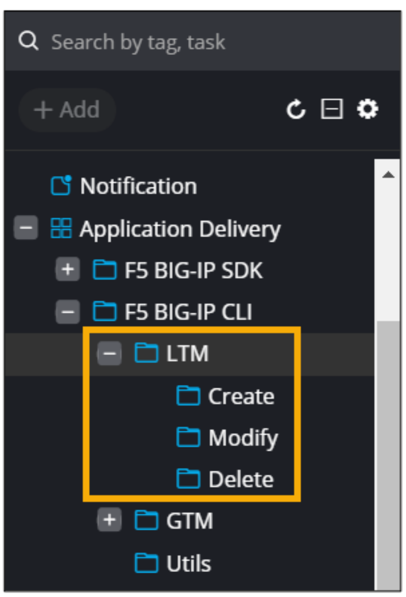
F5 BIG-IP Solutions

This section describes some of the workflows/tasks for F5 BIG-IP provisioning available in the Workflow Studio.

- LTM
- GTM
- Utils
- Creating a LTM Virtual Server on F5 BIG-IP

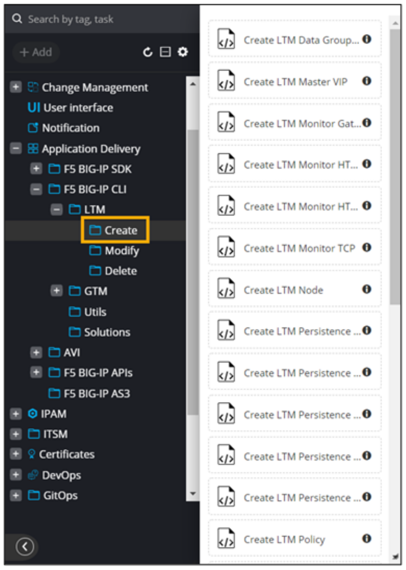
LTM

The workflows and tasks to create, modify, and delete LTM objects and devices on F5 BIG-IP are available under **Application Delivery** > **F5 BIG-IP CLI** > **LTM**.



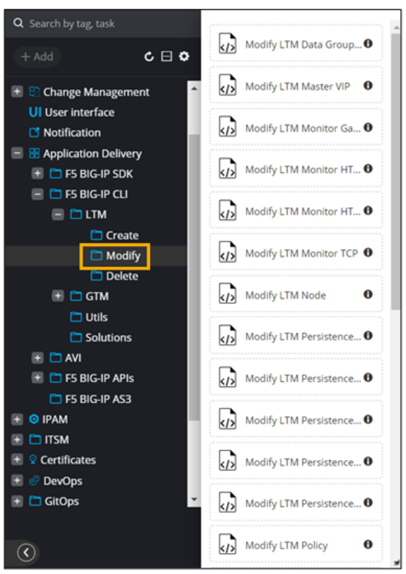
LTM - Create

The **Create** folder under **LTM** contains prebuilt tasks that can be leveraged to create LTM objects on F5 BIG-IP.



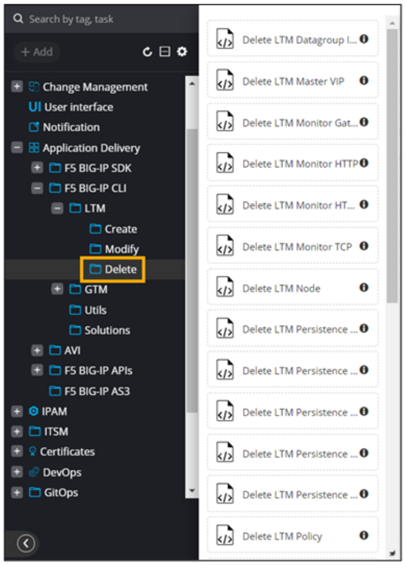
LTM - Modify

The **Modify** folder under **LTM** contains prebuilt tasks that can be leveraged to modify LTM objects on F5 BIG-IP.



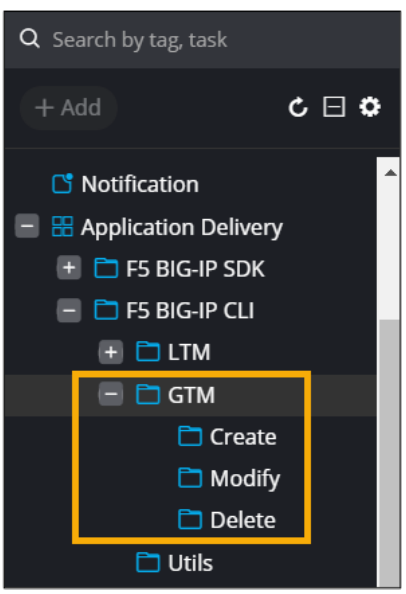
LTM - Delete

The **Delete** folder under **LTM** contains prebuilt tasks that can be leveraged to delete LTM objects on F5 BIG-IP.



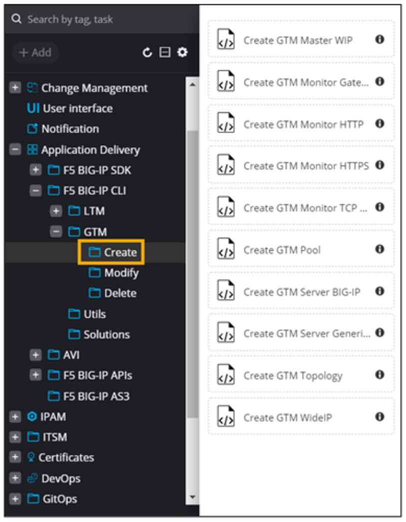
GTM

The workflows and tasks to create, modify, and delete GTM objects and devices on F5 BIG-IP are available under **Application Delivery** > **F5 BIG-IP CLI** > **GTM**.



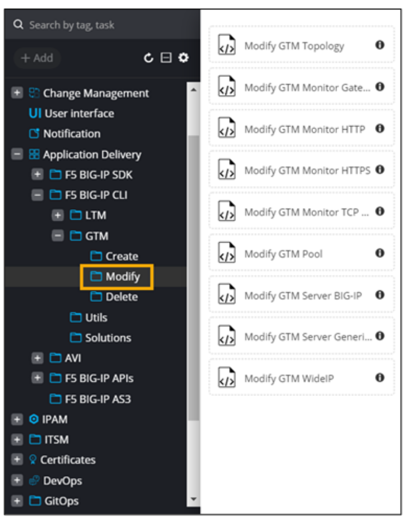
GTM - Create

The **Create** folder under **GTM** contains prebuilt tasks that can be leveraged to create GTM objects on F5 BIG-IP.



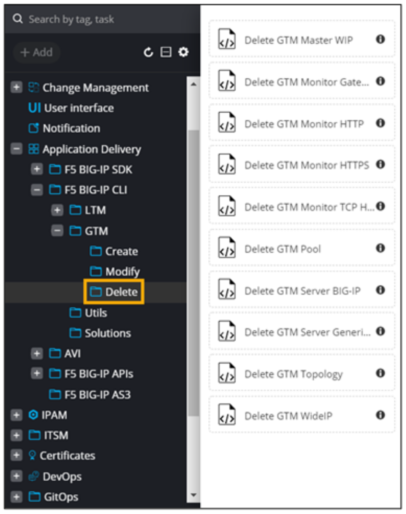
GTM - Modify

The **Modify** folder under **GTM** contains prebuilt tasks that can be leveraged to modify GTM objects on F5 BIG-IP.



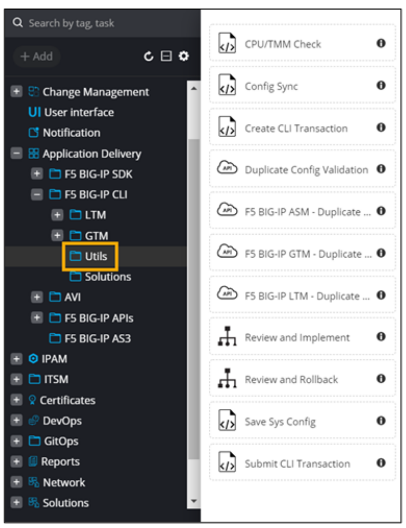
GTM - Delete

The **Delete** folder under **GTM** contains prebuilt tasks that can be leveraged to delete GTM objects on F5 BIG-IP.



Utils

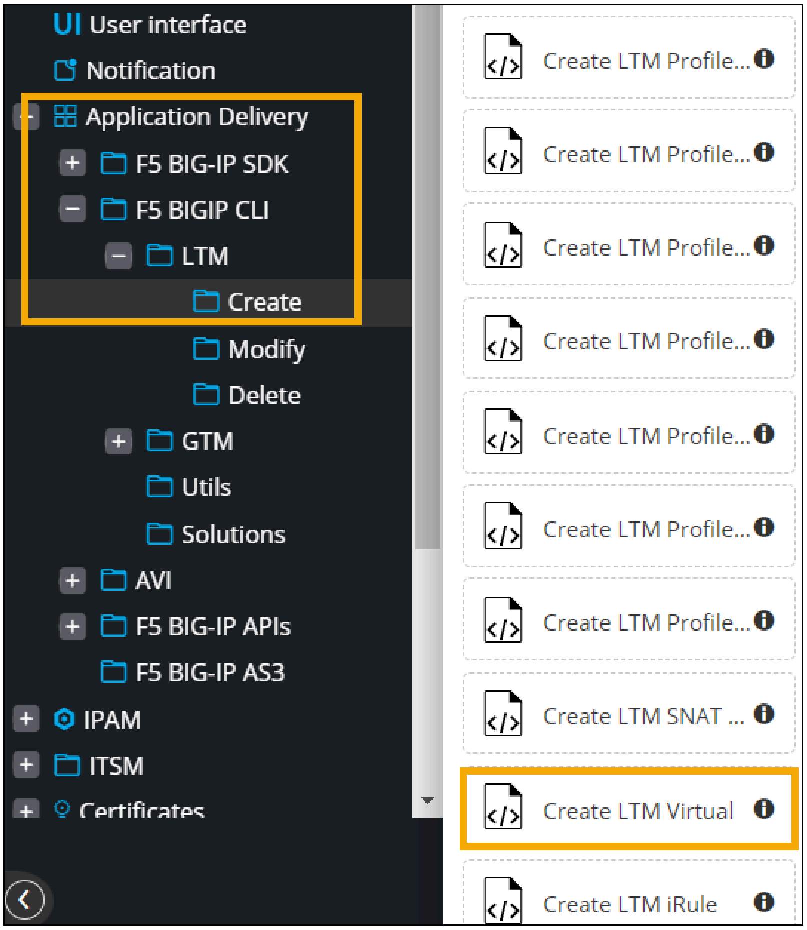
This folder contains common tasks and workflows that are used across workflows for **F5 BIG-IP** provisioning.



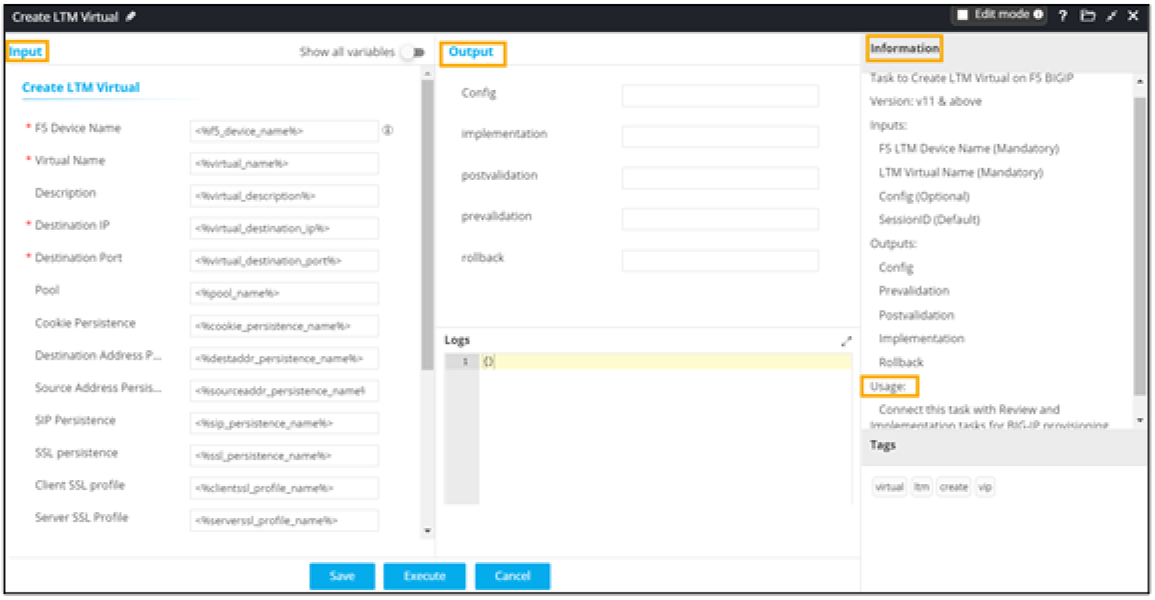
Creating a LTM Virtual Server on F5 BIG-IP

You can leverage the prebuilt LTM tasks to design a workflow for creating a LTM Virtual Server on F5 BIG-IP, add a CPU/TMM check, and also design a rollback workflow.

1. Design a new workflow.
2. In the design workspace, from the **Application Delivery** > **F5 BIG-IP CLI** > **LTM** > **Create** folder, drag and drop the **Create LTM Virtual** task.

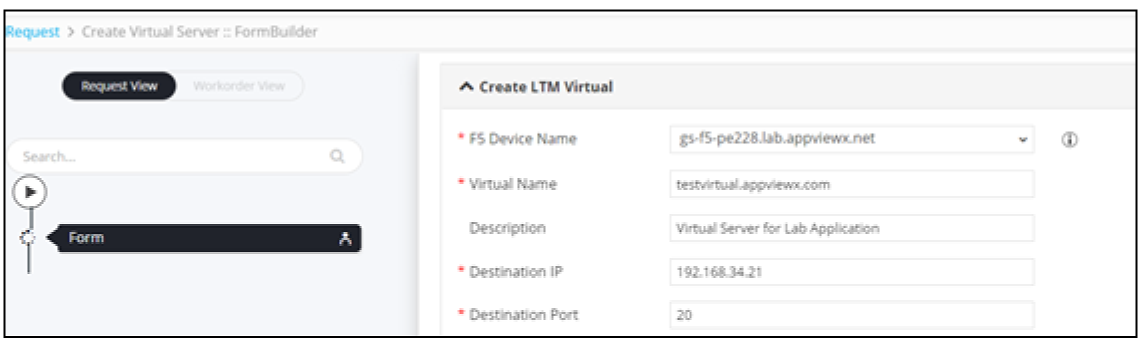


The **Create LTM Virtual** task window shows the defined **Input** and **Output** variables. The **Information** section shows what the task is for and also informs the user to connect this task with Review and Implementation tasks for BIG-IP provisioning.



3. Click **Save**.
4. From the **F5 BIG-IP CLI** > **Utils** folder, drag and drop the **Review and Implement** workflow.
5. Auto generate a self-service form for this workflow.
6. Connect the workflow tasks and enable the workflow.
7. Trigger the workflow.

   The workflow execution page is displayed with the workflow inputs requested at the first stage.
8. Enter the details in the input form.

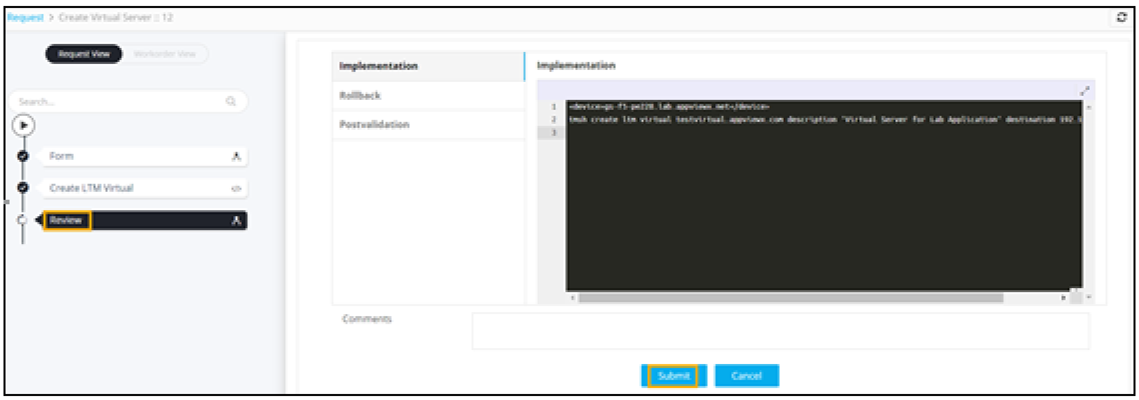


Note: Ensure that the device is available in the Device Inventory.

9. Click **Next**.
10. In the **Confirmation** pop-up window, click **Ok**.

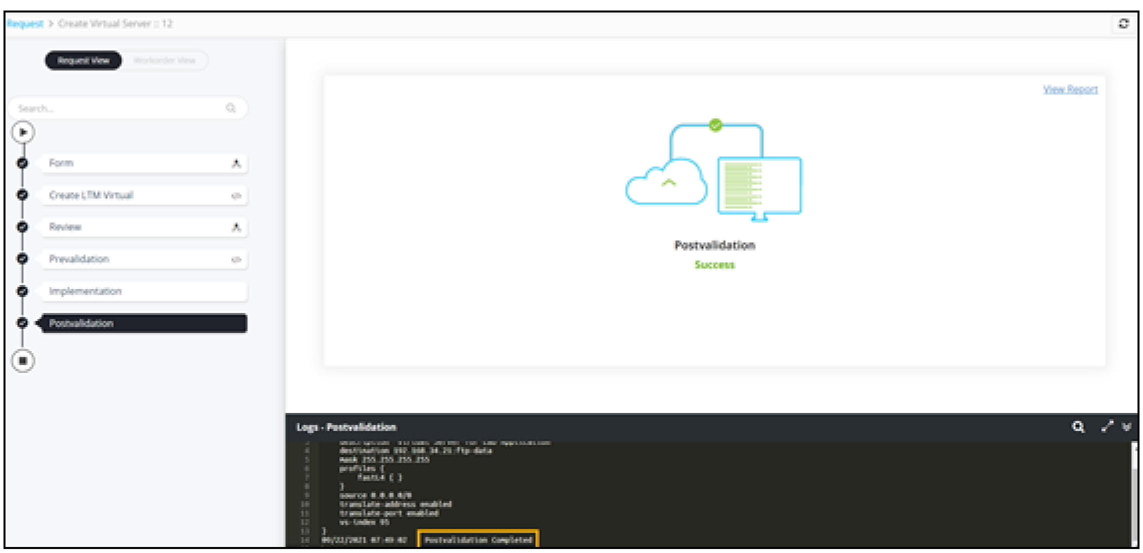
    The **Create LTM Virtual** task is successfully completed.

11. At the **Review** stage, click **Submit**.



12. In the **Confirmation** pop-up window, click **Ok**.

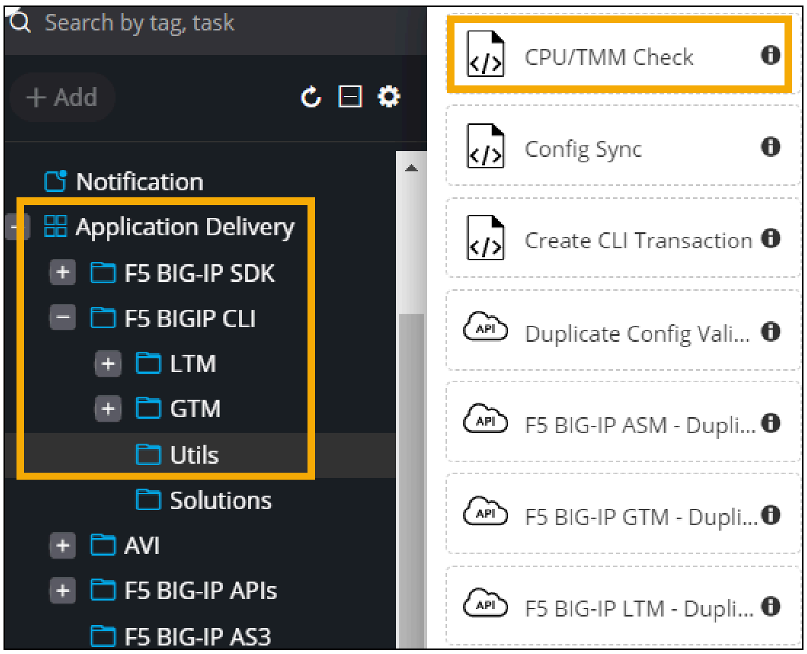
Workflow is successfully executed.



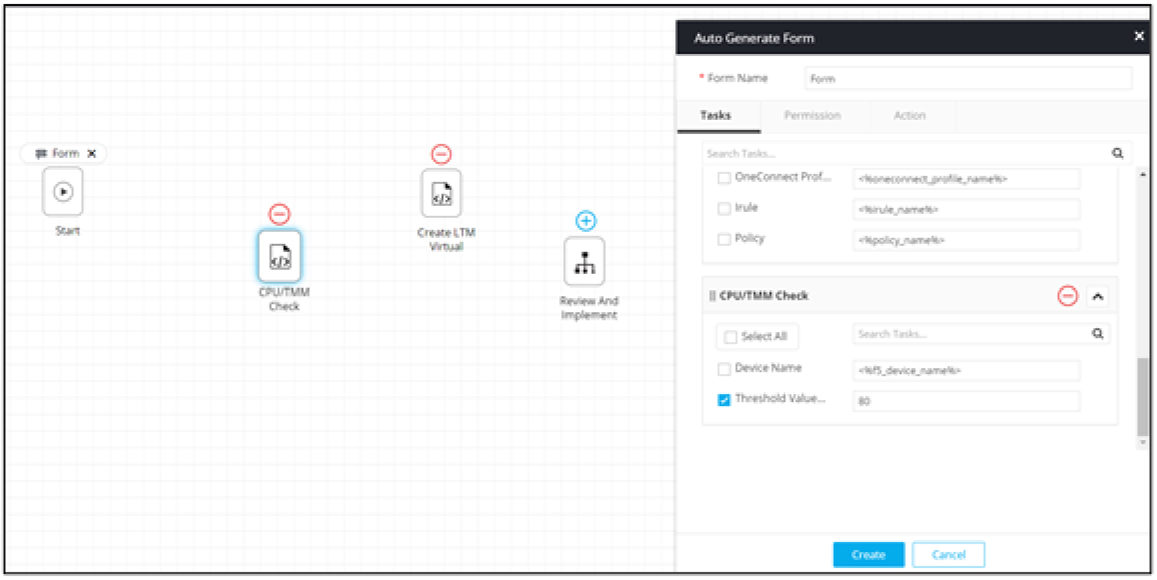
- Performing a CPU/TMM Check
- Adding a Rollback Workflow

Performing a CPU/TMM Check

1. Design a workflow to create a LTM virtual server on F5 BIG-IP.
2. In the design workspace, from the **Utils** folder, under **F5 BIG-IP CLI**, drag and drop the **CPU/TMM Check** task. This task performs a CPU/TMM check on F5 BIG-IP.

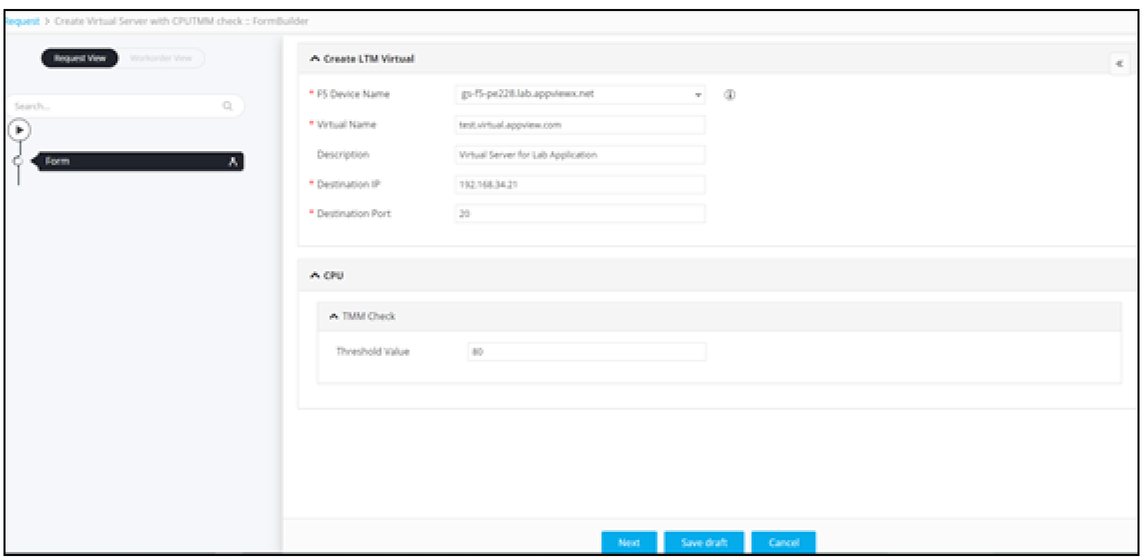


   The **CPU TMM Check** task opens in the Edit mode.
3. Click **Save**.
4. Auto-generate a self-service form for this workflow.



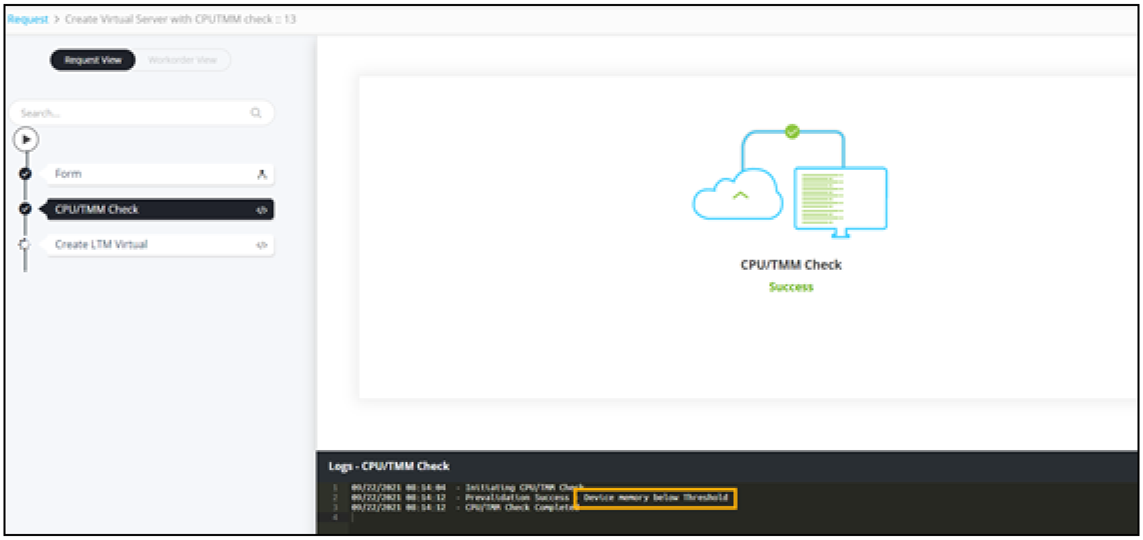
5. Connect the workflow tasks and enable the workflow.
6. Trigger the workflow.

   The workflow execution page is displayed with the workflow inputs requested at the first stage.
7. Enter the details in the input form.

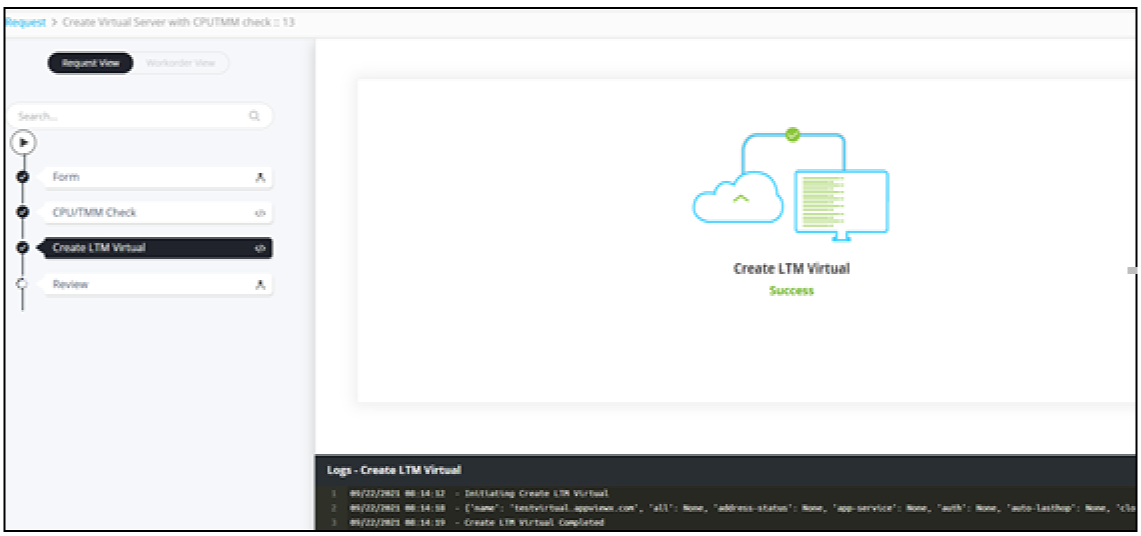


8. Click **Next**.

- **CPU/TMM Check** is successful - Device memory is below threshold.

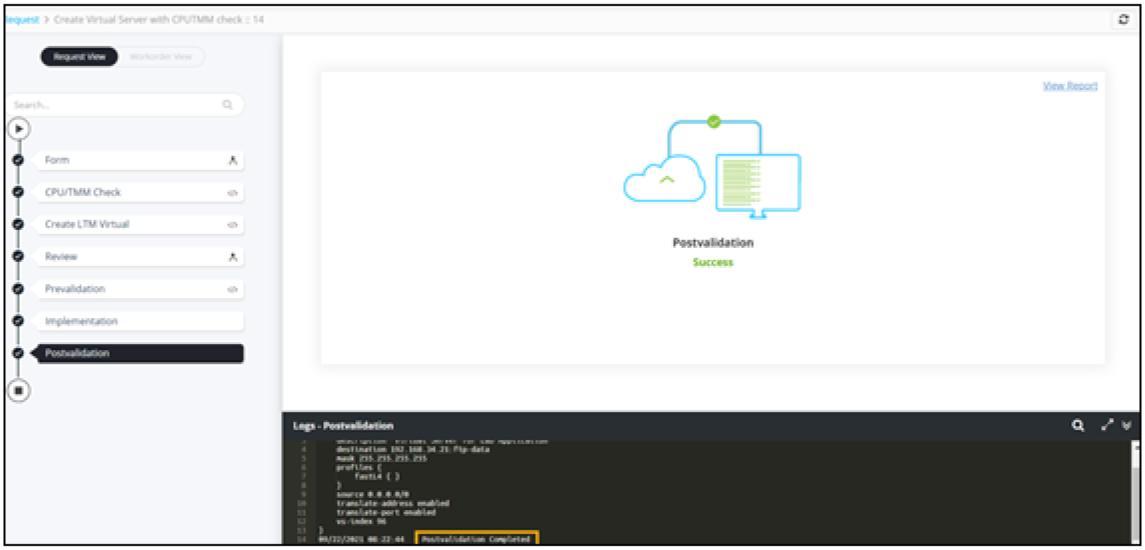


- **Create LTM Virtual** task is successfully completed.



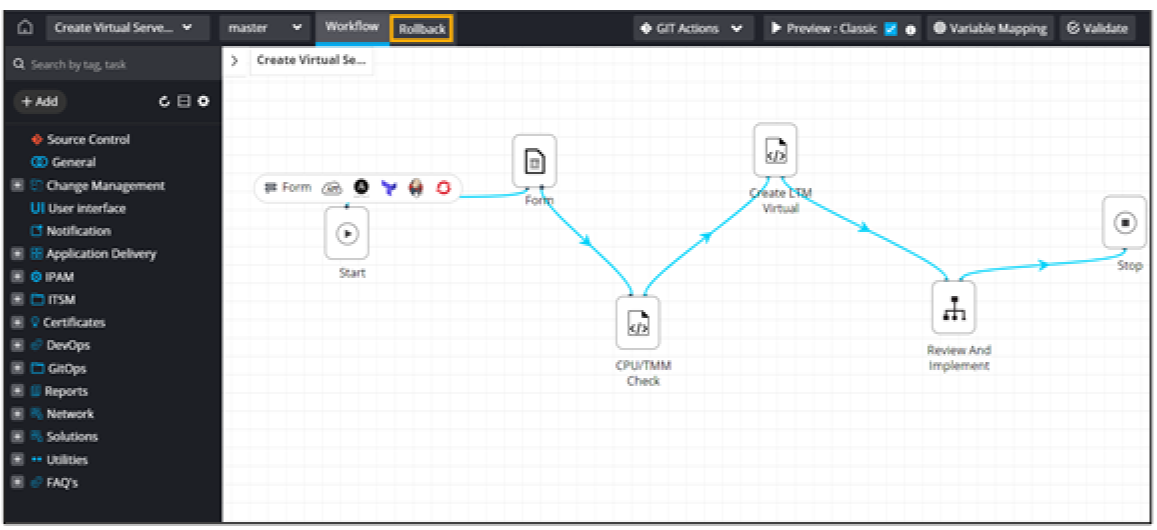
9. At the **Review** stage, click **Submit**.

The workflow is executed successfully.

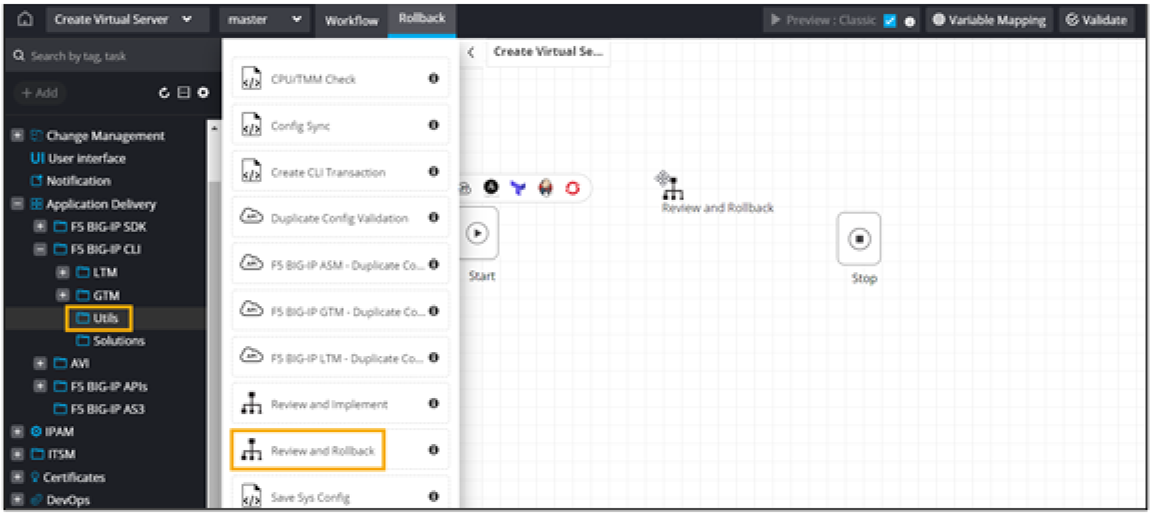


Adding a Rollback Workflow

1. Design a workflow to create a LTM virtual server on F5 BIG-IP with CPU/TMM check.
2. In the design workspace, from the menu on the top, click **Rollback**.

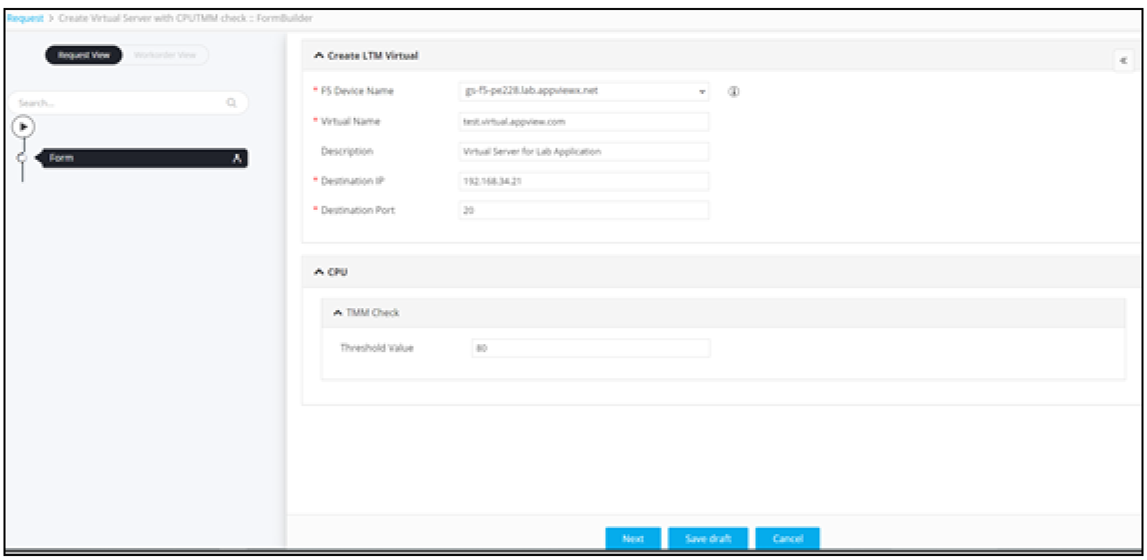


3. From the **Utils** folder in the left menu, drag and drop the **Review and Rollback** workflow.



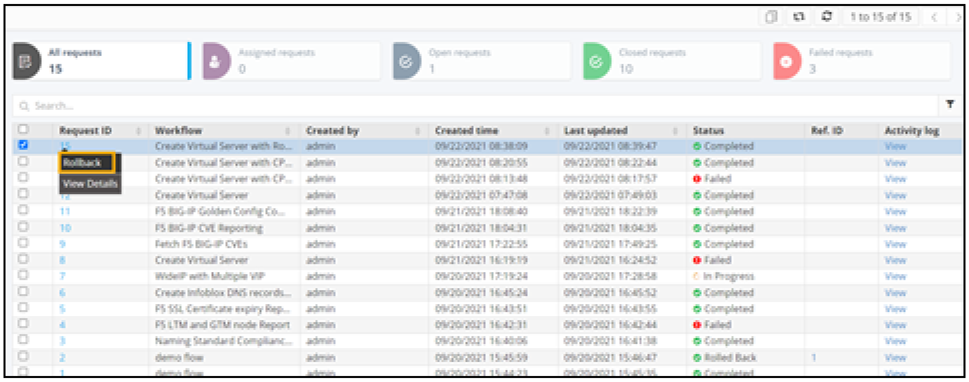
4. Connect the workflow tasks and enable the workflow.
5. Trigger the workflow.

   The workflow execution page is displayed with the workflow inputs requested at the first stage.
6. Enter the details in the input form.



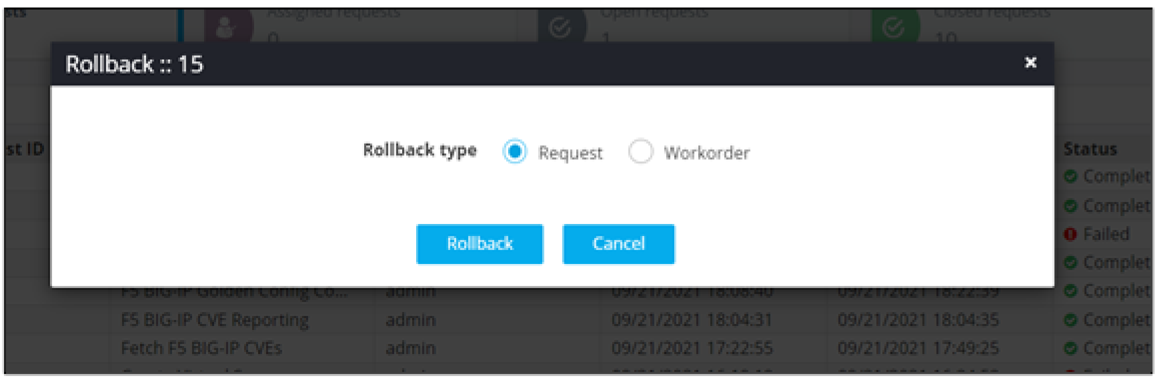
7. At the **Review** stage, click **Submit**.

   The workflow is executed successfully.
8. To trigger the rollback workflow, on the **Request :: All** page, right-click on the **Request ID** of the workflow.
9. From the options displayed, select **Rollback**.



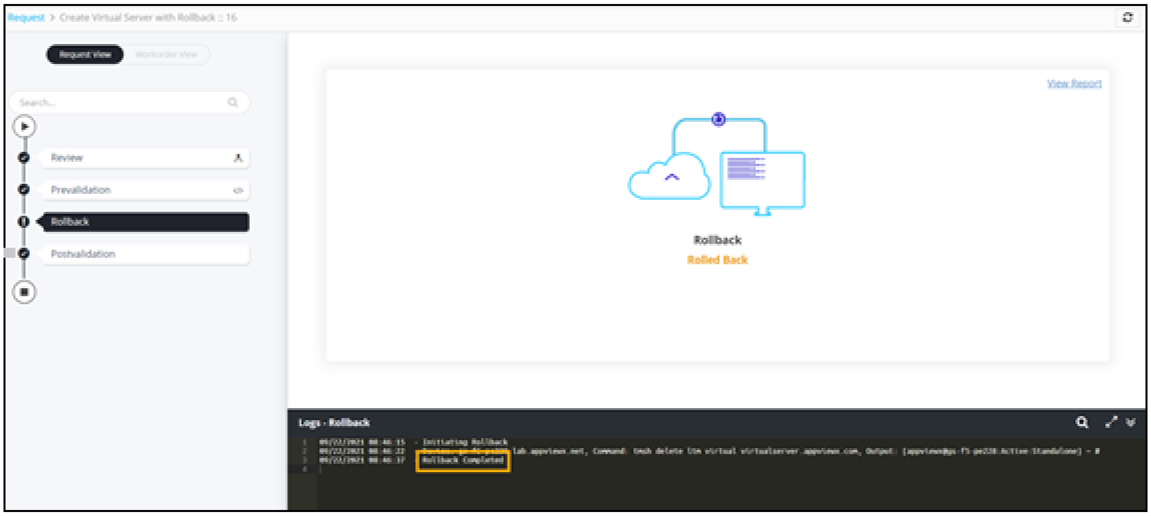
Tip: You can also select the workflow and click **(Rollback)** icon from the command bar in the top right corner of the screen.

10. In the **Rollback** pop-up window, click **Yes** and select **Rollback type** as **Request**.



11. Click **Rollback**.
12. At the **Review** stage, click **Submit**.

Rollback completed.



Related information
[Creating a LTM Virtual Server on F5 BIG-IP]

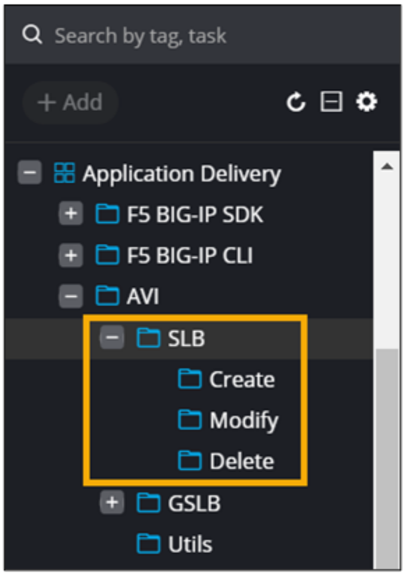
AVI Solutions

This section describes some of the workflows/tasks for AVI provisioning available in the Workflow Studio.

- SLB
- GSLB
- Utils

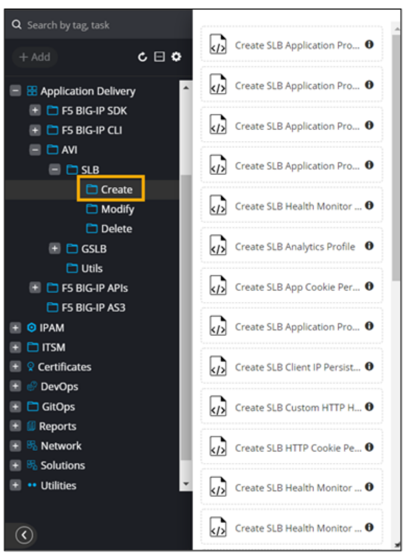
SLB

The workflows and tasks to create, modify, and delete SLB objects and devices on AVI are available under **Application Delivery** > **AVI** > **SLB**.



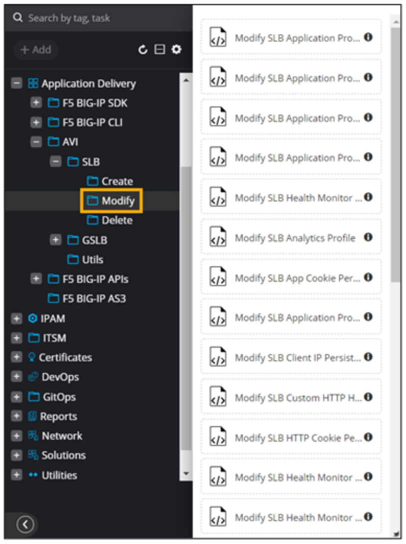
SLB - Create

The **Create** folder under **SLB** contains prebuilt tasks that can be leveraged to create SLB objects on AVI.



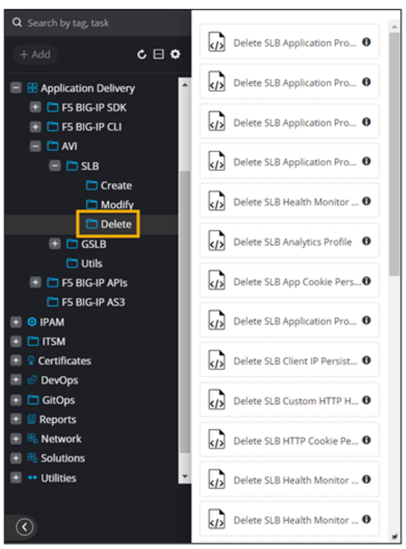
SLB - Modify

The **Modify** folder under **SLB** contains prebuilt tasks that can be leveraged to modify SLB objects on AVI.



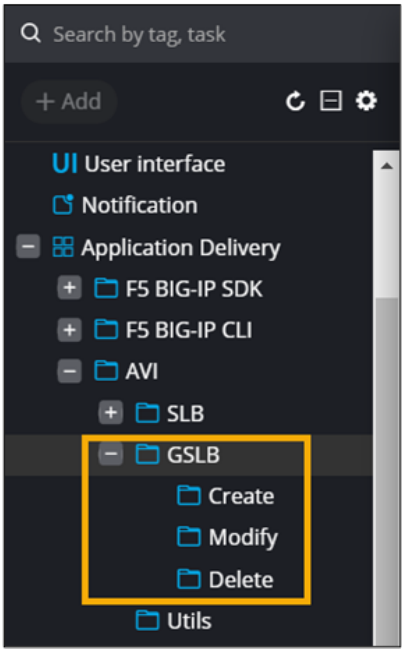
SLB - Delete

The **Delete** folder under **SLB** contains prebuilt tasks that can be leveraged to delete SLB objects on AVI.



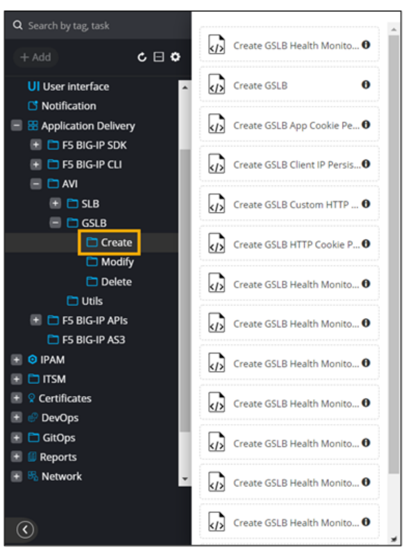
GSLB

The workflows and tasks to create, modify, and delete GSLB objects and devices on AVI are available under **Application Delivery** > **AVI** >**GSLB** GSLB.



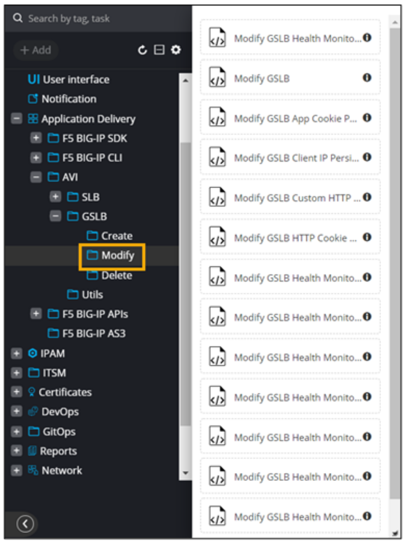
GSLB - Create

The **Create** folder under **GSLB** contains prebuilt tasks that can be leveraged to create GSLB objects on AVI.



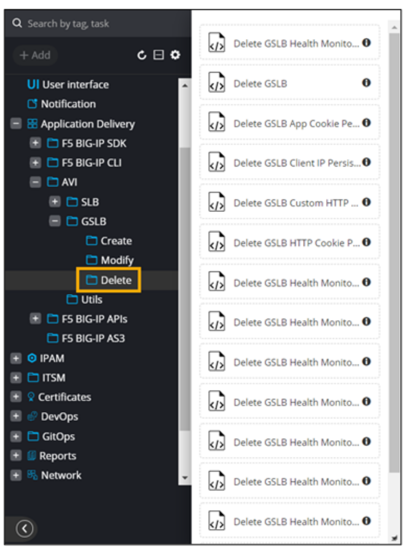
GSLB - Modify

The **Modify** folder under **GSLB** contains prebuilt tasks that can be leveraged to modify GSLB objects on AVI.



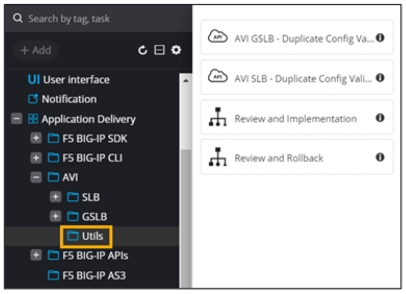
GSLB - Delete

The **Delete** folder under **GSLB** contains prebuilt tasks that can be leveraged to delete GSLB objects on AVI.



Utils

This folder contains common tasks and workflows that are used across workflows for **AVI** provisioning.



Automate DNS Services

You can accelerate application delivery with prebuilt DNS automation tasks and solutions by integrating with IPAM tools such as Infoblox and Bluecat to fetch and reserve free IP addresses automatically when creating virtual servers.

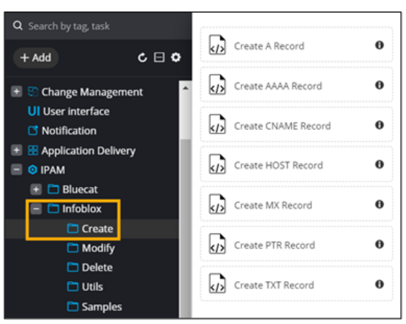
- Infoblox Solutions
- Bluecat Solutions

Infoblox Solutions

This section describes the tasks/workflows that can be leveraged to create, modify, and delete DNS records on Infoblox.

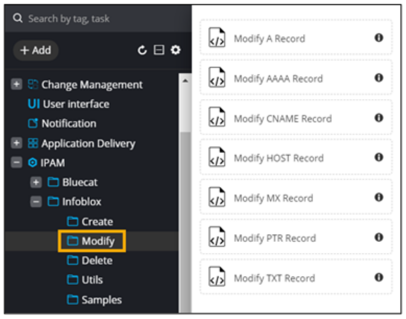
Create

The **Create** folder under **Infoblox** contains prebuilt tasks that can be leveraged to create DNS records such as A Record, CNAME record, and PTR Record on Infoblox.



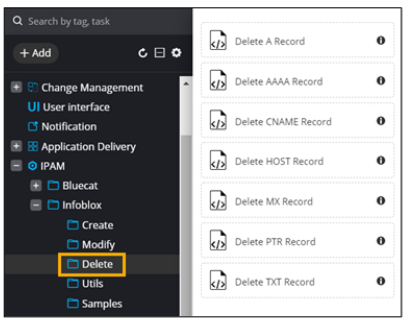
Modify

The **Modify** folder under **Infoblox** contains prebuilt tasks that can be leveraged to modify DNS records such as A Record, CNAME record, and PTR Record on Infoblox.



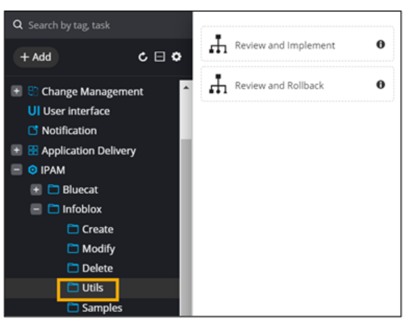
Delete

The **Delete** folder under **Infoblox** contains prebuilt tasks that can be leveraged to delete DNS records such as A Record, CNAME record, and PTR Record on Infoblox.



Utils

This folder contains common workflows that are used across workflows for **Infoblox** provisioning.

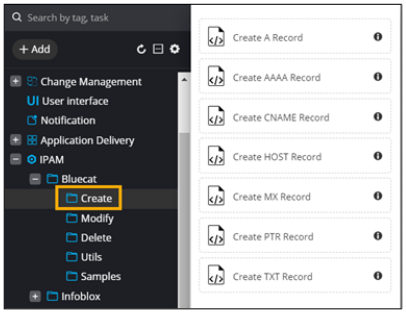


Bluecat Solutions

This section describes the tasks/workflows that can be leveraged to create, modify, and delete DNS records in Bluecat.

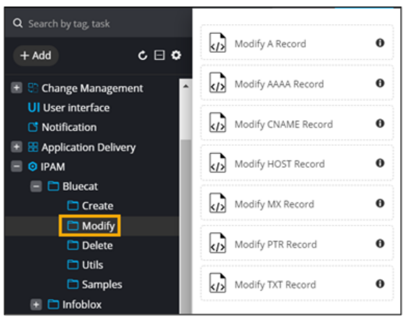
Create

The **Create** folder under **Bluecat** contains prebuilt tasks that can be leveraged to create DNS records such as A Record, CNAME record, and PTR Record on Bluecat.



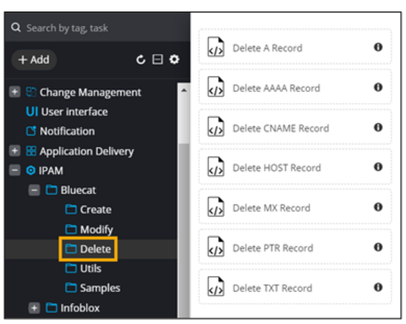
Modify

The **Modify** folder under **Bluecat** contains prebuilt tasks that can be leveraged to modify DNS records such as A Record, CNAME record, and PTR Record on Bluecat.



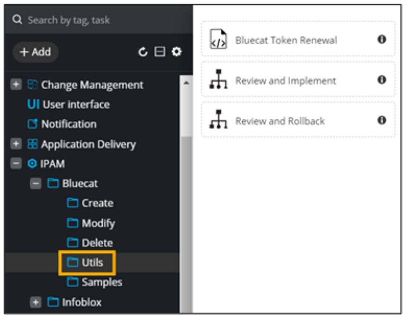
Delete

The **Delete** folder under **Bluecat** contains prebuilt tasks that can be leveraged to delete DNS records such as A Record, CNAME record, and PTR Record on Bluecat.



Utils

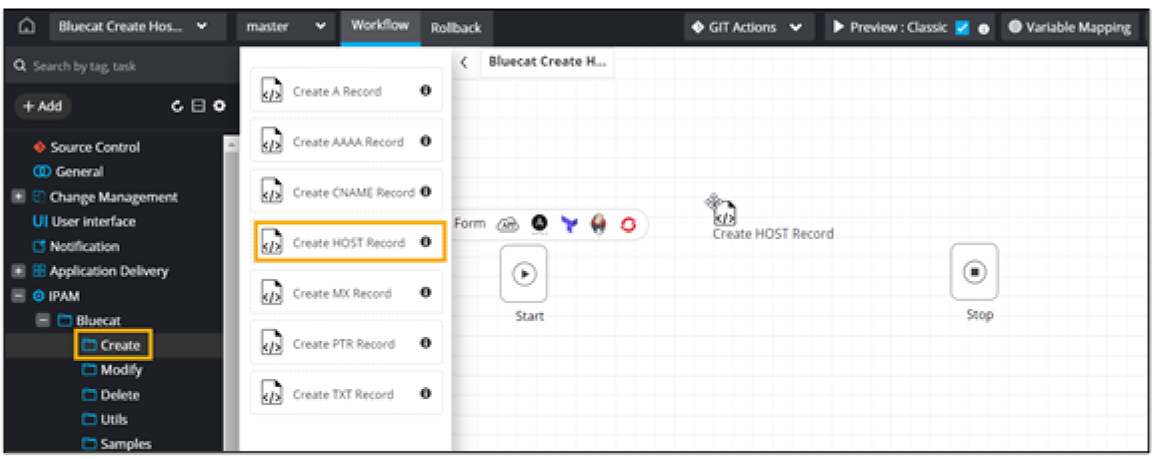
This folder contains common tasks and workflows that are used across workflows for **Bluecat** provisioning.



- Creating a HOST Record in Bluecat

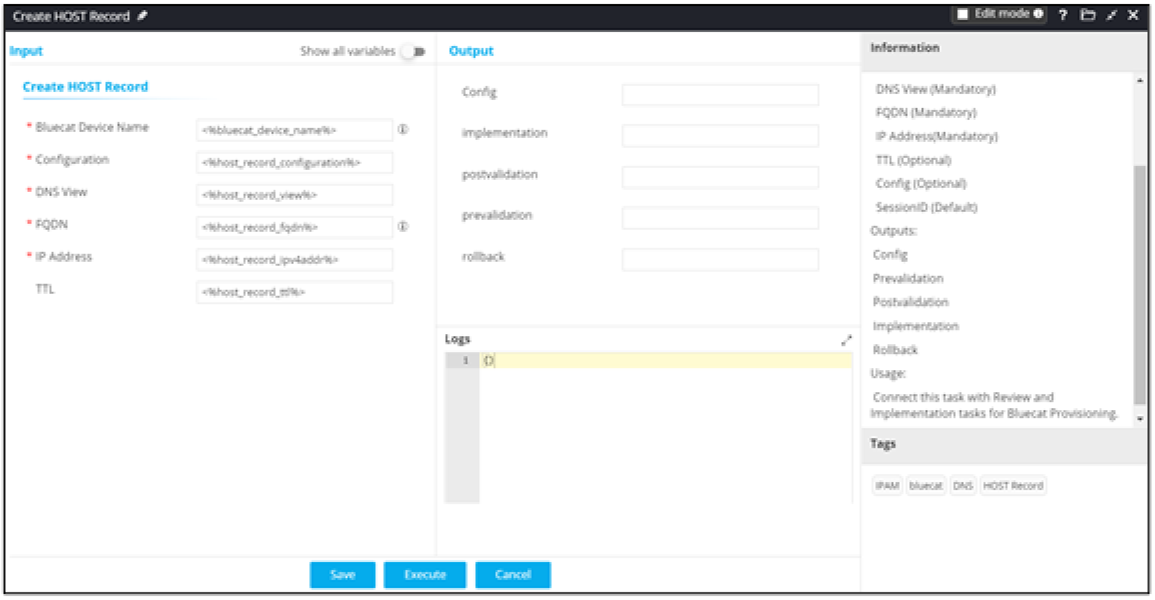
Creating a HOST Record in Bluecat

1. Design a new workflow.
2. In the design workspace, from **IPAM** > **Bluecat** > **Create** folder, drag and drop the **Create HOST Record** task.

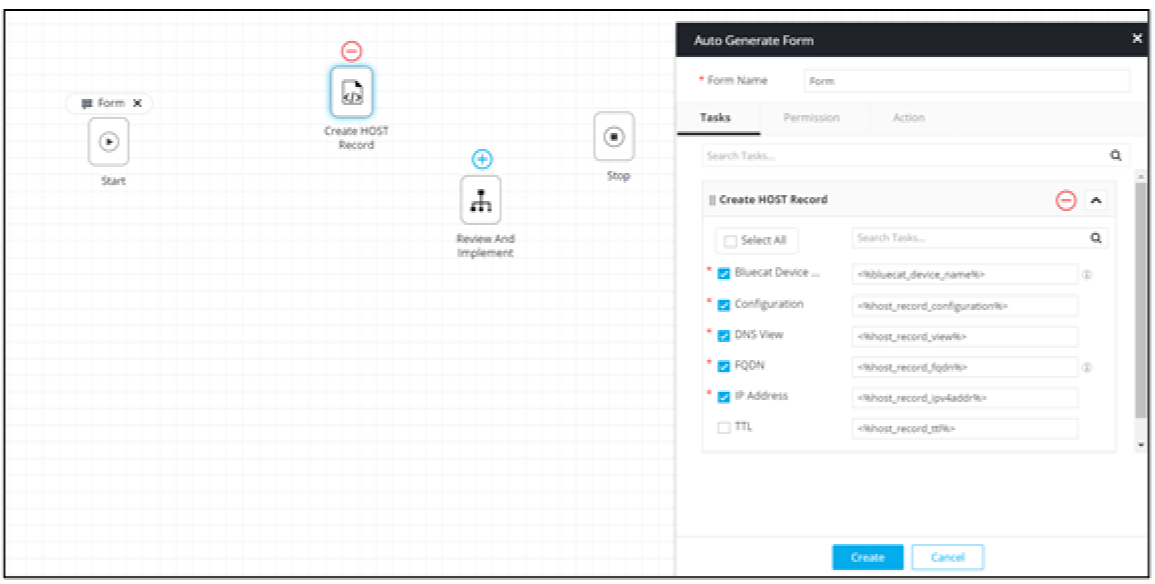


.

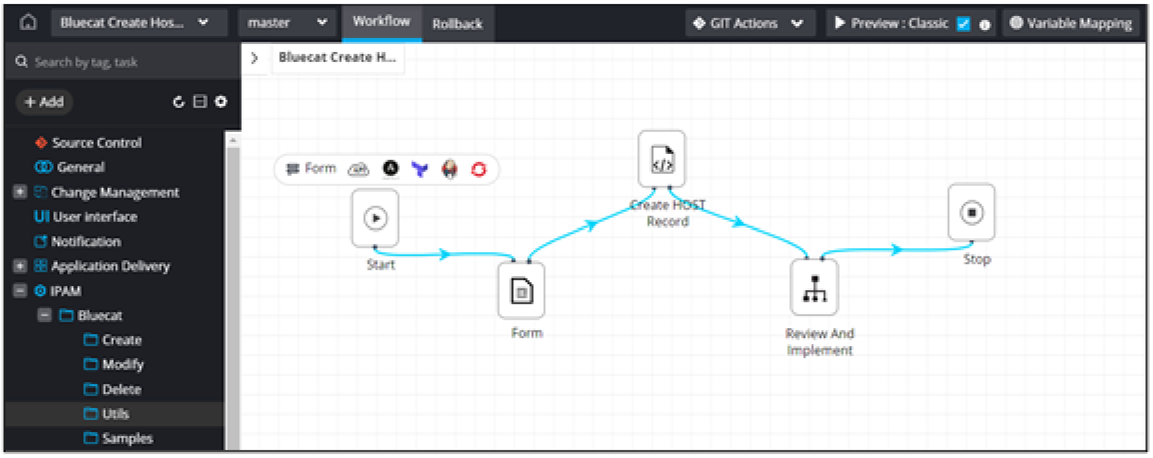
> **Note:** The **Create HOST Record** task window shows the defined **Input** and **Output** variables. The **Information** section shows what the task is for and also informs the user to connect this task with Review and Implementation tasks for Bluecat provisioning.



3. Click **Save**.
4. From the **Utils** folder in the left menu, under **Bluecat**, drag and drop the **Review and Implement** workflow.
5. Auto-generate a self-service form for this workflow.



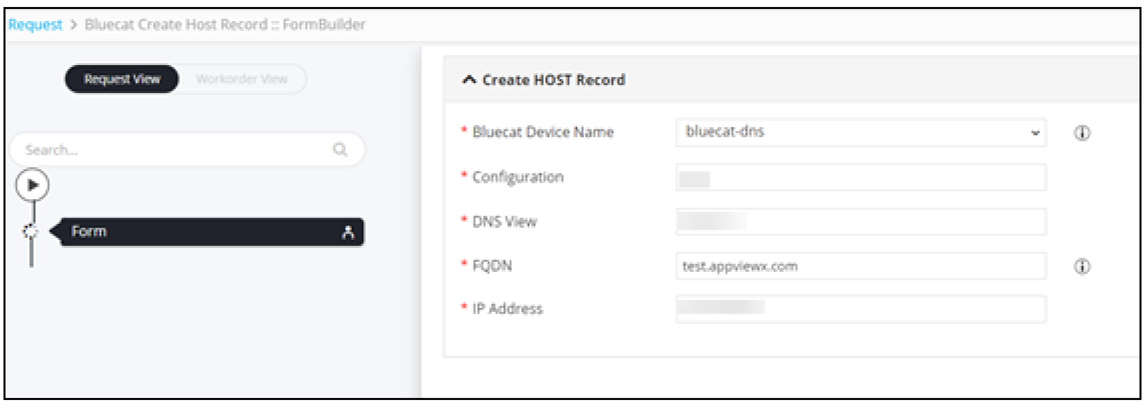
6. Connect the workflow tasks and enable the workflow.



7. Trigger the workflow.

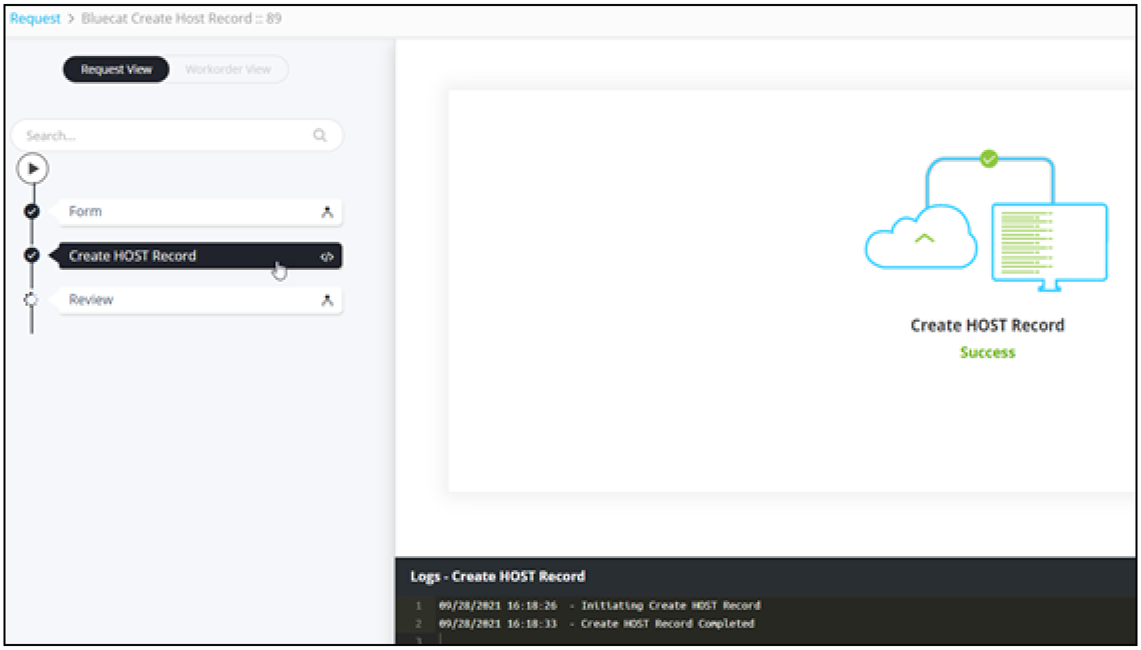
   The workflow execution page is displayed with the workflow inputs requested at the first stage.

8. Enter the details in the input form.



9. Click **Next**.

   The **HOST Record** is created.



10. At the **Review** stage, click **Submit**.

    The workflow is executed successfully.

Change Automation

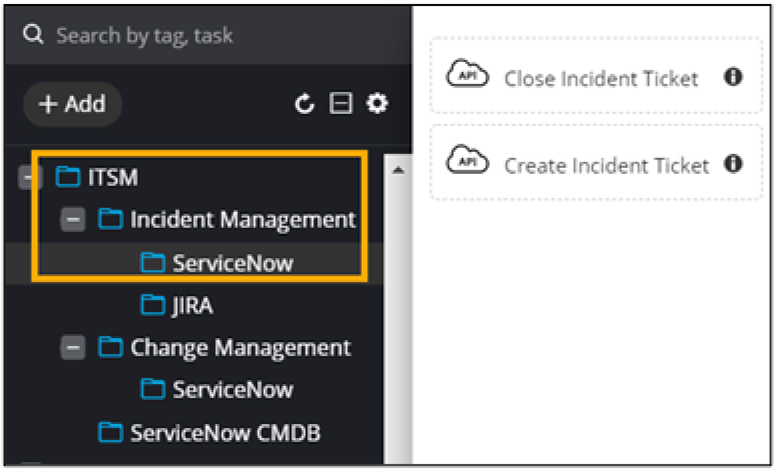
This folder consists of prebuilt tasks and workflows that allow you to integrate with ITSM tools. AppViewX supports the following vendors:

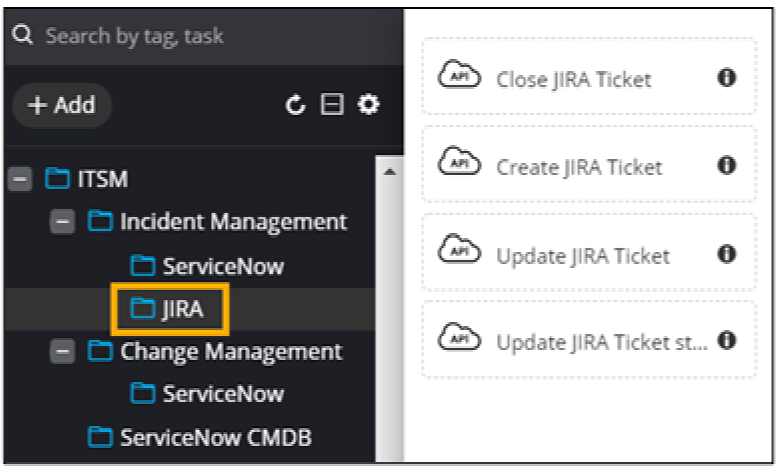
- ServiceNow
- Jira

- Incident Management
- Change Management
- ServiceNow CMDB

Incident Management

The tasks available in the Incident Management folder allow you to design workflows for creating, updating, and closing incident tickets on ServiceNow and Jira.

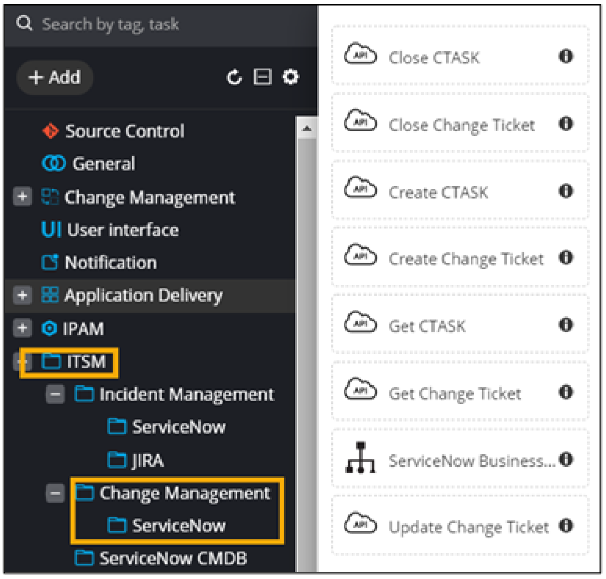




Note: For more information on using these tasks in a workflow, click here.

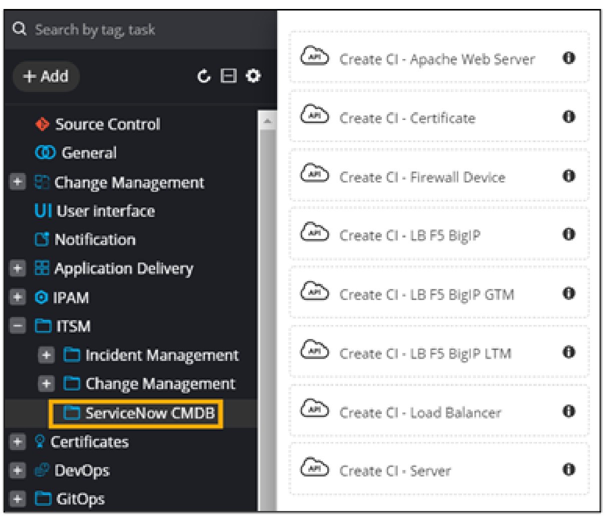
Change Management

The tasks available in the **Change Management** folder allow you to design workflows for creating, updating and closing change tickets on ServiceNow.



ServiceNow CMDB

This folder consists of tasks and workflows that can be leveraged to push different CIs (configuration items) such as certificates, devices, and firewalls to ServiceNow.



DevOps and Continuous Configuration Automation

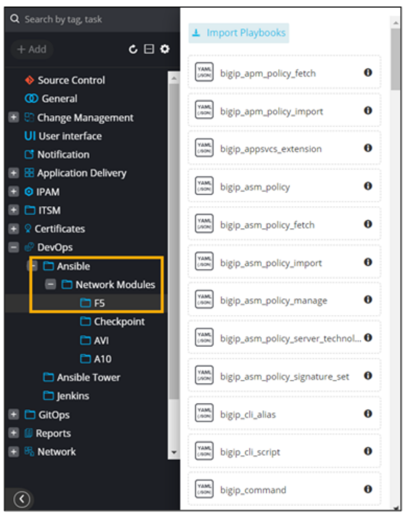
This folder consists of tasks and workflows that allow you to integrate with multiple vendors for network automation. AppViewX supports DevOps integration with Ansible and Jenkins.

- Ansible
- Ansible Tower
- Jenkins

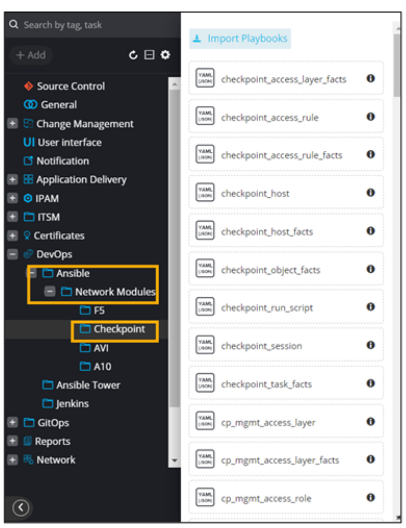
Ansible

This folder consists of the following Ansible Network Modules, which can be executed from playbooks:

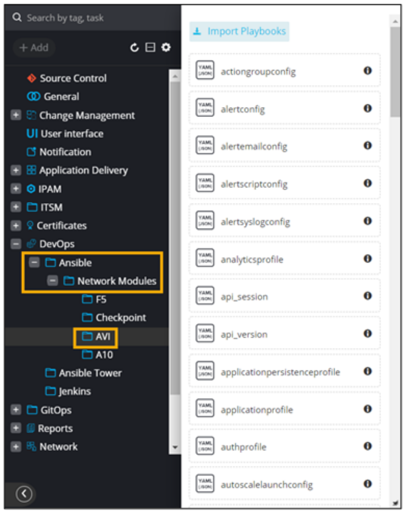
- **F5**



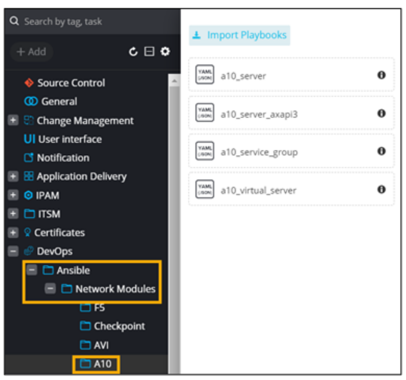
- **Checkpoint**



- **AVI**



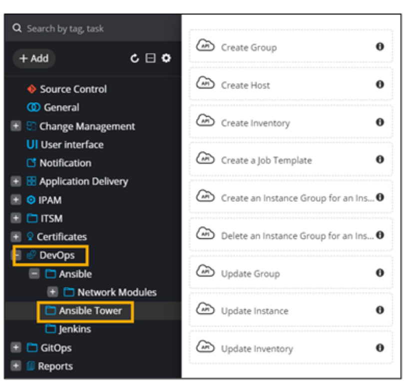
- **A10**



There is also a provision to import playbooks from Ansible.

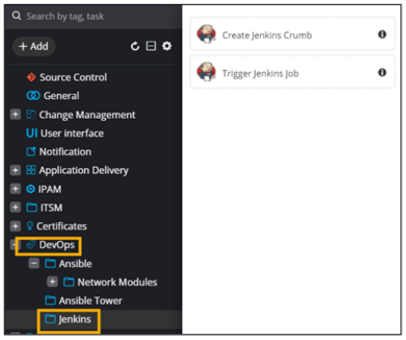
Ansible Tower

This folder consists of API tasks that can be used to create configurations on Ansible Tower and automate your tasks.



Jenkins

This folder contains tasks to allow you to create Jenkins jobs.

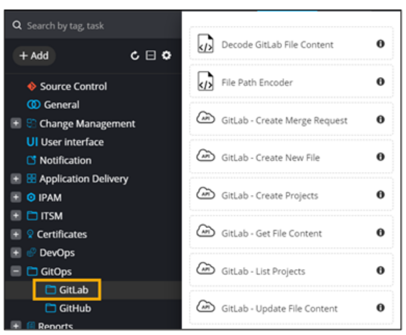


GitOps Integration

This folder consists of tasks that allow you to perform operations on GitHub and GitLab instances.

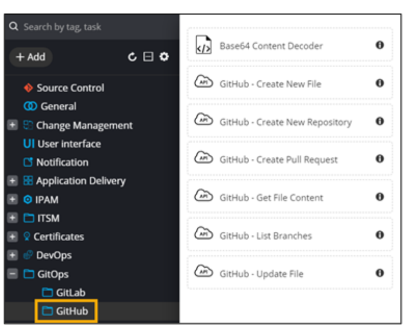
GitLab

The **GitLab** folder contains tasks that can be used to perform actions on a GitLab instance.



GitHub

The **GitHub** folder contains tasks that can be used to perform actions on a GitHub instance.

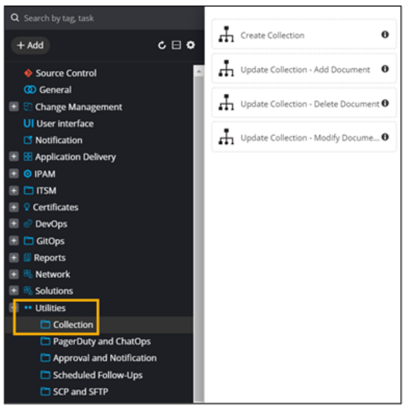


Utilities

This folder consists of prebuilt flows that are commonly used across workflows, such as for creating collections, approvals and notifications, follow-ups, and file transfers.

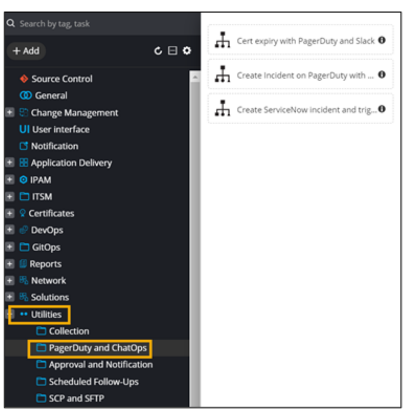
Collection

This folder contains prebuilt workflows for creating and updating collections with metadata. These workflows can be used to add, delete or modify documents within an existing collection.



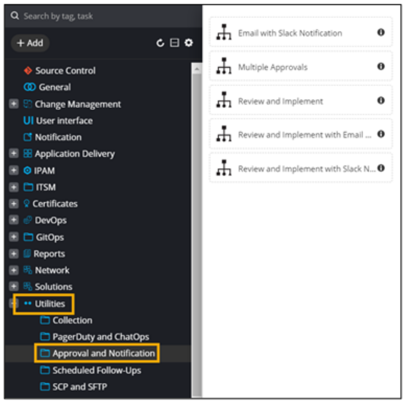
Pagerduty and ChatOps

This folder contains prebuilt workflows for triggering alerts and notifications on PagerDuty, Slack, and other ChatOps after task execution.



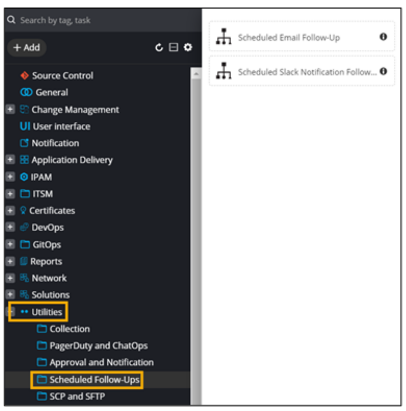
Approval and Notification

This folder contains prebuilt workflows for sending approvals and notifications via email. These workflows can be used for seeking multiple approvals for task execution (review and implementation) and sending email notifications once a task is complete.



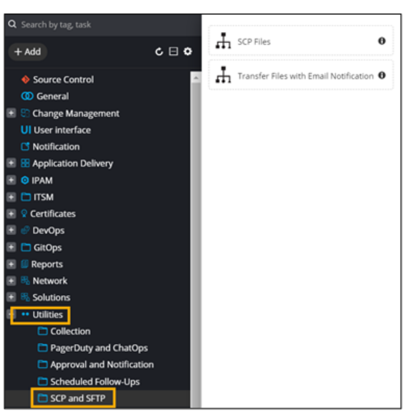
Scheduled Follow-Ups

This folder contains prebuilt workflows that check for pending approvals with scheduled email or Slack notifications.



SCP and SFTP

This folder contains prebuilt workflows for transferring files (SCP and SFTP) with or without email notifications.



SDK Automation

AppViewX's SDK Hub allows the user to import any Python SDK libraries, check for versions, and automate quickly across multiple vendors.

AppViewX SDKs are available for the following vendors:

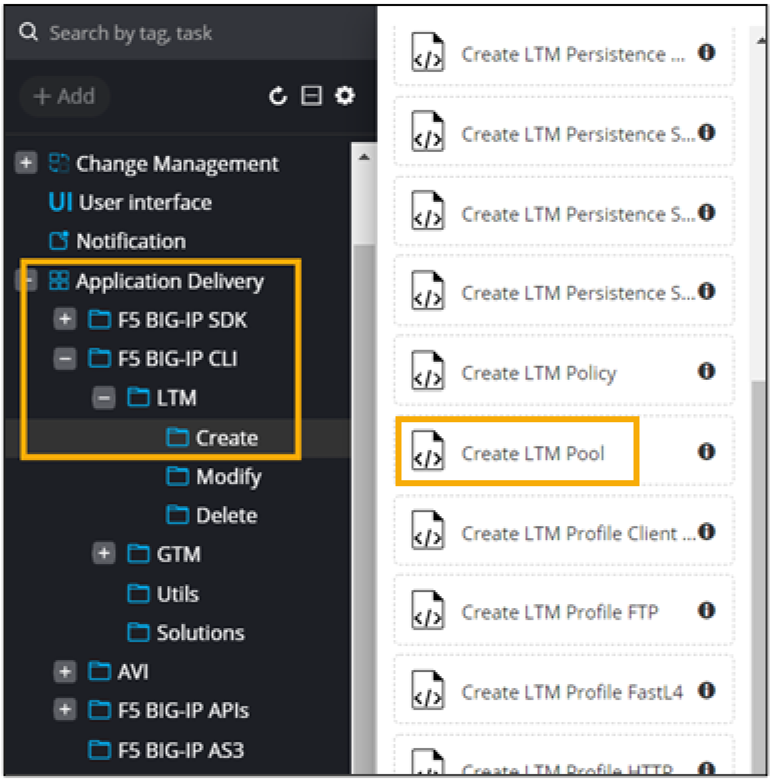
- AVI
- Bluecat
- F5
- Infoblox.

The pathway for the AppViewX folder is:

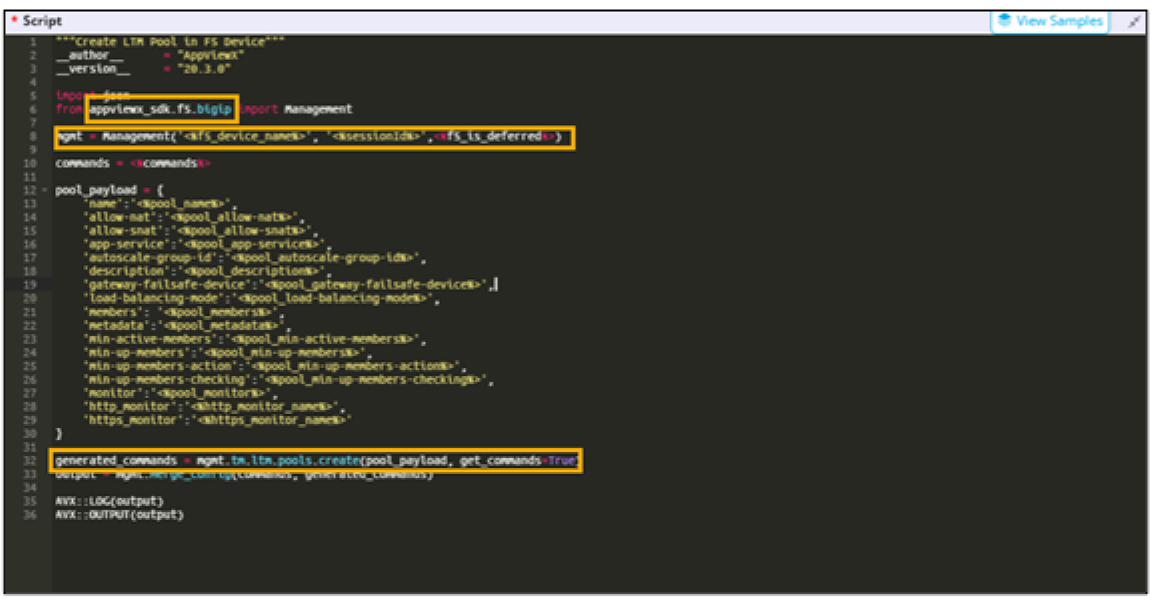
`“<Installation Directory>/appviewx-dependencies/vw/dependencies/appviewx_sdk”`

```
-bash-4.2$ pwd
/home/appviewx/appviewx/appviewx_dependencies/vw/dependencies
-bash-4.2$ ls -ltr | grep appviewx_sdk
drwxrwxr-x  8 appviewx appviewx      125 Jun 29 02:47 appviewx_sdk
-bash-4.2$ ls -ltr appviewx_sdk/
total 4
-rw-rw-r-- 1 appviewx appviewx 249 Jun  9 07:12 README.md
-rw-rw-r-- 1 appviewx appviewx   0 Jun  9 07:12 __init__.py
drwxrwxr-x 4 appviewx appviewx  51 Jun  9 07:12 f5
drwxrwxr-x 4 appviewx appviewx  88 Jun  9 07:12 bluecat
drwxrwxr-x 3 appviewx appviewx  81 Jun  9 07:12 avi
drwxr-xr-x 2 appviewx appviewx  37 Jun 29 02:47 __pycache__
drwxrwxr-x 3 appviewx appviewx 316 Jun 29 02:47 util
drwxrwxr-x 5 appviewx appviewx 107 Jul  7 05:25 infoblox
-bash-4.2$
```

1. Design a new workflow. a workflow.
2. In the design workspace, from the General section, select **Application Delivery** > **F5 BIG-IP CLI** > **LTM** > **Create**.
3. From the **Create** folder, drag and drop the **Create LTM Pool** task.



AppViewX SDK has been used in the script along with a command for creating an LTM pool.



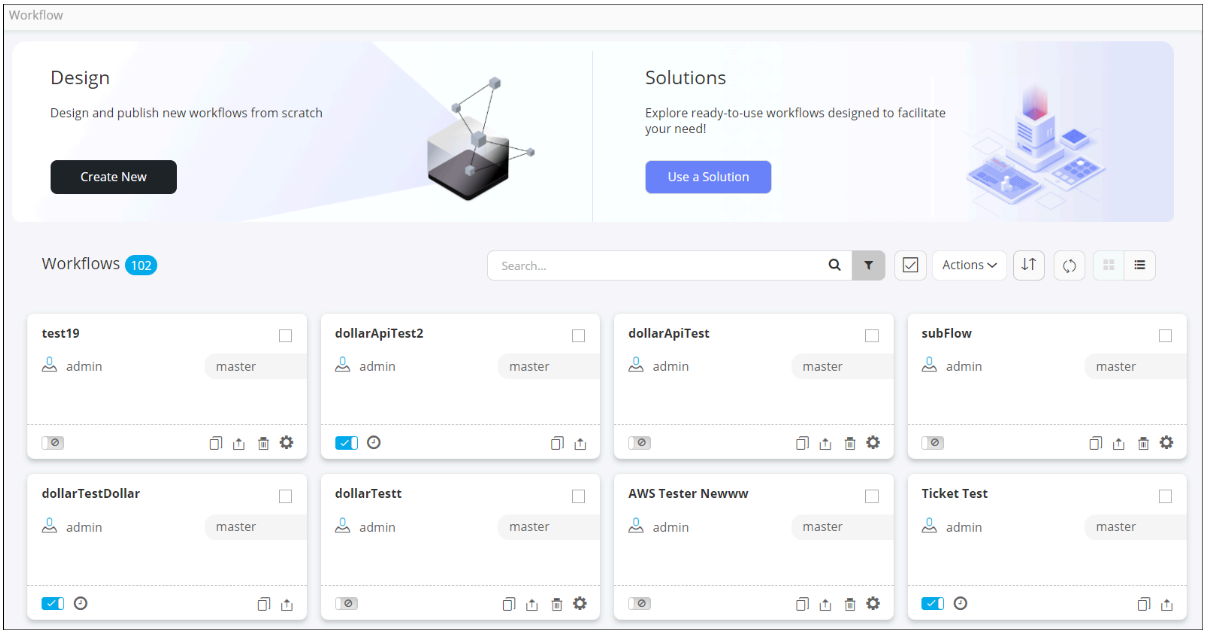
Note: For more information on out of the box (OOB) workflows, refer to the ADC+ Automation Workflow User Guide.

Workflow Studio

Accessing the Workflow Page

New Menu	Old Menu
1. From the main navigation menu, select **Automation**. 2. From the navigation pane on the left, under **WORKFLOW**, select **Design**. The **Workflow** page is displayed.	From the main navigation menu, select **Studio** > **Workflow**. The **Workflow** page is displayed.
Note: For more information on how to switch between menus, click here.	

The **Workflow** page gives a consolidated view of the important aspects of the Visual Workflow platform, and inventory of all the custom-built workflows, and the actions that can be performed from the Workflow studio console.



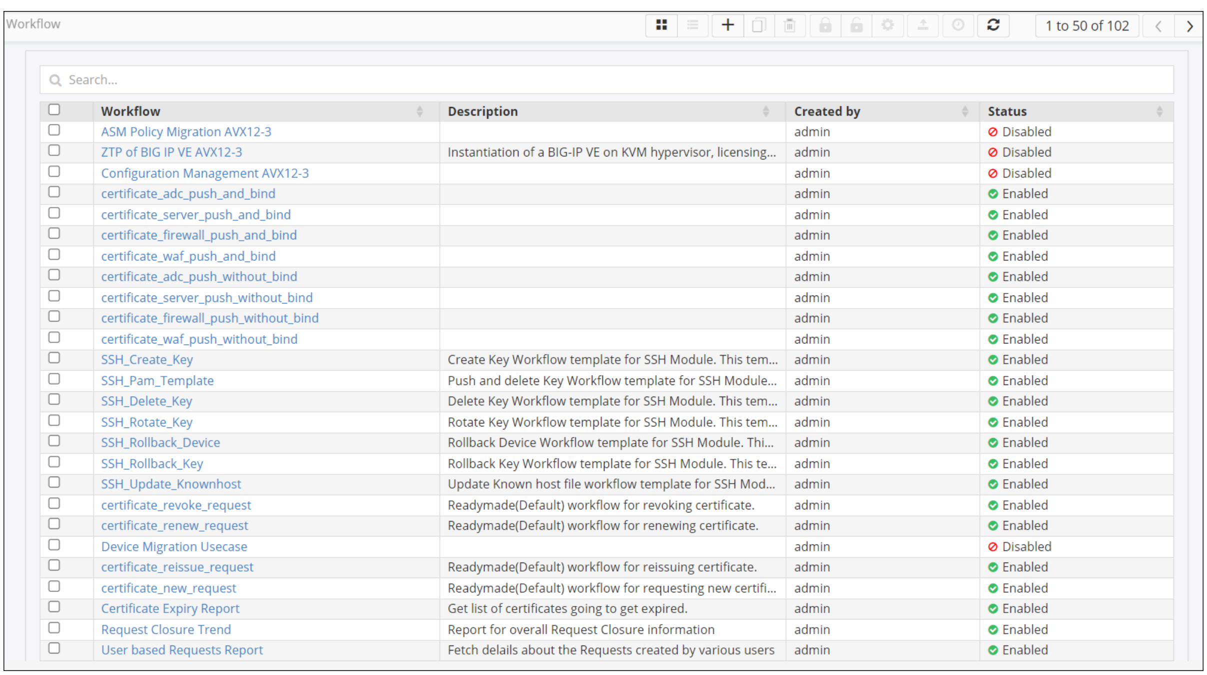
Various options available on the Workflow page when the workflows are in Card view

Option	Description
Design	Allows you to create a new workflow from scratch. **Note:** For more information, click here.
Solutions	Allows you to access the ready-to-use OOB Solutions and FAQs. **Note:** For more information, click here.
Search bar	Allows you to search for a particular workflow by typing in the keyword.
	Allows you to filter workflows based on the category selected in the dropdown.
	Allows you to select all workflows displayed on the page with a single click.
Actions	Allows you to perform the following actions on workflows: • **Export** • **Enable** • **Disable** • **Delete**

Various options available on the Workflow page when the workflows are in Card view (continued)

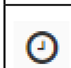
Option	Description
	Note: For more information, click here.
	Allows you to rearrange the workflows as: • Ascending • Descending • Recently created
	Refreshes the page.
	Displays the workflows in a card view.
	Displays the workflows in a list view.

Note: The **Workflow** inventory page is displayed in **Card view** by default. You can switch to the **List view** by clicking  **(List View)** icon.



	Workflow	Description	Created by	Status
☐	ASM Policy Migration AVX12-3		admin	Disabled
☐	ZTP of BIG IP VE AVX12-3	Instantiation of a BIG-IP VE on KVM hypervisor, licensing...	admin	Disabled
☐	Configuration Management AVX12-3		admin	Disabled
☐	certificate_adc_push_and_bind		admin	Enabled
☐	certificate_server_push_and_bind		admin	Enabled
☐	certificate_firewall_push_and_bind		admin	Enabled
☐	certificate_waf_push_and_bind		admin	Enabled
☐	certificate_adc_push_without_bind		admin	Enabled
☐	certificate_server_push_without_bind		admin	Enabled
☐	certificate_firewall_push_without_bind		admin	Enabled
☐	certificate_waf_push_without_bind		admin	Enabled
☐	SSH_Create_Key	Create Key Workflow template for SSH Module. This tem...	admin	Enabled
☐	SSH_Pam_Template	Push and delete Key Workflow template for SSH Module...	admin	Enabled
☐	SSH_Delete_Key	Delete Key Workflow template for SSH Module. This tem...	admin	Enabled
☐	SSH_Rotate_Key	Rotate Key Workflow template for SSH Module. This tem...	admin	Enabled
☐	SSH_Rollback_Device	Rollback Device Workflow template for SSH Module. Thi...	admin	Enabled
☐	SSH_Rollback_Key	Rollback Key Workflow template for SSH Module. This te...	admin	Enabled
☐	SSH_Update_Knownhost	Update Known host file workflow template for SSH Mod...	admin	Enabled
☐	certificate_revoke_request	Readymade(Default) workflow for revoking certificate.	admin	Enabled
☐	certificate_renew_request	Readymade(Default) workflow for renewing certificate.	admin	Enabled
☐	Device Migration Usecase		admin	Disabled
☐	certificate_reissue_request	Readymade(Default) workflow for reissuing certificate.	admin	Enabled
☐	certificate_new_request	Readymade(Default) workflow for requesting new certifi...	admin	Enabled
☐	Certificate Expiry Report	Get list of certificates going to get expired.	admin	Enabled
☐	Request Closure Trend	Report for overall Request Closure information	admin	Enabled
☐	User based Requests Report	Fetch delails about the Requests created by various users	admin	Enabled

Various options available on the Workflow page when the workflows are in List view

Option	Description
Search bar	Allows you to search for a particular workflow by typing in the keyword.
	Displays the workflows in a card view.
	Displays the workflows in a list view.
	Allows you create a new workflow from scratch. **Note:** For more information, click here.
	Allows you to clone a workflow. **Note:** For more information, click here.
	Allows you to delete selected workflow(s). **Note:** For more information, click here.
	Allows you to enable selected workflow(s). **Note:** For more information, click here.
	Allows you to disable selected workflow(s). **Note:** For more information, click here.
	Allows you to define RBAC and other settings for a particular workflow. **Note:** For more information, click here.
	Allows you to export selected workflow(s). **Note:** For more information, click here.
	Allows you to schedule the selected workflow.

Various options available on the Workflow page when the workflows are in List view (continued)

Option	Description
	Note: Only enabled workflows can be scheduled. For more information, click here.
	Refreshes the page.

Workflow Actions

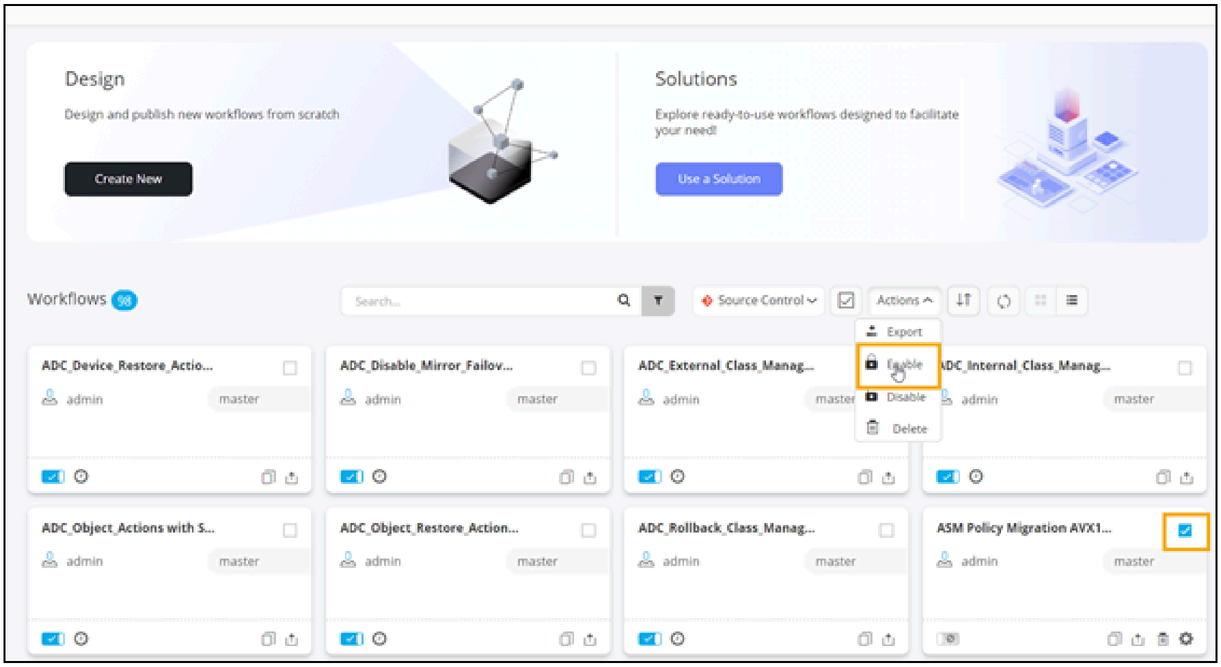
The **Actions** menu on the **Workflow** page (Card view) allows you to perform various actions on the workflows such as export, enable, disable, and delete.

> **Note:** In the **List view** of the Workflow inventory, these actions are available in the command bar on the top right corner of the page.

- Enabling a Workflow
- Disabling a Workflow
- Deleting a Workflow
- Exporting a Workflow
- Cloning a Workflow

Enabling a Workflow

1. Go to the Workflow page.
2. On the **Workflow** page (Card view), select the workflow to be enabled.
3. To enable the selected workflow, from the **Actions** dropdown menu, select **Enable**.



4. Click **Yes** in the **Confirmation** pop-up window.

   The workflow is enabled.

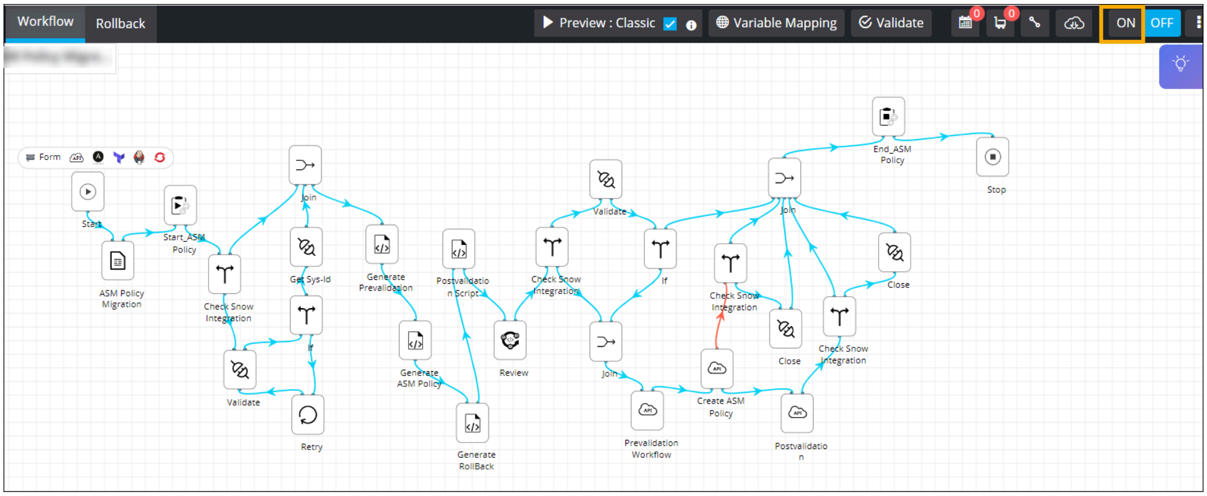
> **Tip:** You can also enable the workflow by turning on the toggle on the selected workflow card.

5. To enable multiple workflows, you can either individually select the workflows to be enabled or click **(Selection)** icon to select all workflows displayed on the page and click **Enable** in the **Actions** dropdown menu.
6. To enable a workflow, when the **Workflow** page is in **List view**, select the workflow(s) to be enabled and click **(Enable)** icon.

- Enabling a Workflow in Design Mode

Enabling a Workflow in Design Mode

1. In the workflow design space, from the right upper corner of the screen, click **ON**.



2. In the **Confirmation** pop-up window, click **Validate & Enable**.

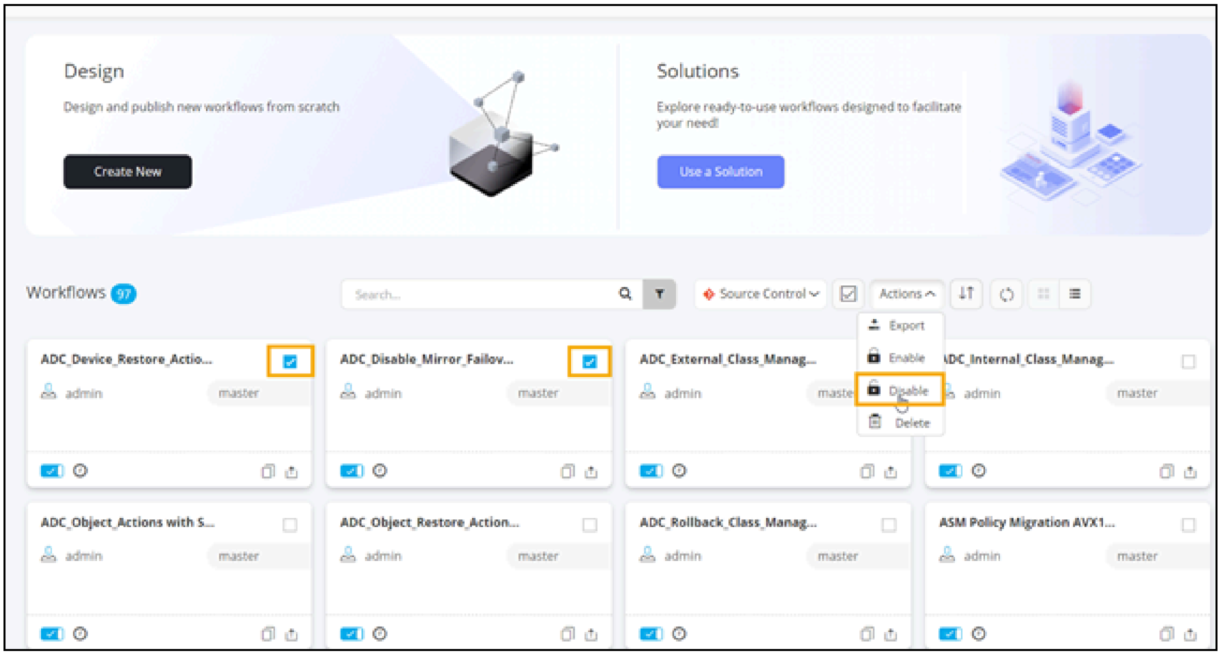
Note: For more information on validating a workflow, click here.

The workflow is enabled.

Important: You cannot modify a workflow when it is in enabled state.

Disabling a Workflow

1. Go to the Workflow page.
2. On the **Workflow** page (Card view), select the workflow to be disabled.
3. To disable the selected workflow, from the **Actions** dropdown menu, select **Disable**.

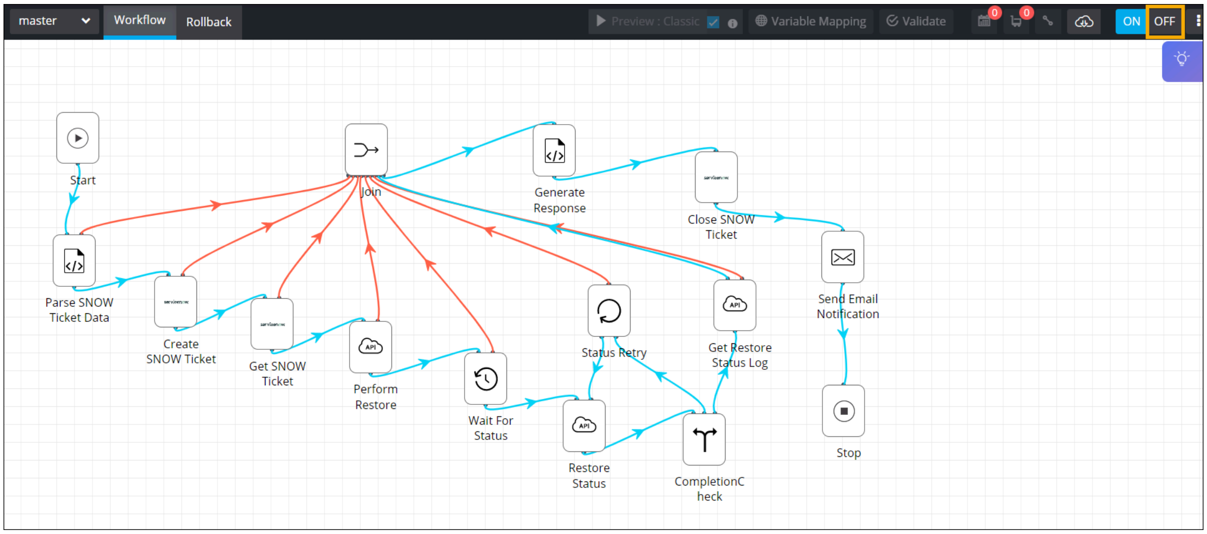


4. Click **Yes** in the **Confirmation** pop-up window.

   The workflow is disabled.

> **Tip:** You can also turn off the toggle on the workflow card to disable the workflow.

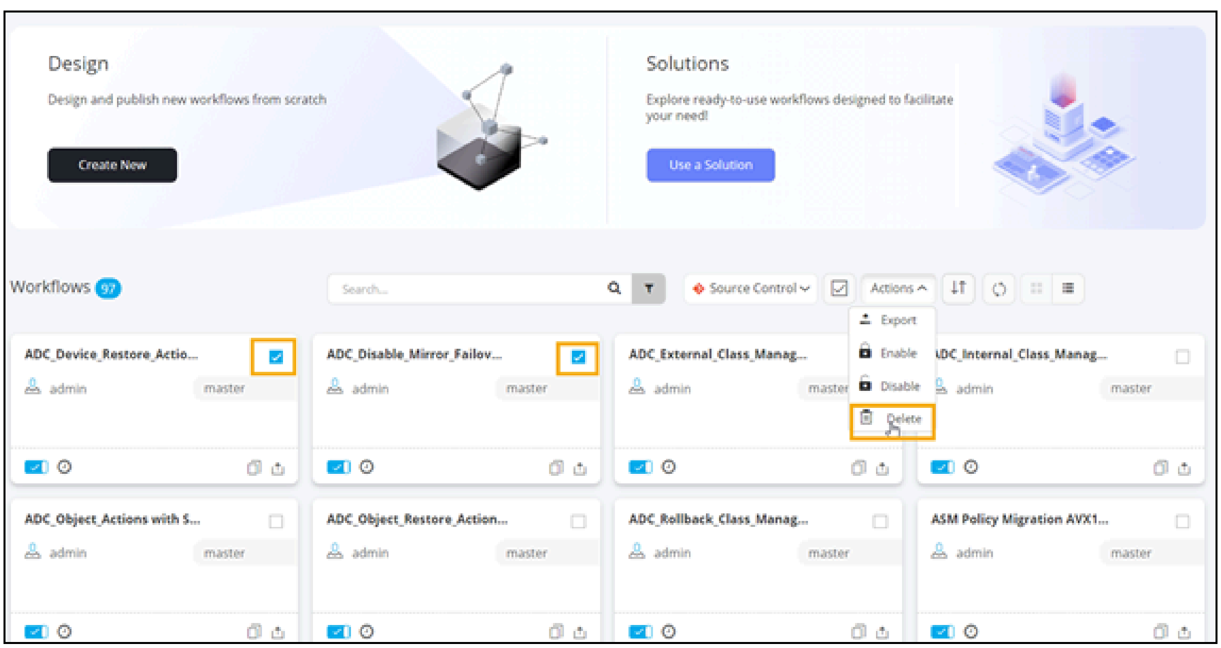
5. To disable multiple workflows, you can either select the workflows to be disabled individually or click  **(Selection)** icon to select all workflows displayed on the page and click **Disable** in the **Actions** dropdown menu.
6. To disable a workflow in the design space, from the upper right corner of the screen, click **OFF**.



7. To disable a workflow when the **Workflow** page is in **List view**, select the workflow(s) and click  **(Disable)** icon.

Deleting a Workflow

1. Go to the Workflow page.
2. On the **Workflow** page (Card view), disable the workflow to be deleted.
3. Select the workflow.
4. To delete the workflow, from the **Actions** dropdown menu, select **Delete**.



5. To delete multiple workflows, either select them individually or click **(Selection)** icon to select all workflows displayed on the page and click **Delete** in the **Actions** dropdown menu.

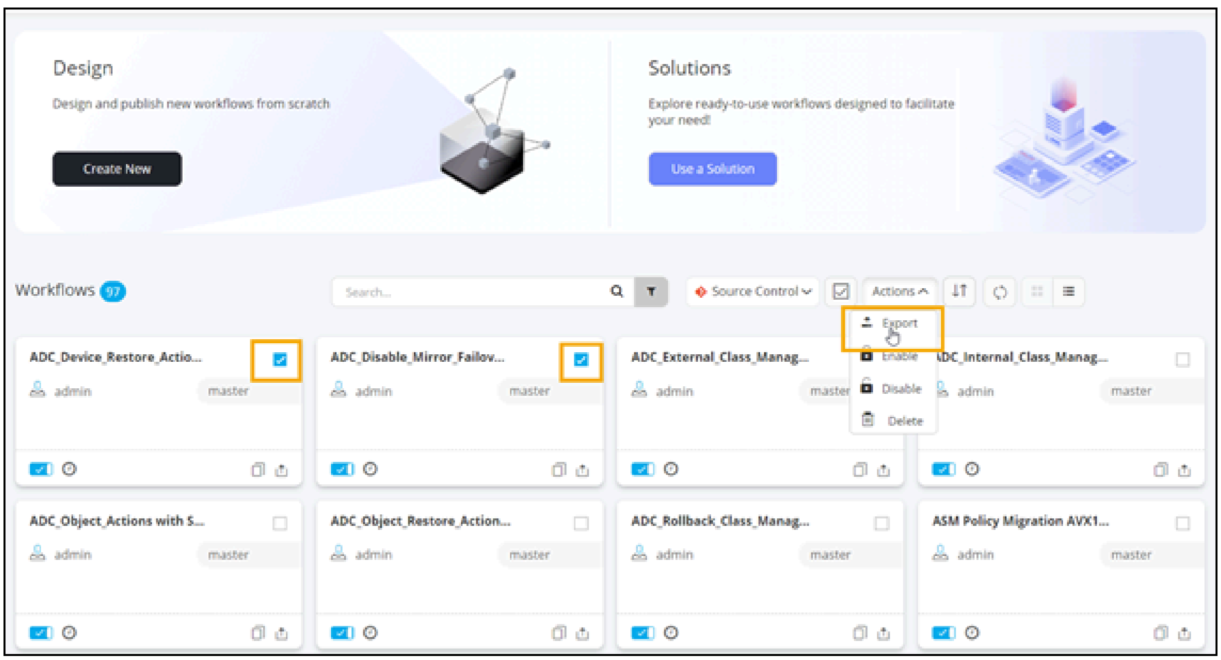
> **Note:** Ensure that the workflows to be deleted are disabled.

6. To delete a workflow when the **Workflow** page is in **List view**, select the workflow(s) to be deleted and click **(Delete)** icon.

> **Remember:** Ensure that the workflow(s) to be deleted are disabled.

Exporting a Workflow

1. Go to the Workflow page.
2. On the **Workflow** page (Card view), select the workflow(s) to be exported.
3. To export the workflow(s), from the **Actions** dropdown menu, select **Export**.



4. Click **Ok** in the **Export** window.

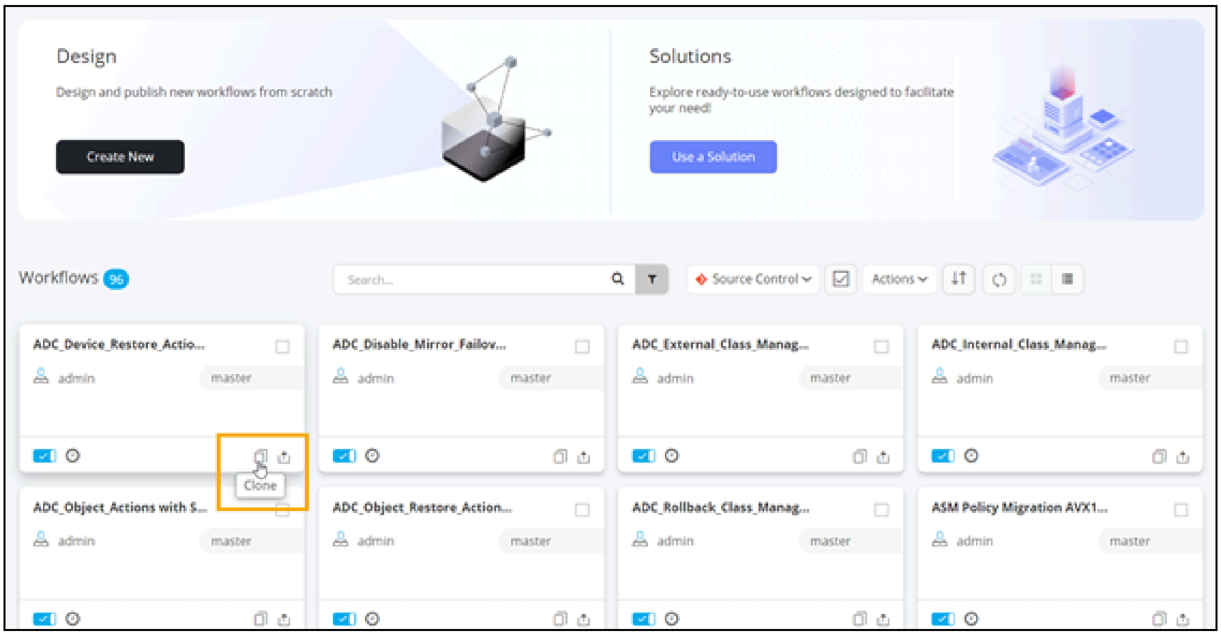
   The selected workflow is saved to your machine as a .zip file.

5. To export a flow, you can also click **(Export)** icon on the workflow card.

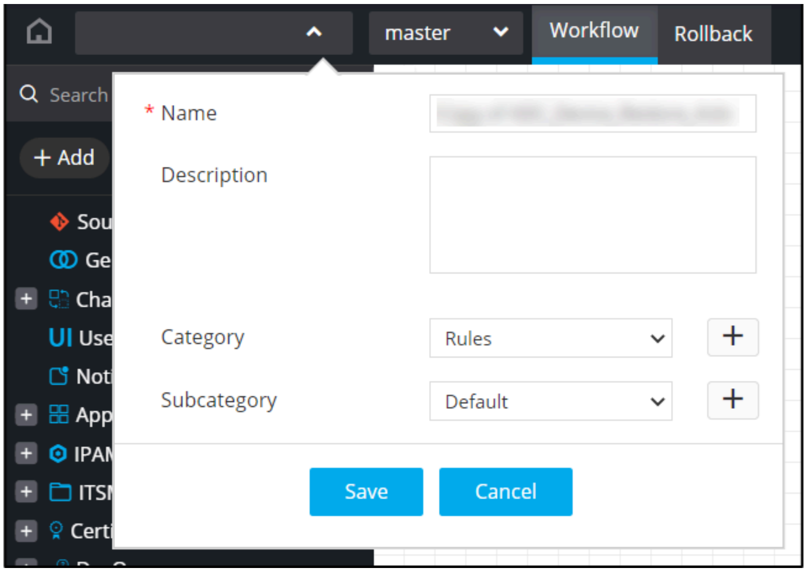
6. To export multiple workflows, you can either select them individually or click **(Selection)** icon to select all workflows displayed on the page and click **Export** in the **Actions** dropdown menu.
7. To export a workflow, when the **Workflow** page is in **List view**, select the workflow(s) to be enabled and click **(Export)** icon.

Cloning a Workflow

1. Go to the Workflow page.
2. On the **Workflow** page (Card view), click **(Clone)** icon on the workflow to be cloned.



   A copy of the cloned workflow opens in the design canvas.
3. Enter a **Name** for the cloned workflow and click **Save**.



A copy of the cloned workflow is added to the **Workflow** page.

> **Note:** To clone a workflow when the **Workflow** page is in **List view**, select the workflow to be cloned and click  **(Clone)** icon.

Designing a Workflow 

The Design section in the Workflow studio allows you to create a new workflow or modify an existing workflow. The canvas is a placeholder to drag and drop tasks, to define custom logic, and stitch together an automation workflow.

The following options are available when designing a new workflow:

- Provision to design a new workflow and assign a category and a sub category.
- Provision to search and view the inventory of tasks by category.
- Provision to drag and drop in-built, custom tasks to design a workflow.
- Provision to clone existing tasks and use them in a workflow.
- Provision to delete tasks.
- Provision to link and update existing links and connect tasks within a workflow.
- Provision to validate a workflow for errors.
- Provision to assign a workflow to user role(s).
- Provision to enable or disable a workflow.
- Provision to bookmark workflow task(s).

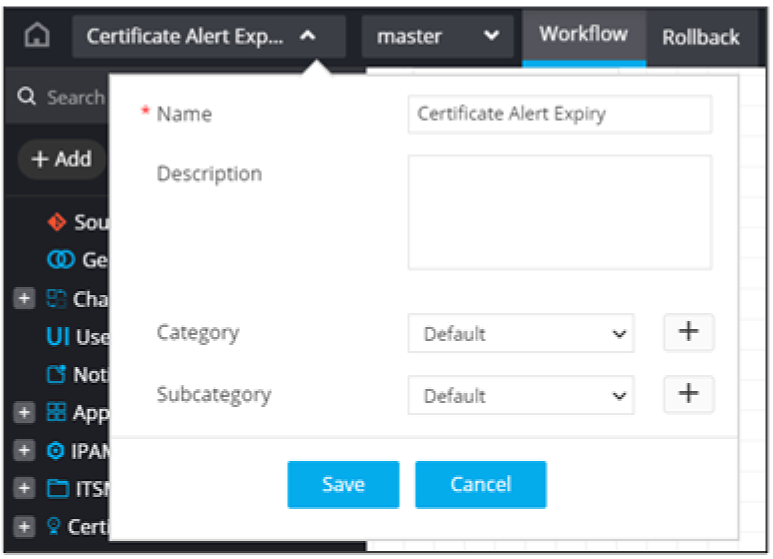
- Provision to preview a workflow task.
- Provision to preview a workflow/nested workflow.
- Provision to reuse and/or import workflow and workflow task(s).

To design a new workflow:

1. Go to the Workflow page.
2. On the **Workflow** page (Card view), click **Create New**.

> **Tip:** When the **Workflow** inventory page is in List View, you can click + **(Add)** icon in the command bar to design a new workflow.

3. In the workflow design space, enter or select the requested field information.



Field descriptions for workflow design space

Field	Description
***Name**	Enter a name for the workflow.
Description	Enter a description for the workflow.
Category	Select a category from the options displayed in the dropdown. **Note:** You can create a new category by clicking + **(Add)** icon.
Subcategory	Select a subcategory from the options displayed in the dropdown.

Field	Description
	Note: You can create a new subcategory by clicking + **(Add)** icon.

Note: The asterisk (*) symbol indicates mandatory fields.

4. Click **Save**.
5. Drag and drop tasks from the folders in the menu on the left to design workflows from scratch.

Note: For more information on getting started quickly with building workflows, click here.

- Modifying a Workflow

Modifying a Workflow

You can modify a pre-existing workflow to suit user-specific requirements.

Note: Before modifying a workflow, ensure that it is disabled.

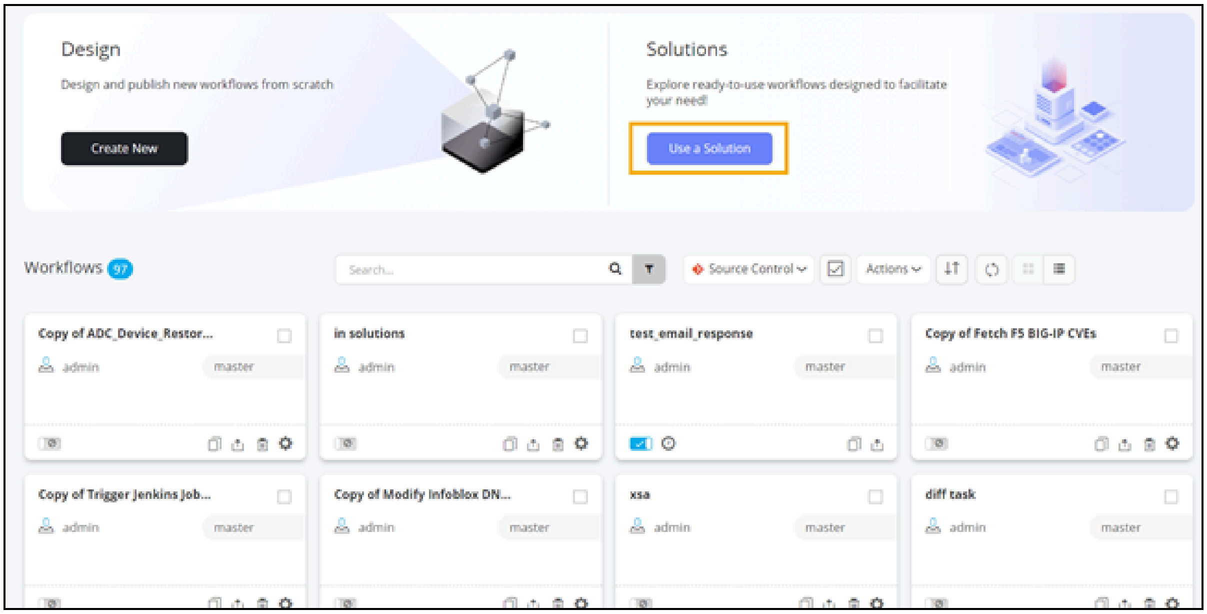
1. Go to the Workflow page.
2. On the **Workflow** page, click on the workflow that you want to modify.

   The workflow opens in the design space.
3. Modify the task(s) in the workflow as required

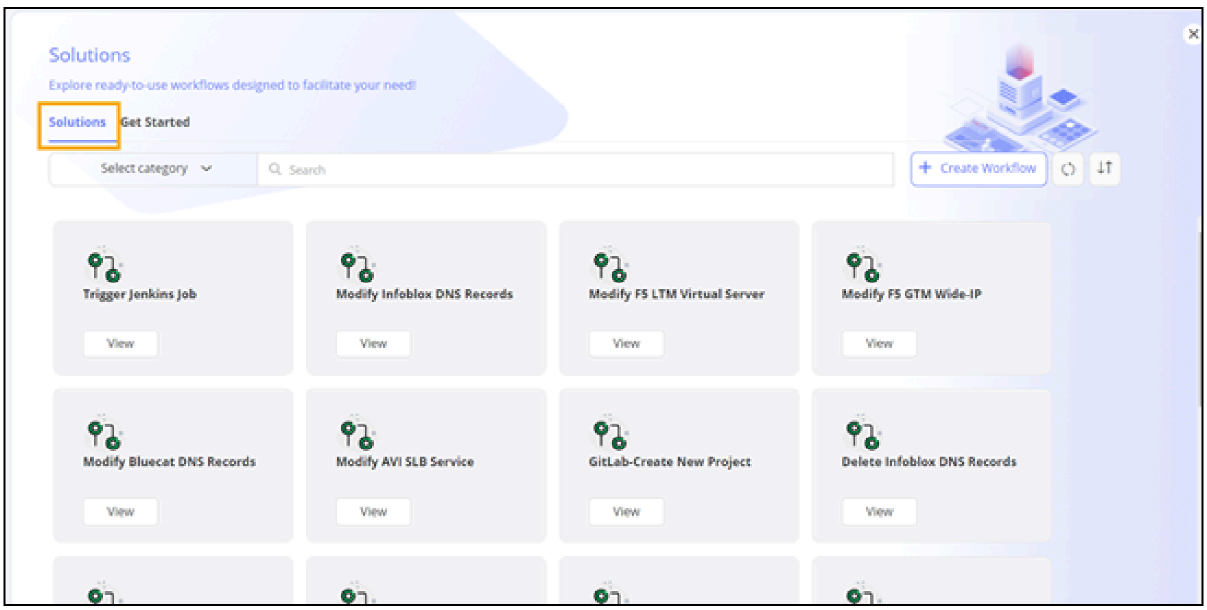
Accessing Solutions

This section allows you to easily access the out of the box (OOB) solutions and FAQs from the Workflow page. You can use any of these solutions to build a workflow quickly and easily.

1. Go to the Workflow page.
2. On the **Workflow** page, click **Use a Solution**.

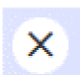


The **Solutions** page is displayed with the **Solutions** tab open by default.



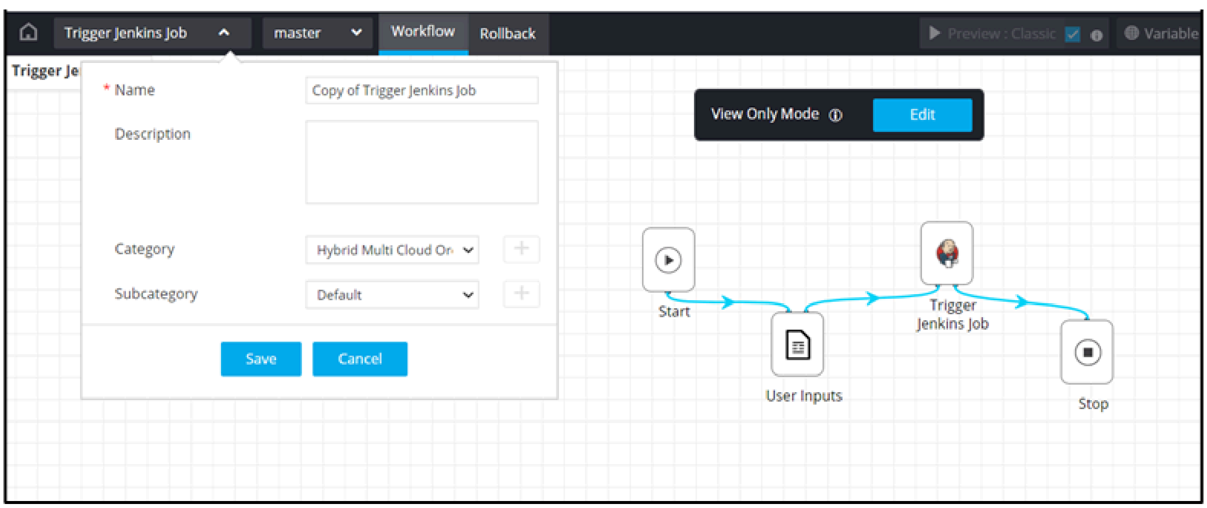
Options available on Solutions tab

Option	Description
Solutions	The **Solutions** tab displays the ready-to-use **solutions** available in the Workflow Studio.
Get Started	The **Get Started** tab displays the **FAQs** available in the Workflow Studio.

Option	Description
Select Category	Allows you to filter solutions according to the category selected in the dropdown list.
Search	Allows you to search for a solution by typing the keyword(s).
Create Workflow	Allows you to create a workflow.
	Refreshes the page.
	Allows you to rearrange the workflows as • Ascending • Descending • Recently created.
	Allows you to exit the **Solutions** page and return to the **Workflow** page (Card view).

3. To open a ready-to-use solution in the design space, click **View** on the corresponding card.

   The workflow opens in a **View Only Mode** in the design workspace.
4. To use this solution, click **Edit**.
5. Enter a new **Name** for the workflow, if required.



6. Click **Save**.

   The workflow can now be modified as required.

Note: You can also access solutions from within a workflow open in the design canvas by clicking icon in the top right corner.

7. To access quick out of the box solutions that can be consumed within workflows, on the Solutions page, click **Get Started** tab.
   The **Get Started** tab displays these OOB solutions.

Accessing the Store

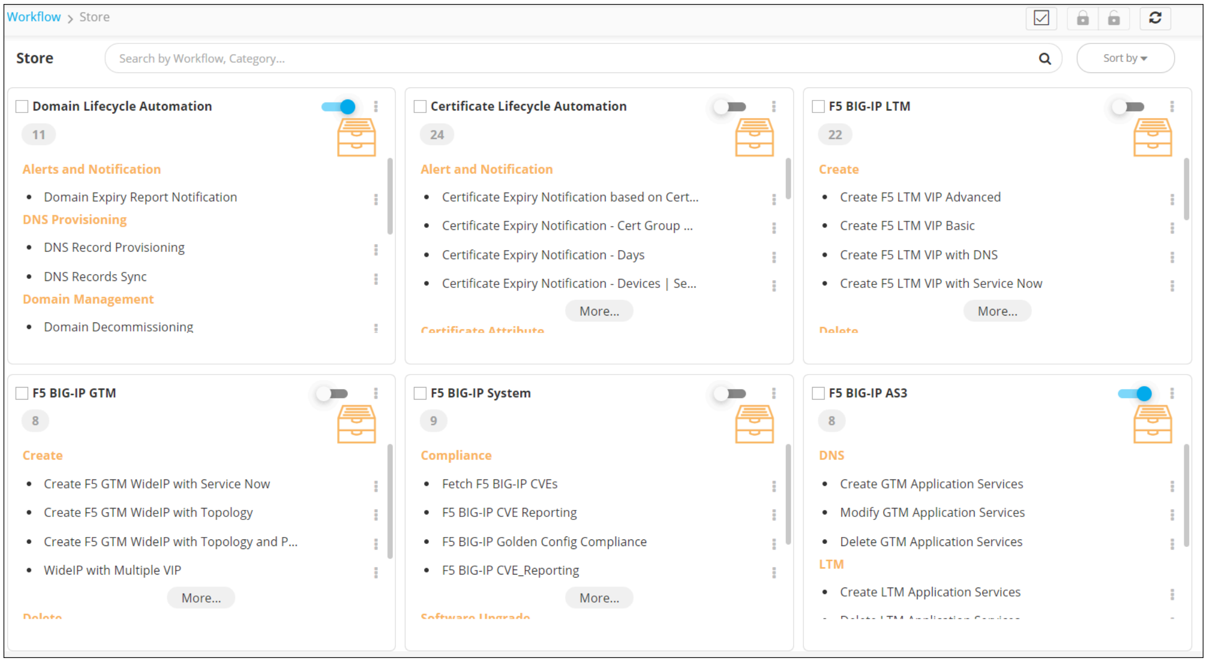
<table>
<tr><th>New Menu</th><th>Old Menu</th></tr>
<tr><td>In the Automation module, from the navigation pane on the left, under WORKFLOW, select Store.

The Workflow > Store page is displayed.</td><td>1. From the main navigation menu, select Studio > Workflow.

The Workflow page is displayed with the Design tab open by default.
2. From the navigation pane on the left, select Store.

The Workflow > Store page is displayed.</td></tr>
<tr><td colspan="2">Note: For more information on how to switch between menus, click here.</td></tr>
</table>

The **Store** is an inventory of all out of the box (OOB) workflows made available to users. All OOB workflows are presented in a catalog view with related workflows grouped together under one catalog. The **Store** can be accessed under **WORKFLOW** section.



Actions you can perform on the Store page

Action	Description
Search bar	Allows you to search for a specific workflow in the Store.
	Allows you to select all workflow catalogs on the Store page.
	Allows you to enable all the workflows in a selected workflow catalog(s). **Note:** This action is enabled only when you select a workflow catalog(s).
	Allows you to disable all the workflows in a selected workflow catalog(s). **Note:** This action is enabled only when you select a workflow catalog(s).
	Refreshes the page
Sort by	Allows you to sort the catalogs.

The following options are available in the **Store**.

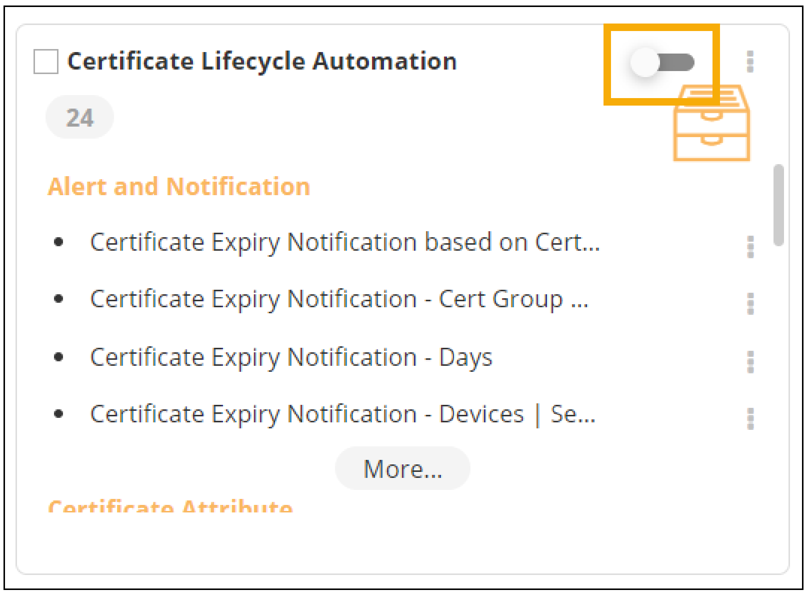
- Enabling/Disabling OOB Workflow Catalogs
- Enabling/Disabling individual OOB Workflows
- Cloning an OOB Workflow

- Scheduling an OOB Workflow
- Configuring RBAC for OOB Workflows
- Sorting the Workflow Catalogs

Enabling/Disabling OOB Workflow Catalogs

This feature allows you to enable/disable all the workflows in a given catalog. When you disable a workflow catalog, all the workflows categorized under the catalog will not be available under **Service Requests** and therefore cannot be scheduled or triggered.

1. Go to the Store.
2. To disable a workflow catalog, on the **Workflow > Store** page. turn off the toggle for the catalog to be disabled.



3. To enable a workflow catalog, on the **Workflow > Store** page, turn on the toggle for the catalog to be enabled.

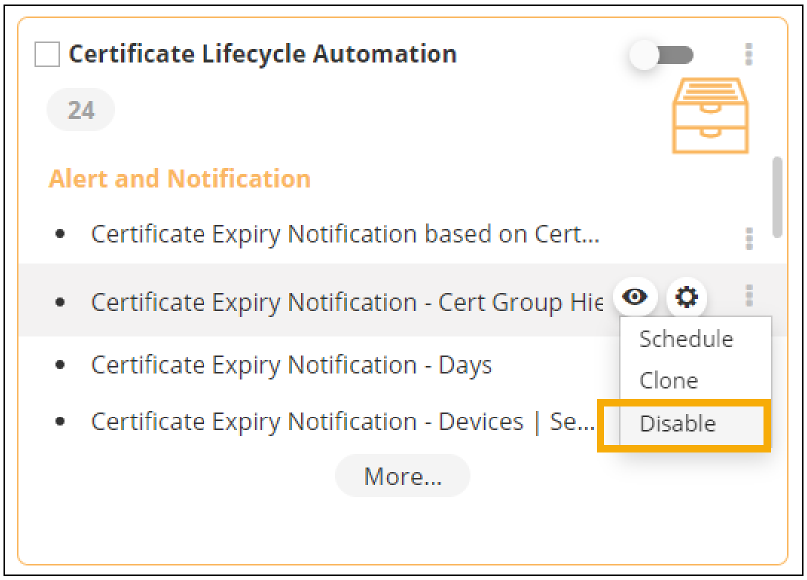
> **Tip:** You can also enable or disable multiple catalogs by selecting the checkbox for the catalog(s) and clicking **(Enable)** icon to enable or **(Disable)** icon to disable the catalog(s).

Enabling/Disabling individual OOB Workflows

You can also enable or disable individual workflow from the Store. When you disable a workflow, it will not be available on the Workflow Request page and therefore cannot be scheduled or triggered.

To disable an OOB workflow:

1. Go to the Store.
2. On the **Workflow > Store** page, hover your mouse over a workflow name, and then click **(Dots)** icon.
3. From the available options, click **Disable**.



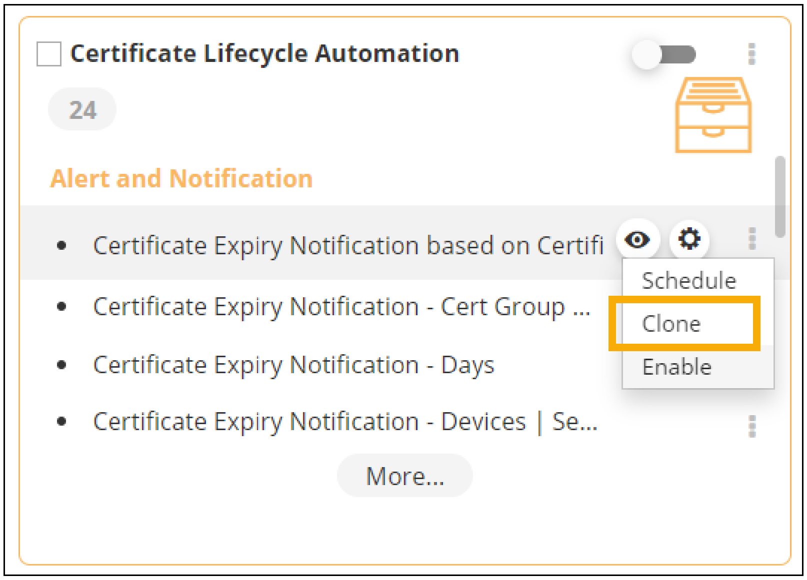
The workflow is disabled.
4. To enable a workflow, on the **Workflow > Store** page, hover your mouse over the workflow to be enabled, and then click **(Dots)** icon.
5. From the options available, click **Enable**.
The workflow is enabled.

Cloning an OOB Workflow

To clone an individual OOB workflow inside a workflow catalog:

1. Go to the Store.
2. On the **Workflow > Store** page, hover your mouse over the workflow to be cloned, and then click **(Dots)** icon.

3. From the options available, click **Clone**.

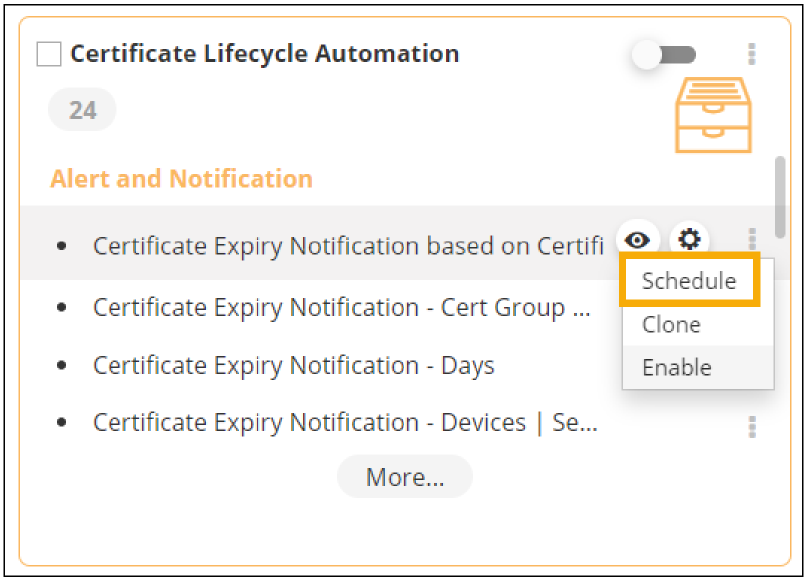


A copy of the cloned workflow is added to the **Workflow** inventory page.

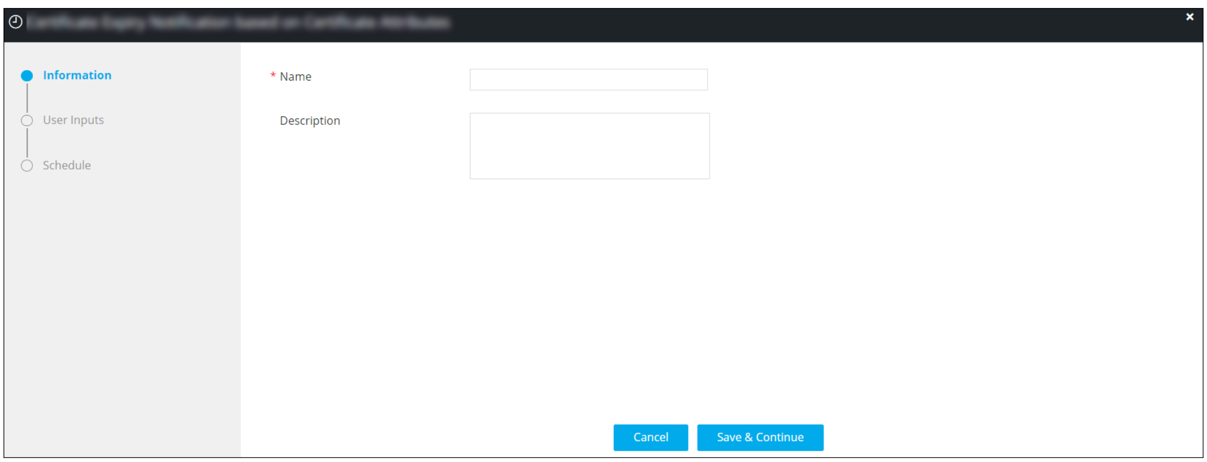
Scheduling an OOB Workflow

To schedule an OOB workflow from the Store:

1. Go to the Store.
2. On the **Workflow** > **Store** page, hover your mouse over the workflow you want to schedule, and then click **(Dots)** icon.
3. From the options available, click **Schedule**.



4. In the pop-up window that is displayed, under required field information for scheduling this workflow.



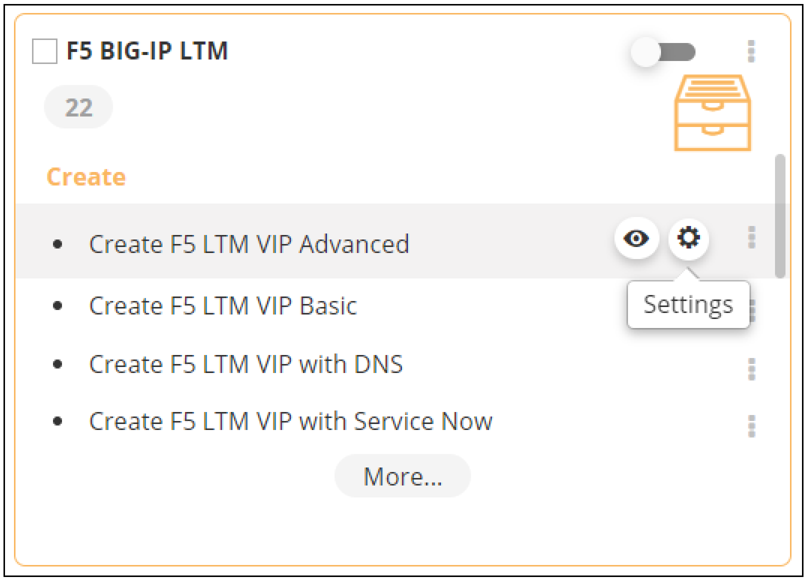
5. Click **Schedule**.

   The scheduled workflow is added to the **Scheduled jobs** on the Request :: Schedule jobs page.

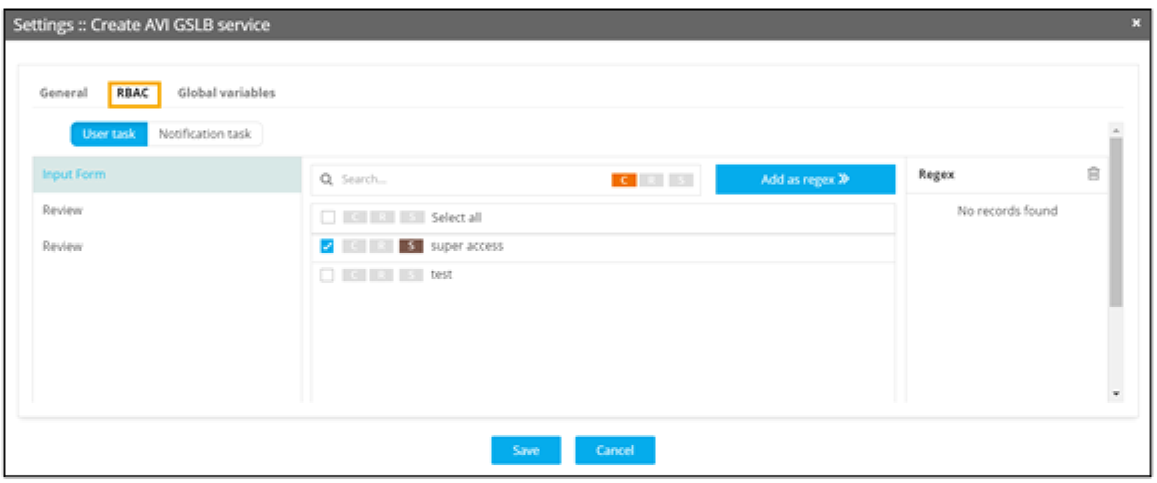
Configuring RBAC for OOB Workflows

To configure RBAC settings for a particular OOB workflow:

1. Go to the Store.
2. On the **Workflow > Store** page, hover your mouse over the workflow, and then click  **(Settings)** icon.



3. In the **Settings** window, under **RBAC**, you can map the access control settings for user(s) for all tasks within the workflow.

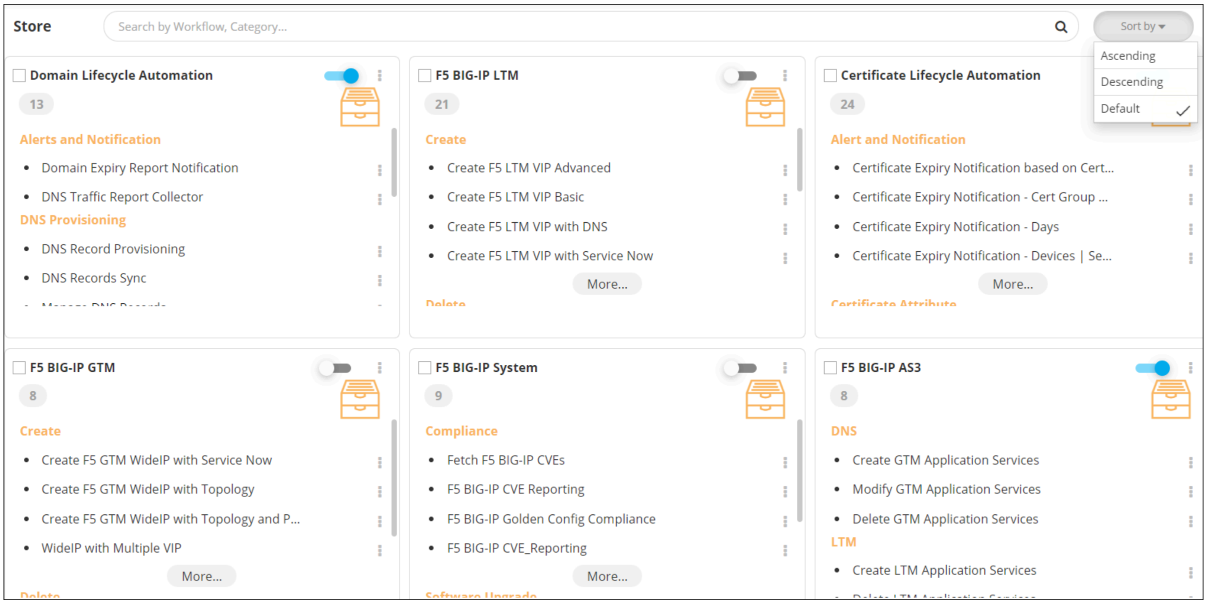


When trying to add the regex "*", a validation error will be triggered, resulting in the disabling of the "Add as regex" option.

Sorting the Workflow Catalogs

You can sort the workflow catalogs in ascending or descending order.

1. On the Store page, from the top right corner of the page, click **Sort by**.



2. From the **Sort by** dropdown menu, select how you want to arrange the workflow catalogs. The options available are:
   - **Ascending**: All workflow catalogs will be sorted in an ascending order as per the catalog name.
   - **Descending**: All workflow catalogs will be sorted in a descending order as per the catalog name.
   - **Default**: Workflows catalogs will be displayed in any random order.

Importing Workflows

You can import workflow(s) from one environment to the other.

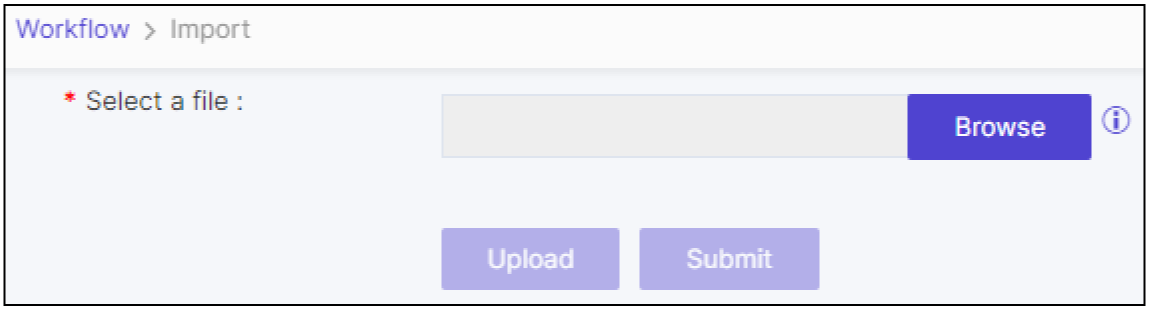
To import a workflow:

New Menu	Old Menu
In the Automationmodule, from the navigation pane on the left, under **WORKFLOW**, select **Import**. The **Workflow** > **Import** page is displayed.	1. From the main navigation menu, select **Studio** > **Workflow**. The **Workflow** page is displayed with the **Design** tab open by default. 2. From the navigation pane on the left, select **Import**.

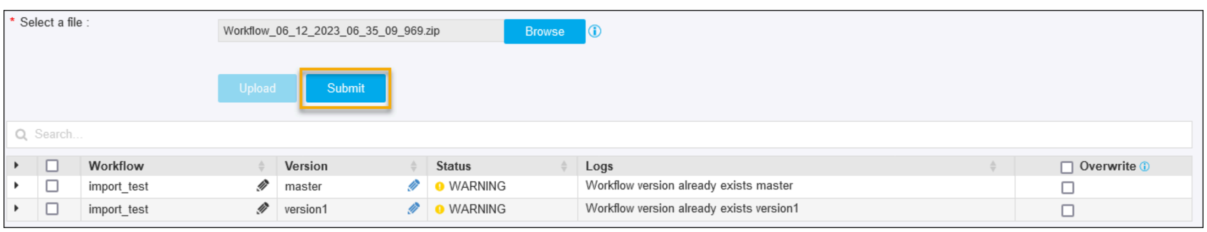
New Menu	Old Menu
	The **Workflow** > **Import** page is displayed.

Note: For more information on how to switch between menus, click here.

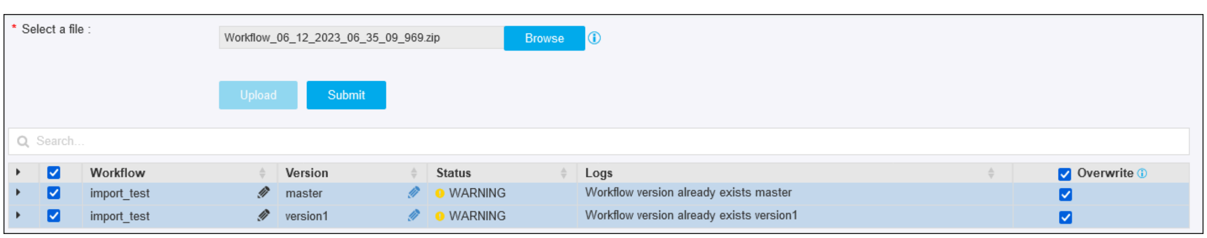
1. On the **Workflow** > **Import** page, to select the file to be imported, click **Browse**.



2. Select the file from the file explorer on your computer.
3. To upload the selected .zip file, click **Upload**.
4. Select the workflow, and then click **Submit**.

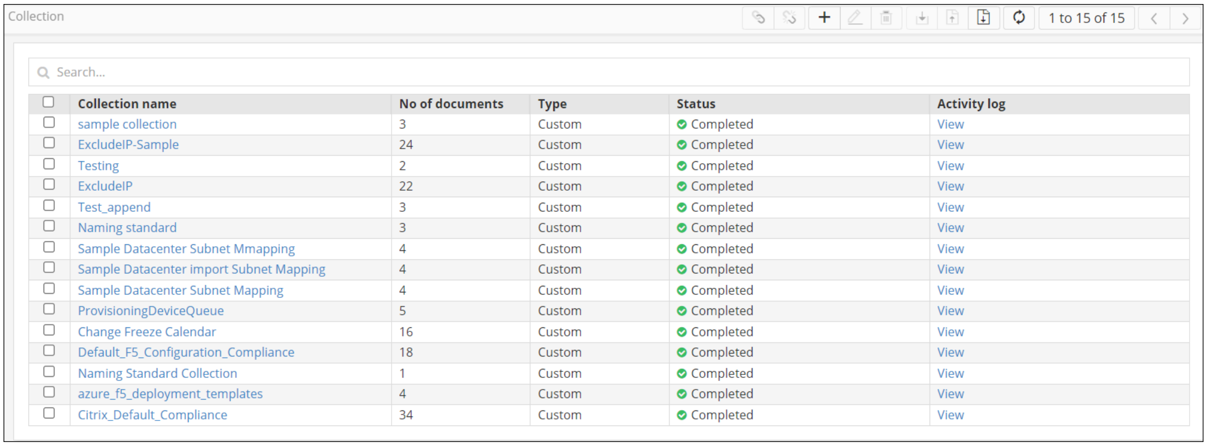


Warning: If you are importing a workflow that already exists, the workflow status will show a Warning. You need to either edit the workflow name or the version to be able to import this workflow or you can choose the overwrite option to replace its existing content and the RBAC permission for the specific task will remain same with the existing workflow.



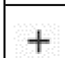
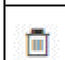
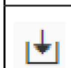
Collection

Collection is a placeholder for storing custom data which can be used in other workflows, reports, and service catalogs. Data generated from one workflow can be stored as a collection and used as a part of another workflow through APIs or Hooks.



	Collection name	No of documents	Type	Status	Activity log
☐	sample collection	3	Custom	Completed	View
☐	ExcludeIP-Sample	24	Custom	Completed	View
☐	Testing	2	Custom	Completed	View
☐	ExcludeIP	22	Custom	Completed	View
☐	Test_append	3	Custom	Completed	View
☐	Naming standard	3	Custom	Completed	View
☐	Sample Datacenter Subnet Mmapping	4	Custom	Completed	View
☐	Sample Datacenter import Subnet Mapping	4	Custom	Completed	View
☐	Sample Datacenter Subnet Mapping	4	Custom	Completed	View
☐	ProvisioningDeviceQueue	5	Custom	Completed	View
☐	Change Freeze Calendar	16	Custom	Completed	View
☐	Default_F5_Configuration_Compliance	18	Custom	Completed	View
☐	Naming Standard Collection	1	Custom	Completed	View
☐	azure_f5_deployment_templates	4	Custom	Completed	View
☐	Citrix_Default_Compliance	34	Custom	Completed	View

Options available on the Collection page

Option	Description
	Allows you to enable the selected collection(s) for building a custom query. **Note:** This option is enabled only when you select a collection.
	Allows you to disable the selected collection(s) for building a custom query. **Note:** This option is enabled only when you select a collection.
	Allows you to create a new collection manually.
	Allows you to edit/update the selected collection(s). **Note:** This option is enabled only when you select a collection.
	Allows you to delete the selected collection(s). **Note**: This option is enabled only when you select a collection.
	Allows you to download the selected collection(s) as a .xls file.

Options available on the Collection page (continued)

Option	Description
	Note: This option is enabled only when you select a collection.
	Allows you to export the selected collection(s) as a .zip file. **Note:** This option is enabled only when you select a collection.
	Allows you to import collection(s) as .xls or .zip files.
	Refreshes the page.

- Creating Collections
- Importing Collections
- Modifying Collections
- Using Collections

Creating Collections

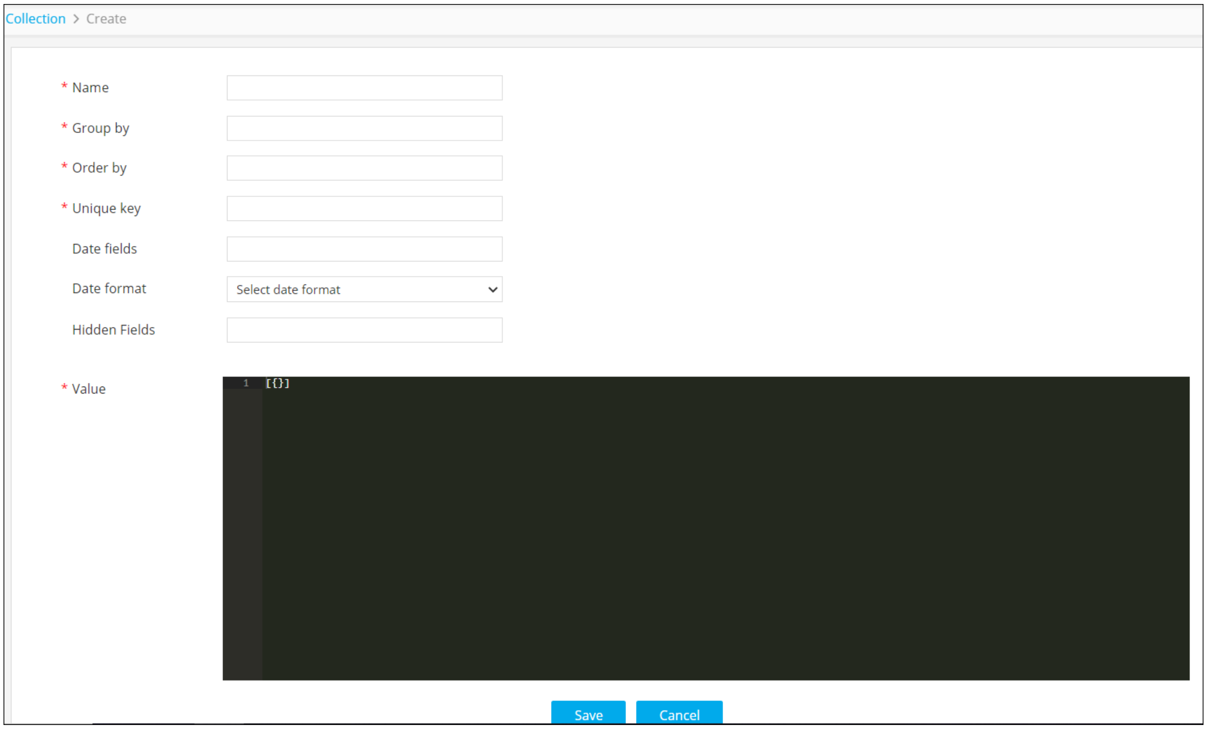
You can create a collection either by defining all the custom data for the collection from scratch or by using the sample collection file.

- Defining Custom Data
- Using Sample Collection File

Defining Custom Data

To create a new collection from scratch:

1. On the **Collection** page, from the command bar on the top right of the screen, click + **(Add)** icon.
2. On the **Collection** > **Create** page, enter the required field information.



Field descriptions for create page

Field	Description
***Name**	Enter the name for the collection. For example: ExcludeIP-Sample
***Group by**	Define how the collection will be grouped. For example: datacenter All the IPs will be grouped according to the datacenter.
***Order by**	Define how the data in the collection will be ordered. For example: ip The data in the collection will be ordered by the IPs
***Unique key**	Define the unique key or identifier for the collection.
Date fields	Define the date fields in the metadata, if required.
Date format	Define the date format, if required. For example: dd/mm/yyyy

Field	Description
Hidden Fields	Enter the hidden fields, if required.

***Value**	Define the data for the collection.
	
	For example:
	`[`
	`{`
	`"ip": "192.168.42.123",`
	`"datacenter": "Texas"`
	`},`
	`{`
	`"ip": "192.168.42.124",`
	`"datacenter": "Texas"`
	`}`
	`]`

Note: The asterisk (*) symbol indicates mandatory fields.	

3. Click **Save**.
   The collection is added to the **Collection** page.

Using Sample Collection File

To create a collection using the sample collection file:

1. On the **Collection** page, from the command bar on the top right of the screen, click **(Download)** icon.
2. On the **Collection** > **Import** page, to download a sample collection file, from the top right corner of the screen, click **(Download)** icon.
   A sample collection file <.xls> is downloaded to your computer.
3. Open the downloaded file and define the keys and metadata as required. A sample is shown here:

	A	B	C	D	E
1	MetaKeys	MetaValues	subnet	datacenter	zone
2	collectionNam	Collection Sample	192.168.33.0/26	NYE	Red
3	groupBy	zone	192.168.32.0/25	CBE	Green
4	orderBy	subnet			
5	uniqueKey	zone			
6					

4. Save the collection and import it into your instance.

Importing Collections

There are three ways to import a collection into your instance:

- Importing the Sample file
- Importing a <.xls> or <.zip> file
- Importing an Exported File

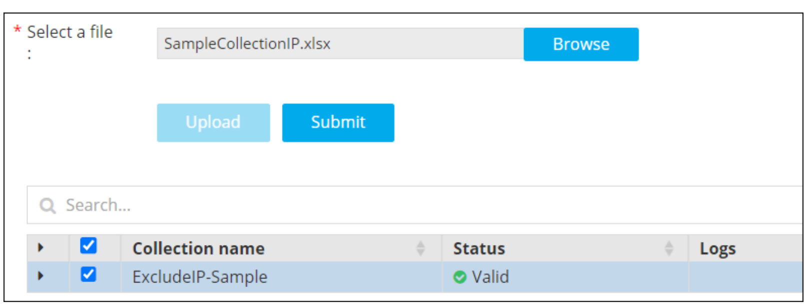
Importing the Sample file

For steps on how to import and use the sample collection file to create custom collections, click here.

Importing a <.xls> or <.zip> file

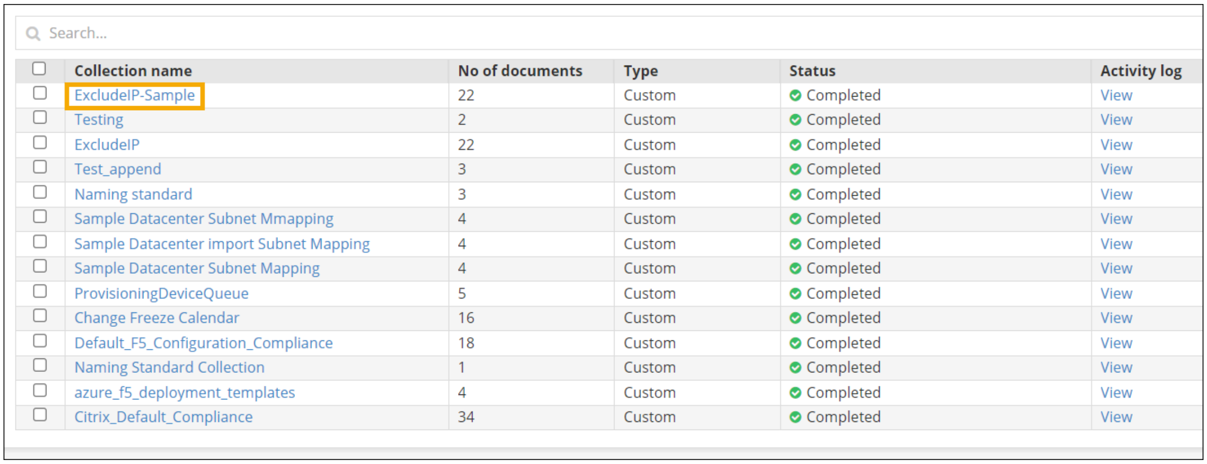
To import a collection into AppViewX:

1. On the **Collection** page, from the command bar on the top right of the screen, click **(Download)** icon.
2. On the **Collection** > **Import** page, click **Browse**.
3. Select the file to be imported into AppViewX and click **Upload**.



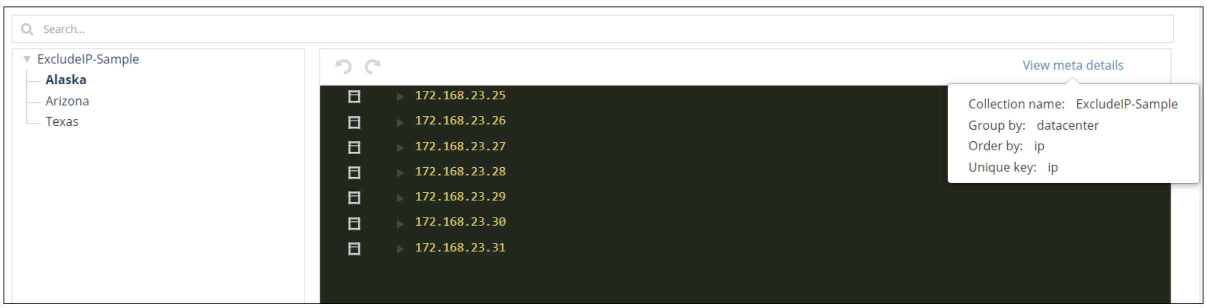
4. Select the checkbox next to the **Collection name**, and then click **Submit**.

   The imported collection is added to the **Collection** page.



Collection name	No of documents	Type	Status	Activity log
ExcludeIP-Sample	22	Custom	Completed	View
Testing	2	Custom	Completed	View
ExcludeIP	22	Custom	Completed	View
Test_append	3	Custom	Completed	View
Naming standard	3	Custom	Completed	View
Sample Datacenter Subnet Mmapping	4	Custom	Completed	View
Sample Datacenter import Subnet Mapping	4	Custom	Completed	View
Sample Datacenter Subnet Mapping	4	Custom	Completed	View
ProvisioningDeviceQueue	5	Custom	Completed	View
Change Freeze Calendar	16	Custom	Completed	View
Default_F5_Configuration_Compliance	18	Custom	Completed	View
Naming Standard Collection	1	Custom	Completed	View
azure_f5_deployment_templates	4	Custom	Completed	View
Citrix_Default_Compliance	34	Custom	Completed	View

5. Click **Collection name** and hover over **View meta details** for details about the collection.



- **Collection name:** This is the name of the collection. For example: **ExcludeIP-Sample**.
- **Group by:** You can define the parameter for grouping the data in the collection. For example: All the IPs in this collection will be grouped by datacenter - Alaska, Arizona, Texas, and so on. For more information, click here.
- **Order by:** You can define how the data will be ordered in the collection For example: If you set the Order by field as ip, then all the IPs will be listed in order in the collection. For more information, click here.
- **Unique key:** This is the unique identifier for the data in the collection.

Importing an Exported File

You can export a collection from a different environment and import it into your environment or export a collection from your instance, modify it, and import it again.

1. To export a collection, on the **Collection** page, from the command bar, click **(Export)** icon. The file is saved to your laptop as a <.zip> file.

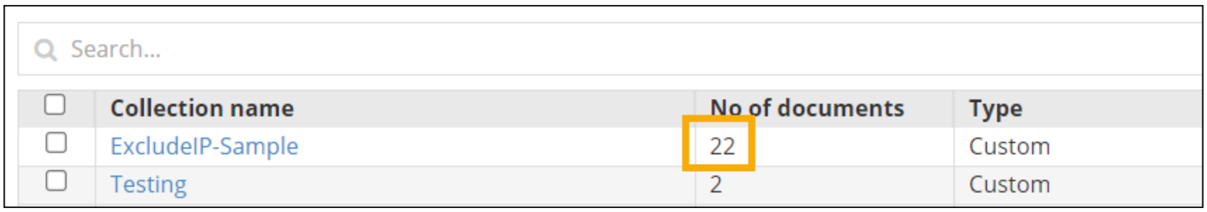
Note: You can also download the collection as a **<.xls>** file by clicking from the command bar.

2. Modify this collection, if required and import it into your instance.

Modifying Collections

To update or modify data in a collection:

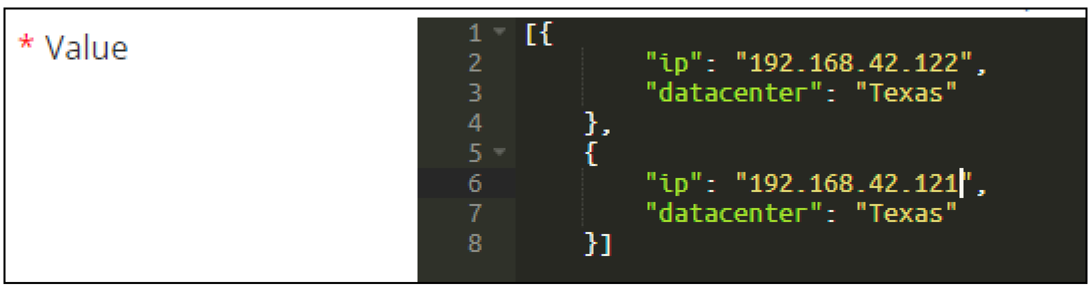
1. On the **Collection** page, select the collection to be modified. For example: ExcludeIP-Sample. This collection currently contains 22 documents.



Search...

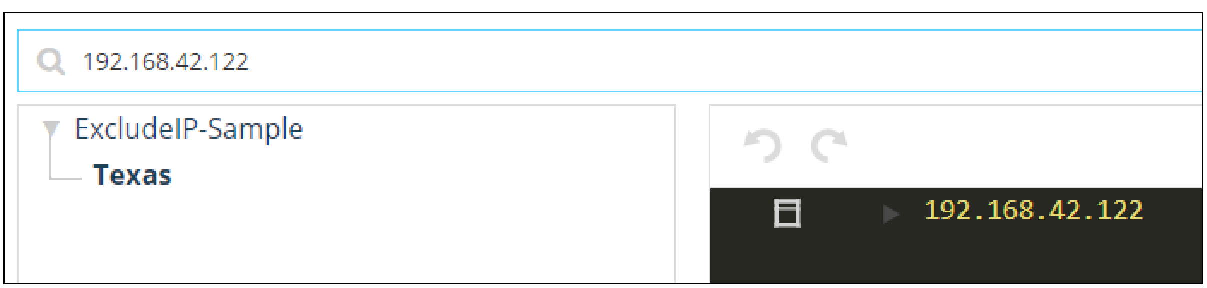
	Collection name	No of documents	Type
☐	ExcludeIP-Sample	22	Custom
☐	Testing	2	Custom

2. From the command bar on the top right of the screen, click **(Modify)** icon.
3. Update the data in the **Value** field.



4. Click **Save**.

The collection is modified with the number of documents in the collection now at 24. You can view the collection details and search for the IPs added to the collection.



Using Collections

Collections can be used in workflows either through APIs or to build a custom query (hooks).

In API

Sample Payload for adding documents to collections


```
{
  "payload": {
          “collection”:”test”
          "action": "add",
          "documents": [
          {
          "deviceName": "F5",
          "ip": "1.2.3.4"
          },
          {
          "deviceName": "A10",
          "ip": "15.25.35.45"
          }
          ]
  }
}
```


In Query Explorer

Click here for more information on importing custom data into the Query Explorer.

Hooks Inventory

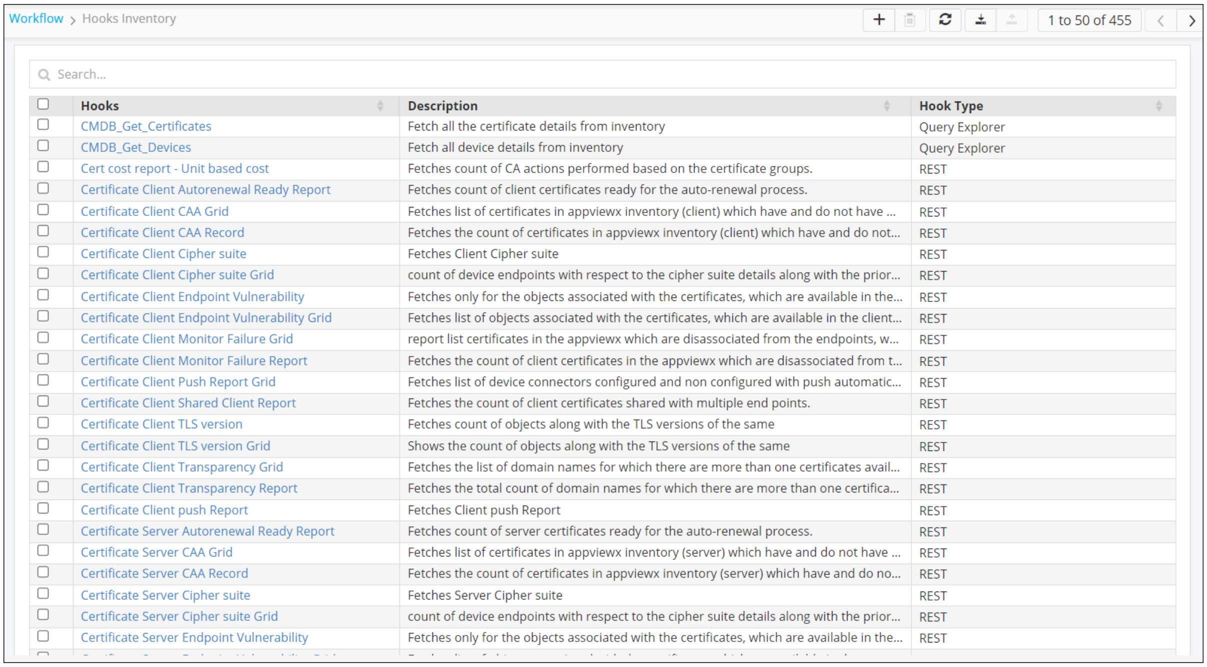
“Design and Define a Query once, reuse them forever…”

A ‘Hook’ is a mechanism used to query and retrieve data from different sources such as database, device, or external vendors; and leverage them as part of the report building process. The mode of query could be either through – Script, REST API and Query Explorer.

For example, a query to retrieve a list of managed BIG-IP F5 devices from the database can be defined once and reused across one or more workflow(s).

There are 3 ways of querying data through hooks:

- [REST API]
- [Script] (Python)
- [Query Explorer] (GUI based)



Hooks	Description	Hook Type
CMDB_Get_Certificates	Fetch all the certificate details from inventory	Query Explorer
CMDB_Get_Devices	Fetch all device details from inventory	Query Explorer
Cert cost report - Unit based cost	Fetches count of CA actions performed based on the certificate groups.	REST
Certificate Client Autorenewal Ready Report	Fetches count of client certificates ready for the auto-renewal process.	REST
Certificate Client CAA Grid	Fetches list of certificates in appviewx inventory (client) which have and do not have ...	REST
Certificate Client CAA Record	Fetches the count of certificates in appviewx inventory (client) which have and do not...	REST
Certificate Client Cipher suite	Fetches Client Cipher suite	REST
Certificate Client Cipher suite Grid	count of device endpoints with respect to the cipher suite details along with the prior...	REST
Certificate Client Endpoint Vulnerability	Fetches only for the objects associated with the certificates, which are available in the...	REST
Certificate Client Endpoint Vulnerability Grid	Fetches list of objects associated with the certificates, which are available in the client...	REST
Certificate Client Monitor Failure Grid	report list certificates in the appviewx which are disassociated from the endpoints, w...	REST
Certificate Client Monitor Failure Report	Fetches the count of client certificates in the appviewx which are disassociated from t...	REST
Certificate Client Push Report Grid	Fetches list of device connectors configured and non configured with push automatic...	REST
Certificate Client Shared Client Report	Fetches the count of client certificates shared with multiple end points.	REST
Certificate Client TLS version	Fetches count of objects along with the TLS versions of the same	REST
Certificate Client TLS version Grid	Shows the count of objects along with the TLS versions of the same	REST
Certificate Client Transparency Grid	Fetches the list of domain names for which there are more than one certificates avail...	REST
Certificate Client Transparency Report	Fetches the total count of domain names for which there are more than one certifica...	REST
Certificate Client push Report	Fetches Client push Report	REST
Certificate Server Autorenewal Ready Report	Fetches count of server certificates ready for the auto-renewal process.	REST
Certificate Server CAA Grid	Fetches list of certificates in appviewx inventory (server) which have and do not have ...	REST
Certificate Server CAA Record	Fetches the count of certificates in appviewx inventory (server) which have and do no...	REST
Certificate Server Cipher suite	Fetches Server Cipher suite	REST
Certificate Server Cipher suite Grid	count of device endpoints with respect to the cipher suite details along with the prior...	REST
Certificate Server Endpoint Vulnerability	Fetches only for the objects associated with the certificates, which are available in the...	REST

Options available on the Hooks Inventory page

Option	Description
Search bar	Allows you to search for a particular hook by typing in keyword(s).
	Allows you to create a new hook.
	Allows you to delete hook(s).
	Refreshes the hooks inventory.
	Allows you to import a hook.
	Allows you to export hook(s).

- [Accessing the Hooks Inventory]
- [Creating a Query Explorer Hook]
- [Creating a Script Hook]

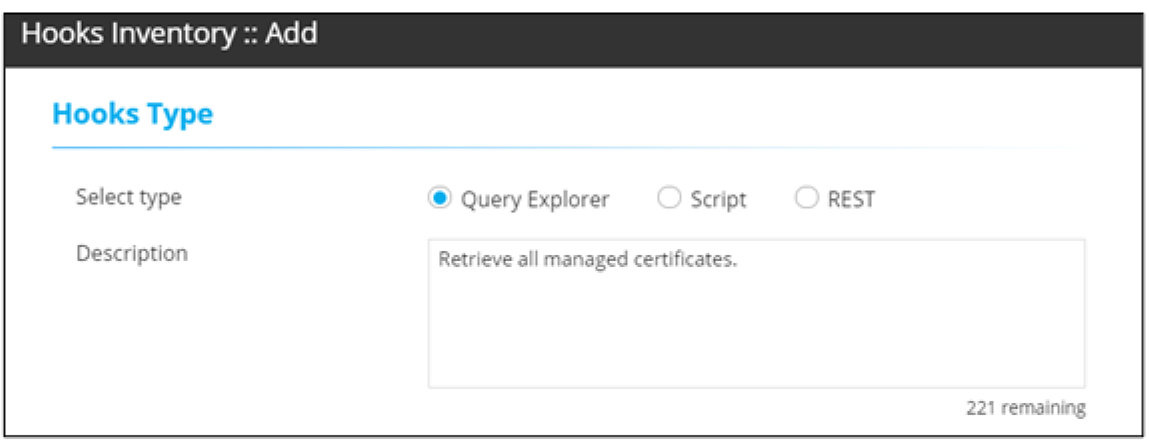
- Creating a REST API Hook
- Defining a Hook and using it within a Workflow

Accessing the Hooks Inventory

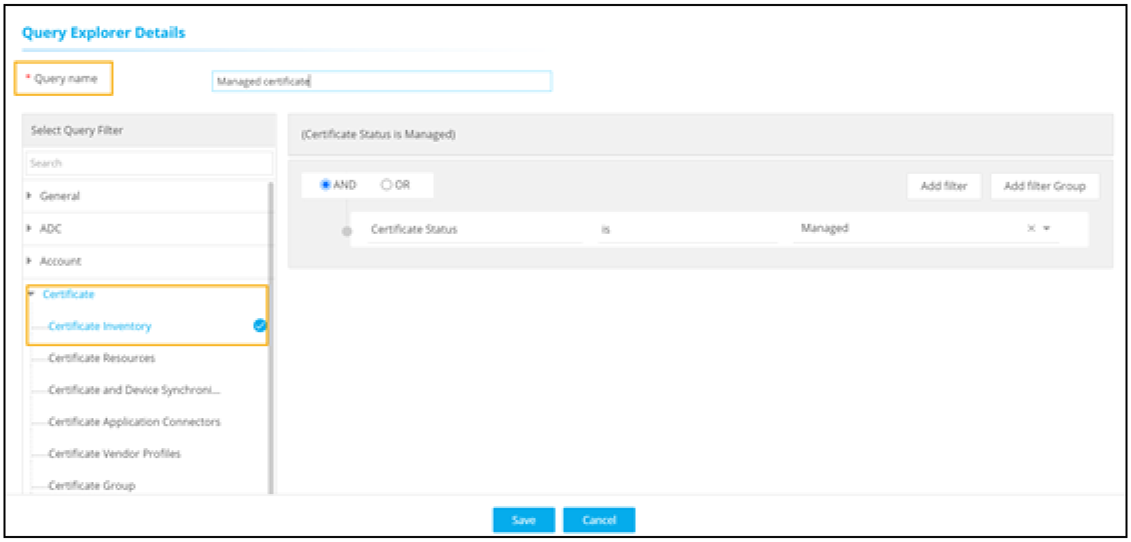
New Menu	Old Menu
In the Automation module, from the navigation pane on the left, under **WORKFLOW**, select **Hooks**. The **Workflow** > **Hooks Inventory** page is displayed.	1. From the main navigation menu, select **Studio** > **Workflow**. The **Workflow** page is displayed with the **Design** tab open by default. 2. From the navigation pane on the left, select **Hooks**. The **Workflow** > **Hooks Inventory** page is displayed.
Note: For more information on how to switch between menus, click here.	

Creating a Query Explorer Hook

1. Go to the Hooks Inventory.
2. On the **Workflow** > **Hooks Inventory** page, from the command bar on the top right corner of the screen, click + **(Add)** icon.
3. On the **Workflow** > **Hooks Inventory** > **Add** page, under **Hooks Type**, select **Query Explorer**.
4. Enter a description for the hook.



5. Under **Query Explorer Details**, enter or select the required field information.



Field Descriptions for Query Explorer Details

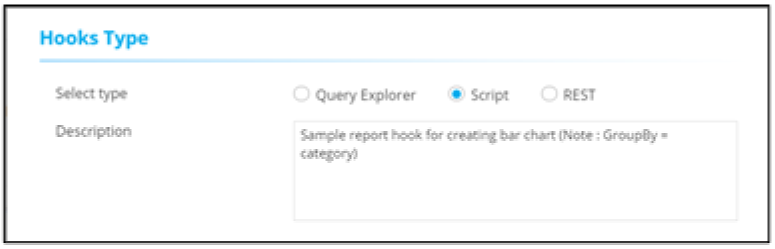
Field	Description
***Query name**	Enter a valid query name. For example, **Managed certificates**.
Select Query Filter	Select an appropriate query filter from the pre-populated list of filters. For example, **Certificate Inventory under Certificate section**.
Add Filter	Add a filter and select appropriate fields from the predefined values. For example, **Certificate Status is Managed**.
AND/OR	Select the appropriate conditional parameters depending on the type of output required in the report. **AND**: Report will reflect data satisfying all the filter conditions. **OR**: Report will reflect data satisfying either of the filter conditions.

Note: The asterisk (*) symbol indicates mandatory fields.

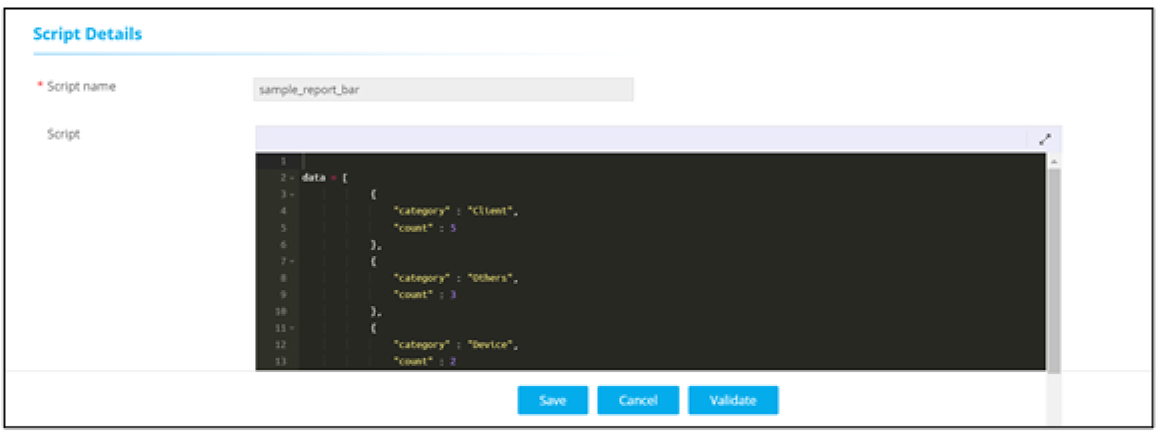
6. To add this new hook to the hooks inventory, click **Save**.

Creating a Script Hook

1. Go to the Hooks Inventory.
2. On the **Workflow** > **Hooks Inventory** page, from the command bar on the top right corner of the screen, click + **(Add)** icon.
3. On the **Workflow** > **Hooks Inventory** > **Add** page, under **Hooks Type**, select **Script**.
4. Enter a description for the hook.



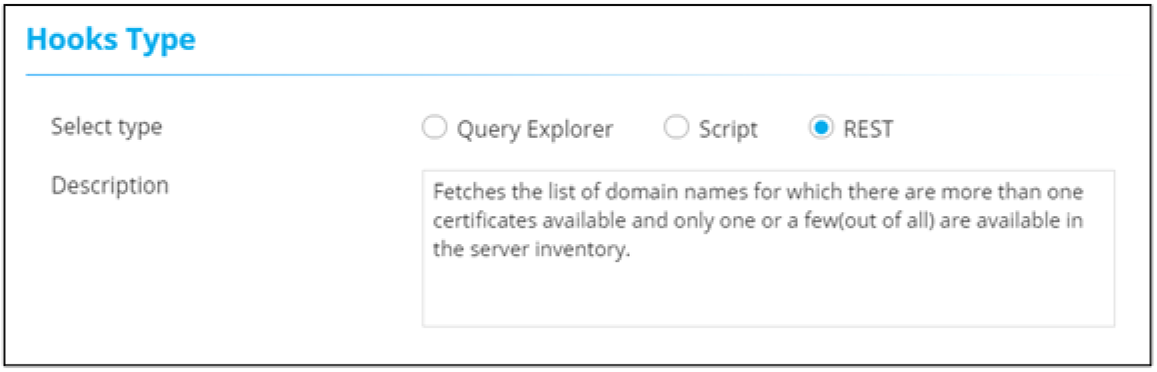
5. Under **Script Details**, enter the **Script name** and define the script logic.



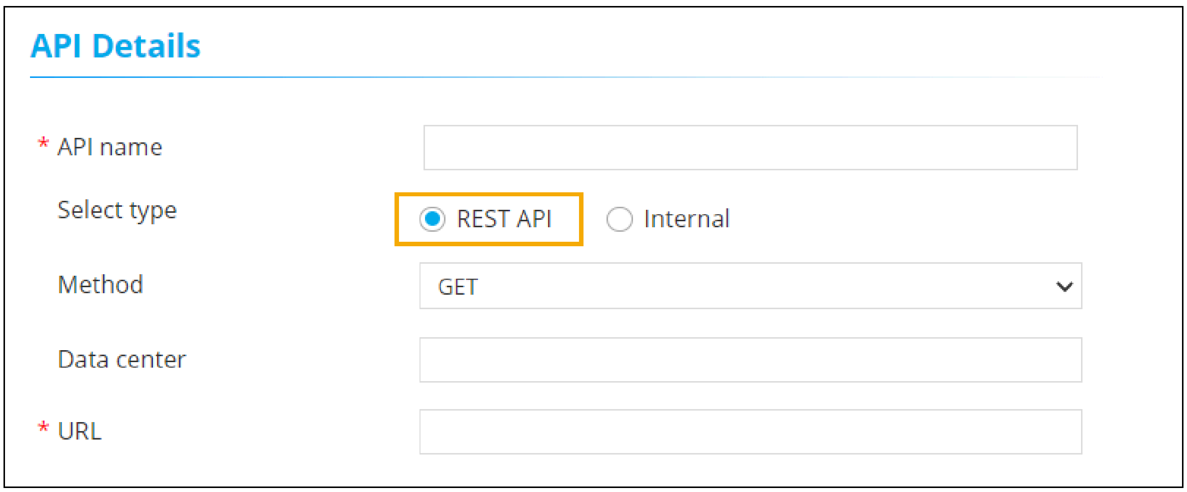
6. To add this new hook to the hooks inventory, click **Save**.

Creating a REST API Hook

1. Go to the Hooks Inventory.
2. On the **Workflow** > **Hooks Inventory** page, from the command bar on the top right corner of the screen, click + **(Add)** icon.
3. On the **Workflow** > **Hooks Inventory** > **Add** page, under **Hooks Type**, select **REST**.
4. Enter a description for the hook.



5. Under **API Details**, enter or select the required field information.
   - **Using REST API**: This is an external facing customer API (For example, ServiceNow, JIRA, GIT)

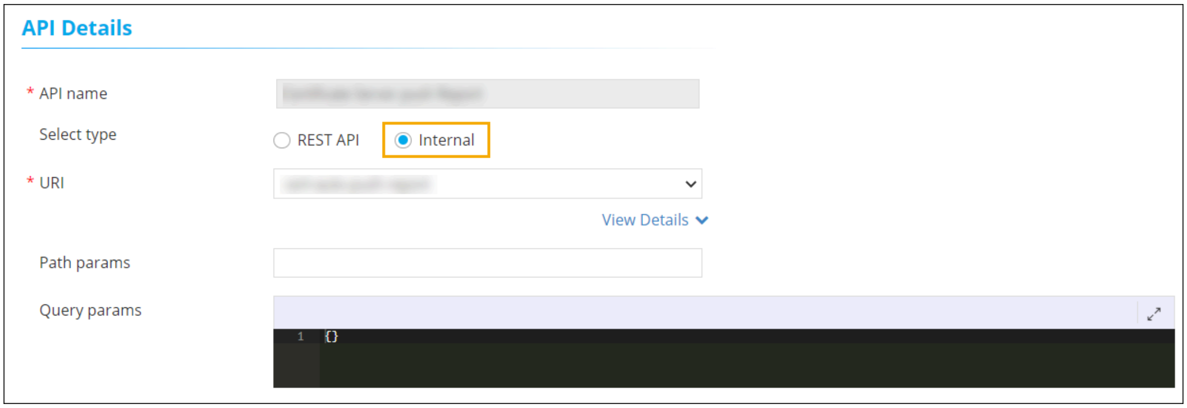


Field descriptions for API Details Select type REST API

Field	Description
API name	Enter a valid name for the API.
Select type	You can select between REST API and Internal. • **REST API:** Use this to select external API and enter the URL. • **Internal:** Use this to select an internal API and select a URI (Uniform Resources Identifier) from the dropdown menu.
Method	Select the API method for submitting requests.
Data center	Enter the name of the data center.
***URL**	Enter the API URL.

Field	Description
Note: The asterisk (*) symbol indicates mandatory fields.	

- **Using Internal API**: API that is integrated internally



Field descriptions for API Details Select type Internal

Field	Description
API name	Enter a valid name for the API.
Select type	You can select between REST API and Internal. • **REST API**: Use this to select external API and enter the URL. • **Internal**: Use this to select an internal API and select a URI (Uniform Resources Identifier) from the dropdown menu.
Path params	Parameters that are used to identify a specific resource(s).
Query params	Parameters that are used to filter resources.
Note: The asterisk (*) symbol indicates mandatory fields.	

6. Under **Request Body**, enter the payload and response mapping details.

Field descriptions for Request Body

Field	Description
Payload	Payload is the actual data that needs to be passed for the API.

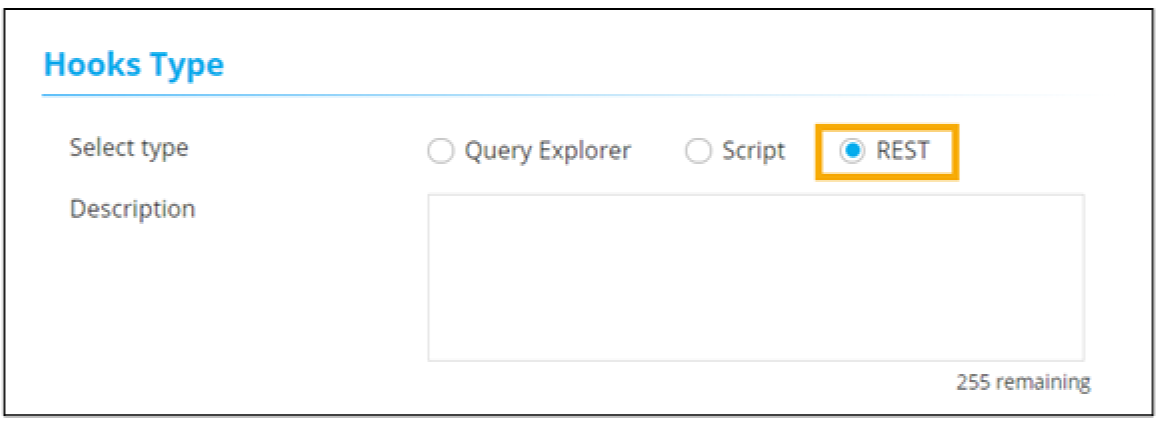
Field	Description
Response mapping	Mapping an integration response from the backend.

7. To add this new REST API hook to the hooks inventory, click **Save**.

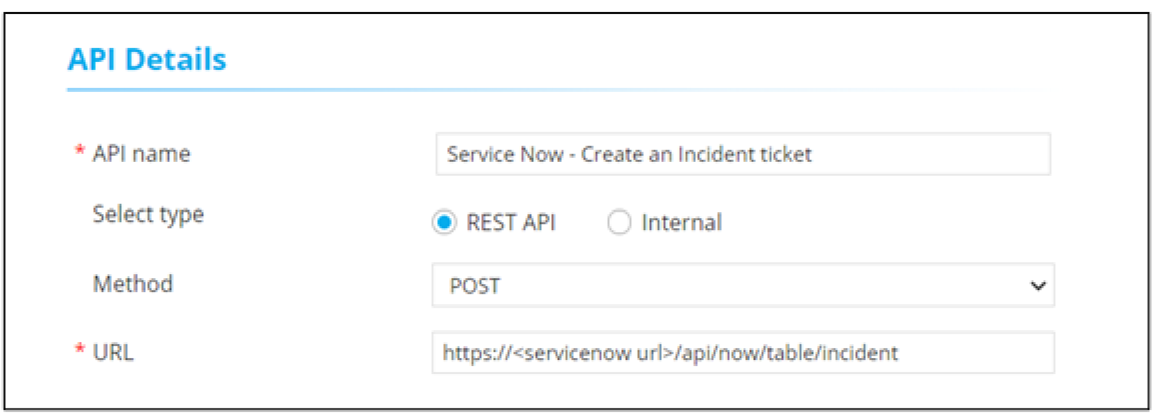
Defining a Hook and using it within a Workflow

To define a common hook to create an incident ticket on ServiceNow and use it within a workflow:

1. Go to the Hooks Inventory.
2. On the **Workflow** > **Hooks Inventory** page, from the command bar on the top right corner of the screen, click + **(Add)** icon.
3. On the **Workflow** > **Hooks Inventory** > **Add** page, under **Hooks Type** select **REST** and enter a **Description** for the hook.

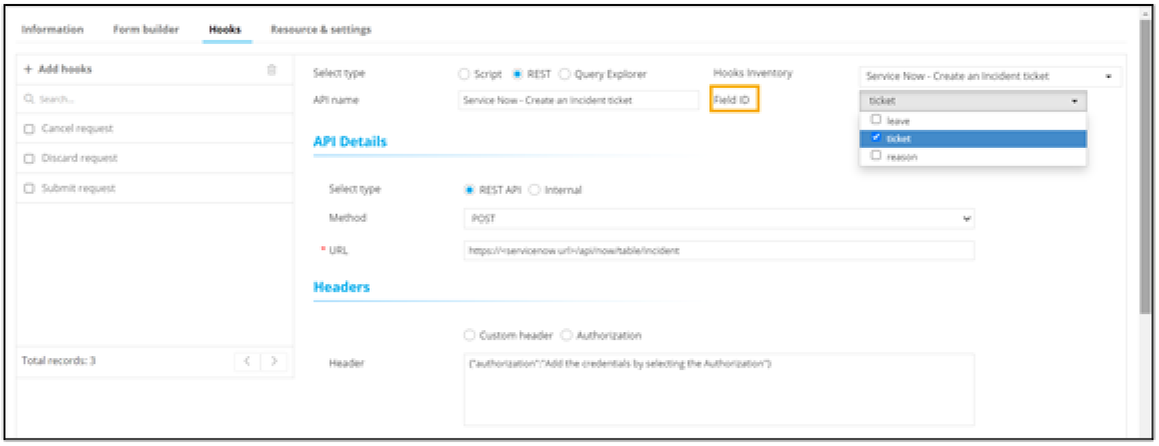


4. In the **API Details** section, enter or select the required field information as shown in the image below.



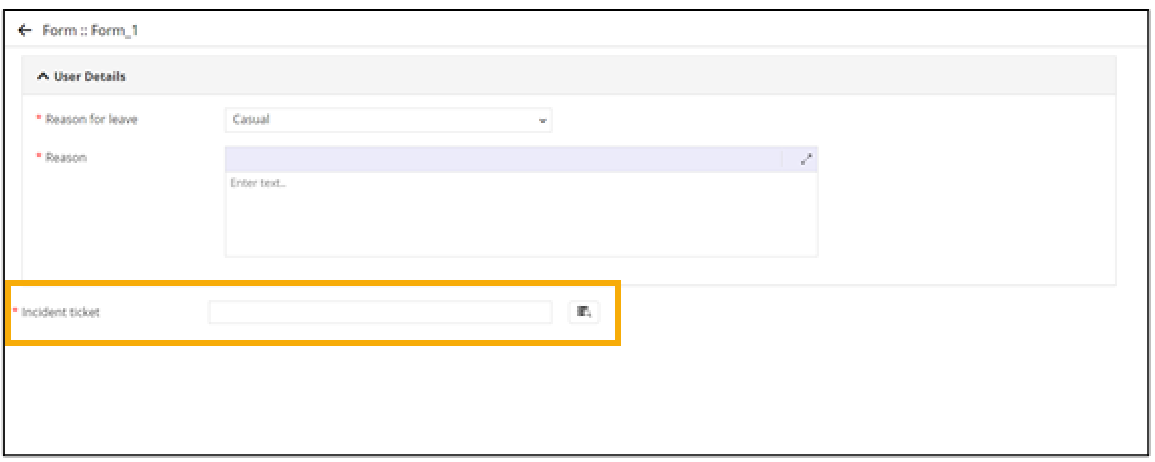
5. Click **Save**.

6. Design a workflow.
7. From the **User Interface** section, drag and drop a **Form** task.
8. Under the **Form builder** tab, define the necessary form fields.
9. Under the **Hooks** tab, select the **Hook type** as **REST**.
10. From the **Hooks Inventory** dropdown, select the API name as defined before.
11. From the **Field ID** dropdown, select the field against which the hook should be mapped.



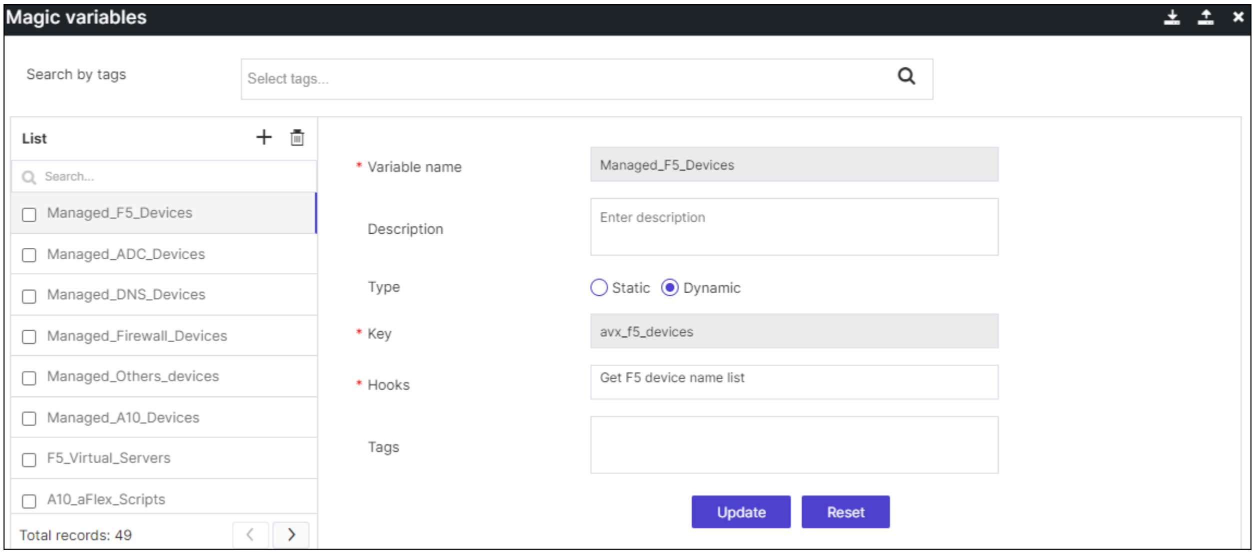
12. Click **Add**.
13. Under the **Form Builder** tab, click **(View)** icon.

    A preview of the form shows the hook added to the form field.



Magic Variable

This shows a list of all the Magic variables defined in various tasks across workflows.



- Accessing Magic Variables
- Defining a Hook and declaring it as a Magic Variable
- Using a Magic Variable within a Workflow

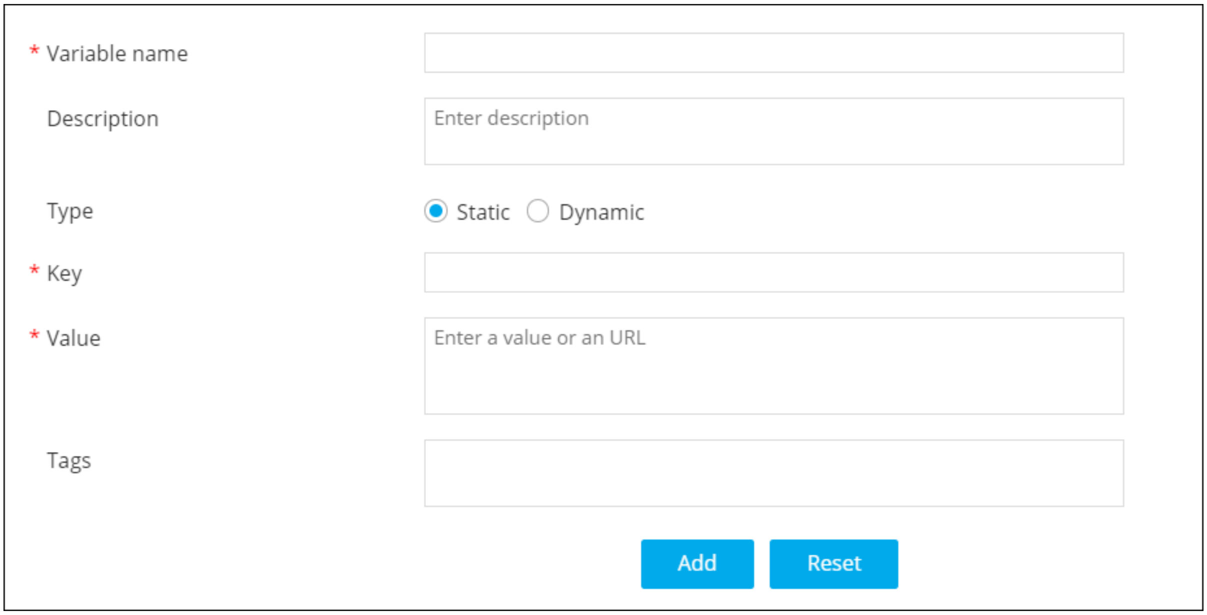
Accessing Magic Variables

New Menu	Old Menu
In the Automation module, from the navigation pane on the left, under **WORKFLOW**, select **Magic Variables**. The **Magic Variables** pop-up window is displayed.	1. From the main navigation menu, select **Studio** > **Workflow**. The **Workflow** page is displayed with the **Design** tab open by default. 2. From the navigation pane on the left, select **Magic Variables**. The **Magic Variables** pop-up window is displayed.
Note: For more information on how to switch between menus, click here.	

Defining a Hook and declaring it as a Magic Variable

You can define a common hook and refer it within a workflow form.

1. From the Hooks inventory, create a Hook using REST API or Script.
2. Go to the Magic Variables section.
3. In the **Magic variables** window, to define a new magic variable, click **(Add)** icon.
4. Enter or select the required field information.



Field descriptions for adding new magic variable

Field	Description
***Variable name**	Enter the name for this magic variable. For example: **Managed_F5_Devices**
Description	Enter a description for the magic variable.
Type	Select if the magic variable is • **Static:** Hooks which are static in nature can be referred as magic variables. For example, Slack webhook, ITSM change ticket URL etc. • **Dynamic:** Hooks that execute dynamically can be referred as magic variables. For example, Query to get list of devices, Query to get list of unused VIPs, Query to get list of managed F5 devices etc.
***Key**	Enter a unique key for the magic variable.
***Value**	Enter the values for the magic variable. These can either be comma-separated values or a URL.

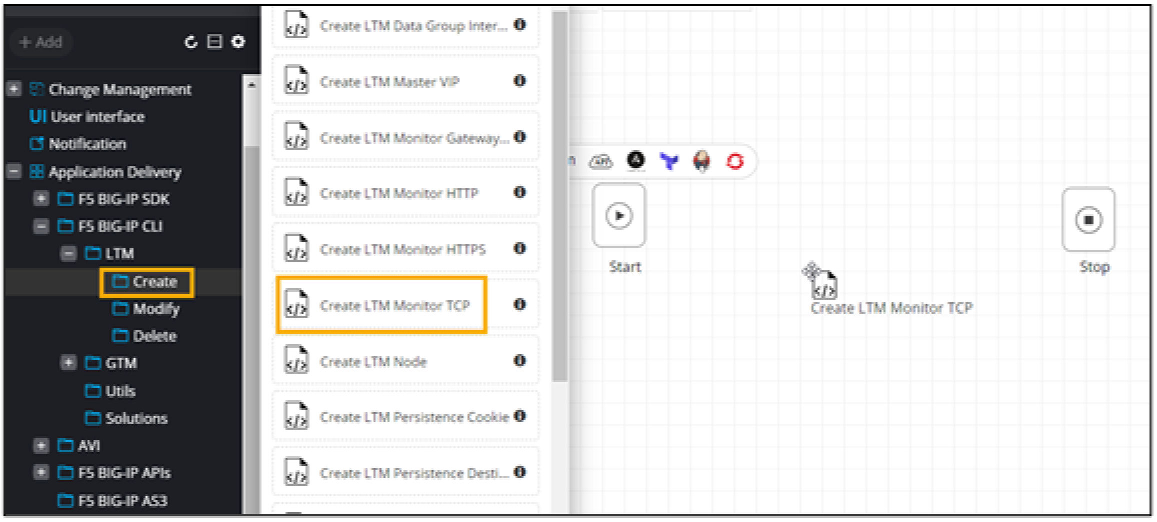
Field	Description
	Note: This field is displayed only when you select the **Type** as **Static**.
***Hooks**	Enter the name of the hook that you want to define as the magic variable. For example, **Get F5 device name list**. **Note:** This field is displayed only when you select the **Type** as **Dynamic**.
Tags	Enter tags to be associated with this magic variable, if any. This will help you to search for magic variables by tags.
Note: The asterisk (*) symbol indicates mandatory fields.	

5. Click **Add**.

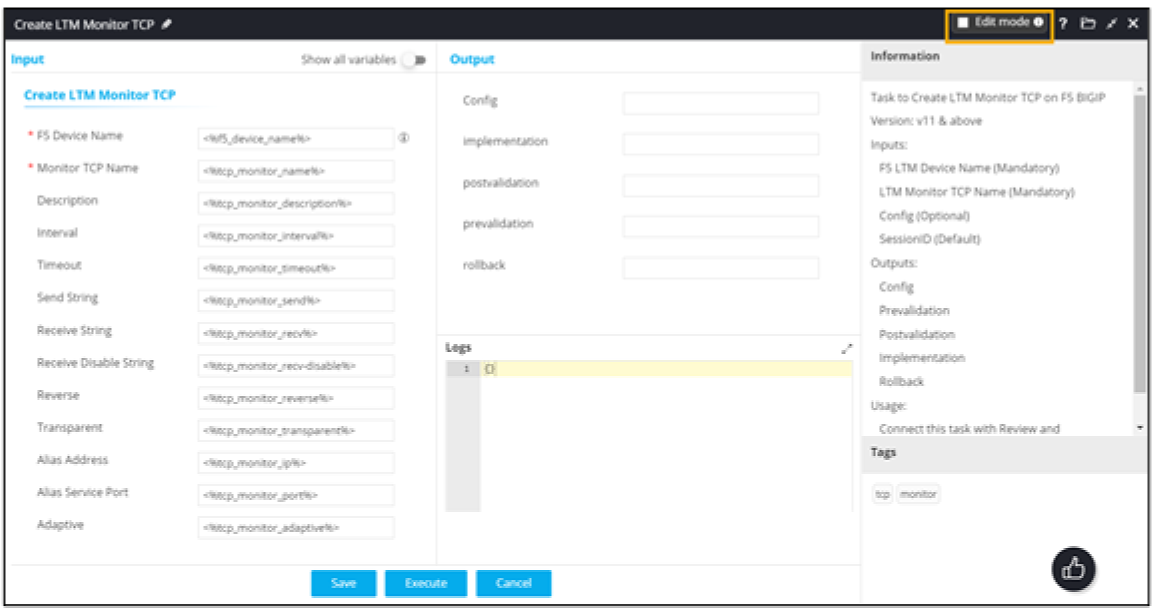
Using a Magic Variable within a Workflow

This section describes the steps to define a hook as a magic variable and use it in the Form task.

1. Design a workflow.
2. In the design workspace, from the menu on the left, navigate to **Application Delivery** > **F5 BIG-IP CLI** > **LTM** > **Create**.
3. From the **Create** folder, drag and drop the **Create LTM Monitor TCP** task.

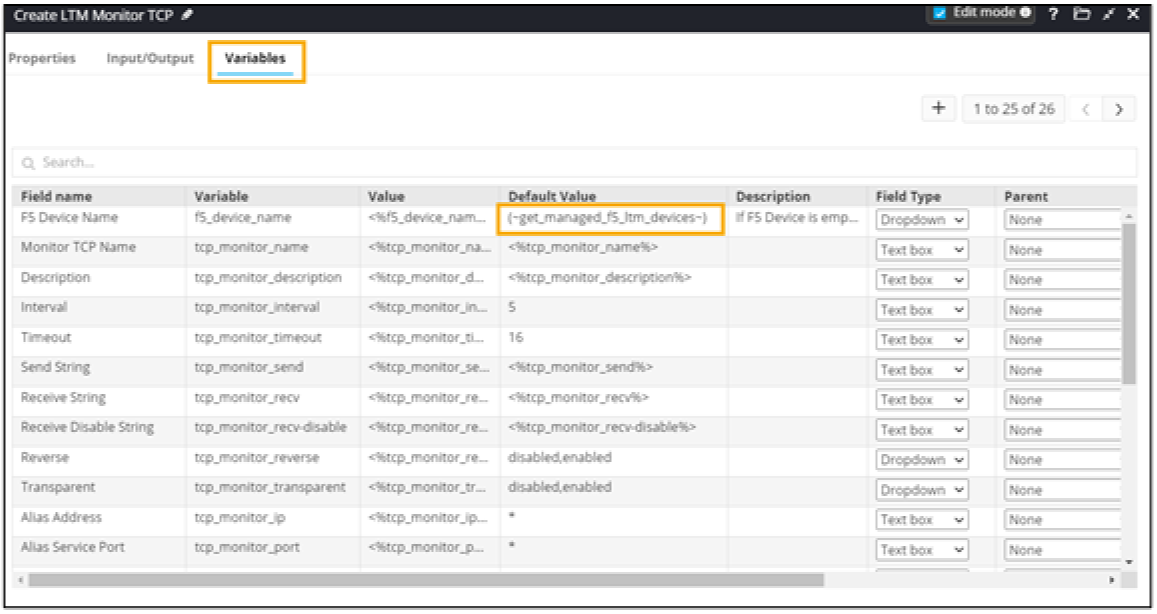


4. In the **Create LTM Monitor TCP** task window, select the **Edit mode** checkbox to see information on the variables used in this task.



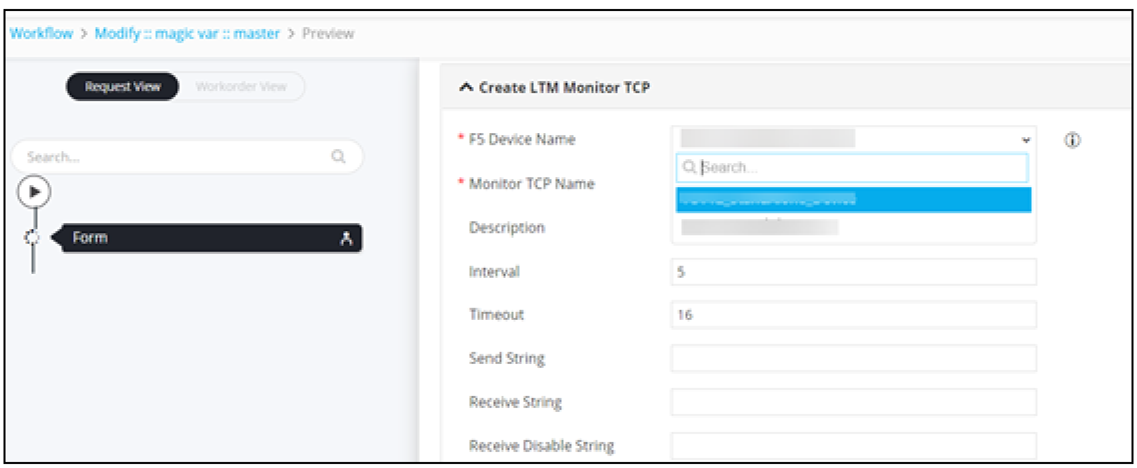
5. To see the magic variable used in this task, in the **Create LTM Monitor TCP** task window, click **Variables**.

   The magic variable `(~get_managed_f5_ltm_devices~)` retrieves the list of all managed F5 LTM devices.



6. Auto-generate a self-service form for this workflow.
7. Connect the workflow tasks and click **Preview**.

   The magic variable auto-populates the **F5 Device Name** form field.



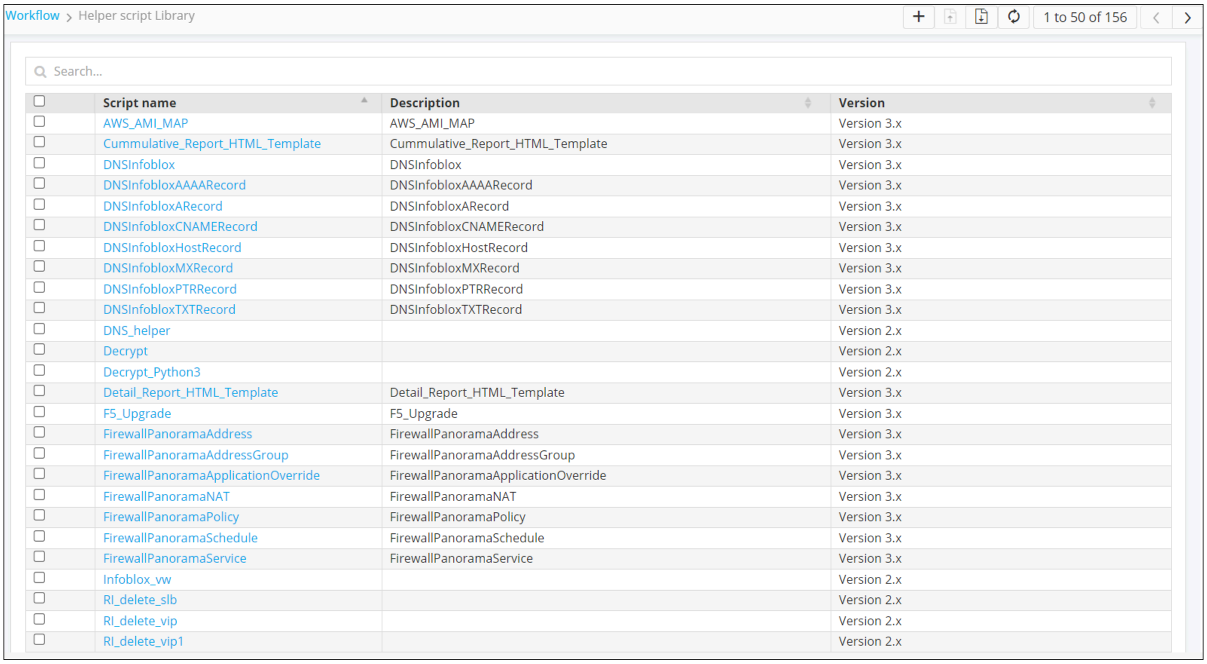
Helper Scripts

“Write once, reuse them forever…”

Helper scripts can be defined once with the relevant logic and can be reused across any workflow scripts with the relevant function name. One python script is essentially reused within another.

Frequently used methods and functions can be written as helper scripts once and reused across other custom scripts within the workflow. Either default helper script(s) can be leveraged and/or new helpers can be defined. For example, logic to establish database connection can be defined once and reused for any other database related queries via script.

The **Helper Script Library** can be accessed by navigating to **Automation** > **WORKFLOW** > **Helper script**



Options available on the Helper Script Library page

Option	Description
Search bar	Allows you to search for a particular helper script by typing in keyword(s).
+	Allows you to create a new helper script.
	Allows you to export a helper script.
	Allows you to import a helper script.
	Refreshes the pages.

- Accessing Helper Script Library
- Creating a Helper Script
- Exporting Helper Scripts
- Importing Helper Scripts

- Configuring RBAC for Helper Scripts
- Using a Default Helper Script

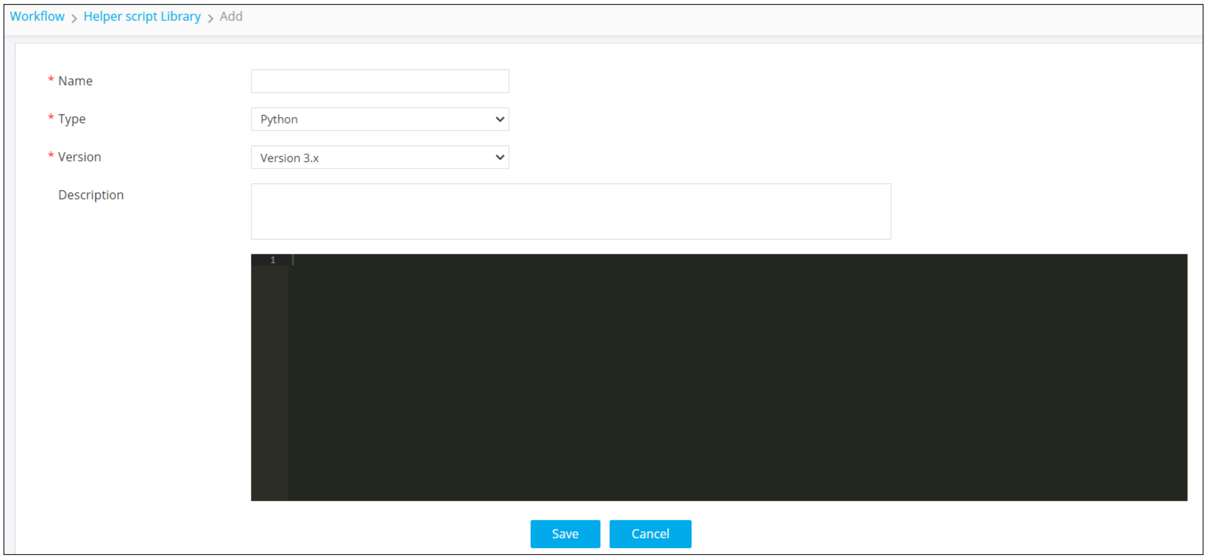
Accessing Helper Script Library

New Menu	Old Menu
In the Automation module, from the navigation pane on the left, under **WORKFLOW**, select **Helper Script**. The **Workflow** > **Helper script Library**page is displayed.	1. From the main navigation menu, select **Studio** > **Workflow**. The **Workflow** page is displayed with the **Design** tab open by default. 2. From the navigation pane on the left, select **Helper Script**. The **Workflow** > **Helper script Library**page is displayed.
Note: For more information on how to switch between menus, click here.	

Creating a Helper Script

1. Go to the Helper Script Library.
2. To create a new Helper Script, on the **Helper Script Library** page, click + **(Add)** icon in the command bar.

3. On the **Workflow** > **Helper Script Library** > **Add** page, enter or select the field information..

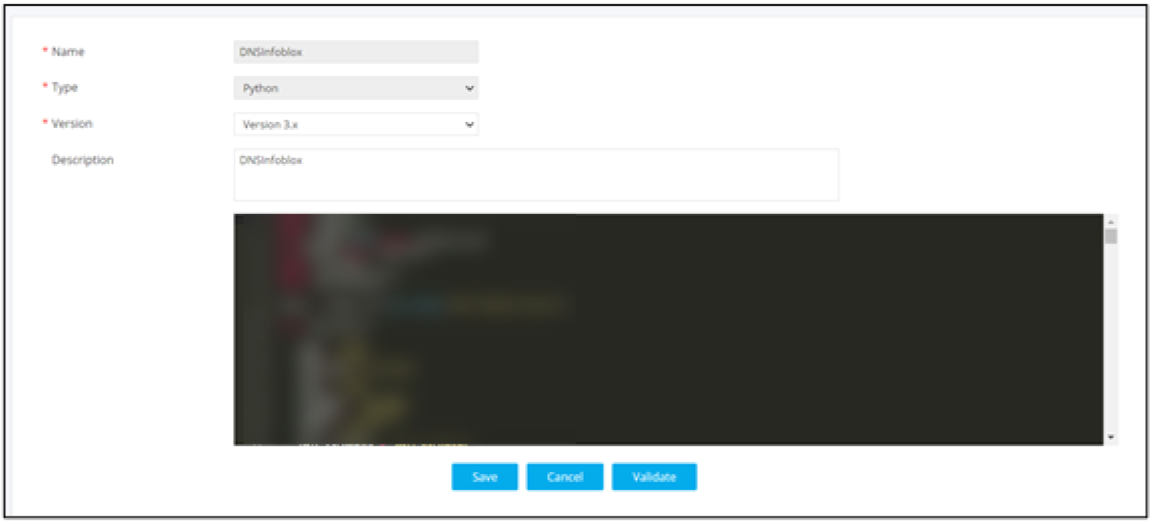


Field descriptions for adding new workflow to the helper script library

Field	Description
***Name**	Provide an appropriate name for the Helper Script.
***Type**	Select script as Python.
***Version**	Select version as 3.x
Description	Enter a description of the script.

Note: The asterisk (*) symbol indicates mandatory fields.

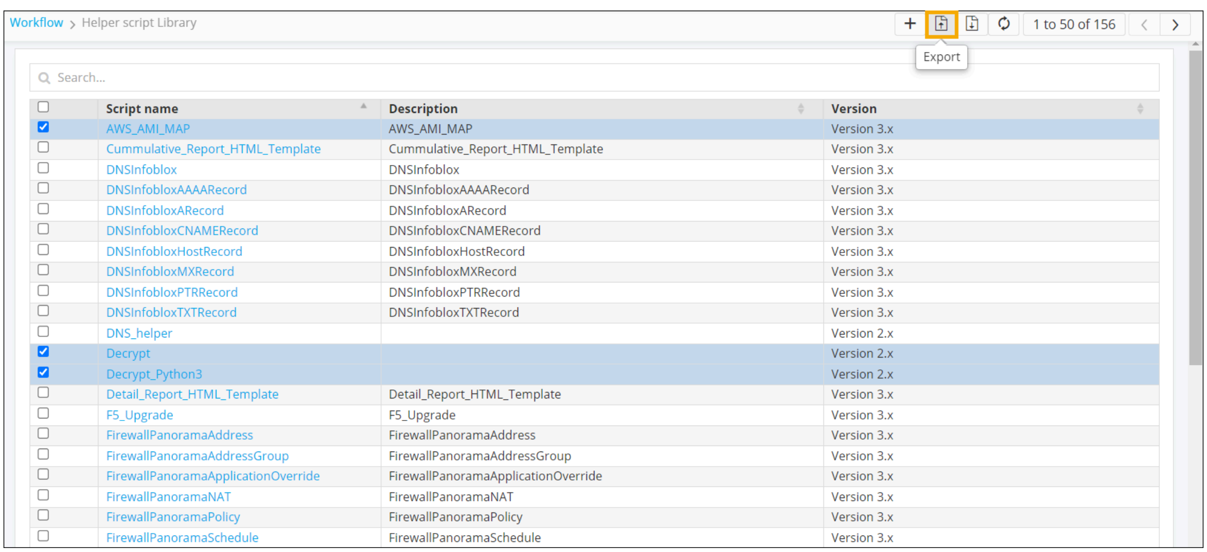
4. Define the script logic.



5. Click **Save**.

Exporting Helper Scripts

1. Go to the Helper Script Library.
2. On the **Helper Script Library** page, select the helper script(s) to be exported.
3. To export the selected helper script(s), click  **(Export)** icon in the command bar.

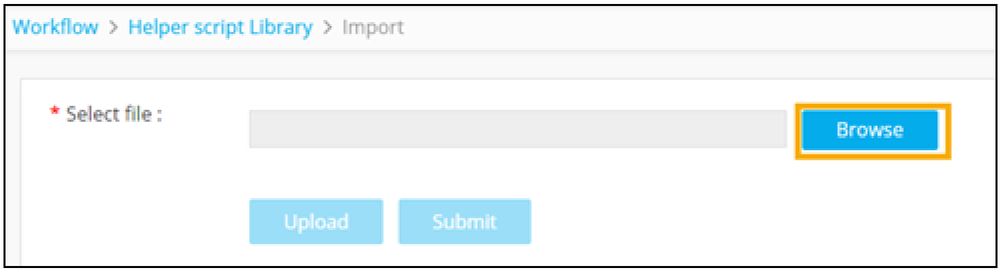


4. Click **Ok** in the **Confirmation** pop-up window.

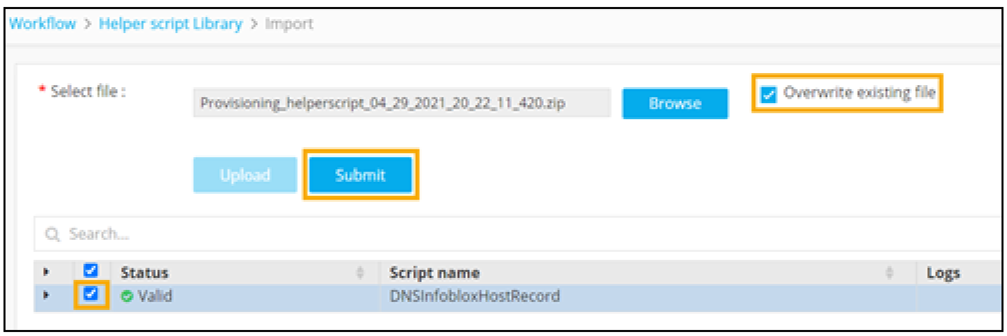
   The helper script(s) is/are saved to your computer as a <.zip> file.

Importing Helper Scripts

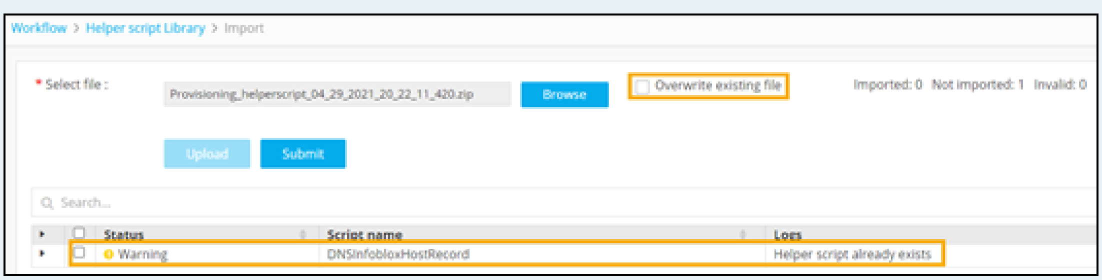
1. Go to the Helper Script Library.
2. On the **Helper Scripts Library** page, click **(Import)** icon in the command bar.
3. To select the helper script file to be uploaded, click **Browse**.



4. To upload the selected file, click **Upload**.
5. Select the script and click **Submit**.



> **Note:**
> - By default, the 'Overwrite existing file' option will be checked. Any existing helper scripts will be overwritten when new files are imported.
> - Only <.zip> files will be supported for import.
> - If the 'overwrite existing file' option is not selected and the script already exists in the destination environment, the following warning will be displayed to the user on clicking Submit.

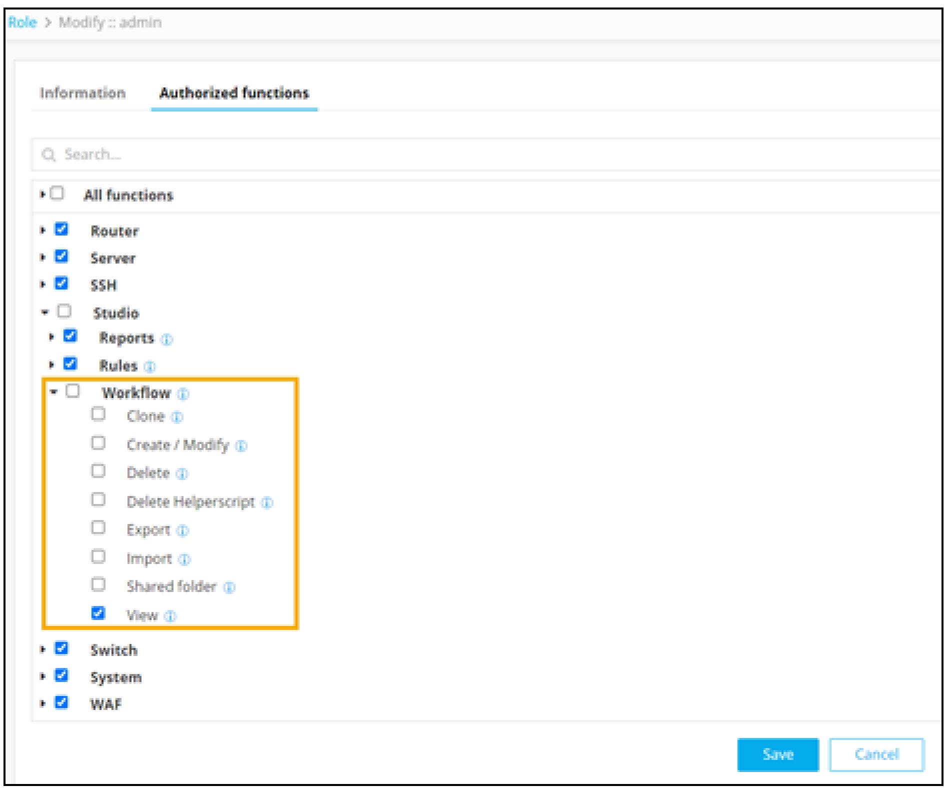


Configuring RBAC for Helper Scripts

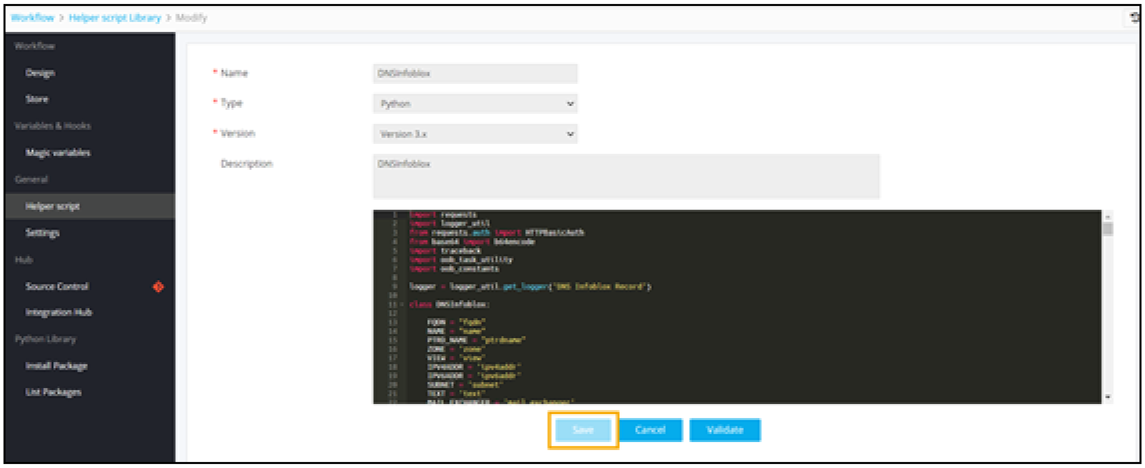
You can allow or restrict a user to modify Helper Scripts. Users can modify helper scripts only when they have the Create/Modify permission for workflows. Users with 'View' permission cannot modify helper scripts.

To restrict a user from modifying Helper Scripts in the Workflow Studio:

1. From the upper left corner of the screen, click **(Menu)** icon.
2. From the menu displayed, select **Platform** > **IDENTITY** > **Role**.
3. Select the role for which you want to enable access. For example, **admin**.
4. Under **Authorized functions**, select **Request** > **Workflow**.
5. Under **Workflow**, only select the **View** checkbox.



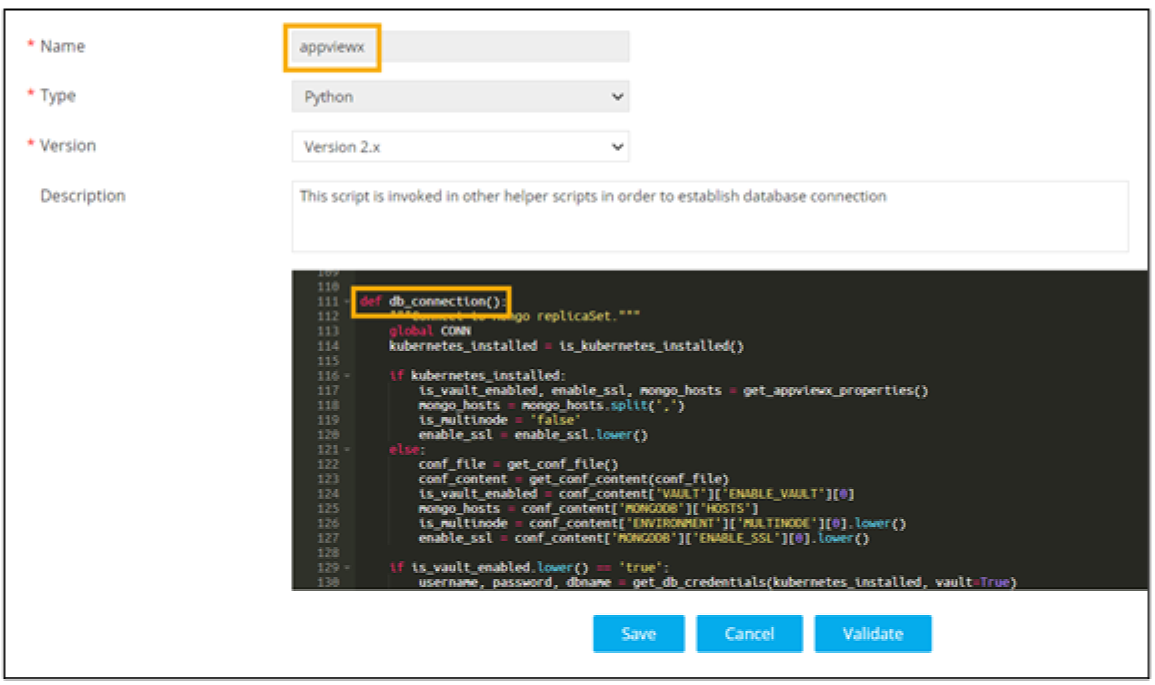
On the **Helper Script Library** > **Modify** page, the script editor and **Save** button are disabled.



Using a Default Helper Script

1. Leverage the helper script called "appviewx", which establishes database connection.

   Function name: `db_connection`



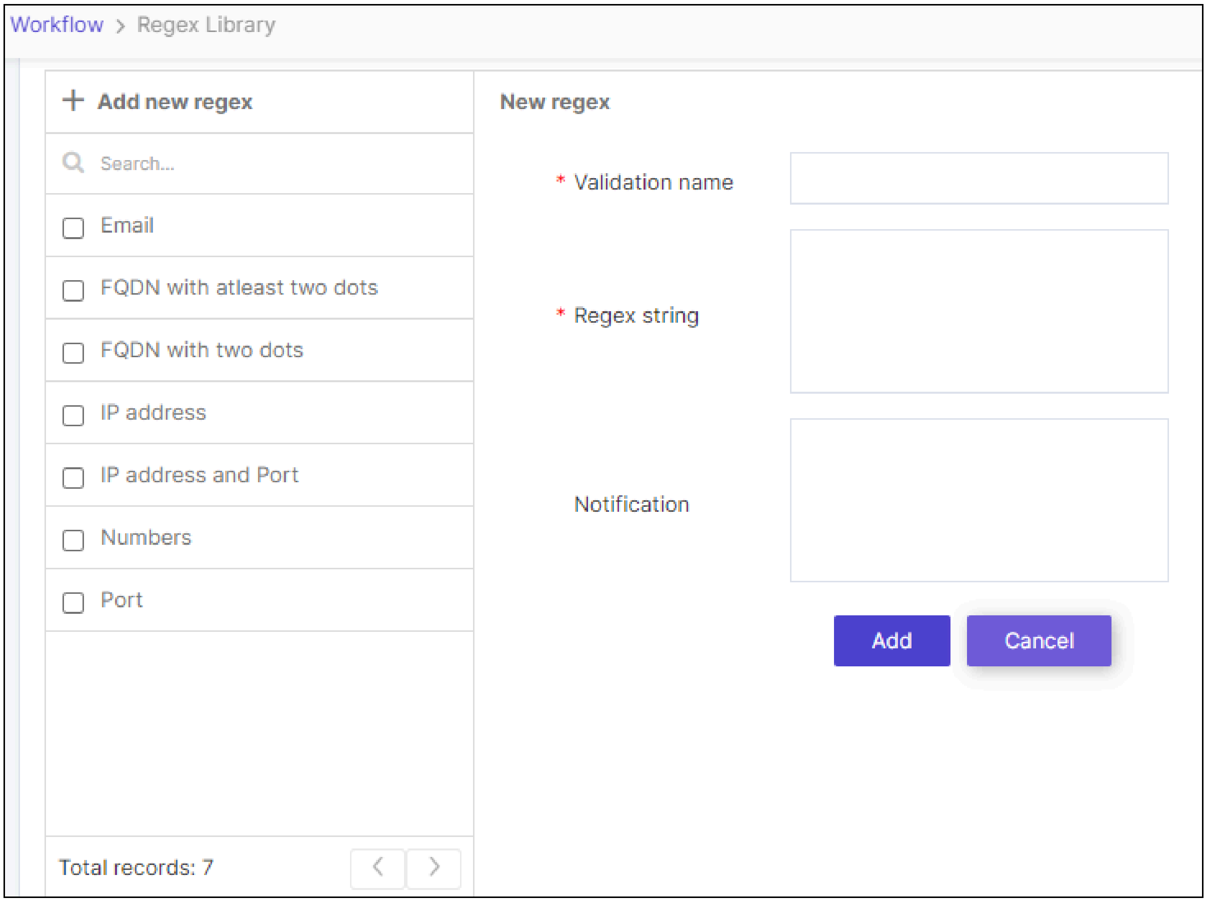
2. Reuse the predefined helper script across other workflow scripts such as ‘Get device list’.

   Get device associate script


```
import sys

import json

sys.path.insert(0, AVX::DEPENDENCIES)

sys.path.insert(0, AVX::HELPER)

import appviewx


def device_list():

    connection = appviewx.db_connection()  _--------------------------------> helper.helperfunction()

    collection = connection.appviewx.device

    dev_list = [value['name'] for value in collection.find({"vendor":"F5", "category" : "ADC"})]

    dev_list = sorted(filter(None,dev_list), key = lambda s: s.lower())

    print (json.dumps([{'Device': name} for name in dev_list])) if dev_list else (json.dumps({'error':"No Device to list"}))


device_list()
```

Regex Library

The Regex Library allows you to define custom regex validation or use the prebuilt regex patterns. Regex validations can be referenced against a form field with specific notifications to the user. You can add, edit any regex in the inventory.



- Accessing the Regex Library
- Adding a New Regex
- Modifying an Existing Regex

Accessing the Regex Library

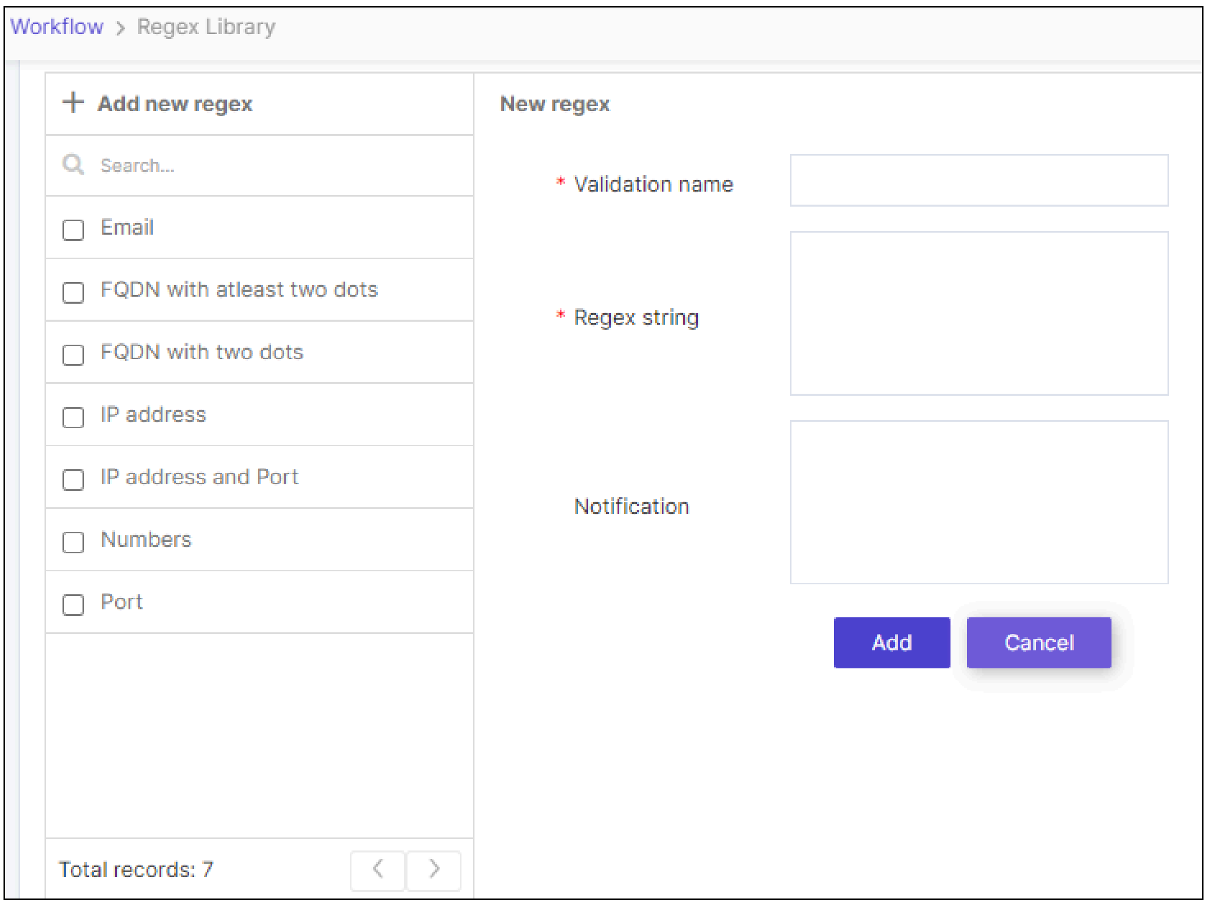
New Menu	Old Menu
In the Automation module, from the navigation pane on the left, under **WORKFLOW**, select **Regex Library**. The **Workflow** > **Regex Library**page is displayed.	1. From the main navigation menu, select **Studio** > **Workflow**. The **Workflow** page is displayed with the **Design** tab open by default. 2. From the navigation pane on the left, select **Regex Library**.

New Menu	Old Menu
	The **Workflow** > **Regex Library**page is displayed.

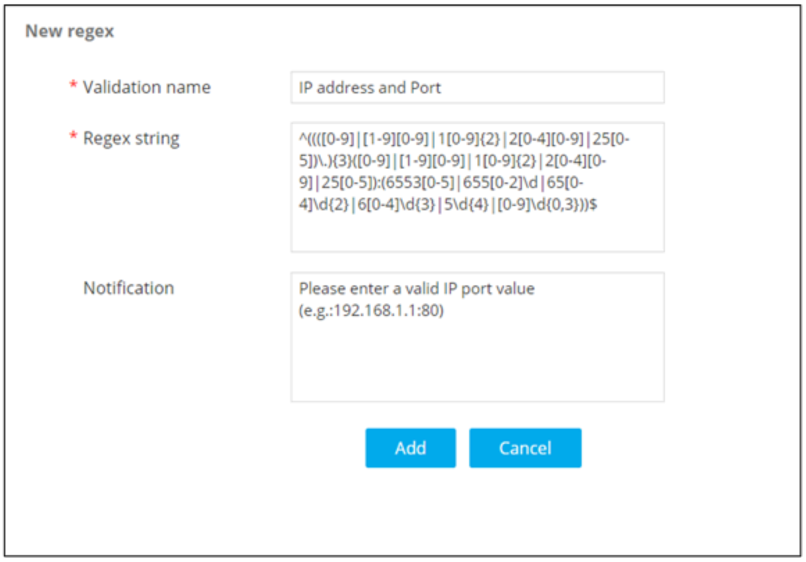
Note: For more information on how to switch between menus, click here.

Adding a New Regex

1. Go to the Regex Library.
2. To add a new regex string, on the **Workflow** > **Regex Library** page, click **Add new regex**.



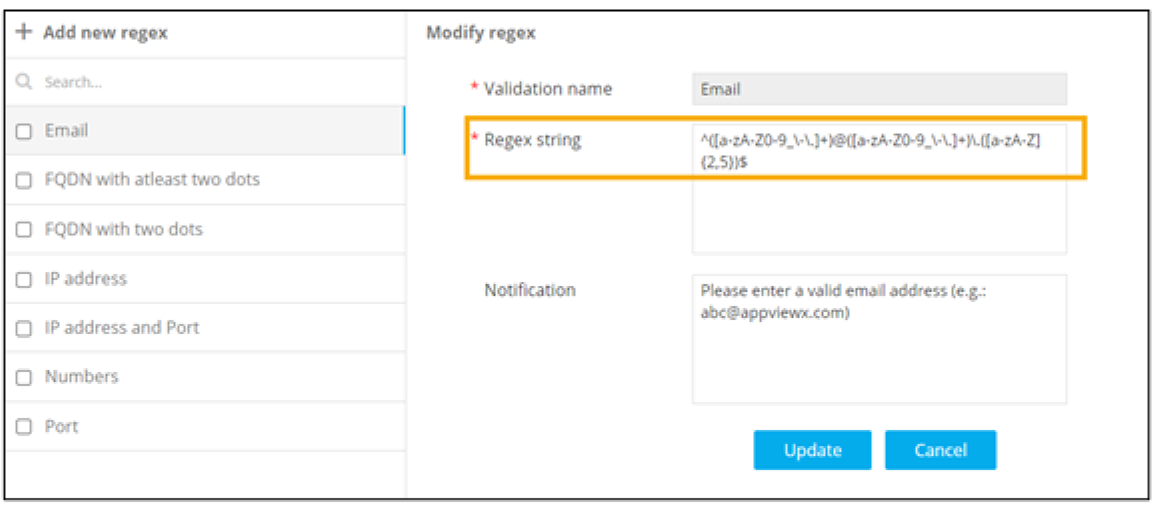
3. Enter the required field information as shown.



4. To save this new regex script to the library, click **Add**.

Modifying an Existing Regex

1. Go to the Regex Library.
2. On the **Workflow** > **Regex Library** page, select an existing Regex from the list. For example: **Email**.
3. To modify the **Regex string**, enter a different regex pattern for email validation.



4. To save this modified regex pattern, click **Update**.

Request Settings

You can request specific customized request settings, allowing tailored configurations for efficient and personalized automation workflows.

To define the request settings:

New Menu	Old Menu
1. Form the main navigation menu, select **Automation**. 2. In the **Automation** module, from the left menu, click **WORKFLOW** > **Request Settings**.	1. Form the main navigation menu, select **Settings**. 2. On the **Design** page, **Request Settings**popup is displayed.
Note: For more information on how to switch between menus, click here.	

1. When you click **Request Settings** the **Settings** popup page is displayed.
2. Under the **Request Settings** section, configure the request settings based on request execution access, assigned request filters, and role-based access.
3. In the **Request Execution Access** field, select the execution access to **All User** or **Requestor** or **Dynamic Auth**.
   - If execution access is set to **All User**, every user with all assigned resources can view the request.
   - If execution access is set to **Requestor**, all users with submitter resource can view the request.
   - If execution access is set to **Dynamic Auth**, users with dynamic authentication can only view the request.
4. In the **Filter Assigned Request** field, select the filter assigned to **All User** or **Requestor** or **Dynamic Auth**.
   - If filter assigned is set to **All User**, every user with all assigned resources can view the request.
   - If filter assigned is set to **Requestor**, all users with submitter resource can view the request.
   - If filter assigned is set to **Dynamic Auth**, users with dynamic authentication can only view the request.
5. Turn on the **Role Based Access** toggle to view role based access to the request.
6. Click **Save**.

   The **Request Settings** will be updated successfully, and only the customized requests will be displayed on the Workflow page.

Provisioning for Workflow Tasks

You can define specific configurations for executing the provisioning workflow tasks such as Implementation, Prevalidation, Postvalidation, and Rollback.

To define the configurations:

New Menu	Old Menu
1. Form the main navigation menu, select **Automation**. 2. In the **Automation** module, from the left menu, click **WORKFLOW** > **Provisioning**.	1. Form the main navigation menu, select **Settings**. 2. On the **Settings** module, from the left menu, click **Provisioning**.
Note: For more information on how to switch between menus, click here.	

1. Under the **Work order configuration** section, define the configurations for Both or Manual or Auto of the **Implementation type**.
2. In the **Implementation type** field, select the Implementation type from the dropdown.
3. Under the **Command configuration** section, define the configurations for success or failure of the implementation task.
4. In the **Failure messages** field, enter the keywords for error responses to identify the task as failed as shown in the image below.

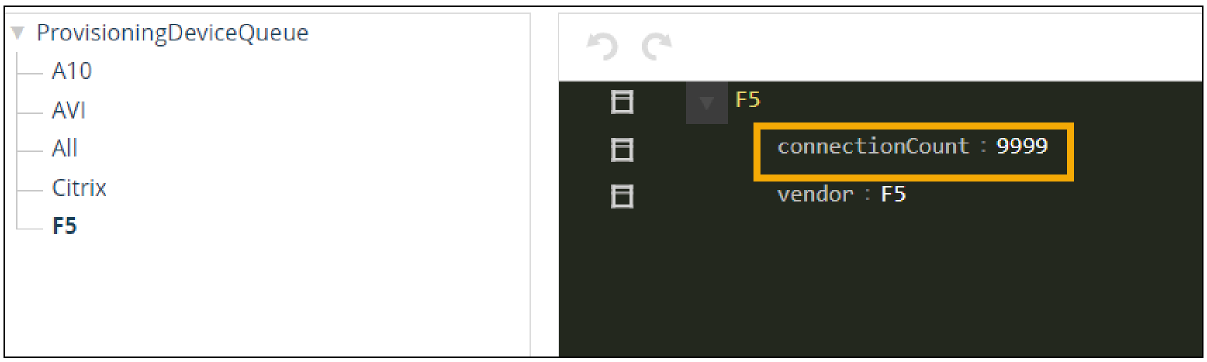
If the workflow execution displays any of the keywords defined above as error messages, the task will be considered as failed.

5. In the **Excluded failure messages** field, enter the keywords to be excluded for identifying the task as failed.

6. In the **Commands to wait** field, you can enter a set of commands and define a time out for the implementation task. If the task contains the configured commands, then it will time out as per the time out defined. This is useful for tasks or commands that take longer than the default time to execute.
7. In the **Time out (mins)** field, enter the timeout value for the tasks having the configured set of commands.

Note: The default value is 0.5 minutes or 30 seconds.

8. Turn on the **Continue on a command failure** toggle, to proceed in case of a command failure.
9. Turn on the **Enable Device Queue** toggle to enable queuing of device connections as per the defined connection count in the Collection.



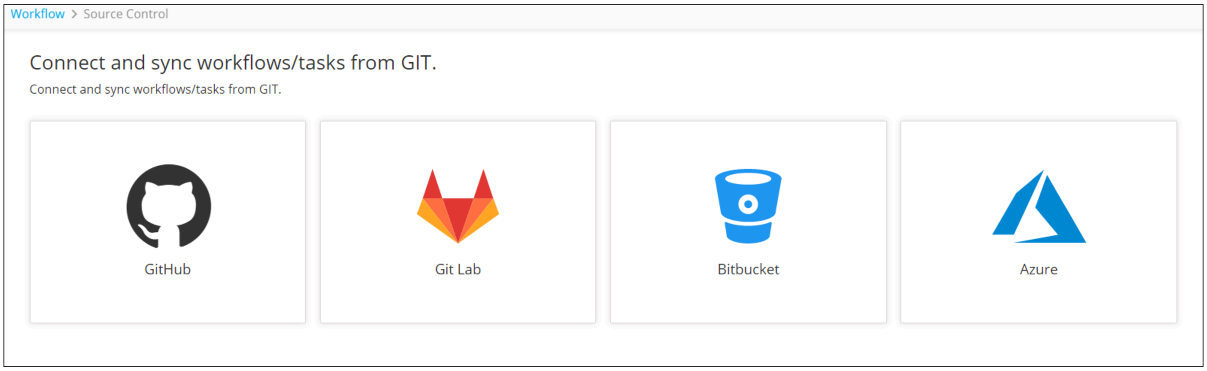
10. Turn on the **Enable Device Session Cache** toggle, to check for any existing open sessions for the device and execute other requests in the same session.
11. Turn on the **Rollback work order** toggle to rollback configurations that are partially implemented.

Note: Configurations that were successfully implemented will be excluded from the rollback work order.

12. Under the **Rest configuration** section, define the different success status codes in the **Rest success status code** field.
13. Under the **Collection configuration** section, turn on the **Enable default collection** toggle to create a collection in the name of the device and store all the IPs in the device (using a predefined parsing logic) during the next device parsing.

Source Control

This feature allows you to connect and sync workflows/tasks from GIT. You can integrate with GitHub, GitLab, Bitbucket, and Azure through AppViewX.



- Prerequisites
- Accessing the Source Control Page
- Linking a Repository
- Committing Workflows from the Workflow Page
- Committing Workflows from the Design Canvas
- Committing Tasks and Subflows
- Pushing to a Remote Repository from Workflow Page
- Pushing to a Remote Repository from the Design Canvas
- Pulling from the Repository from Workflow Page
- Pulling from the repository from the Design Canvas
- Unlinking a Repository
- Logs

Prerequisites

- Users should have role based access permission to the workflow studio.
- The user should have a configured remote GIT repository.

Accessing the Source Control Page

<table>
<tr><th>New Menu</th><th>Old Menu</th></tr>
<tr><td>In the Automation module, from the navigation pane on the left, under WORKFLOW, select Source Control.

The Workflow > Source Control page is displayed.</td><td>1. From the main navigation menu, select Studio > Workflow. The Workflow page is displayed with the Design tab open by default.
2. From the navigation pane on the left, select Source Control.

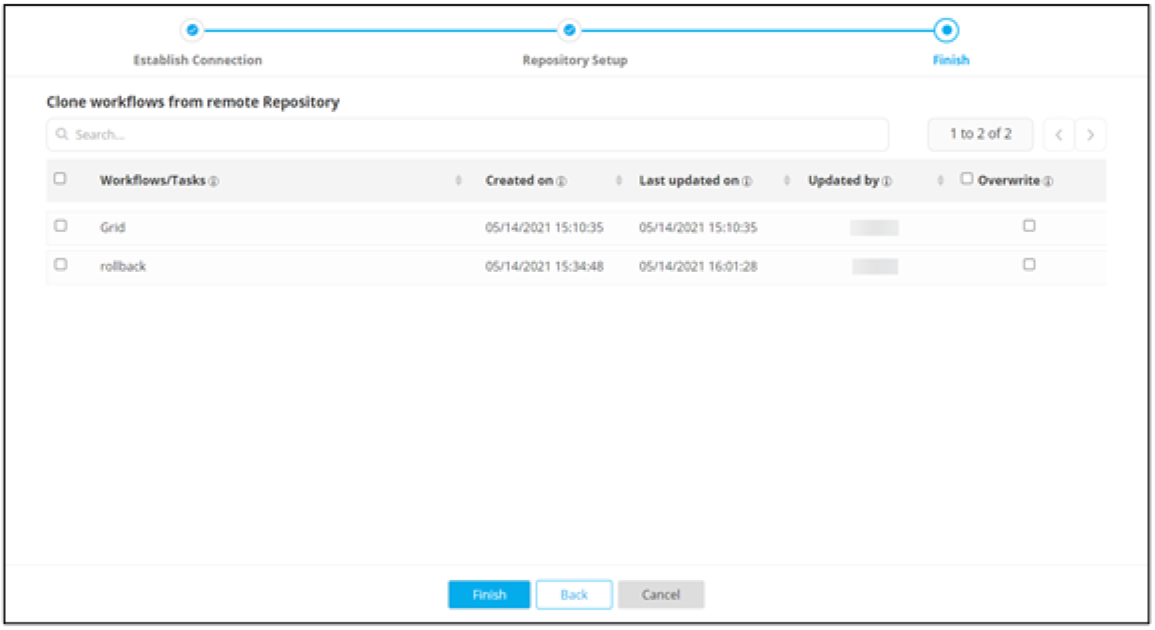
The Workflow > Source Control page is displayed.</td></tr>
<tr><td colspan="2">Note: For more information on how to switch between menus, click here.</td></tr>
</table>

Linking a Repository

1. Go to the Source Control page.
2. On the **Source Control** page, select the integration type as **GitLab**.
3. Select the connection type as HTTPS or SSH.
4. For **HTTPS**, enter the valid credentials along with the HTTPS clone **URL**.
5. For **SSH**, enter the valid private key corresponding to the public key and SSH clone URL.

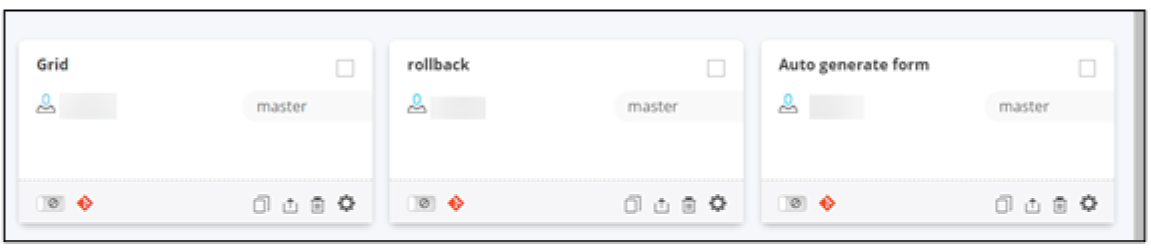
> **Note:** Prerequisite for SSH - Public key is to be added to the remote user account.

6. From the **Branches** dropdown, select **master**.
7. Click **Next**.
   All available workflows, tasks and subflows in the remote repository are displayed.

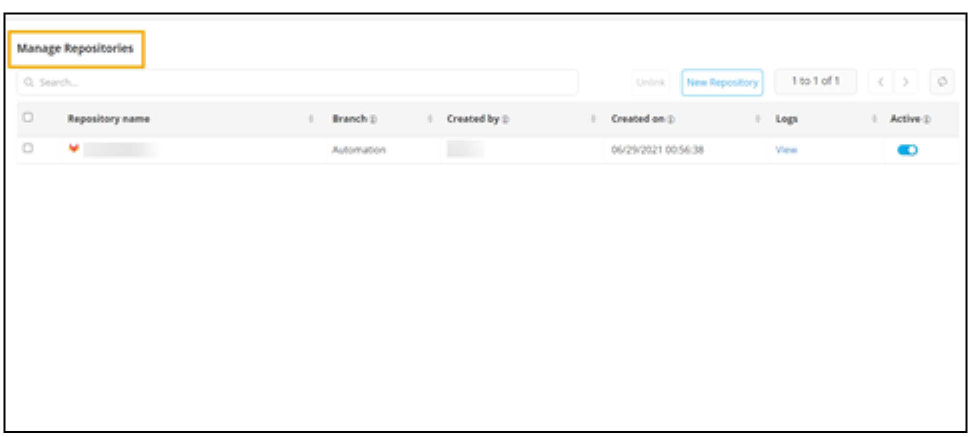


8. Select the workflows to be cloned and click **Finish**.
9. In the **Confirmation** pop-up window, click **Yes**.
   - Cloned tasks and subflows are added to Source Control Directory and Workflows to the inventory.

     The connected workflows reflect the Source Control logo .

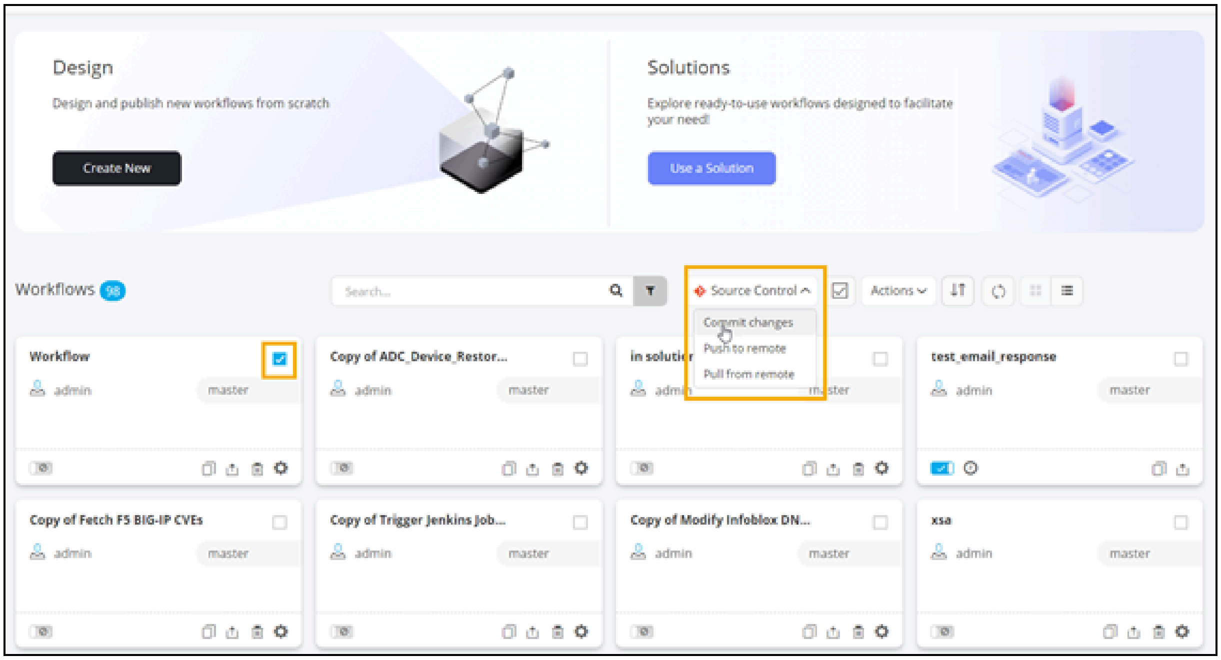


   - Cloned Repository is added to the **Manage Repositories** page.

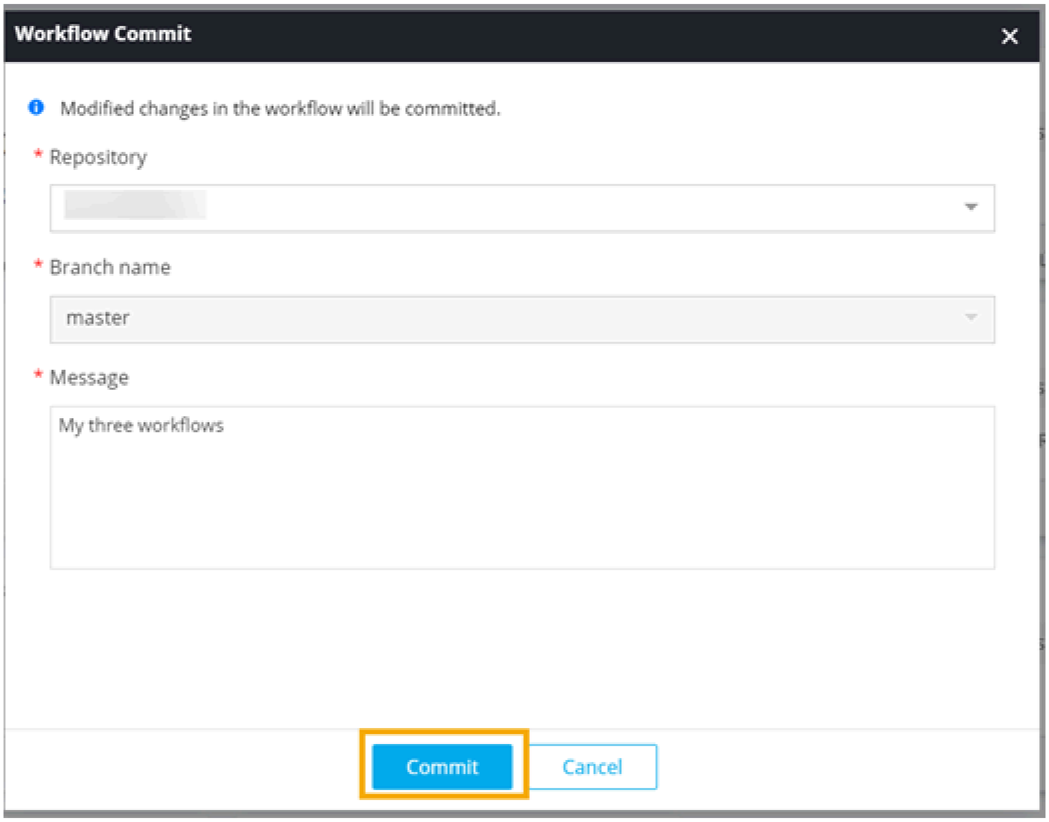


Committing Workflows from the Workflow Page

1. Go to the Workflow page.
2. On the **Workflow** page, select the workflows to be committed.
3. From the **Source Control** dropdown menu, select **Commit changes**.

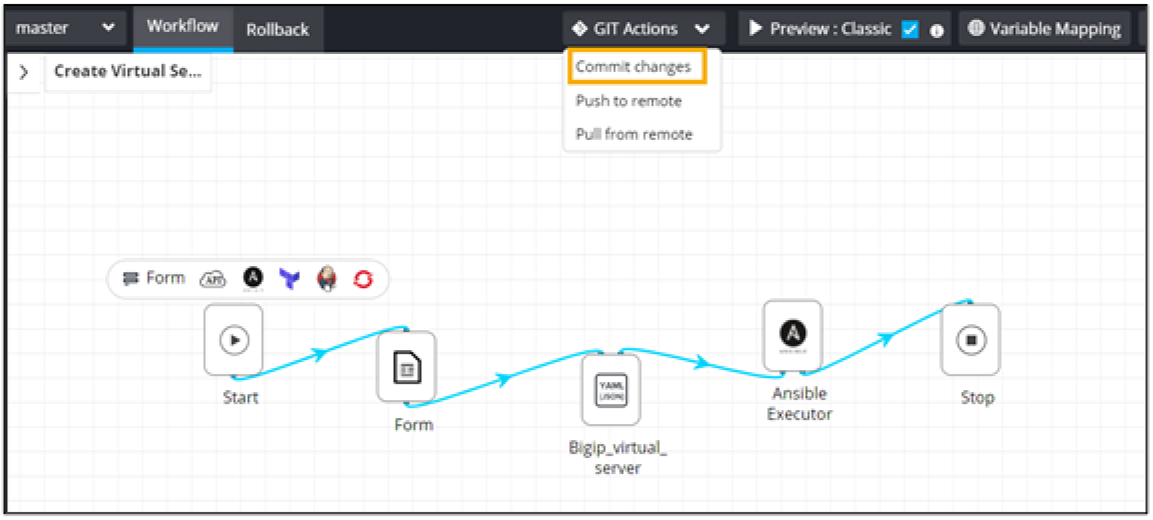


4. In the **Workflow Commit** window, enter a commit **Message** and click **Commit**.



Committing Workflows from the Design Canvas

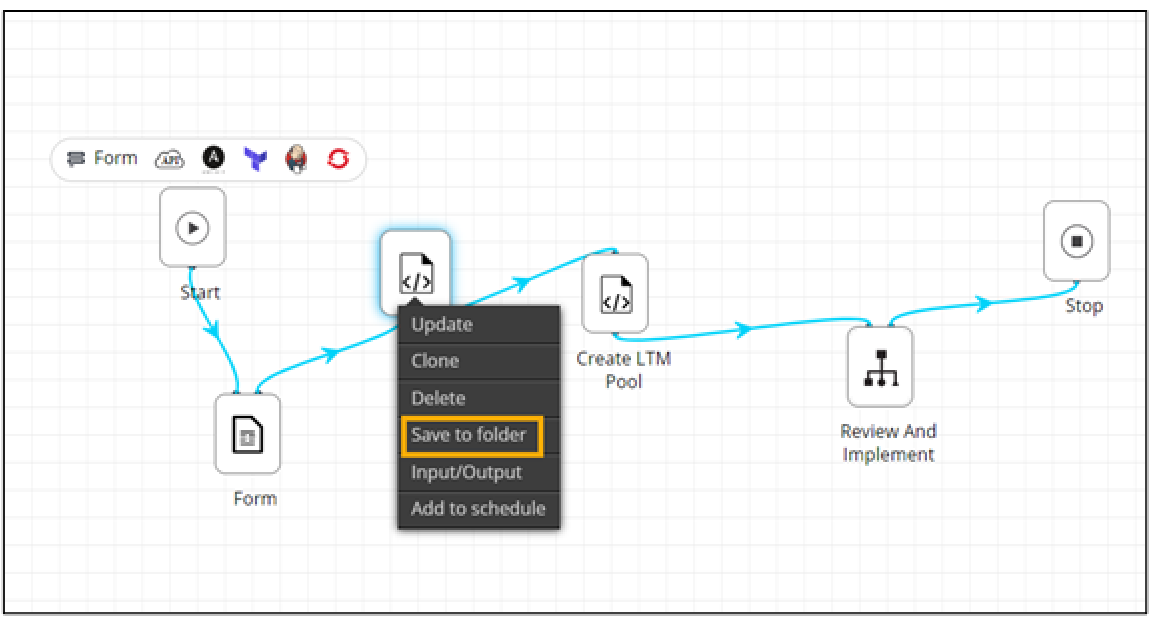
1. Go to the Workflow page.
2. Design a workflow or open an existing workflow that you want to commit to Git.
3. Form the GIT Actions dropdown, select **Commit changes**.



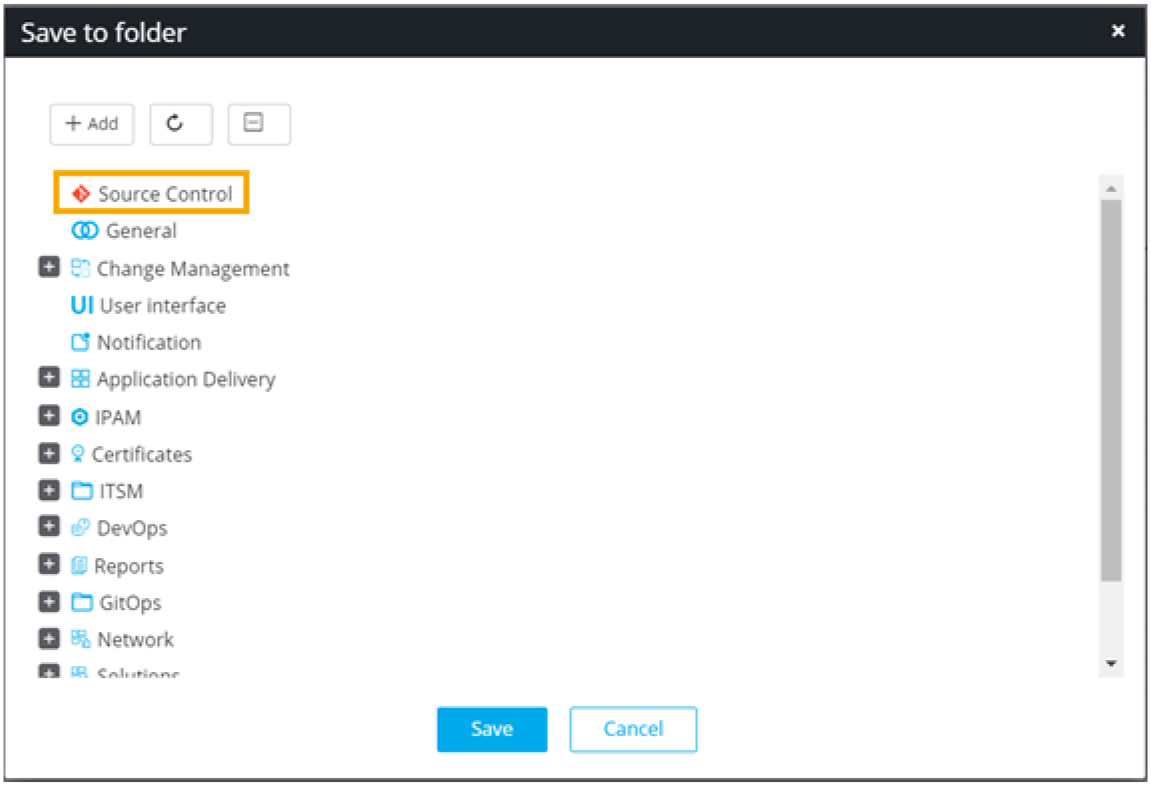
4. In the **Workflow Commit** window, enter a commit **Message** and click **Commit**.

Committing Tasks and Subflows

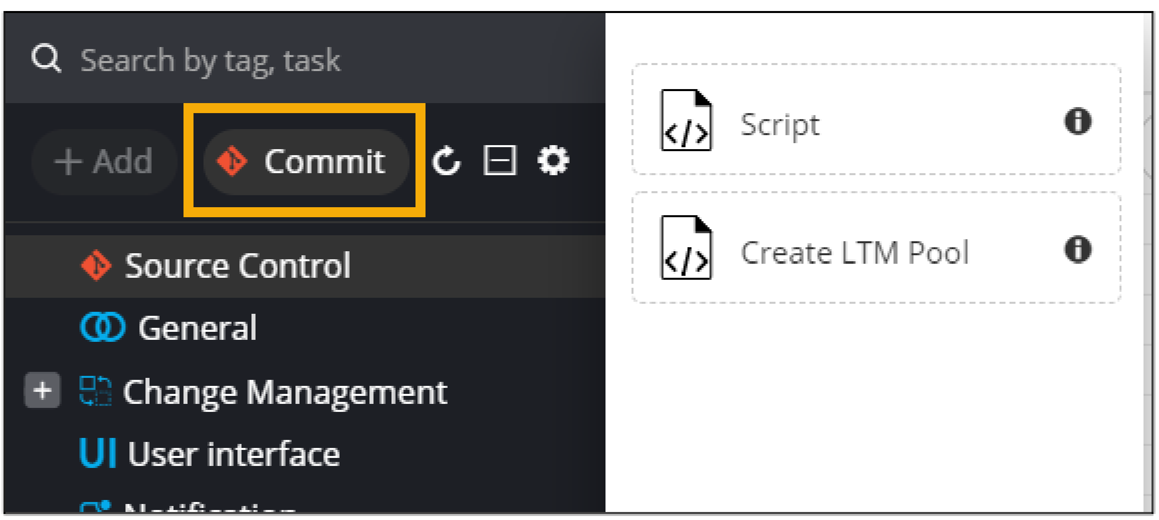
1. Go to the Workflow page.
2. Design a workflow or open an existing workflow.
3. Right-click on a task to see the options.
4. From the options displayed, click **Save to Folder**.



5. In the **Save to folder** window, save the task to the **Source Control** directory.

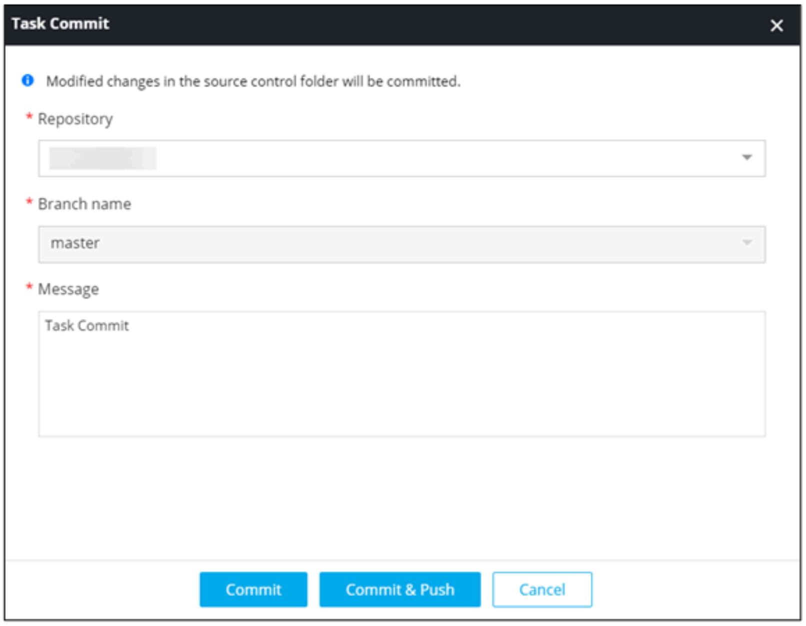


6. Click **Save**.
7. Add the tasks and subflows to be committed to the Source Control Directory.
8. Click **Commit**.



Note: By default, the Commit button is disabled. It is enabled only after selecting the Source Control Directory to save tasks.

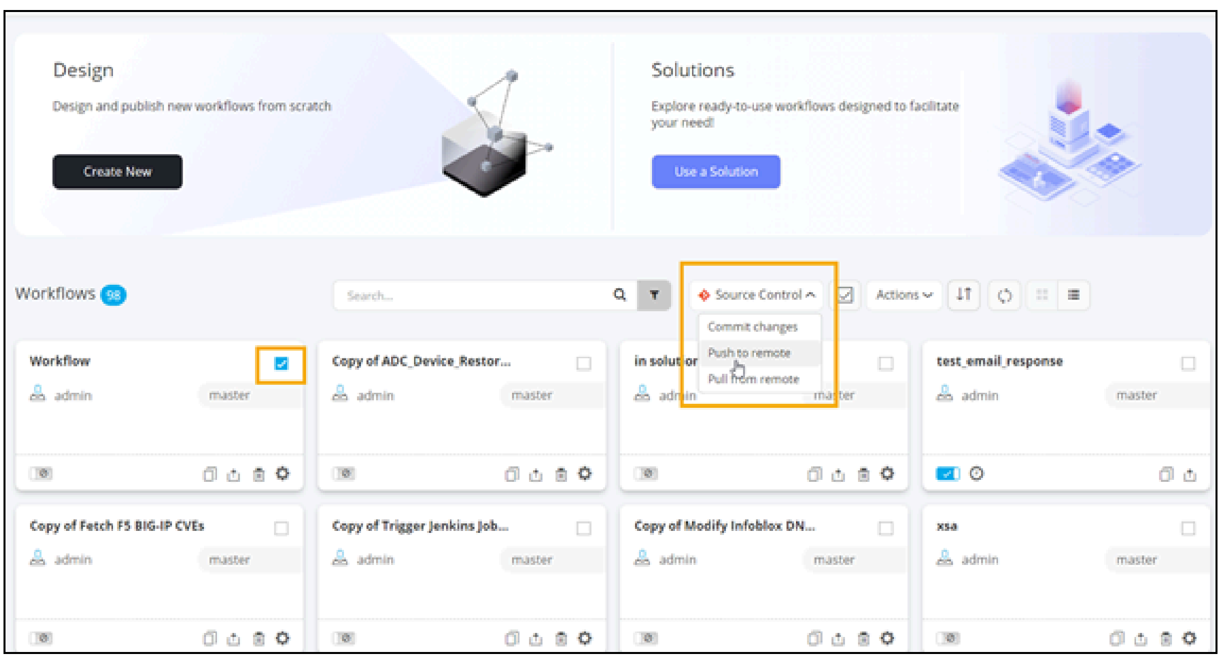
9. In the **Task Commit** window, enter a commit **Message** and click **Commit**.



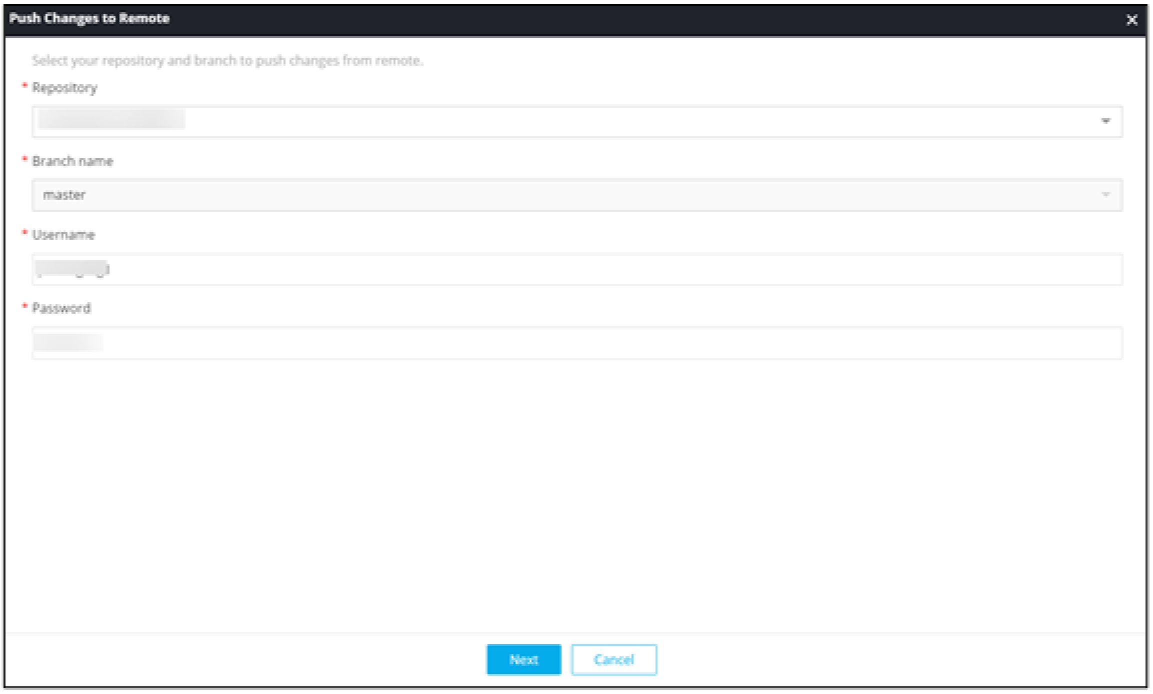
Note: Only the recently added tasks/subflows will be committed.

Pushing to a Remote Repository from Workflow Page

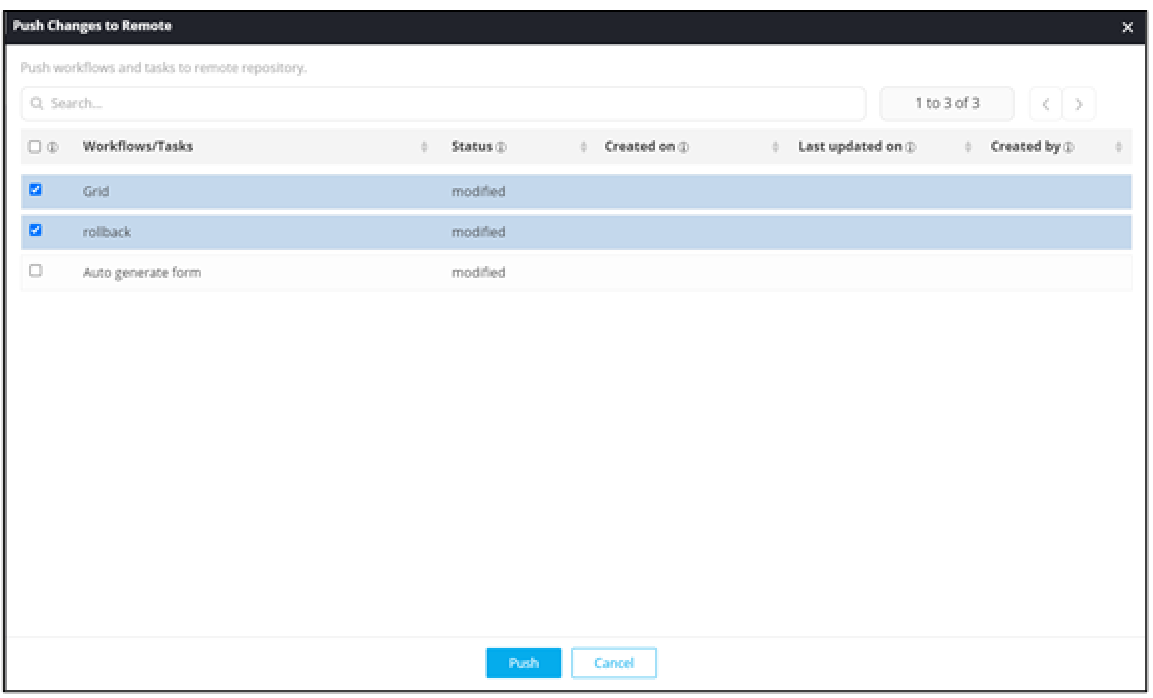
1. Go to the Workflow page.
2. On the **Workflow** page, from the **Source Control** dropdown, select **Push to remote**.



3. In the **Push Changes to Remote** window, enter the **Password**.



4. Click **Next**.
5. In the **Push Changes to Remote** window, from the list of committed tasks, select the task(s) to be pushed.

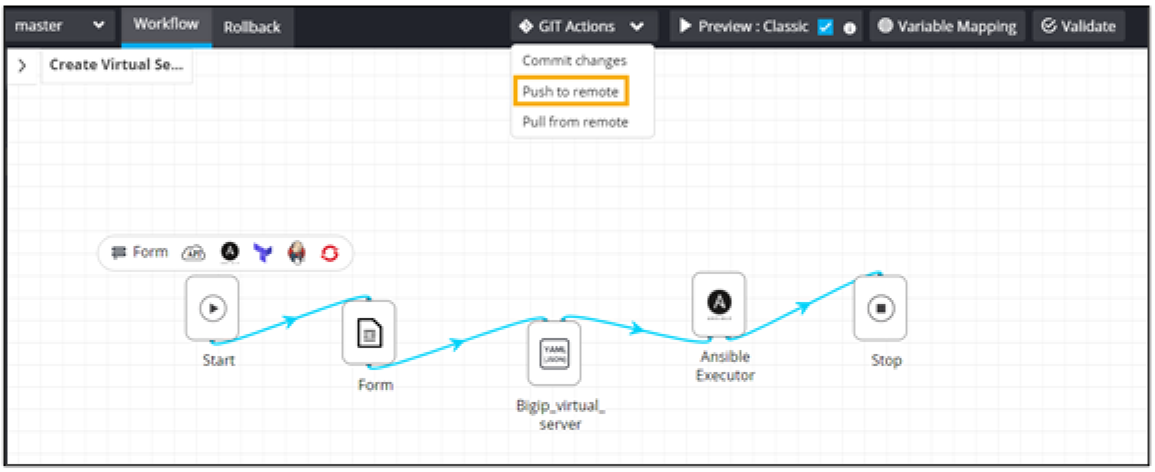


6. Click **Push**.

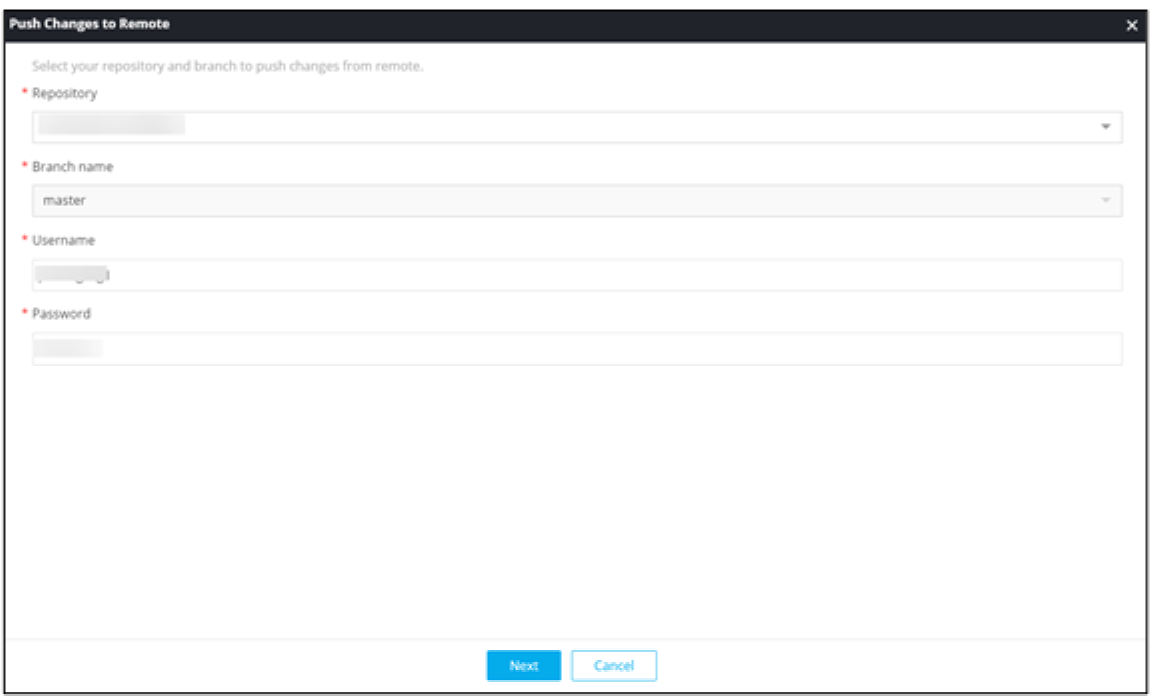
   The selected workflows/tasks are pushed successfully.

Pushing to a Remote Repository from the Design Canvas

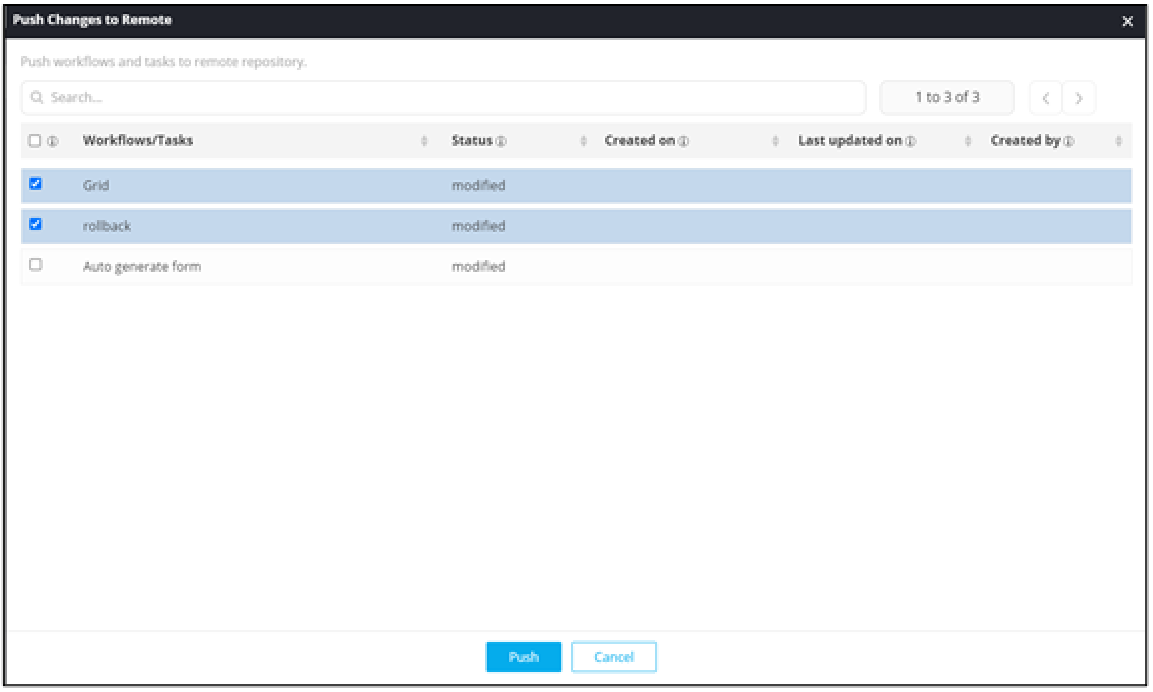
1. Go to the Workflow page.
2. Design a workflow or open an existing workflow that you want to push to Git.
3. From the **GIT Actions** dropdown, select **Push to remote**.



4. In the **Push Changes to Remote** window, enter the **Password**.



5. In the **Push Changes to Remote** window, from the list of committed tasks, select the task(s) to be pushed.

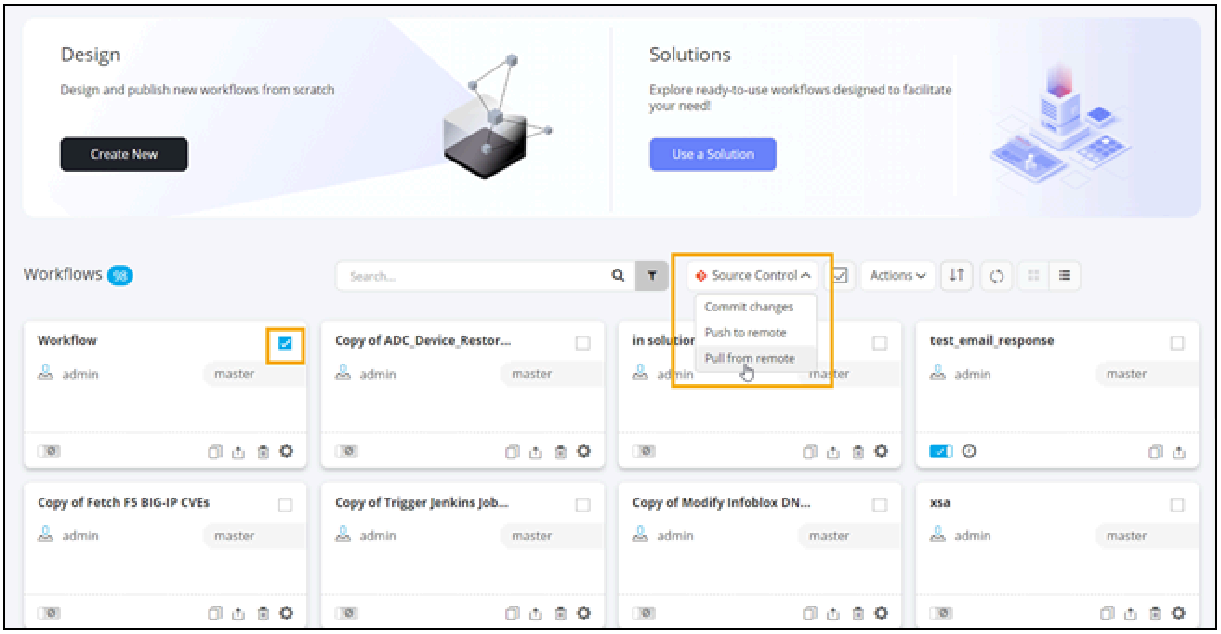


6. Click **Push**.

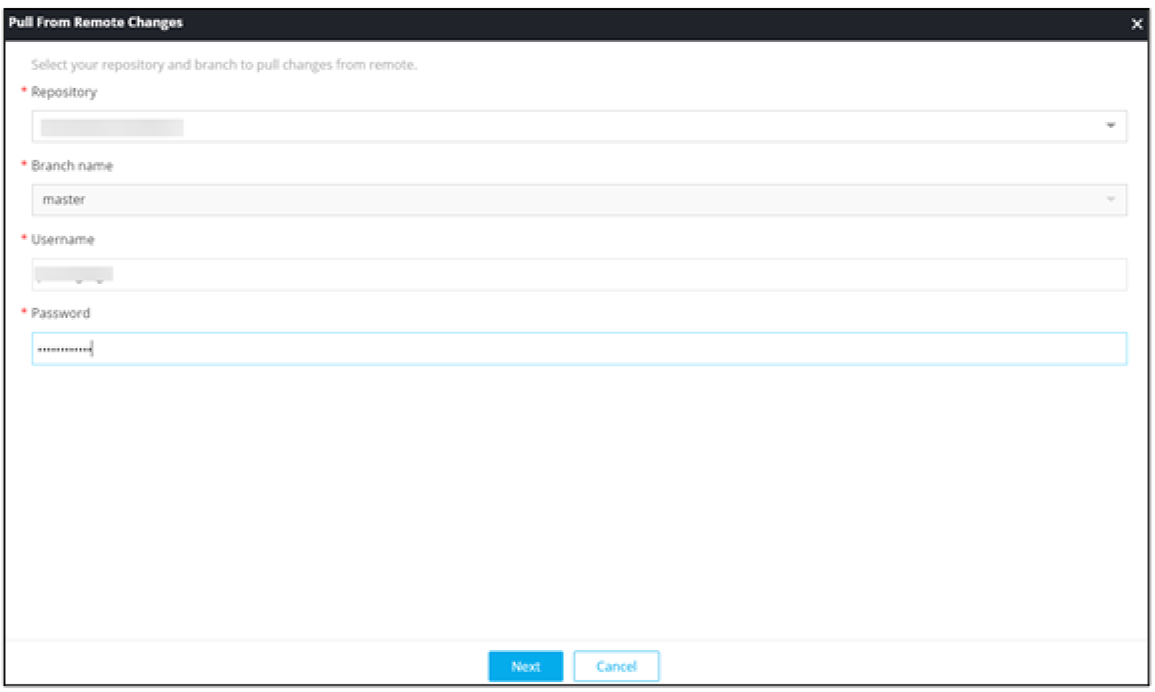
   The selected workflows/tasks are pushed successfully.

Pulling from the Repository from Workflow Page

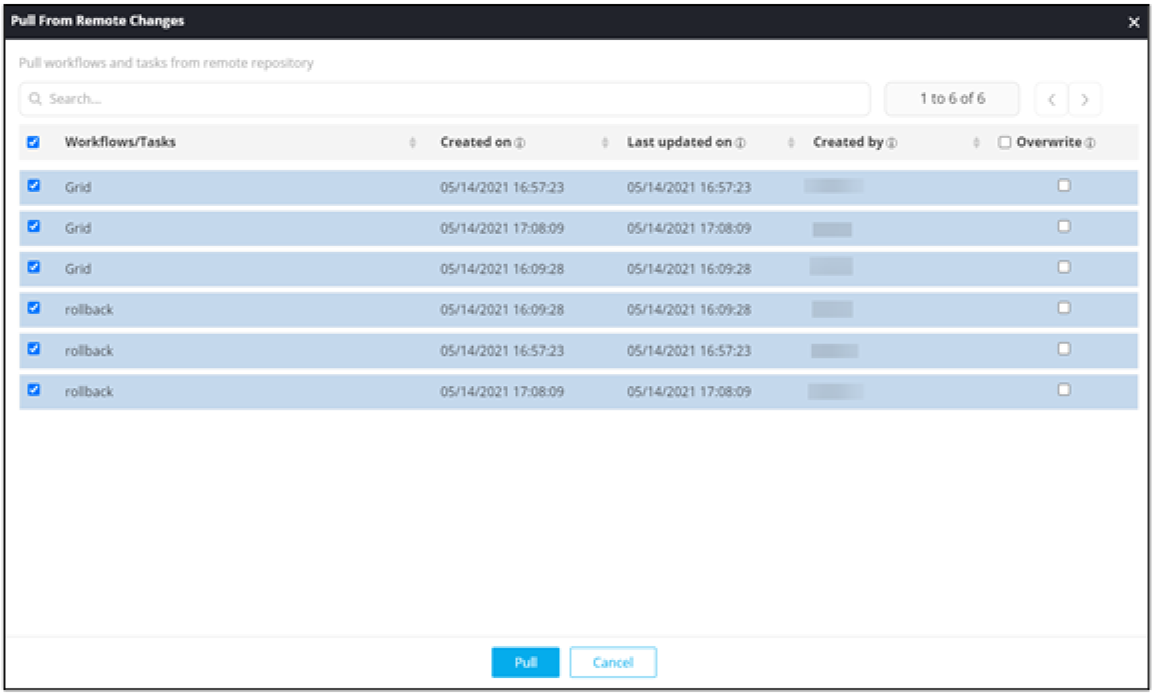
1. Go to the Workflow page.
2. On the **Workflow** page, from the **Source Control** dropdown, select **Pull from remote**.



3. In the **Pull From Remote Changes** window, enter the field information.

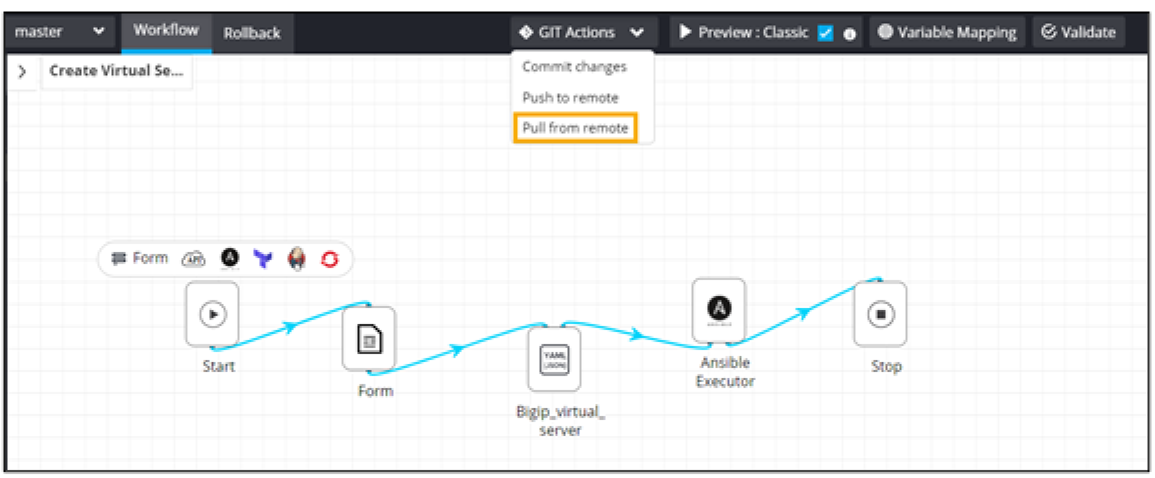


4. In the **Pull From Remote Changes** window, select the workflows to be pulled and click **Pull**.

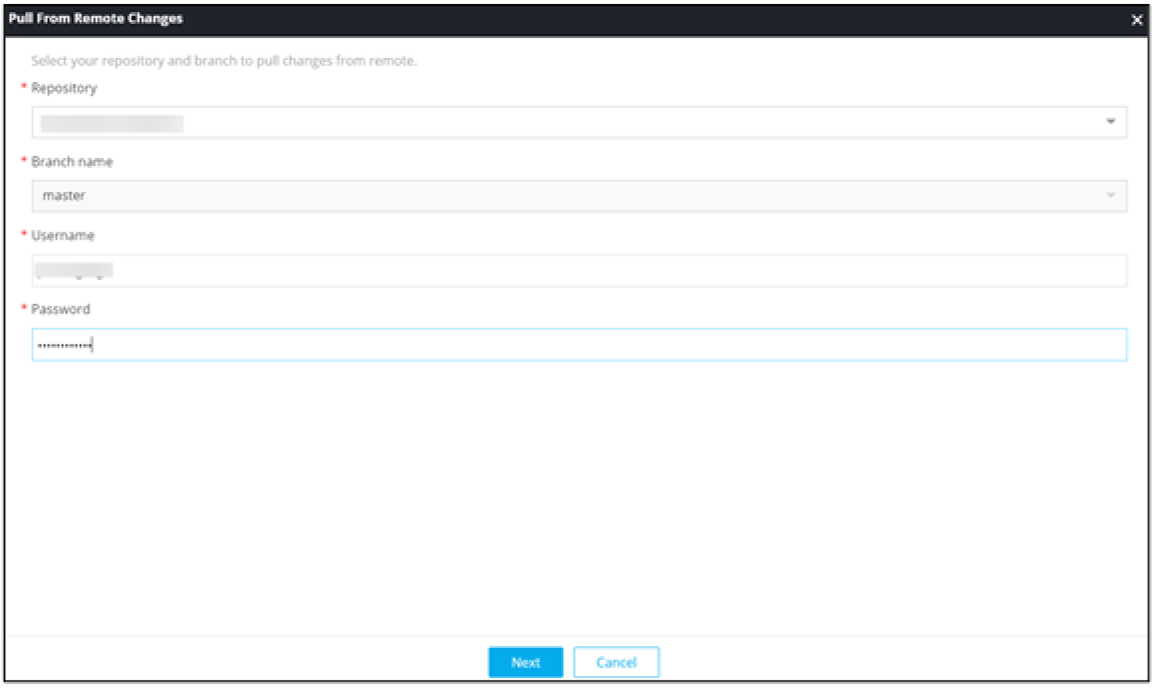


Pulling from the repository from the Design Canvas

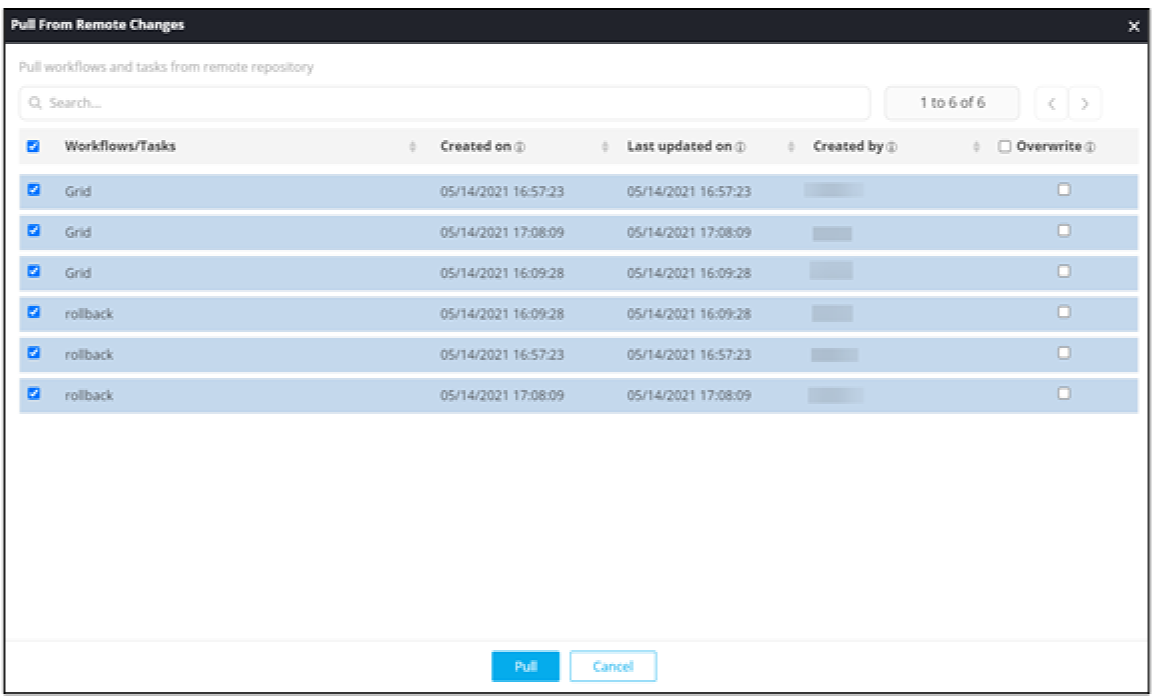
1. Go to the Workflow page.
2. Design a workflow or open an existing workflow into which you want to pull tasks/workflows.
3. From the **GIT Actions** dropdown, select **Pull from remote**.



4. In the **Pull from Remote Changes** window, enter the required field information.

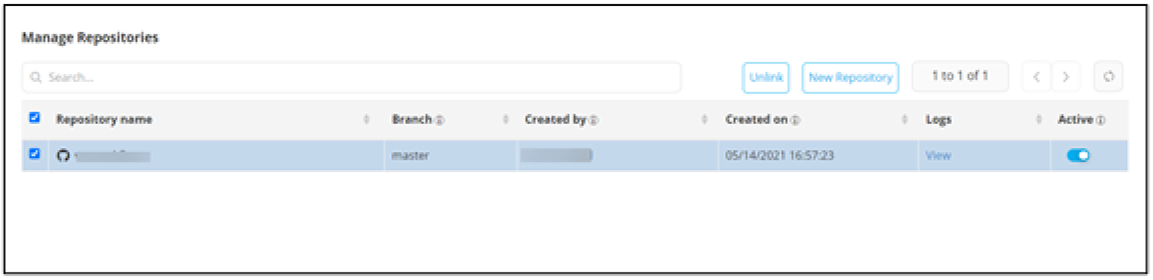


5. In the **Pull from Remote Changes** window, select the workflows to be pulled and click **Pull**.



Unlinking a Repository

1. Go to the Source Control page.
2. On the **Source Control** page, under **Manage Repositories**, select the repository to be unlinked.



3. Deactivate the repository by turning off the toggle.
4. To unlink the repository, click **Unlink**.

5. Click **Ok** in the **Confirmation** pop-up window.

Logs

Logs is an option supported to view the actions performed in the chosen repository. Logs for the latest 10 actions are displayed in the descending order.

The logs can be viewed on the **Manage Repositories** page.

The following actions are **supported** for Source Control:

- Support for Cloning, Pulling, Committing and Pushing Workflows,Tasks, and Subflows.
- Support for both On Premises and Cloud Repositories.

- Support for both public and private repositories.
- Support for SSH and HTTP/HTTPS authentication protocols.
- New branch can be created from AppViewX.
- Support for creating new versions and overwriting existing workflows on selection while importing them.
- Tasks and subflows can be overwritten or duplicated on selection while importing them.
- Support for cloning an empty repository is available.
- Allows switching between active repositories while performing Push, Pull, and Commit actions.
- Support to commit recently added tasks and subflows.
- Support to auto handle the conflicts thus enabling seamless pull and push actions in the repository.
- A concept analogous to Git staging is supported for choosing files to push from a committed list along with the Transaction status.
- Renamed workflows can be pushed and duplicates are deleted in the remote repository if the workflow with “Deleted” status is selected in staging.
- Confirmation for actions like, Overwrite, First Commit inside Design, and Cloning without selecting any files.
- Source Control Repository connection establishment based on repo name, source control type, and working branch name.
- Support for other dependencies like Helpers, Regex, Hooks, and Pages.
- Commit and Push actions for workflow(s) and task(s) can be performed from the Workflow page as well from the workflow Design studio.

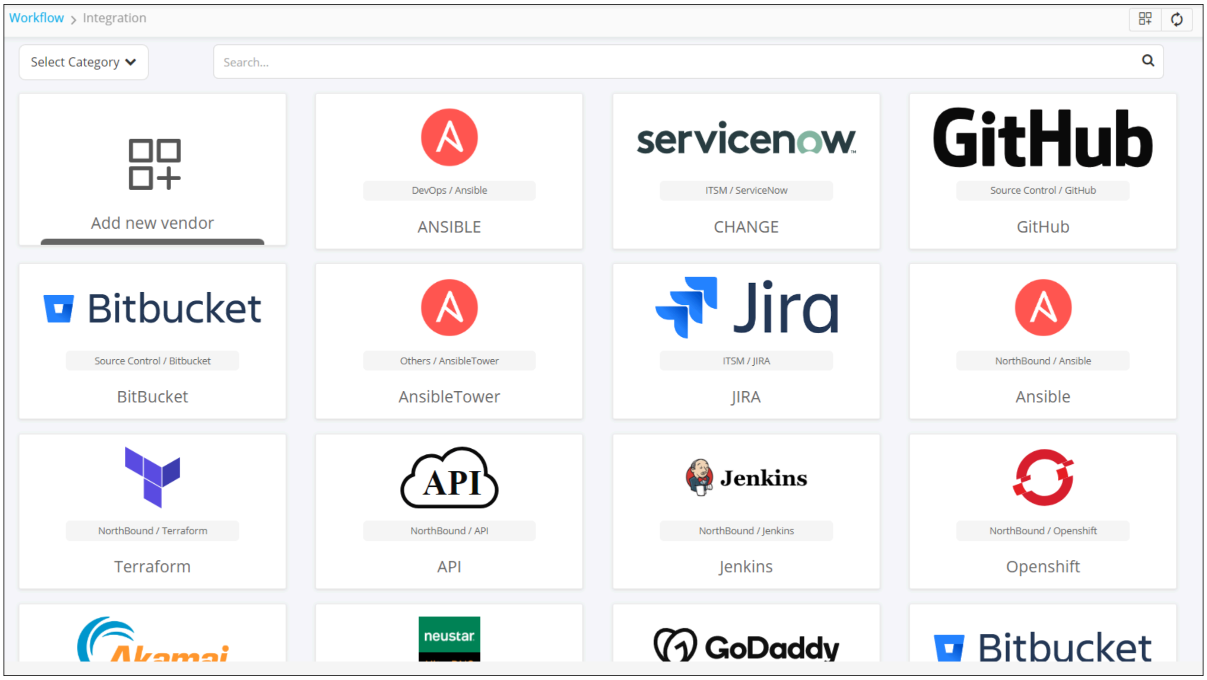
The following actions are **not supported** for Source Control:

- Option for hiding password in tasks and folder segregation along with filtering field values of exported data in remote is not supported.
- Support is not provided for deleting remote tasks and workflows.

Integration

AppViewX facilitates integration with different vendors (ITSM, GIT, DevOps, PaaS) and allows you to use the configurations for speedy automation. Relevant vendor configurations, communication endpoints, closure codes and so on must be pre-configured as part of the ITSM vendor configuration plug-in in order to reference them as part of the workflow task(s).

- Provision to select/reference a specific ITSM instance to be integrated with a workflow.
- Provision to push specific configurations and logs as part of the automation process using variables.



- Accessing the Integration Hub
- ITSM Vendor Configuration
- Jenkins Vendor Configuration
- Ansible Vendor Configuration
- Terraform Vendor Configuration
- Using Variables in Integrations

Accessing the Integration Hub

New Menu	Old Menu
In the Automation module, from the navigation pane on the left, under **WORKFLOW**, select **Integration**. The **Workflow** > **Integration** page is displayed.	1. From the main navigation menu, select **Studio** > **Workflow**. The **Workflow** page is displayed with the **Design** tab open by default. 2. From the navigation pane on the left, select **Integration**.

New Menu	Old Menu
	The **Workflow** > **Integration** page is displayed.

Note: For more information on how to switch between menus, click here.

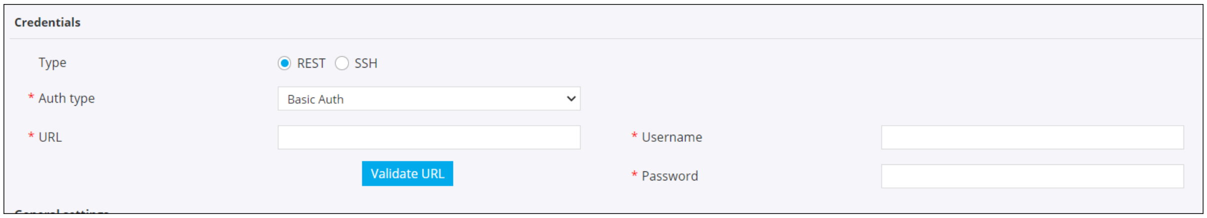
ITSM Vendor Configuration

You can configure the relevant ITSM configuration - communication end points, field mappings, closure codes - required to integrate a workflow automation flow with an IT Service Management (ITSM) tool.

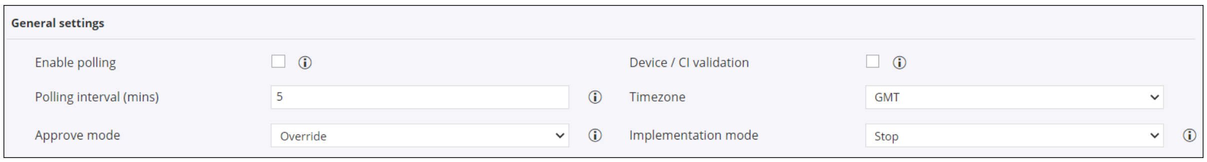
- Provision to configure the ITSM vendor name and the instance (For example, ServiceNow, Redmine, JIRA etc.) which can be referenced within the workflow tasks.
- Provision to define credentials to connect to any ITSM system.
- Provision to define relevant REST API details in order to do the following:
  - Create an ITSM ticket
  - Poll ITSM ticket details
  - Update an ITSM ticket
  - Close an ITSM ticket.
- Provision to define mapping of custom change ticket closure codes between AppViewX and the ITSM vendor.
- Provision to define custom validation parameters between AppViewX vis-a-vis ITSM system, for example, Device/CI, Implementation date/time, ITSM state/status etc.

Configuring a ServiceNow Instance

1. Go to the Integration Hub.
2. On the **Workflow** > **Integration** page, click **servicenow/change**.
3. On the **Vendor Configuration** page that is displayed, under **Credentials**, enter the required field information.



4. Under **General Settings**, define the relevant field information.



Field descriptions for General Settings

Field	Description
Enable polling	Allows you to enable periodic polling when integrated with an ITSM system.
Polling Interval (mins)	Allows you to define the frequency of polling interval when integrated with an ITSM system.
Device / CI validation	Allows you to validate for any mismatch between device/CI (configuration item) details between the AppViewX work order and the change ticket at the time of the work order approval and implementation.
Timezone	Allows you to select the timezone from the list.
Approve mode	Allows you to enable either a ‘hard stop’ or ‘override’ (the work order) in the event of any mismatch between the AppViewX work order and the change ticket details at the time of work order approval, peer review.
Implementation mode	Allows you to enable either a ‘hard stop’ or ‘override’ (the work order) in the event of any mismatch between the AppViewX work order and the change ticket details at the time of work order implementation post all peer reviews.

5. Define the **Log/Configuration** settings.

Field descriptions for Log/Configuration

Field	Description
Select configuration type	Allows you to enable pushing of configuration commands and/or logs to the ITSM system. For example, pushing config and configurations to the ‘journal field’ on ServiceNow.
Auto close	Allows you to enable automatic ticket closure or otherwise on the ITSM system once the AppViewX work order is executed.

6. Under **Configurations**, turn on the **Enable config** toggle and define relevant API parameters in the JSON configurator that is displayed.
   Sample Schema:

```
{
 "ServiceNowConfig": {
  "serviceName": "ServiceNow",
  "serviceApiList": {
   "getTicket": {
    "url": "/api/now/table/change_request?sysparm_query=number=<ticketNumber>",
    "responseDataMapping": {
     "state": "result~approval",
     "startTime": "result~start_date",
     "endTime": "result~end_date",
     "sysId": "result~sys_id",
     "ticketNumber": "result~number"
    },
    "apiListToCallAfter": [
     "getDeviceList"
    ],
    "name": "getTicket",
    "method": "GET"
   },
   "getDeviceList": {
    "url": "/api/now/table/task_ci?sysparm_display_value=true&sysparm_query=task=<sysId>",
    "responseDataMapping": {
     "deviceList": "result~ci_item~display_value"
    },
```



```
   "name": "getDeviceList",
   "method": "GET"
  },
  "closeTicket": {
   "url": "/api/now/table/change_request/<sysId>",
   "responseDataMapping": {
    "state": "result~state"
   },
   "payloadDataMapping": {
    "state": "closureCode",
    "close_notes": "comment"
   },
   "name": "closeTicket",
   "method": "PUT"
  },
  "updateTicket": {
   "url": "/api/now/table/change_request/<sysId>",
   "responseDataMapping": {},
   "payloadDataMapping": {
    "work_notes": "updateData"
   },
   "name": "updateTicket",
   "method": "PUT"
  }
 },
 "validation": {
  "state": "approved"
 },
 "tooltipData": {
  "Affected CI's": "deviceList",
  "Start Date": "startDate",
  "End Date": "endDate",
  "Status": "state"
 },
 "authorizationConfig": {
  "type": "Basic"
 },
```

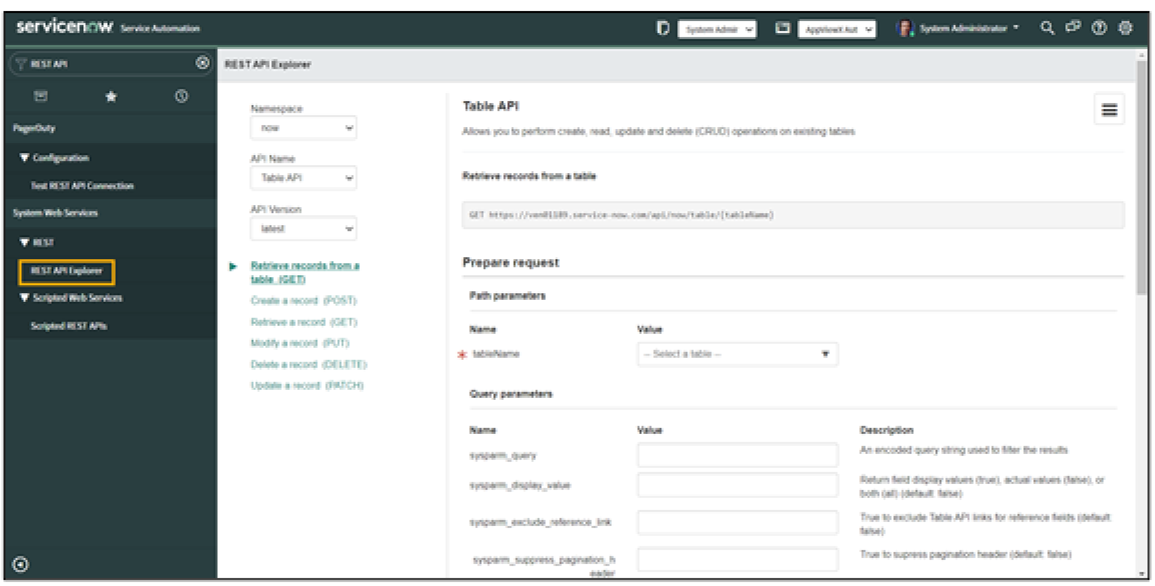
```
"mandatoryHeaders": {
 "Content-Type": "application/json"
},
"constants": {
 "ticketCloseStates": {
  "Success": 3,
  "Not implemented": 4,
  "Withdrawn": 7
 },
 "dateFormat": "yyyy-MM-dd HH:mm:ss"
}
}
}
```

7. Click **Add**.

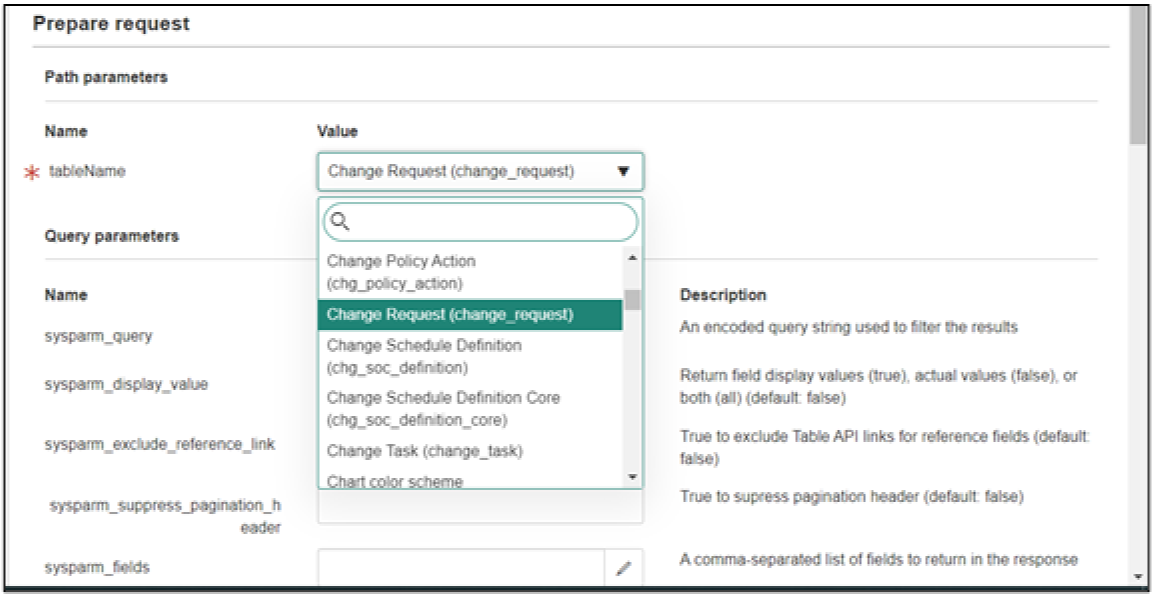
Checking ServiceNow REST API details

To retrieve details using REST API explorer for a given Change ticket on ServiceNow:

1. Login to ServiceNow using your credentials.
2. Search or navigate to **REST API Explorer**.

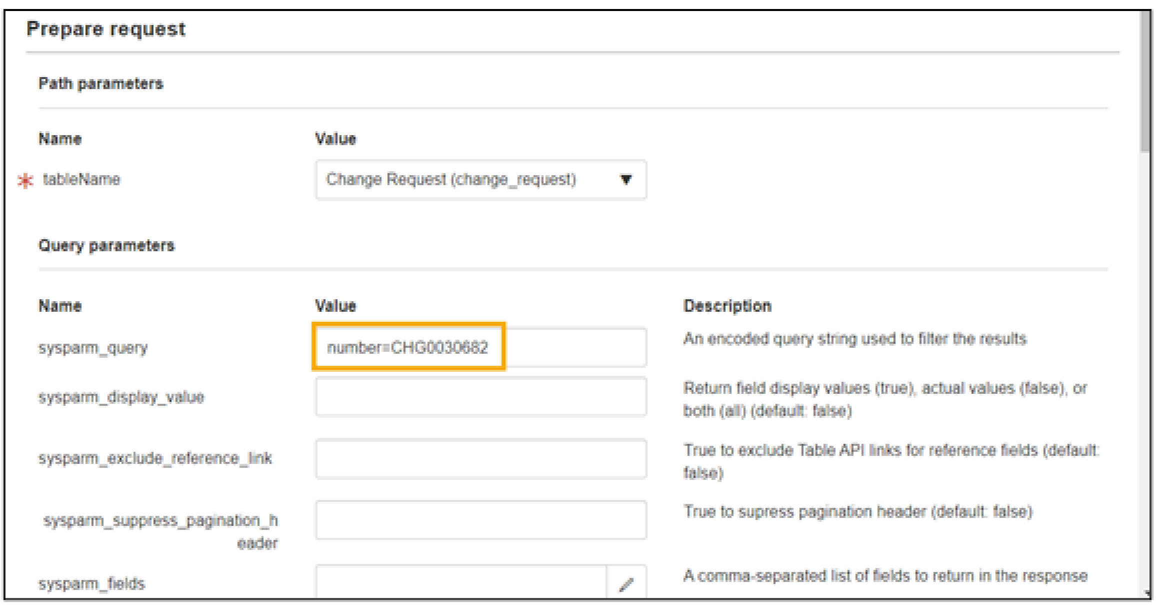


3. Select **tableName** value from the dropdown list.



4. Enter the value for **sysparm_query** (number=<change ticket>).

   For example, number=CHG0030682



5. Get ‘sysID’ API response body for the change ticket.


```
{
 "result": [
  {
   "u_glide_date_1": "",
   "parent": "",
   "reason": "",
   "made_sla": "true",
   "backout_plan": "",
   "watch_list": "",
   "u_integer_4": "",
   "upon_reject": "cancel",
   "sys_updated_on": "2016-11-11 07:58:35",
   "type": "Comprehensive",
   "conflict_status": "Not Run",
   "approval_history": "",
   "number": "CHG0030682",
   "test_plan": "",
   "sys_updated_by": "admin",
   "opened_by": {
    "link": "https://ven01189.service-now.com/api/now/table/sys_user/6816f79cc0a8016401c5a33be04be441",
    "value": "6816f79cc0a8016401c5a33be04be441"
   },
```



```
"user_input": "",
"requested_by_date": "",
"sys_created_on": "2016-11-11 07:08:22",
"sys_domain": {
  "link": "https://ven01189.service-now.com/api/now/table/sys_user_group/global",
  "value": "global"
},
"state": "4",
"sys_created_by": "admin",
"knowledge": "false",
"order": "",
"phase": "requested",
"closed_at": "2016-11-11 07:58:35",
"cmdb_ci": {
  "link": "https://ven01189.service-now.com/api/now/table/cmdb_ci/0ca262a80f4fa200c3e6cd8ce1050e68",
  "value": "0ca262a80f4fa200c3e6cd8ce1050e68"
},
"delivery_plan": "",
"impact": "3",
"active": "false",
"review_comments": "kk_vs1",
"work_notes_list": "",
"business_service": "",
"priority": "2",
"time_worked": "",
"cab_recommendation": "",
"expected_start": "",
"production_system": "false",
"rejection_goto": "",
"opened_at": "2016-11-11 07:08:22",
"review_date": "",
"business_duration": "",
"group_list": "",
"requested_by": {
  "link": "https://ven01189.service-now.com/api/now/table/sys_user/kk",
  "value": "kk"
},
```

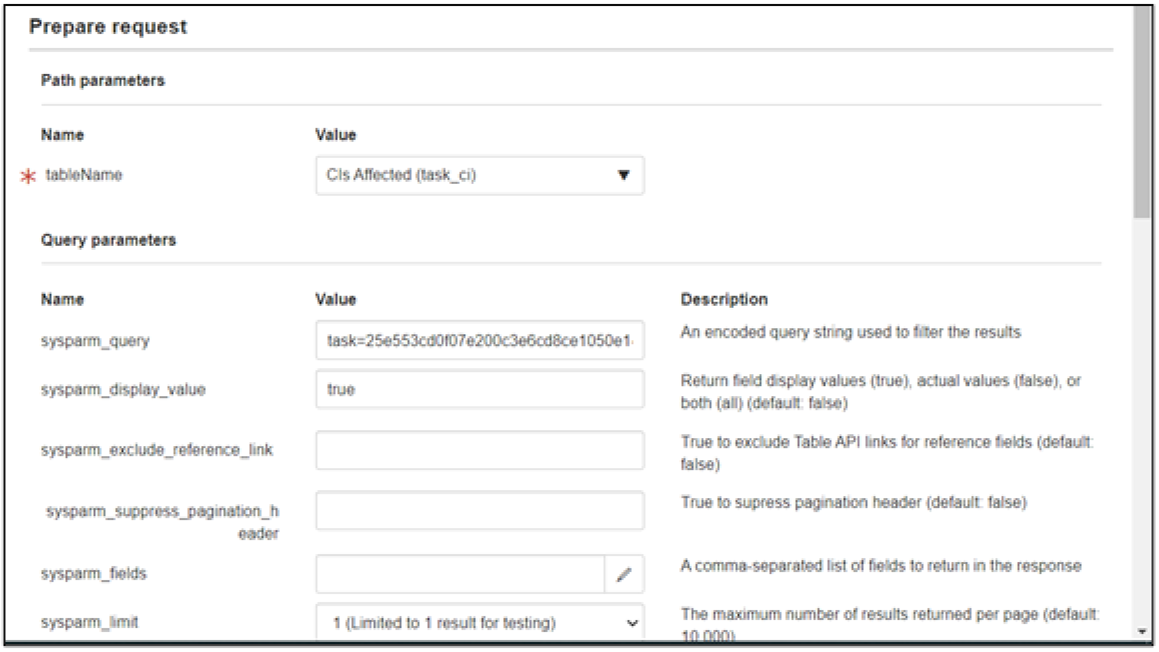


```
"work_end": "",
"change_plan": "",
"phase_state": "open",
"approval_set": "",
"cab_date": "",
"wf_activity": "",
"work_notes": "",
"implementation_plan": "",
"u_table_name_1": "",
"end_date": "2016-11-12 00:00:00",
"short_description": "Create VS on UAT",
"correlation_display": "",
"delivery_task": "",
"work_start": "",
"assignment_group": "",
"additional_assignee_list": "",
"outside_maintenance_schedule": "false",
"description": "Create VS",
"calendar_duration": "",
"close_notes": "Closing the ticket from AppViewX.",
"sys_class_name": "change_request",
"closed_by": {
  "link": "https://ven01189.service-now.com/api/now/table/sys_user/6816f79cc0a8016401c5a33be04be441",
  "value": "6816f79cc0a8016401c5a33be04be441"
},
"follow_up": "",
"sys_id": "25e553cd0f07e200c3e6cd8ce1050e14",
"contact_type": "phone",
"urgency": "3",
"scope": "3",
"company": "",
"justification": "",
"reassignment_count": "0",
"review_status": "3",
"activity_due": "",
"assigned_to": "",
"start_date": "2016-11-07 00:00:00",
```

```
      "u_field_name_3": "",
      "u_field_name_1": "",
      "comments": "",
      "u_field_name_2": "",
      "u_journal_1": "",
      "approval": "approved",
      "sla_due": "",
      "comments_and_work_notes": "",
      "due_date": "",
      "sys_mod_count": "4",
      "sys_tags": "",
      "conflict_last_run": "",
      "escalation": "0",
      "upon_approval": "proceed",
      "correlation_id": "",
      "location": "",
      "risk": "4",
      "category": "Hardware"
    }
  ]
}
```


6. Enter or select the following field information to get ‘CI’ specific details.

tableName	CIs Affected (task_ci)
sysparm_query	task=’sysID’ (task=25e553cd0f07e200c3e6cd8ce1050e14)
sysparm_display_value	true



Response for sysid from task_ci table.


```
{
  "result": [
    {
      "sys_id": "39e593010f47e200c3e6cd8ce1050e8a",
      "sys_updated_by": "admin",
      "task": {
        "display_value": "CHG0030682",
        "link": "https://ven01189.service-now.com/api/now/table/task/25e553cd0f07e200c3e6cd8ce1050e14"
      },
      "applied": "false",
      "sys_created_on": "2016-11-11 07:08:23",
      "xml": "",
      "sys_mod_count": "0",
      "sys_updated_on": "2016-11-11 07:08:23",
      "sys_tags": "",
      "applied_date": "",
      "ci_item": {
        "display_value": "117",
        "link": "https://ven01189.service-now.com/api/now/table/cmdb_ci/0ca262a80f4fa200c3e6cd8ce1050e68"
      },
```



```
      "sys_created_by": "admin"
    }
  ]
}
```

ITSM JSON Configuration

The IT Service Management (ITSM) JavaScript Object Notation (JSON) configurator is a generic method through which an ITSM vendor that supports a JSON based REST communication can integrate with AppViewX. This allows AppViewX to establish communication with an external tool and achieve the necessary change management integration.

The key components of the ITSM JSON configuration are:

- Application Programming Interface (API) Definitions
- Validation parameter configuration
- Tooltip data configuration
- Mandatory headers configuration
- Authorization configuration
- Constants.
  - Change ticket closure states.
  - Date format used by the vendor.

The API definition can have the following configurations:

- **Name**: The API Name
- **Method**: The HTTP method - **GET, POST, PUT, DELETE**
- **URL**: The relative URL for the REST endpoint of the API
- responseDataMapping
- payloadDataMapping
- apiListToCallAfter
- apiListToCallBefore.

- ITSM APIs
- URL and Response Data Mapping
- Response Data Mapping
- Payload Data Mapping
- Validation Parameters

- Tooltip Data Configuration
- Mandatory Headers
- Authorization Configuration

ITSM APIs

The following APIs need to be defined within the configurator in order to integrate Change management vendor with AppViewX:

- **Create ticket**: This API is used to create an ITSM change ticket and pass some basic ticket details defined in the payloadDataMapping.
- **Get ticket**: This API is used to fetch the ITSM change ticket for a given RFC ID (Request for change). All the fields mapped in the responseDataMapping will be stored internally in AVX for reference.
- **Close ticket**: This API is used to close the ITSM change ticket based on the AppViewX workorder status. The corresponding state in the ITSM system has to be mapped in the constants, under ticketCloseStates.
  - The value of “closureCode” is picked from ticketCloseStates based on the workorder status.

```
"close": {
  "url": "/api/now/table/change_request/<sysId>",
  "responseDataMapping": {
   "state": "result~state"
  },
  "payloadDataMapping": {
   "state": "closureCode",
   "close_notes": "comment"
  },
  "name": "closeTicket",
  "method": "PUT"
 },
```

- **Update ticket**: This API is used to update data back to the change ticket. Commonly the update data are workorder configurations (Implementation, Rollback, Pre-validation, and Post-validation) and workorder logs.
  - Specific configurations such as Implementation, Pre-validation, and Post-validation can be mapped against the relevant ITSM field, and pushed.

```
"update": {
  "url": "/api/now/table/change_request/<sysId>",
  "responseDataMapping": {},
  "payloadDataMapping": {
   "work_notes": "updateData"
  },
  "name": "update",
  "method": "PUT"
 },
 "updateConfig": {
  "responseDataMapping": {},
  "payloadDataMapping": {
   "user_input": [
    "rollback_config"
   ],
   "close_notes": [
    "postvalidation_config"
   ],
   "work_notes": [
    "prevalidation_config"
   ]
  },
  "url": "/api/now/table/change_request/<sysId>",
  "name": "updateConfig",
  "method": "PUT"
 },
 "updateLog": {
  "responseDataMapping": {},
  "payloadDataMapping": {
   "user_input": [
    "rollback"
   ],
   "close_notes": [
    "postvalidation"
   ],
   "work_notes": [
    "prevalidation"
```



```
  ]
 },
 "url": "/api/now/table/change_request/<sysId>",
 "name": "updateLog",
 "method": "PUT"
```

If the API for withdrawal of the ticket is different from the close API, it can be defined under the name “withdrawTicket”.

```
"method": "GET"
 },
 "withdraw": {
  "responseDataMapping": {
   "status": "status"
  },
  "payloadDataMapping": {
   "change_number": "ticketNumber",
   "comment": ""
  },
  "url": "/withdraw",
  "name": "withdrawTicket",
  "method": "POST"
 },
```

URL and Response Data Mapping

The relative URL for the REST endpoint of the API and the relevant query parameters for the URL can be specified within <> representation.

For example, In the getTicket API only `<ticketNumber>` can be used. But for other API any mapped data can be used with the <> notation.

Response Data Mapping

This is a guideline for the data to be mapped for internal use by AppViewX based on the JSON response obtained from the ITSM system. Response data mapping (key-value pair) is used to define all the relevant fields from the ITSM system into AppViewX.


```
{
 "ServiceNowConfig": {
  "serviceName": "ServiceNow",
  "serviceApiList": {
   "getTicket": {
    "url": "/api/now/table/u_vip_create?sysparm_query=u_number=<ticketNumber>",
    "responseDataMapping": {
     "state": "result~u_approval",
     "startTime": "result~u_start_date",
     "endTime": "result~u_end_date",
     "sysId": "result~sys_id",
     "ticketNumber": "result~u_number"
    },
    "apiListToCallAfter": [
     "getDeviceList"
```

- The key on the left indicates the values that are used in AppViewX as part of the provisioning, ticket validation process.
- The values on the right indicate the response received from an external tool and used within AppViewX as part of the provisioning, ticket validation process.

Payload Data Mapping

These are attributes required to be sent to an ITSM system as part of the payload data (via REST).


```
"closeTicket": {
   "url": "/api/now/table/u_vip_create/<sysId>",
   "responseDataMapping": {
    "state": "result~u_state"
   },
   "payloadDataMapping": {
    "u_state": "closureCode",
    "u_notes": "comment"
   },
```

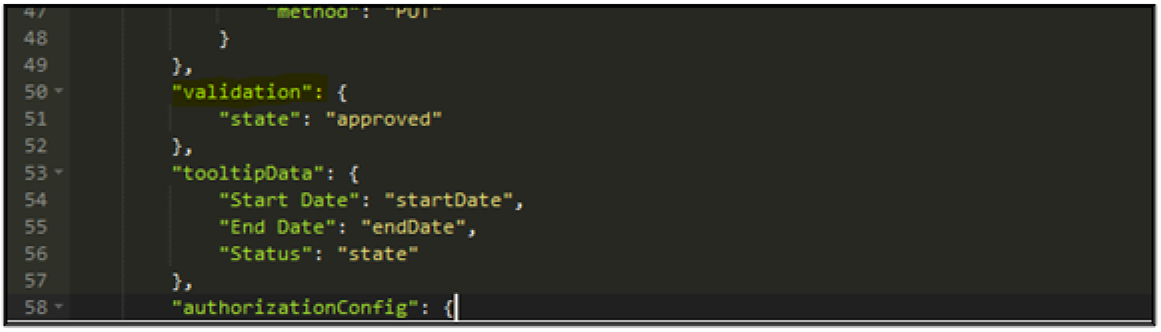
- The key on the left indicates the specific field values on the ITSM system.
- The values on the right indicate the specific values with which the fields on ITSM have to be updated by AppViewX.

Validation Parameters

This is a set of JSON key value pairs that define what attributes of a change ticket needs to be validated within AppViewX. Any custom attributes can be defined as part of the validation. The values on the left side of the assignment denote the attribute to be validated and the right side of the assignment denotes the expected value for the parameter to be considered as valid.

Note: In case an attribute that is not mapped to the system is validated, these will be ignored and will be treated as invalid. The validation message on the GUI will be displayed accordingly.

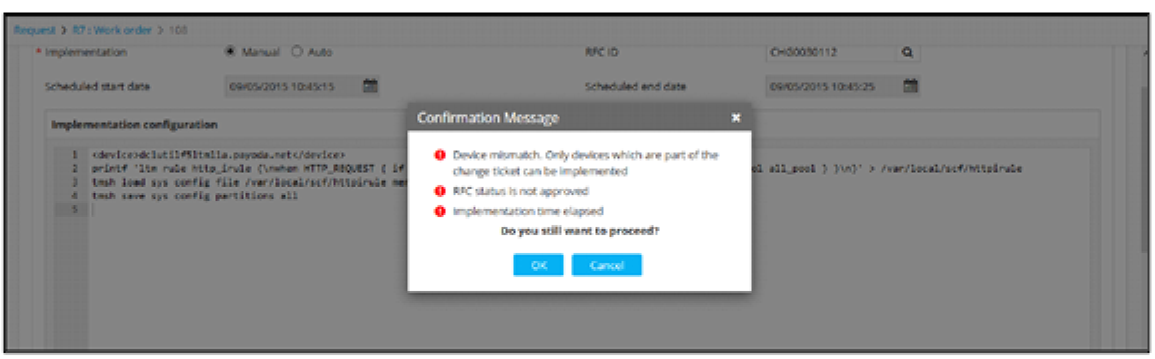
Example validation parameter configuration:




```
"name": "updateTicket",
   "method": "PUT"
  }
 },
 "validation": {
  "state": "approved"
 },
 "tooltipData": {
  "Start Date": "startDate",
  "End Date": "endDate",
```

• Validation message sample

In this case, AppViewX executes a work order based on the validation parameters. For example, the state of the change ticket on the ITSM tool must be ‘approved’, Device/CI must be valid, Change window must be valid, and so on.



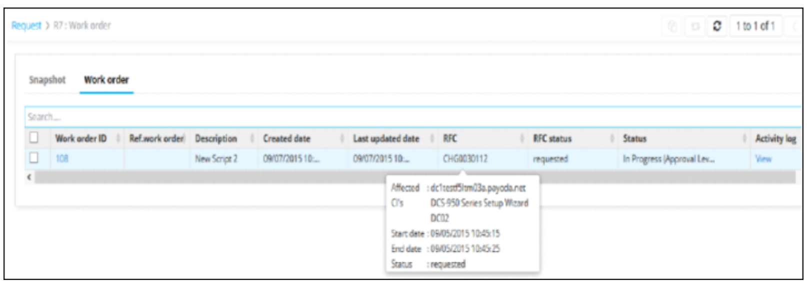
Tooltip Data Configuration

This placeholder gives you the flexibility to customize the tooltip data that is to be displayed in the workorder review page against a specific work order (on hovering over the RFC ID field).

The left side of the assignment denotes the Label to be displayed in the tooltip, and the right side of the assignment denotes the data that is to be shown in the tooltip.

Example tooltip config:

```
"tooltipData":{
        "Affected CI's":"deviceList",
        "Start Date":"startDate",
        "End Date":"endDate",
        "Status":"state"
    }
```



Mandatory Headers

These are key-value pairs that are needed for every HTTP request sent to the defined APIs. Support is available for “Content-type”:”application/json”. There can be additional headers depending on the system.

For example:

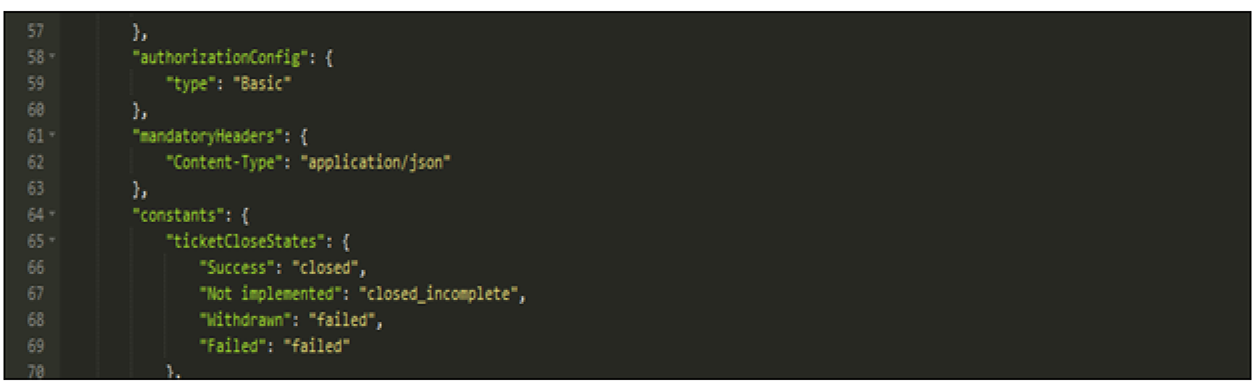
```
"mandatoryHeaders":{
        "Content-Type":"application/json"
    }
```


Authorization Configuration

This part of the configuration contains the authentication details to be performed in order to communicate with the ITSM system. Support is available for “no authentication” and “BASIC authentication”. This is to be specified in the “type” key under in configuration.

For example:

```
"authorizationConfig":{
      "type":"Basic"
   }
```



- Ticket close states: A placeholder to map the AppViewX work order states - Completed, Failed, Not Implemented - with the appropriate ITSM change ticket closure states. This can be defined either as a String or Integer format depending on the ITSM system.

- Vendor date format: Date format is mandatory for AppViewX to interpret the date values from the ITSM system.

Example: Ticket closure states in integer format

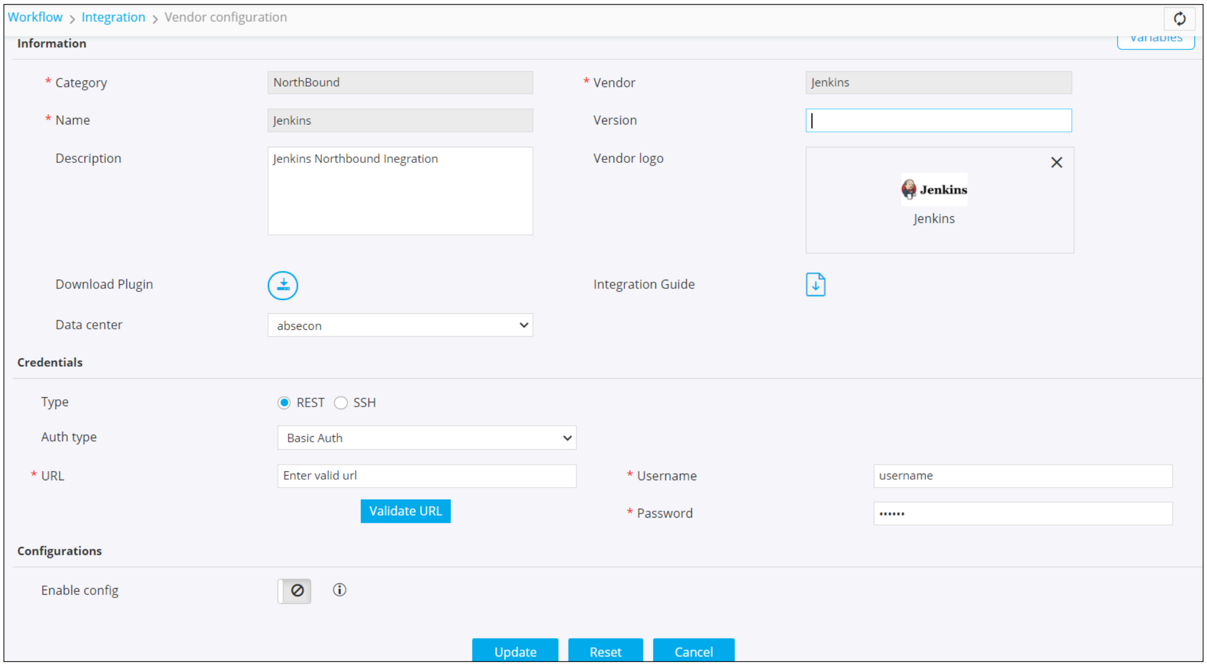
```
"constants":{
          "ticketCloseStates":{
           "Success":3,
           "Not implemented":4,
           "Withdrawn":7
     },
     "dateFormat":"yyyy-MM-dd HH:mm:ss"
   }
```

Example: Ticket closure states in string format

```
"constants":{
          "ticketCloseStates":{
           "Success":3,
           "Not implemented":4,
           "Withdrawn":7
     },
     "dateFormat":"yyyy-MM-dd HH:mm:ss"
   }
```

Jenkins Vendor Configuration

1. Go to the Integration Hub.
2. On the **Workflow** > **Integration** page, click **Jenkins**.
3. On the **Vendor configuration** page, enter the required field information.



4. Click **Save**.

Note: For more information on Jenkins - Visual Workflow Northbound integration, click here.

Ansible Vendor Configuration

AppViewX allows you to integrate with Ansible in order to automate and orchestrate the network.

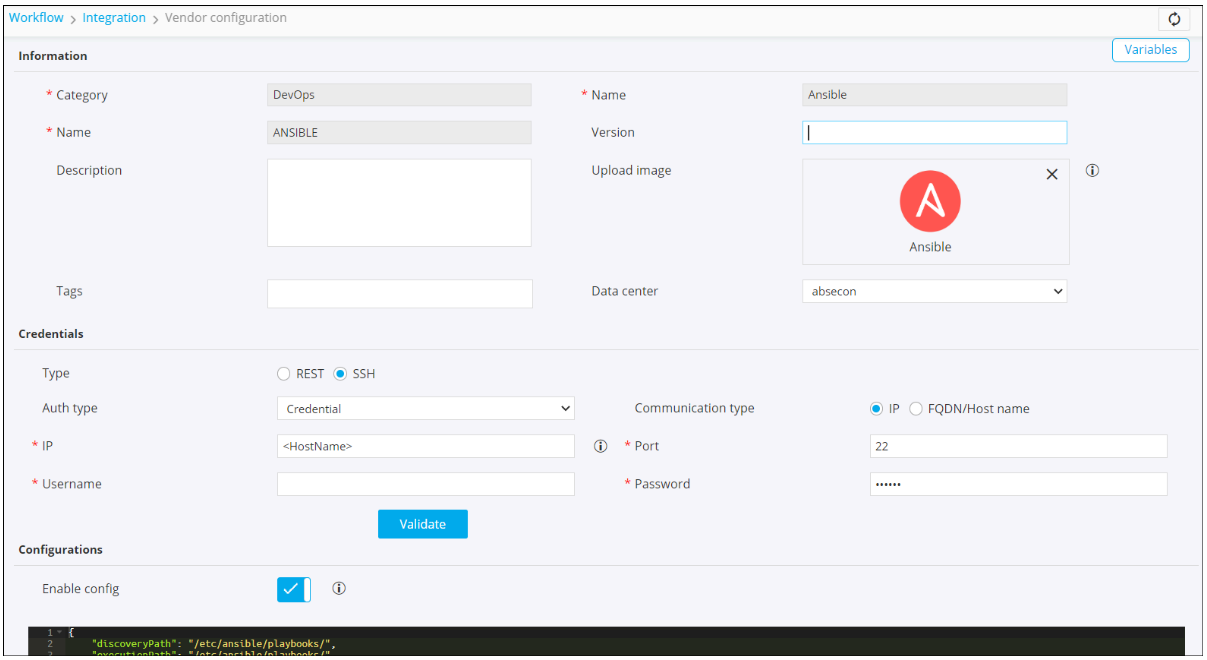
- Provision to integrate with an existing Ansible instance.
- Provision to use the Ansible executor task when there is a need for automating network configurations through Ansible as the southbound.
- Provision to Discover and Import playbooks from a specific path within the visual workflow studio.
- Drag & drop playbooks and automate using Ansible modules via visual workflow.

- Configuring an Ansible Instance
- Discovering Playbooks
- Getting Device Credentials Dynamically
- Ansible Executor

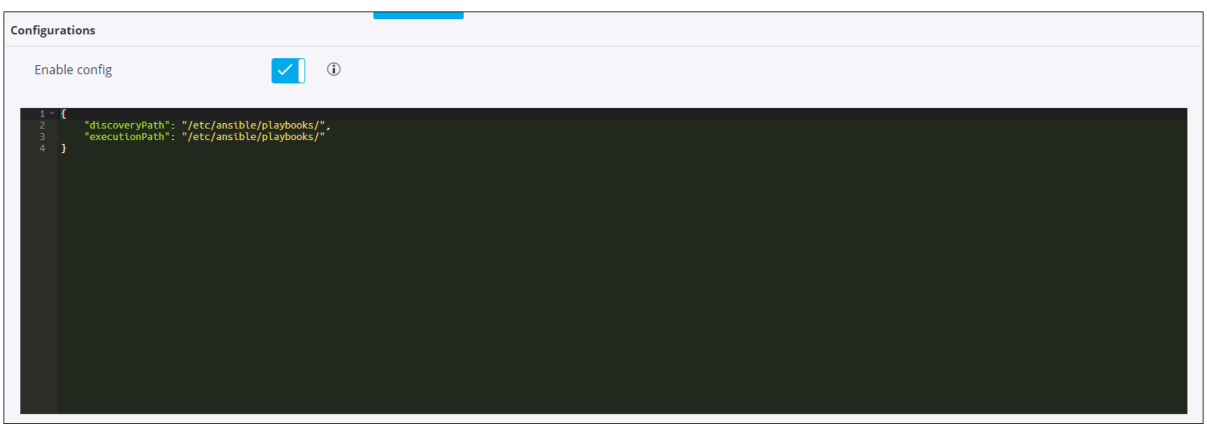
Configuring an Ansible Instance

An existing Ansible instance can be configured and integrated, which allows for reusing playbooks, and for south bound implementation via AppViewX. One or more instances can be configured.

1. Go to the Integration Hub.
2. On the **Workflow** > **Integration** page, click **Ansible**.
3. On the **Vendor configuration** page, enter the required field information.



4. Under **Configurations**, turn on the **Enable config** toggle and define the folder/path from where existing playbooks can be discovered within the workflow studio.



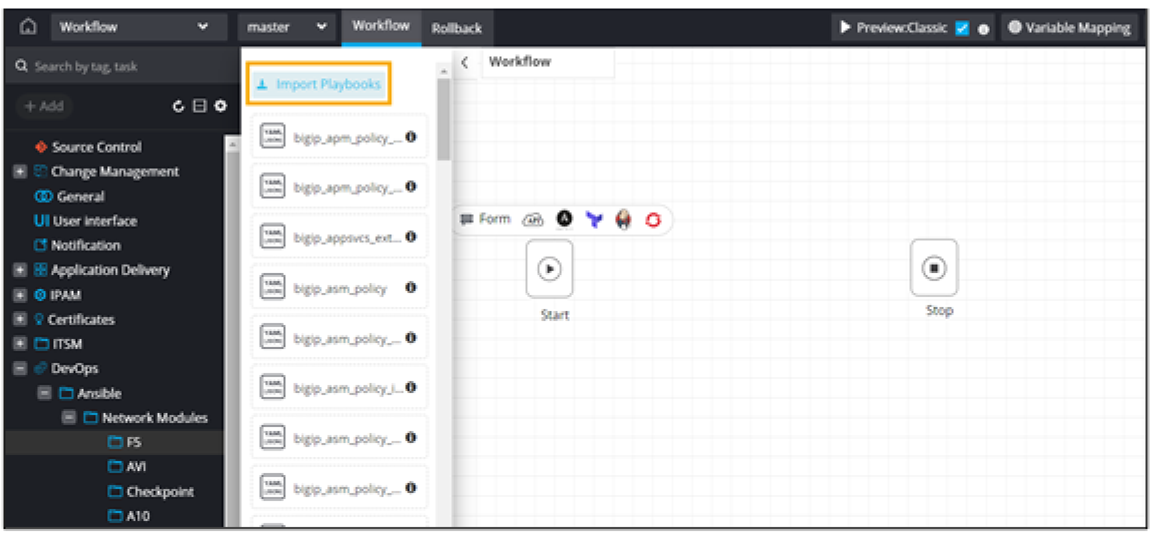
5. Click **Add**.

Note: For more information on Ansible - Visual Workflow Northbound integration, click here.

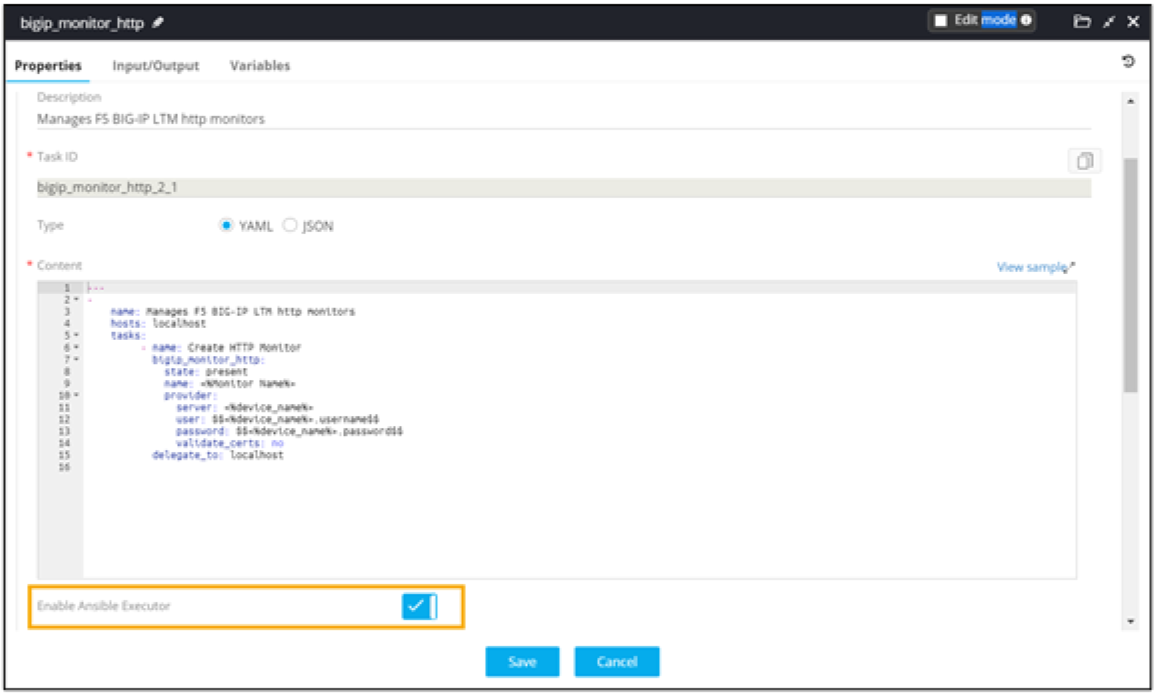
Discovering Playbooks

Once an Ansible instance is configured, playbooks can be discovered from the defined path and reused within the workflow design studio for further automation.

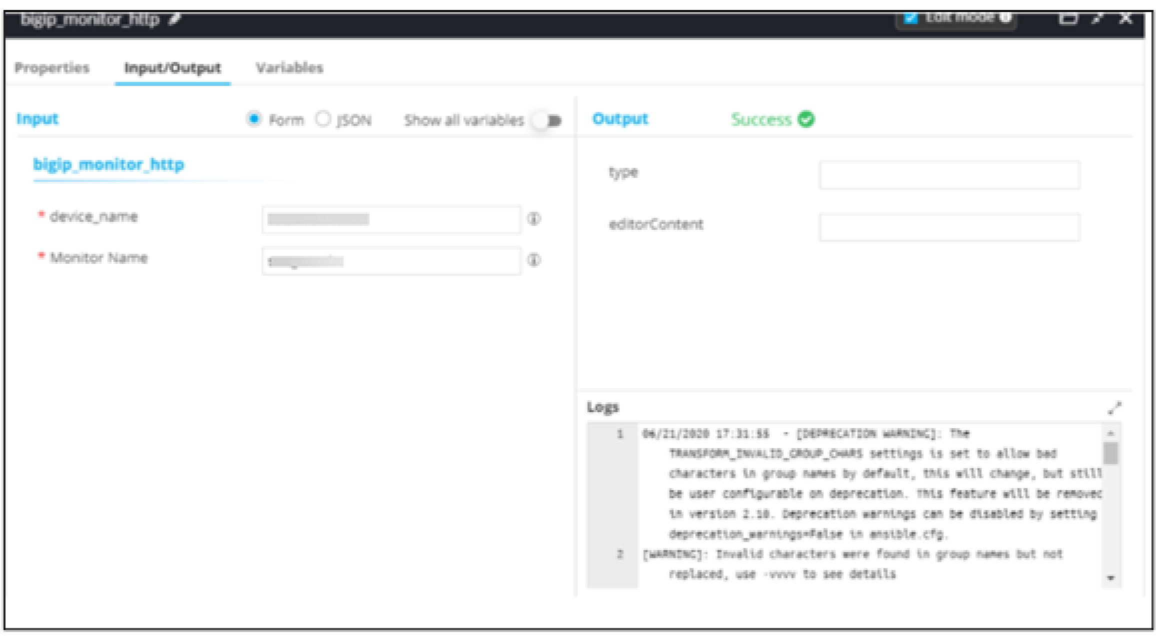
1. Go to the Workflow page.
2. Design a workflow.
3. From the menu on the left, under **DevOps**, select **Ansible** > **Network Modules** > **F5**.
4. From the **F5** folder, click **Import Playbooks**.



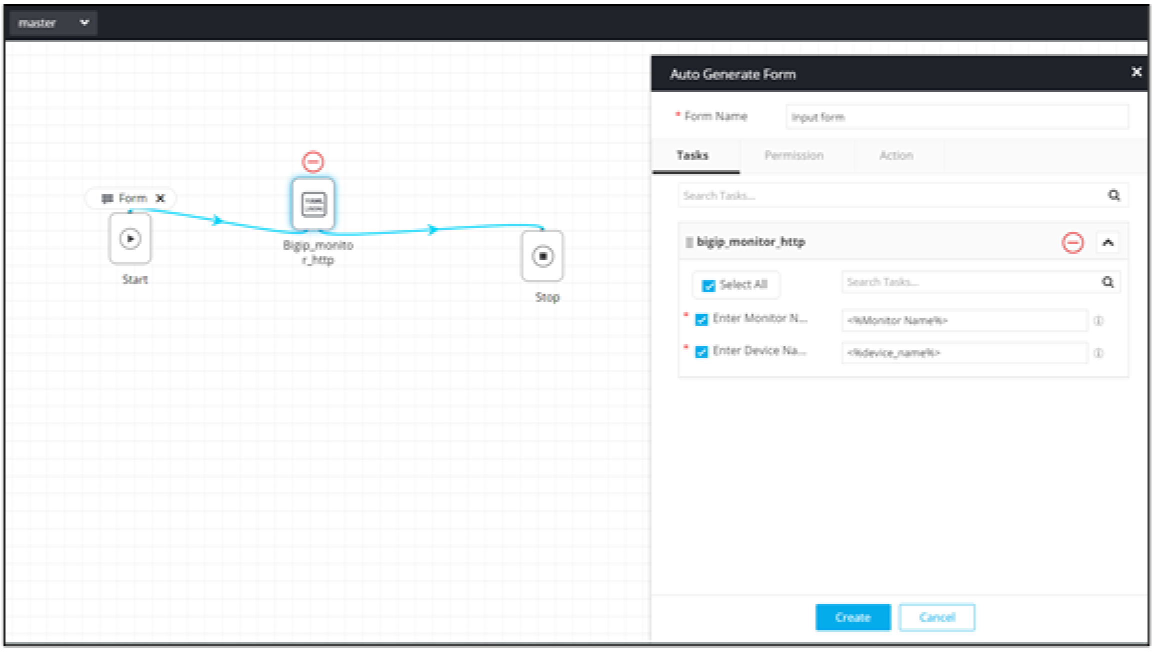
5. In the **Import playbooks** window, select import **Type**.
   - **Upload file**: Upload playbooks from a local folder.
   - **Import playbooks**: Discover playbooks from Ansible folder.
6. Save playbooks to a custom folder.
7. Drag and drop the playbooks into the workspace.
8. In the task window, under **Properties**, to enable the Ansible instance, turn on the toggle.



9. Under **Input/Output**, enter the **Input/Output** and **Variables** details.
10. Execute the task to view the execution log.

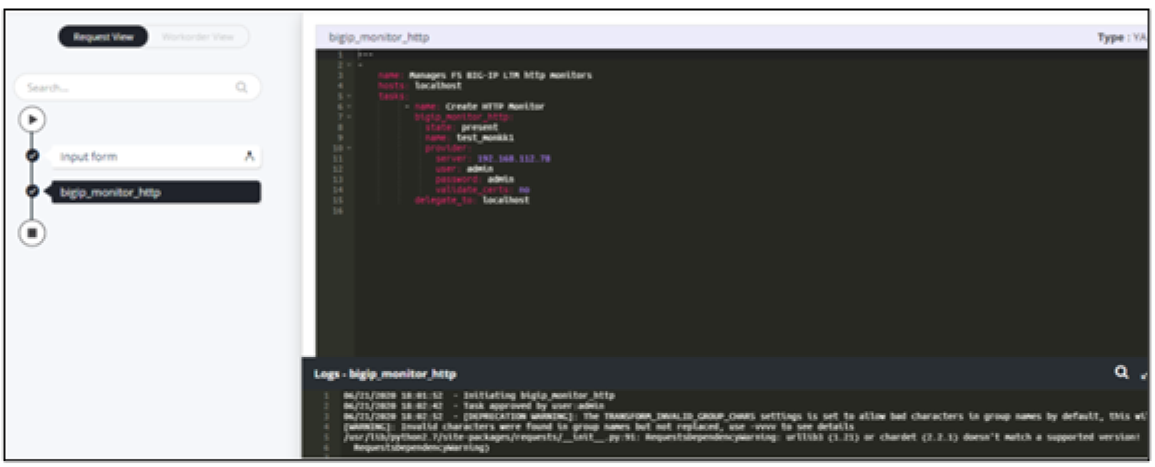


11. Auto generate a self-service form.



12. Connect the workflow tasks and enable the workflow.
13. Trigger the workflow.

    Workflow is executed.



14. To view the Request activity log, on the Request :: View/Run page, under the **Activity** column, click **View**.

    A **Summary** of the workflow request execution is displayed.

Getting Device Credentials Dynamically

Device credentials can be dynamically retrieved as part of the workflow execution process through a YAML (JSON) task. The following syntax must be used in the task in order to get device credentials.

- Get device detail: `$$devicename.username$$`
- Get device credential: `$$devicename.password$$`

> **Note:** Device(s) must be added in the Appviewx.

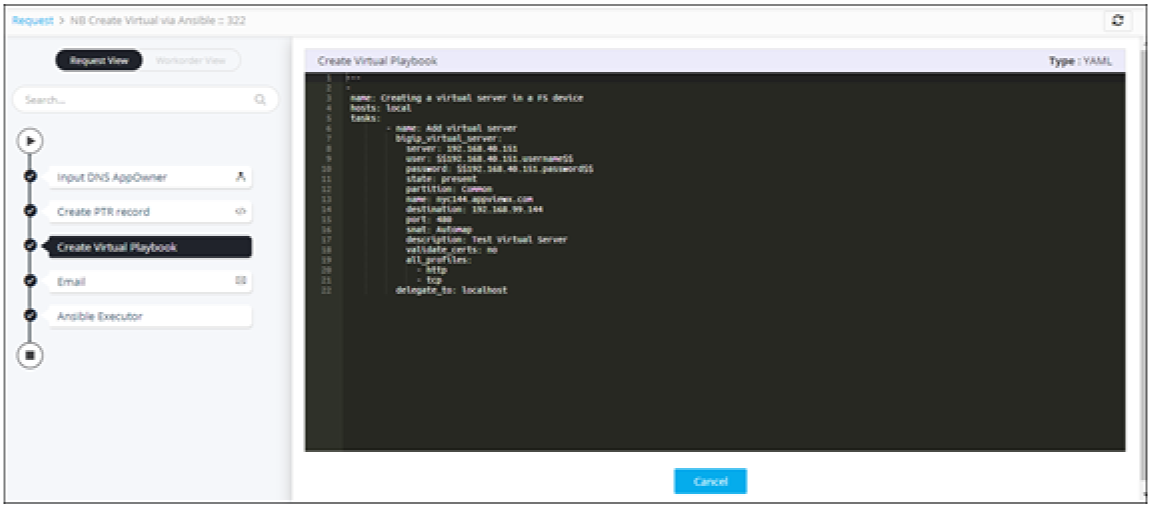
To get the device credentials dynamically during the workflow execution process:

1. Go to the Integration Hub.
2. Design a new workflow.
3. Drag and drop relevant tasks.
4. From the **User Interface** section in the left menu, drag and drop the **YAML** task.
5. Add input data in the YAML format.
6. Enter the following syntax to get the device credentials dynamically.

Syntax	Usage Sample
--- - name: Creating a virtual server in a F5 device hosts: local tasks: - name: Add virtual server bigip_virtual_server: device: <Devicename or IP address> username: $$devicename.username$$ password: $$devicename.password$$ state: present	--- - name: Creating a virtual server in a F5 device hosts: local tasks: - name: Add virtual server bigip_virtual_server: device: bigip.ltm.12.1 username: $$ bigip.ltm.12.1.username$$ password: $$ bigip.ltm.12.1.password$$ state: present

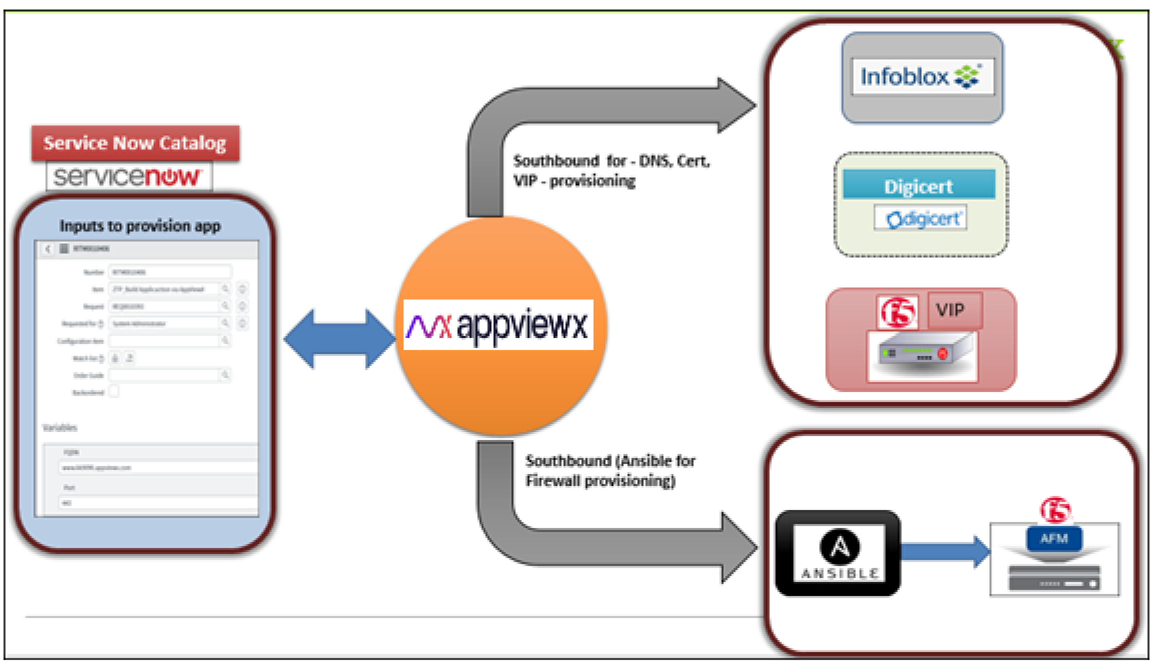
7. Connect the workflow tasks and enable the workflow.
8. Trigger the workflow.

Device details are fetched.



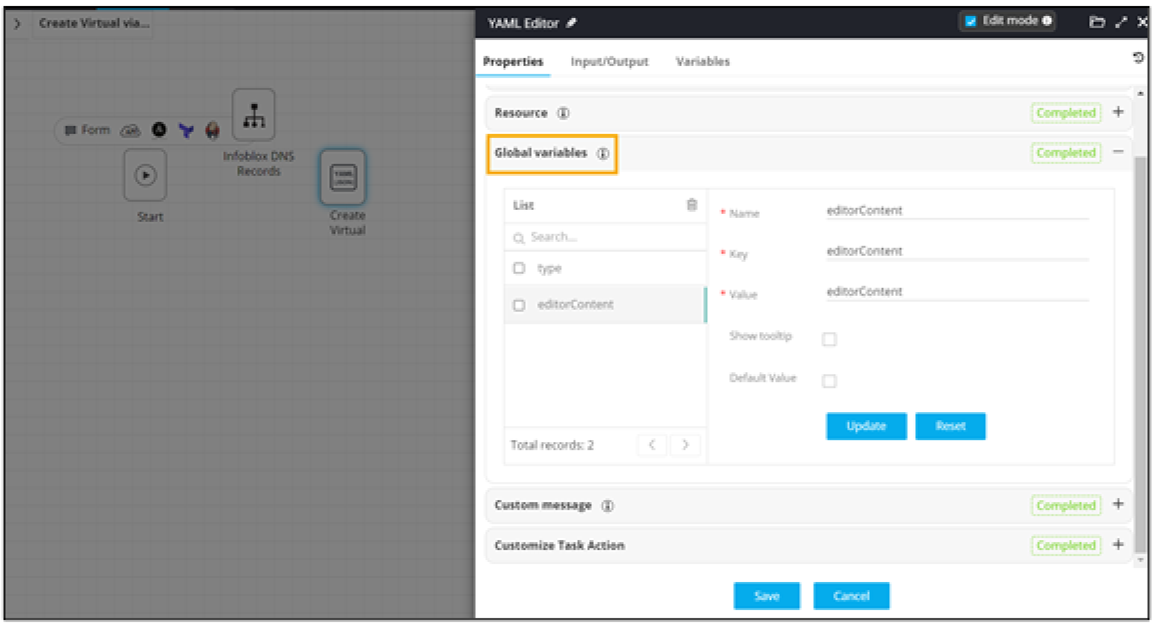
Ansible Executor

The following is a sample illustration to integrate Visual workflow and Ansible with Ansible as a southbound. This will allow discovery of any existing playbooks and further automate and orchestrate with Visual Workflow.

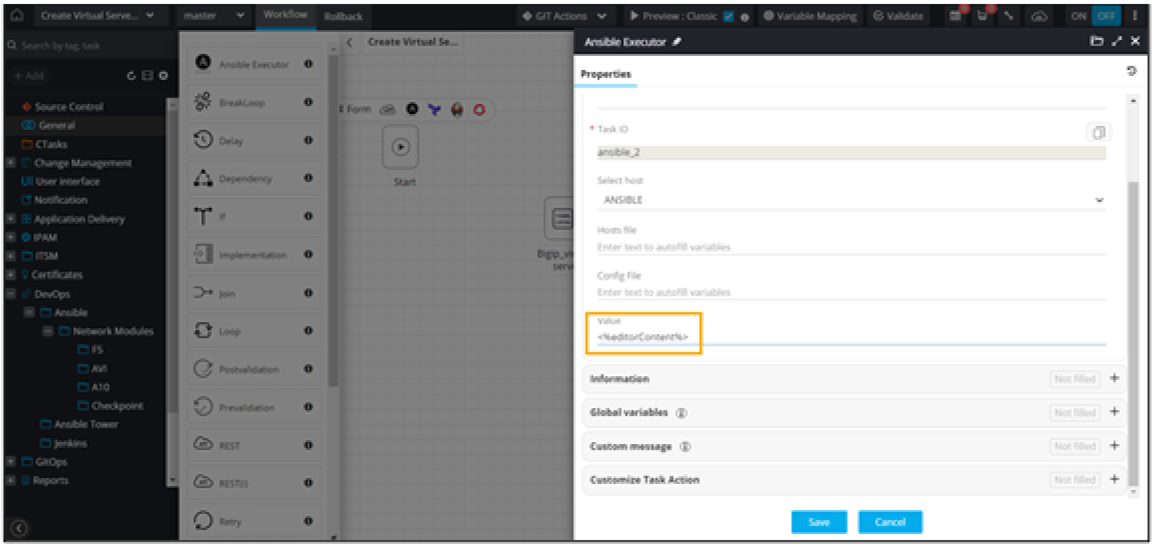


To connect to an Ansible instance, discover and reuse a BIG-IP playbook and integrate with Infoblox IPAM via Visual Workflow:

1. Configure an Ansible instance.
2. Go to the Workflow page.
3. Design a new workflow.
4. In the design workspace, from the **DevOps** folder in the left menu, select **Ansible** > **Network Modules** > **F5**.
5. From the **F5** folder in the left menu, drag and drop the **BigIP_Virtual_Server** task into the workspace.
6. In the **YAML Editor** task window, under **Properties**, in the **Global variables** section, declare the global variable in this task to pass data from the YAML task to the Ansible Executor on the southbound.

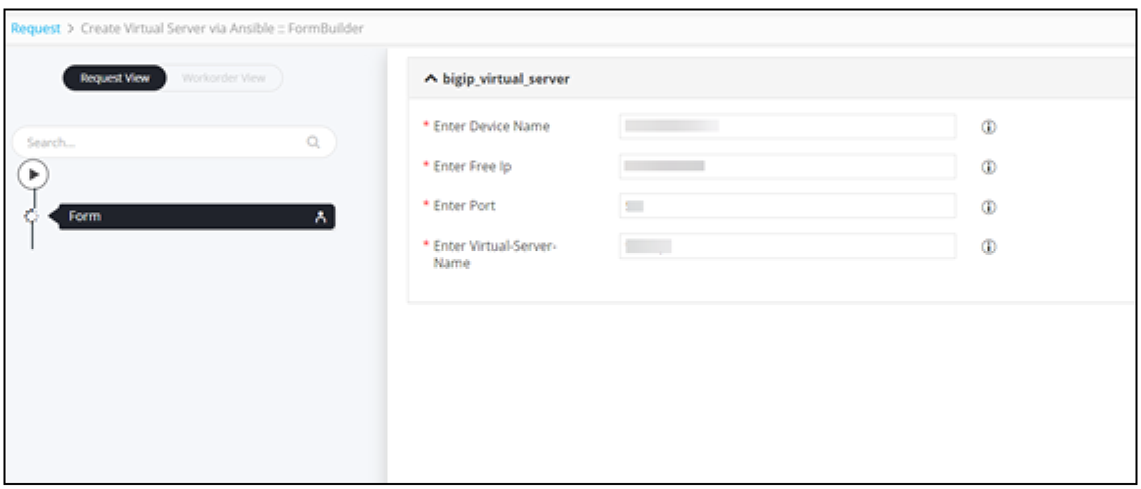


7. From the **General** section, drag and drop the **Ansible Executor** task.
8. In the **Ansible Executor** task window, under **Properties**, select the Ansible host on which implementation is to be done.
9. Map the global variable from the YAML task.



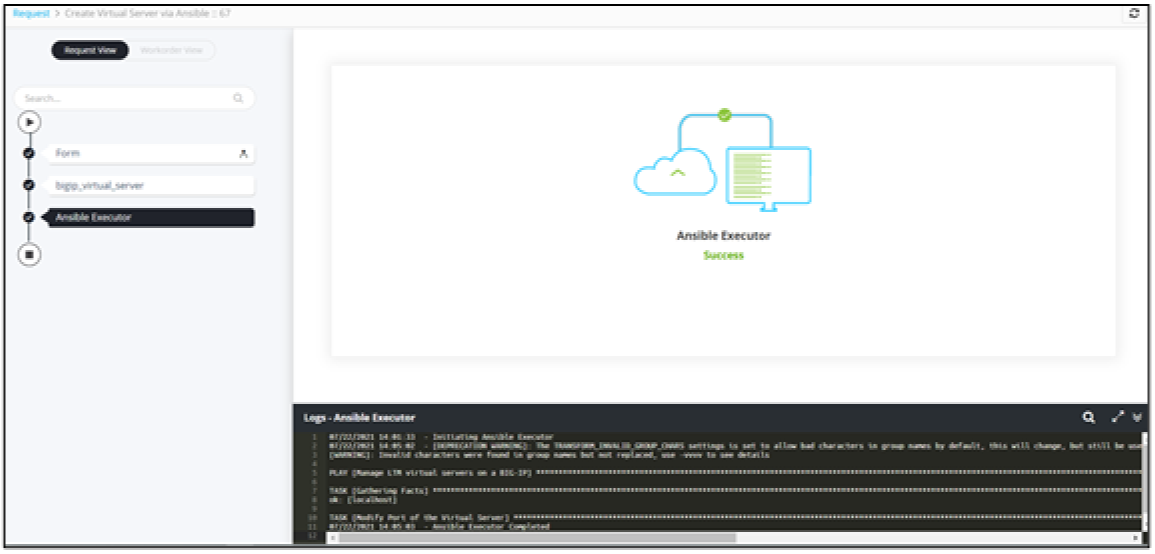
10. Auto-generate a self-service form.
11. Connect the workflow tasks and enable the workflow.
12. Trigger the workflow.

    The workflow execution page is displayed with the workflow inputs requested at the first stage.
13. Enter the input details in the Form.



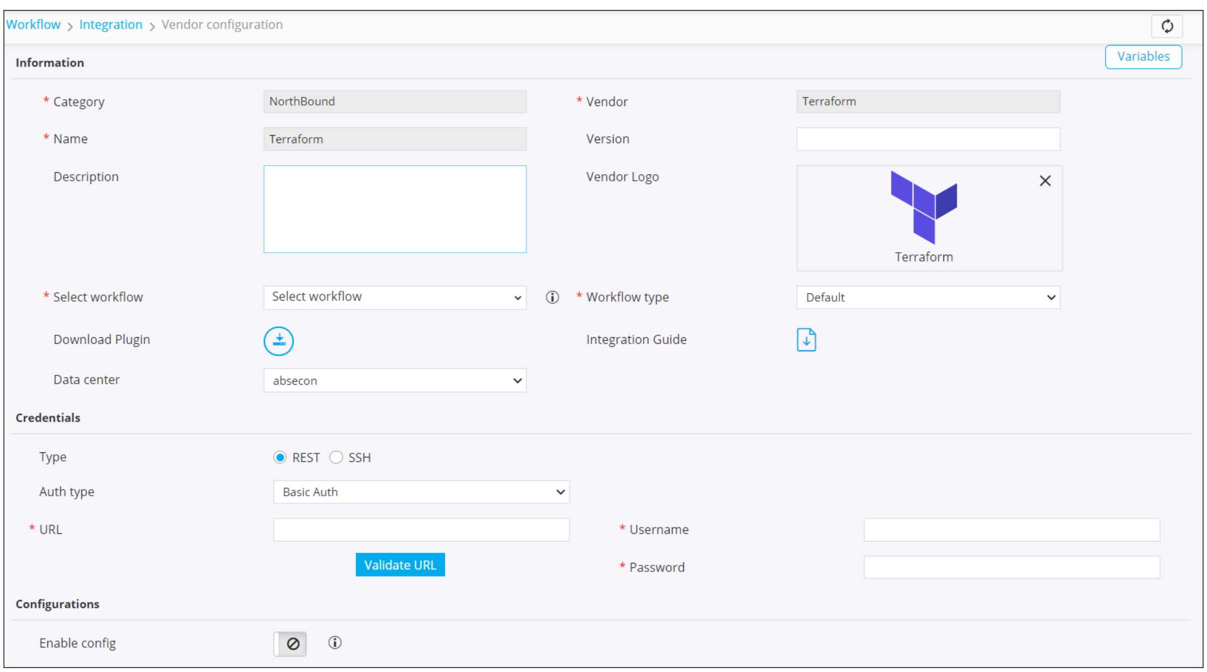
14. Click **Next**.
15. Submit the YAML task.

Ansible Executor task completed.



Terraform Vendor Configuration

1. Go to the Integration Hub.
2. On the **Workflow** > **Integration** page, click **Terraform**.
3. On the **Vendor configuration** page, enter the required field information.



4. Click **Save**.

> **Note:** For more information on Terraform - Visual Workflow Northbound integration, click here.

Using Variables in Integrations

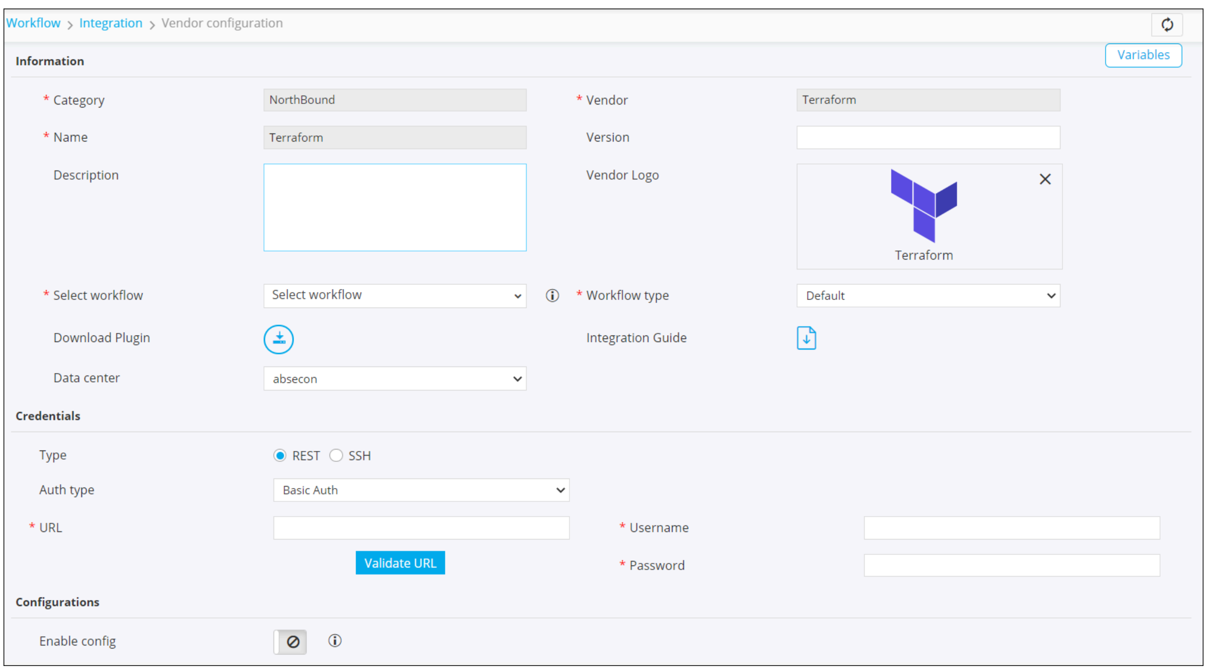
By integrating with third-party vendors such as Ansible, Bitbucket, Terraform and so on, you can define and use variables for your automation workflows.

The following syntax must be used to refer integration variables within a workflow:

`<%Integrations.<integration name>.<key name>%>`

To refer a Terraform integration variable, such as the IP address, in any task within the workflow:

1. Go to the Integration Hub.
2. On the **Workflow** > **Integration** page, click **Terraform**.
3. On the **Vendor Configuration** page, enter the field information to configure the Terraform integration.



4. To refer any of the configurations, for example the IP address, as a variable in any task within a workflow, use the syntax: `<%Integrations.<Terraform>.<key name>%>`

Workflow Tasks

A workflow palette or task is an independent business logic block required to automate and stitch together a workflow.

- Tasks are objective workflow elements required to automate and design a workflow
- Tasks can have custom and/or pre-defined business logic to aid in workflow automation
- Tasks can be defined as either user tasks or automated tasks
- Tasks can be cloned and reused with the workflow
- Tasks will have an Input/output (I/O operation) to pass data (in the form of variables) from one stage of a workflow to another.

Tasks are broadly divided into the following categories:

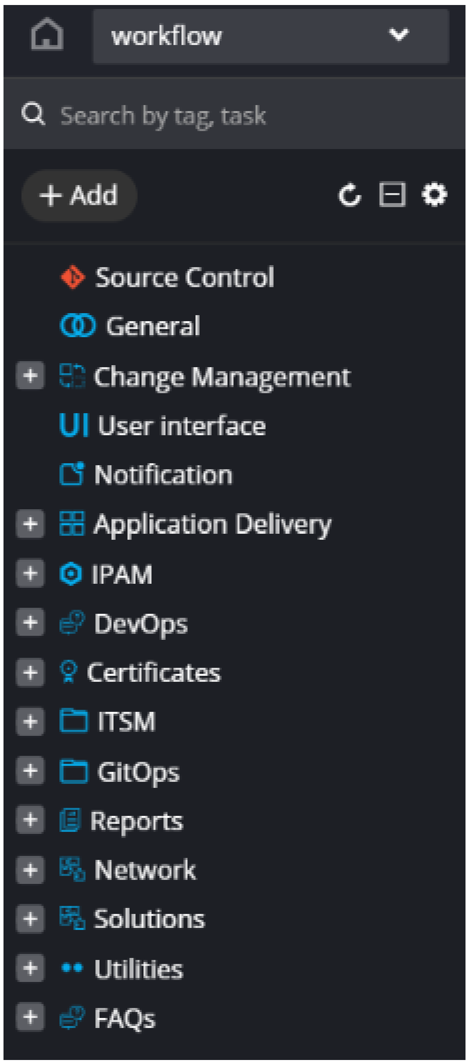
- General
- User Interface
- ChatOps and Notifications
- Change Management.

- Workflow Options
- Northbound Integrations

Accessing Workflow Tasks

1. Go to the Workflow page.
2. From the **Workflow** page, click **Create New**.

   The design space is displayed with the workflow tasks listed in the menu on the left.



Options and Descriptions for Workflow Tasks folder menu

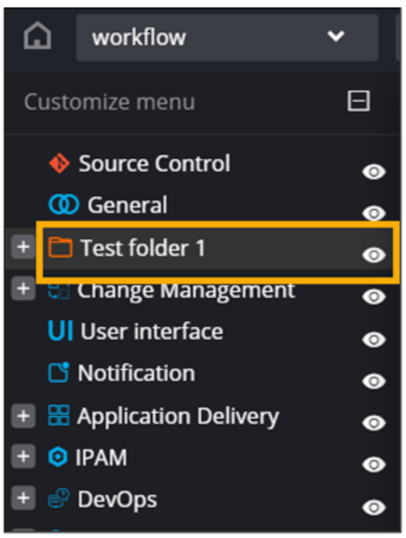
Options	Description
Search	Allows you to search for tasks by typing keyword(s)
Add	Allows you to add a new folder

Options	Description
	Refreshes the tasks folders menu
	Allows you to expand and collapse the task folders
	Allows you to hide folders from the workflow tasks menu and customize task sequence within a folder

Hiding a Task Folder

To hide a folder from the workflow tasks menu:

1. From the top right corner of the menu, click **(Settings)** icon.
2. Click **(View)** icon next to the folder you want to hide from the menu.



3. Click **Save**.

   The folder will not be visible in the menu.

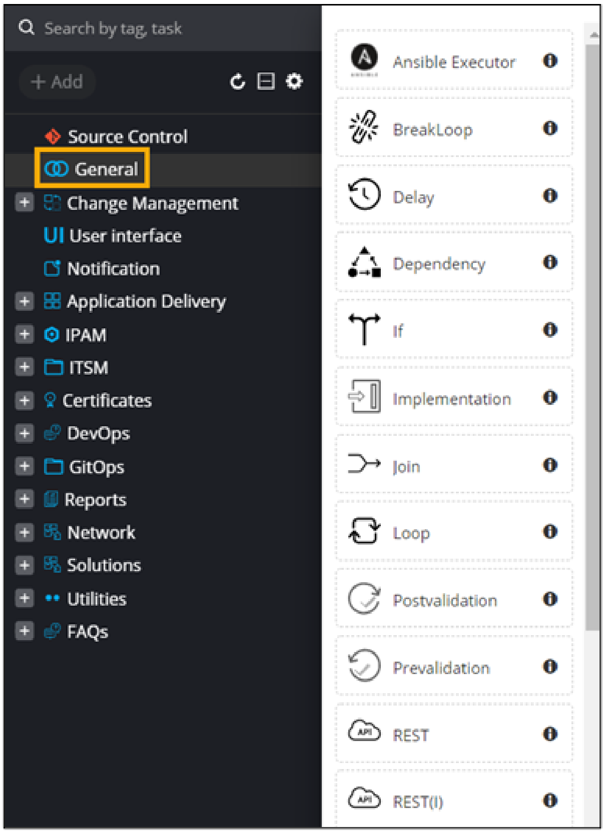
Reordering Task Sequence

To reorder the sequence of tasks within a folder:

1. From the top right corner of the menu, click **(Settings)** icon.
2. Select a folder from the menu.
3. Hold and drag tasks to change their sequence in the folder.

Task Category - General

This folder allows you to incorporate all flow control tasks within a workflow.



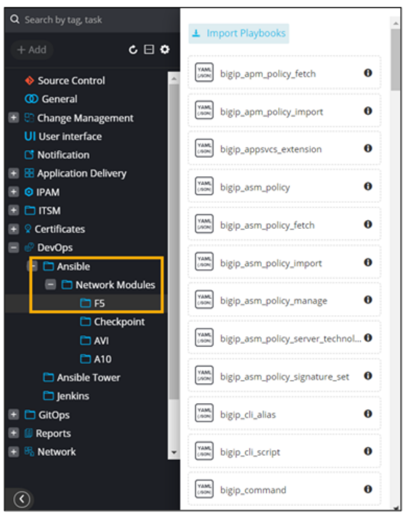
- Ansible
- Break Loop
- Delay
- Dependency
- If
- Implementation
- Join
- Loop
- Postvaildation
- Prevalidation
- REST
- REST(I)
- Retry

- Rollback
- Schedule
- Script
- Split
- Switch
- WorkOrder

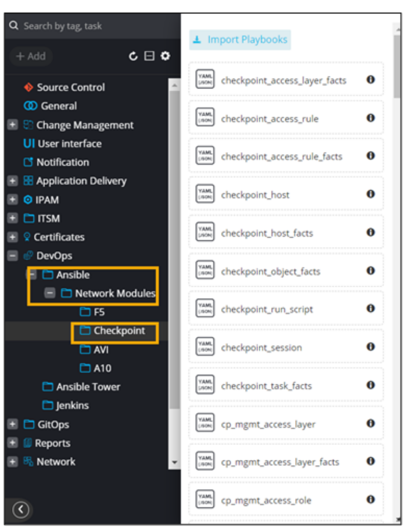
Ansible

This folder consists of the following Ansible Network Modules, which can be executed from playbooks:

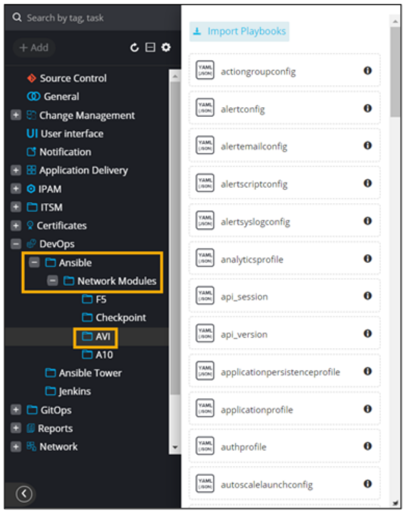
- **F5**



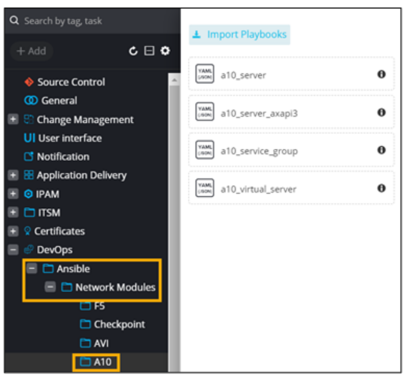
- **Checkpoint**



- **AVI**



- **A10**

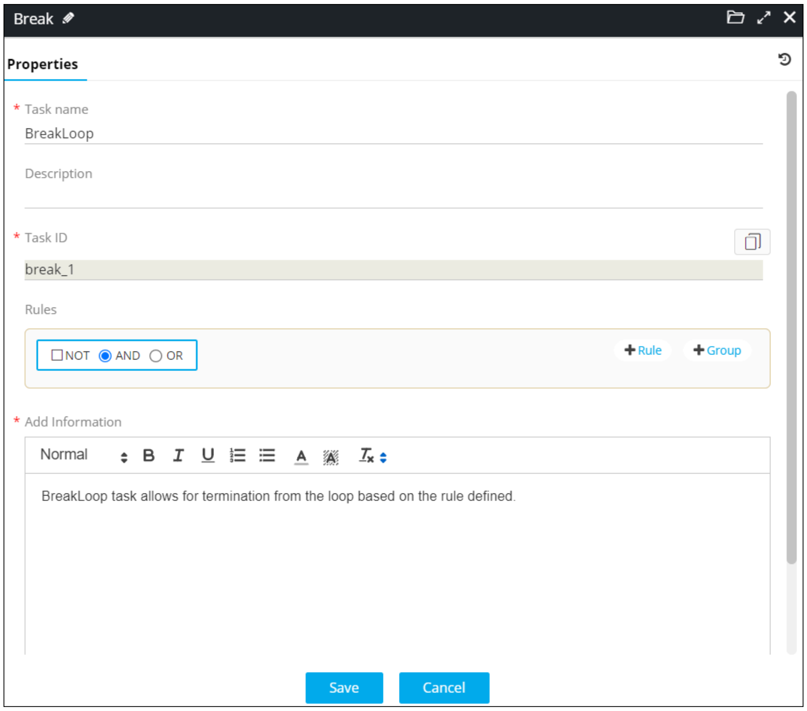


There is also a provision to import playbooks from Ansible.

Break Loop

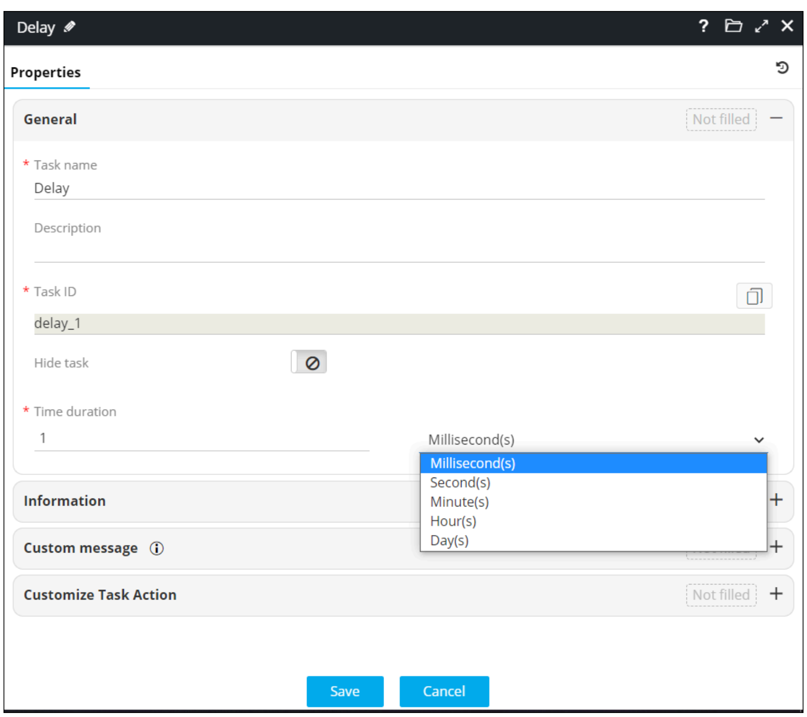
Break Loop task allows for termination from the loop based on the rule defined.

- Provision to define a break condition to terminate the loop.
- Provision to define rule or boolean condition (AND, OR, NOT) to test the loop variable.
  - If the condition holds true, the loop terminates.
  - If the condition holds false, control is passed to the loop.



Delay

Delay task allows you to configure delays between workflow tasks. Delay can be configured for Milliseconds, Seconds, Minutes, Hours and Days.



Dependency

This task allows you to explicitly define dependencies on previous tasks prior to executing the next task in the workflow sequence. This is helpful when you want to check whether a specific task has completed or not before going on to the next task.

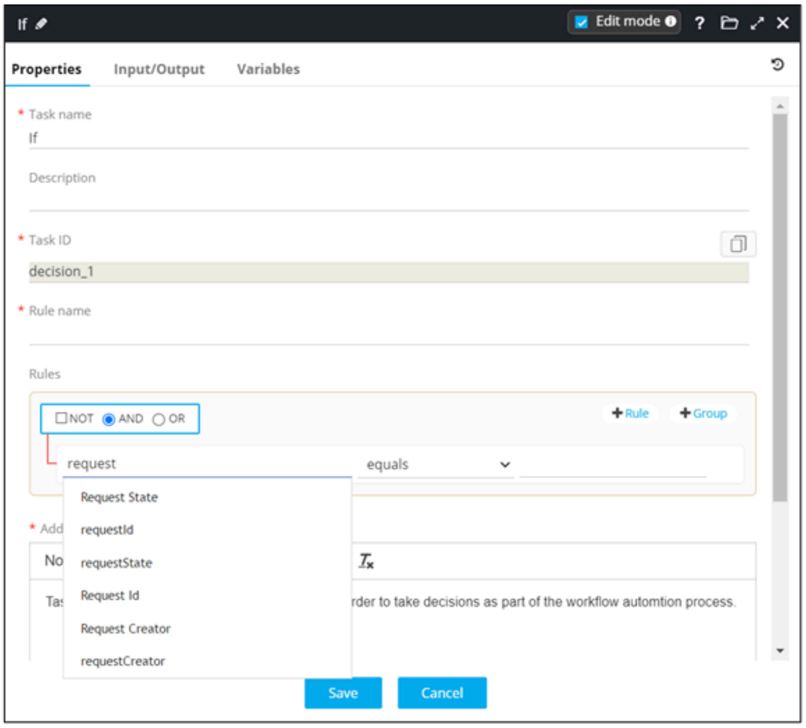
You can assign a wait time or duration for which the next task in the flow can wait for the dependent task to be completed. If the dependent task is completed within the wait time, the next task is executed. However, if the dependent task does not get completed with the assigned wait time, the workflow stops.

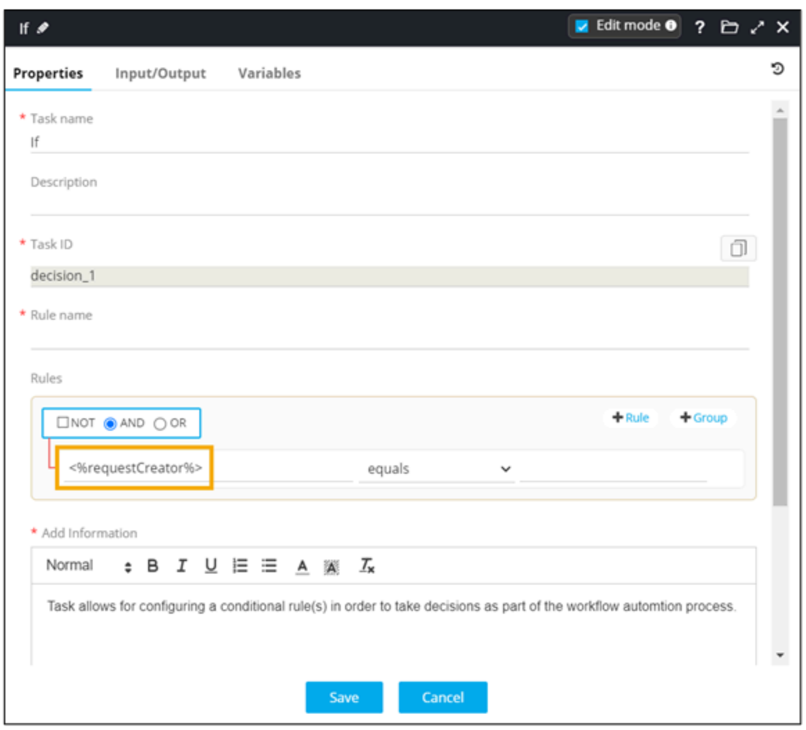
- Provision to select the dependent tasks which are defined within the workflow.
- Provision to assign a wait time or the duration for which the next task in the flow can wait for the dependent task to be completed.
- Task dependency will check for the successful completion of the dependent tasks.

If

The If task allows you to configure a conditional rule(s) in order to take decisions as part of the workflow automation process.

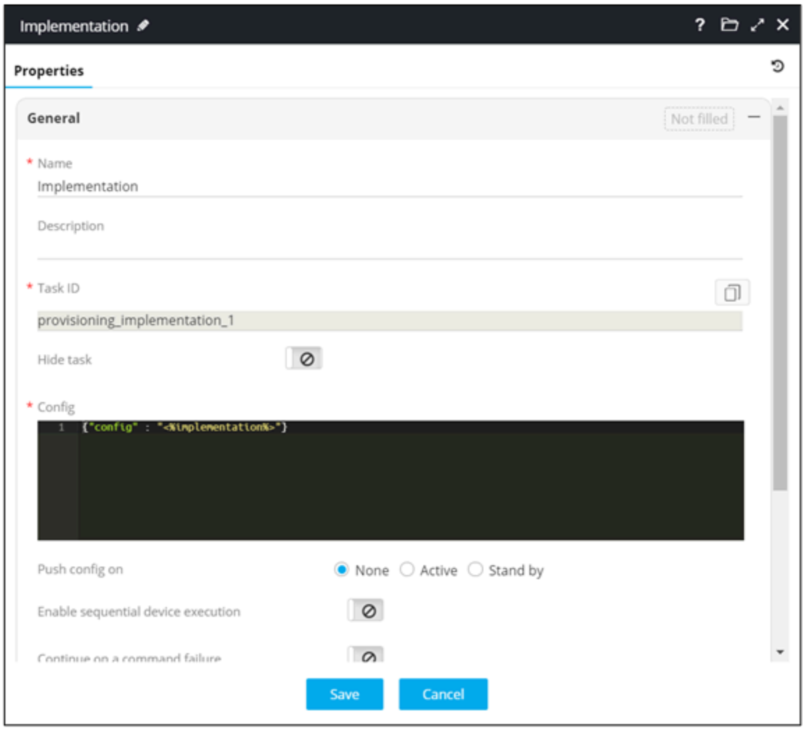
- Provision to define multiple rules based on which a workflow can be routed (maximum of two decisions can be taken).
- Provision to refer variables from a previous task as input.
- Provision to define conditional rule logic as a truth table.
- Provision to copy the task id and use it to reference across workflows.
- Provision to auto-populate the global variables.





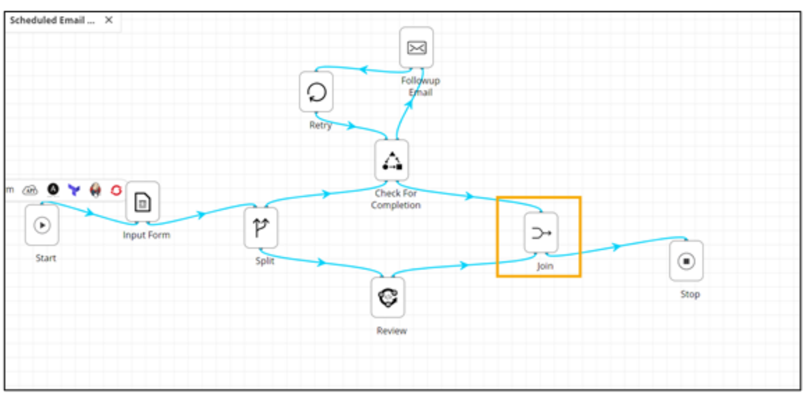
Implementation

This task allows you to perform Implementation changes on the end point.



Join

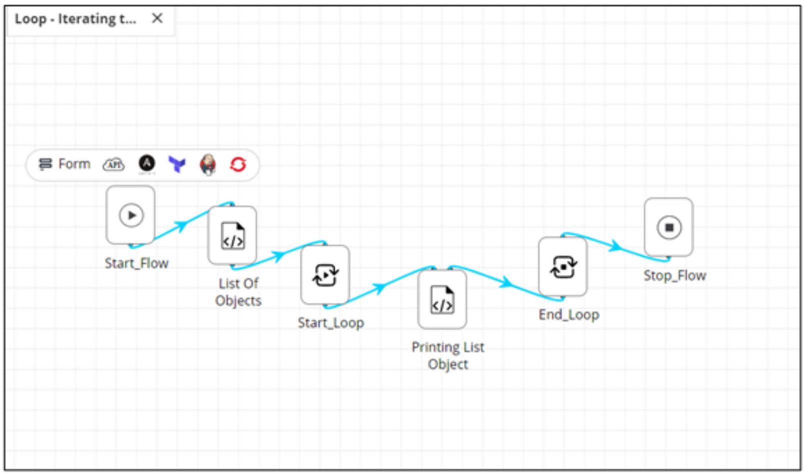
The Join task is used when multiple flows are branched out and executed in parallel. This task allows you to merge one or more branches of a workflow into a logical end.

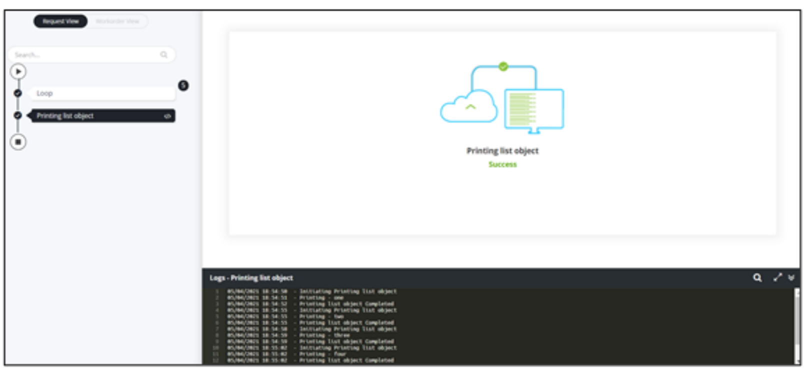


Loop

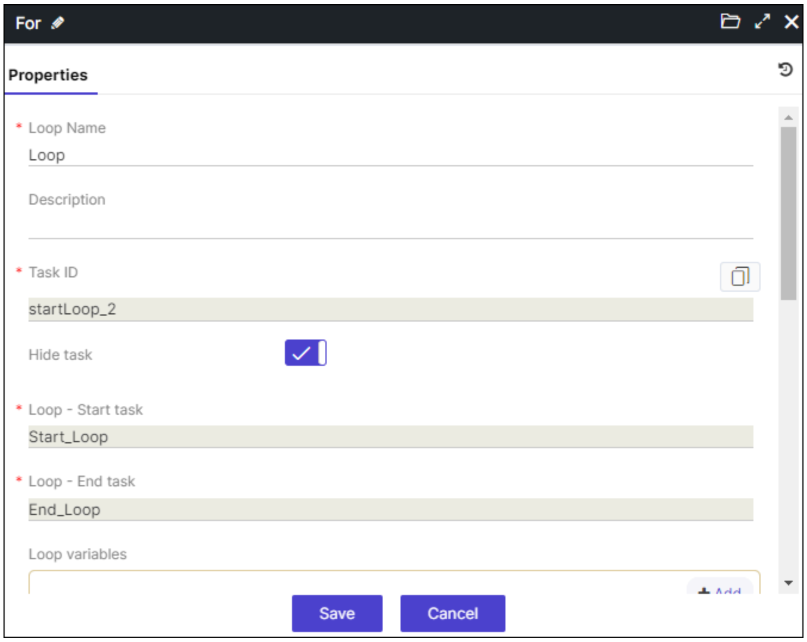
Loop task allows for repeated execution of a task or sub-process based on a specified condition. The Loop task consists of two parts – Start loop and End loop.

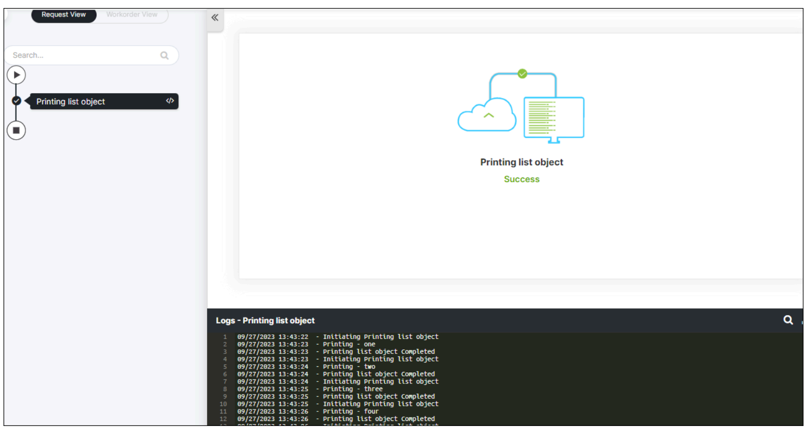
- Provision to define a loop for iterative execution of a task or sub-process.
- Provision to resume control to the next iteration of the loop in the event of failure of any task within the loop.
- Provision to define rule or Boolean condition (AND, OR, NOT) to execute the loop variable. If the condition holds true, the loop continues to iterate. If the condition holds false, the loop skips or exists.
- Provision to display the count of iteration in the stage view.





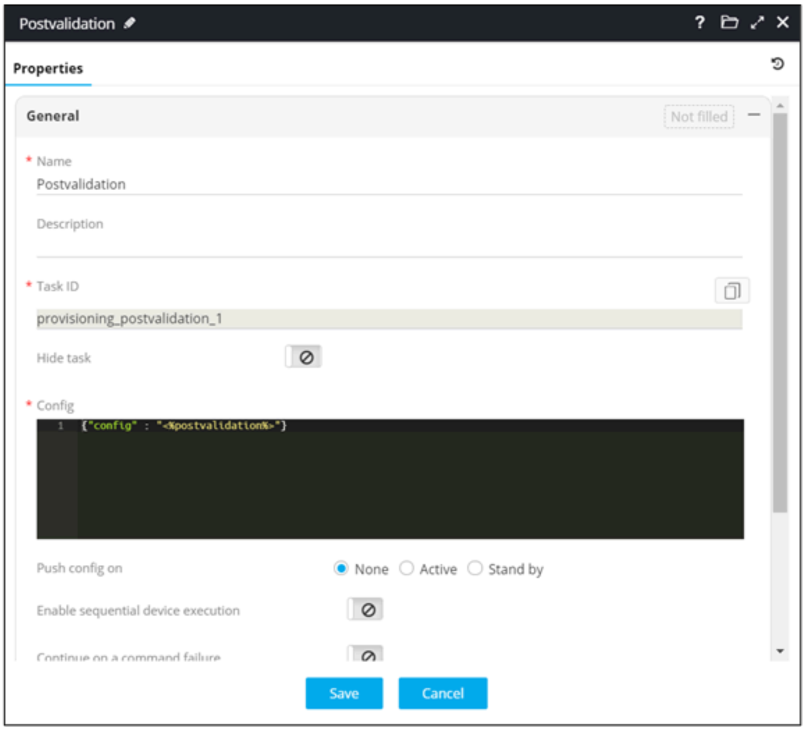
- Provision to enable or disable the visibility of loops in the stage view.





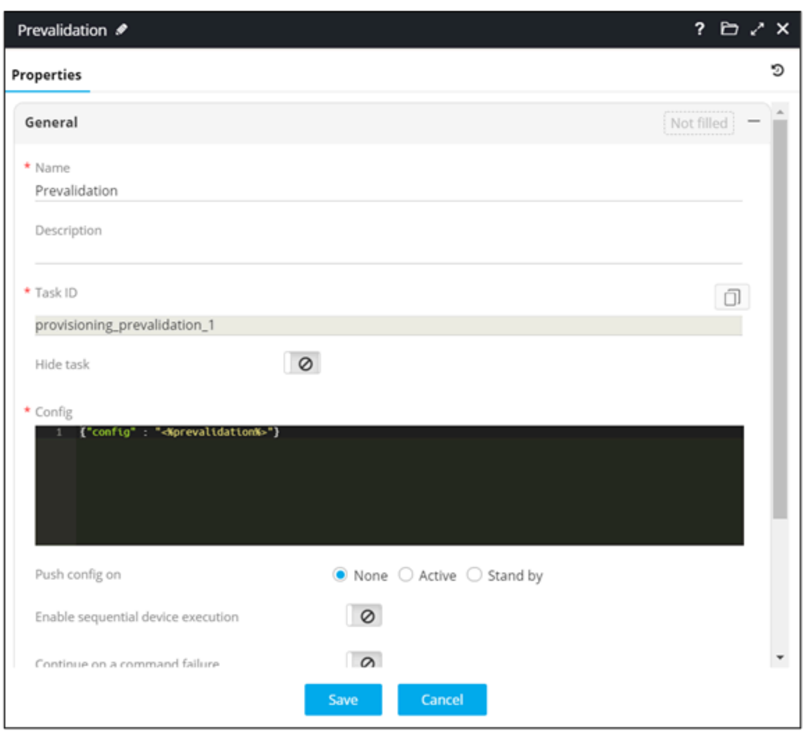
Postvaildation

This task allows you to perform post-validation changes on the end point.



Prevalidation

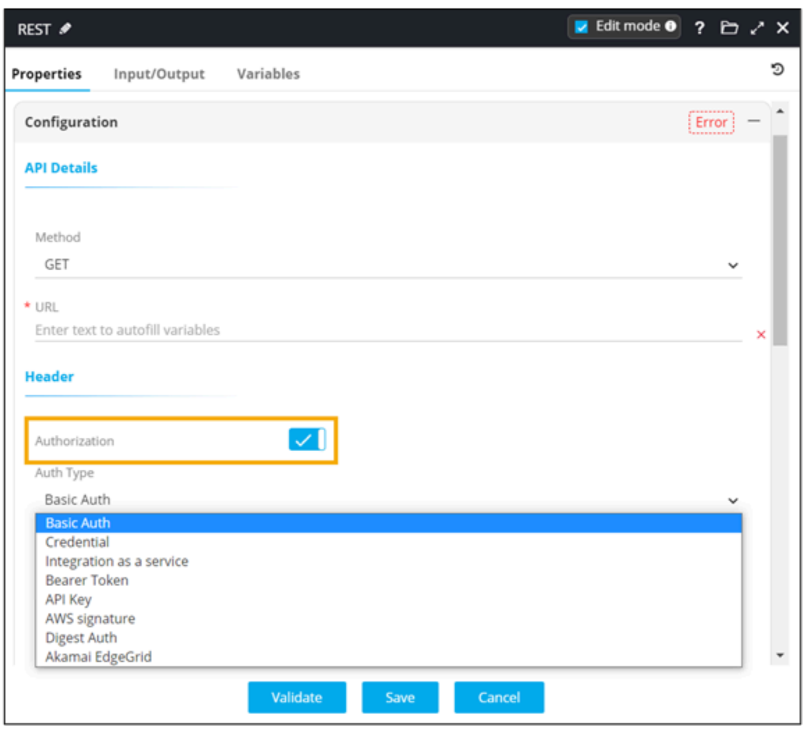
This task allows you to perform pre-validation changes on the end point.



REST

REST API task is used for external web service communications as part of the workflow automation process.

- Enhanced API auth support:
  - Basic Auth
  - Bearer Token
  - API Key
  - Digest Auth
  - AWS Signature
  - Akamai EdgeGrid.
- Support for 'GET', POST', 'PUT', 'DELETE' and 'PATCH' methods
- Support for multiple headers.



- Authorization Modes

Authorization Modes

Provision to support different types of authentication in the visual workflow Rest API task.

- Basic Auth
- Bearer Token
- API Key
- Digest Auth
- AWS Signature
- Akamai EdgeGrid
- oAuth

Basic Auth

The client sends HTTP requests with the Authorization header that contains a base64-encoded value of Username and Password.

1. In the **REST** task, under **Authorization**, select **Auth Type** as **Basic Auth**.

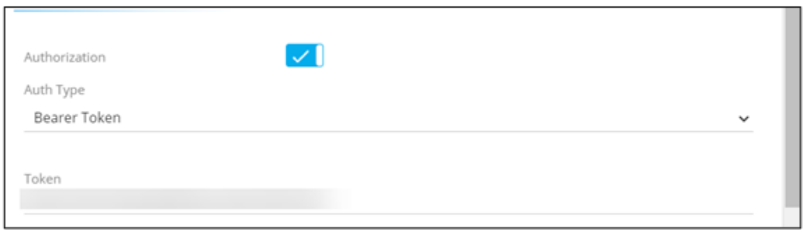
2. Enter the **Username** and **Password**.
3. Click **Save**.

Bearer Token

A bearer token is an HTTP authentication scheme that involves authentication with security tokens. The client can define this token in the Authorization header while making requests to protected resources.

1. In the **REST** task, select the **Authorization type** as **Bearer Token**.

2. Enter the **Token** value.

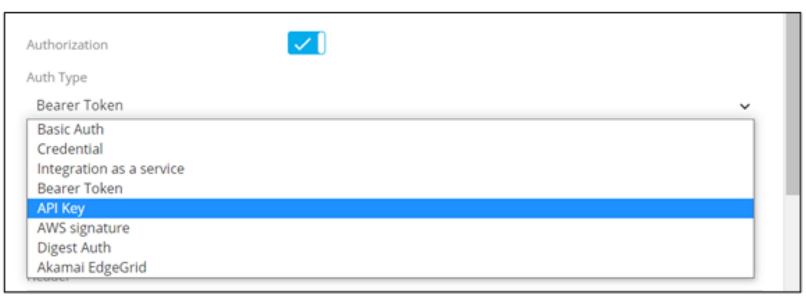


3. Click **Save**.

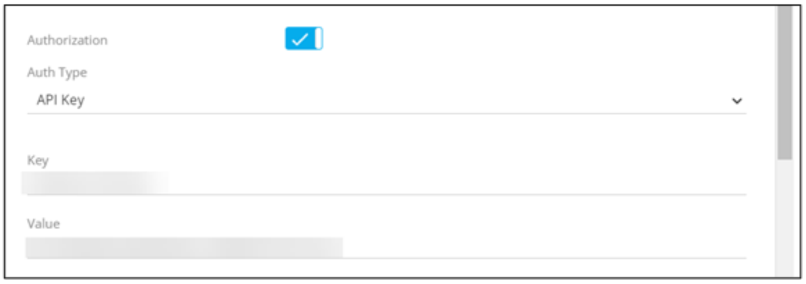
API Key

A unique generated value is assigned to all users to signify that the user is known. API keys can be sent in the query string as part of the URL or through headers.

1. In the **REST** task, under **Authorization**, select **Auth Type** as **API Key**.



2. Enter the **Key** and **Value**.

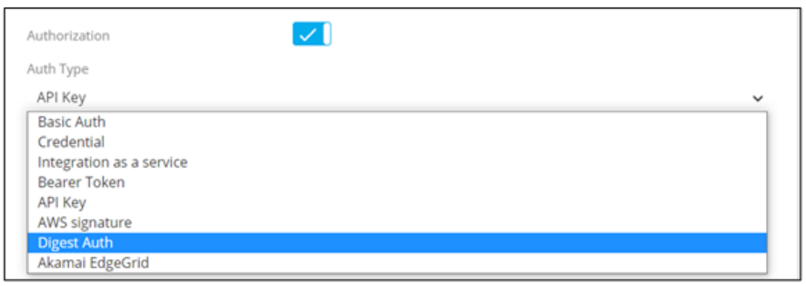


3. Click **Save**.

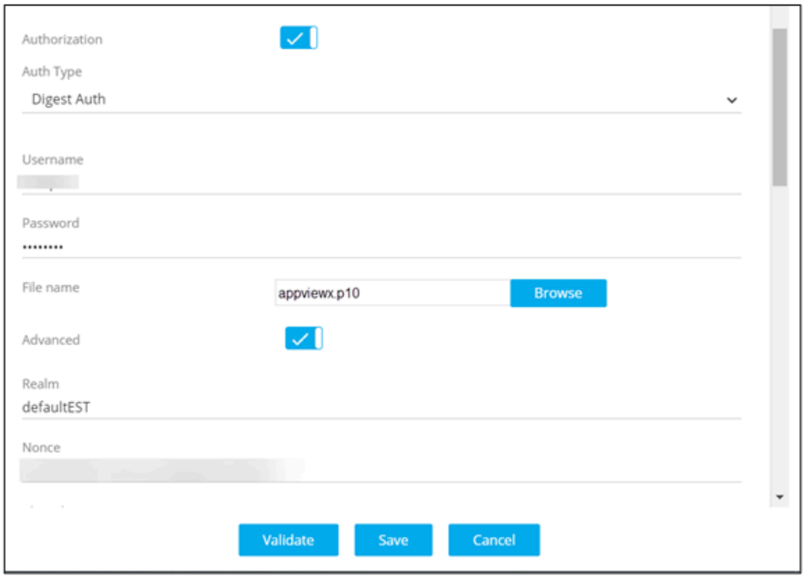
Digest Auth

This authorization is immune to replay attacks as it uses a one time number (a nonce) from the server. The same method is used to generate a hash-key on the server side.

1. In the **REST** task, under **Authorization**, select **Auth Type** as **Digest Auth**.



2. Enter all the field information.

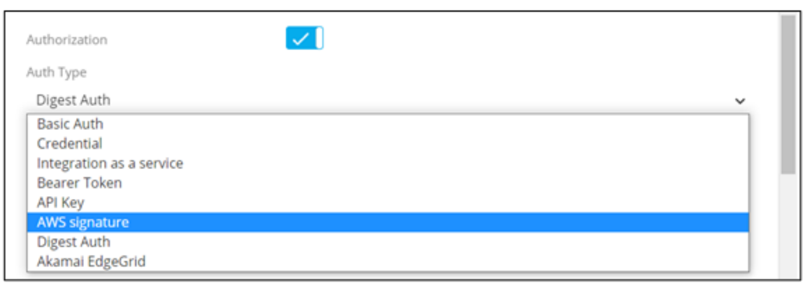


3. Click **Save**.

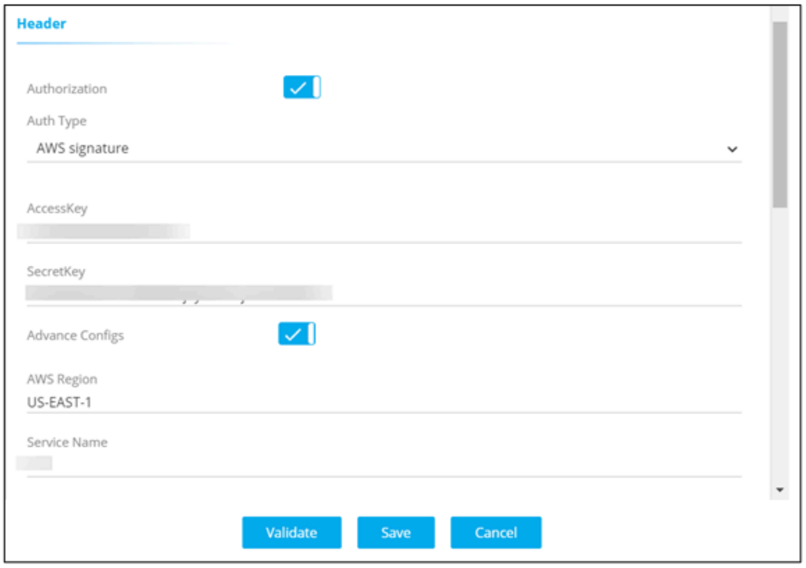
AWS Signature

Authentication information you send must include a signature.

1. In the **REST** task, under **Authorization**, select **Auth Type** as **AWS Signature**.



2. Under **Properties**, enter the field information.

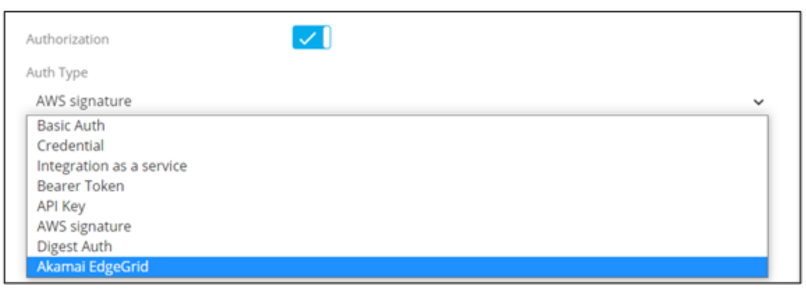


3. Click **Save**.

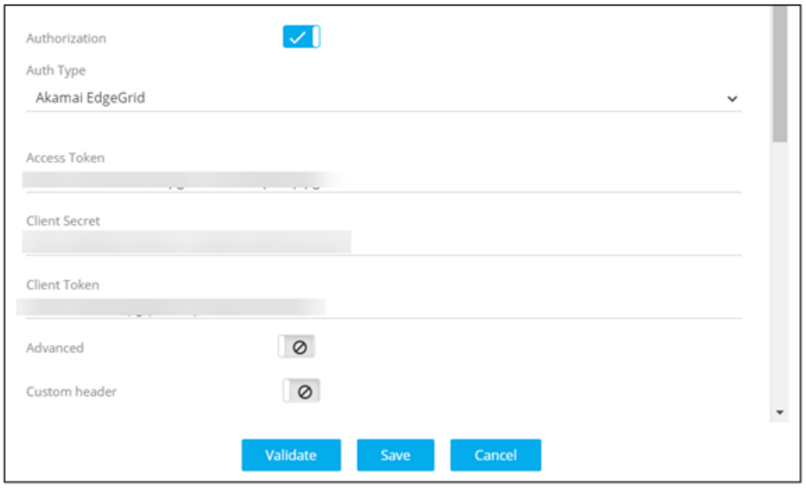
Akamai EdgeGrid

Akamai EdgeGrid uses a signature which is the base-64 encoding of the SHA-256 HMAC of the data to sign with the signing key.

1. In the **REST** task, under **Authorization**, select **Auth Type** as **Akamai EdgeGrid**.



2. Enter all the field information.

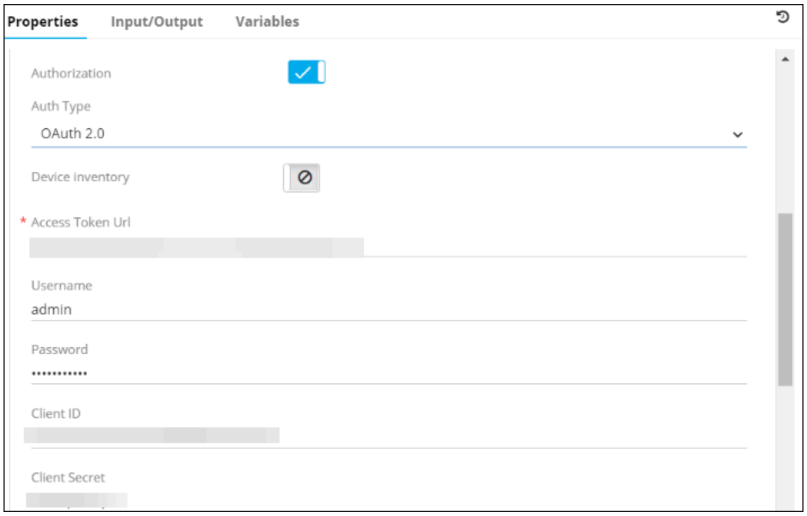


3. Click **Save**.

oAuth

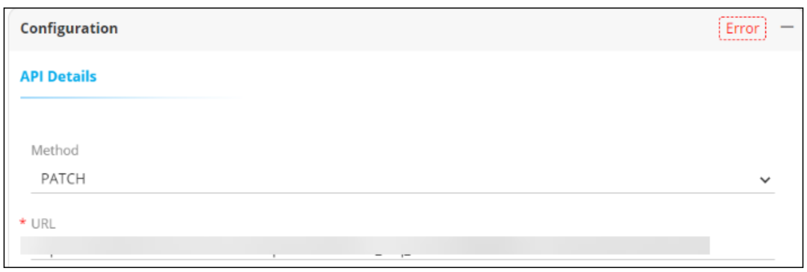
This authentication uses an access token that works as a temporary password (with fixed validity) instead of a regular password.

1. In the **REST** task, under **Authorization**, select **Auth Type** as **OAuth 2.0**.
2. Enter all the required field information.
   - Access Token URL of the oAuth server.
   - Username
   - Password
   - Client ID
   - Client Secret.



Once this information is submitted, the client will receive an authorization code/refresh token.

3. Add this token received from the oAuth server to the **URL** in the **API Details** section.

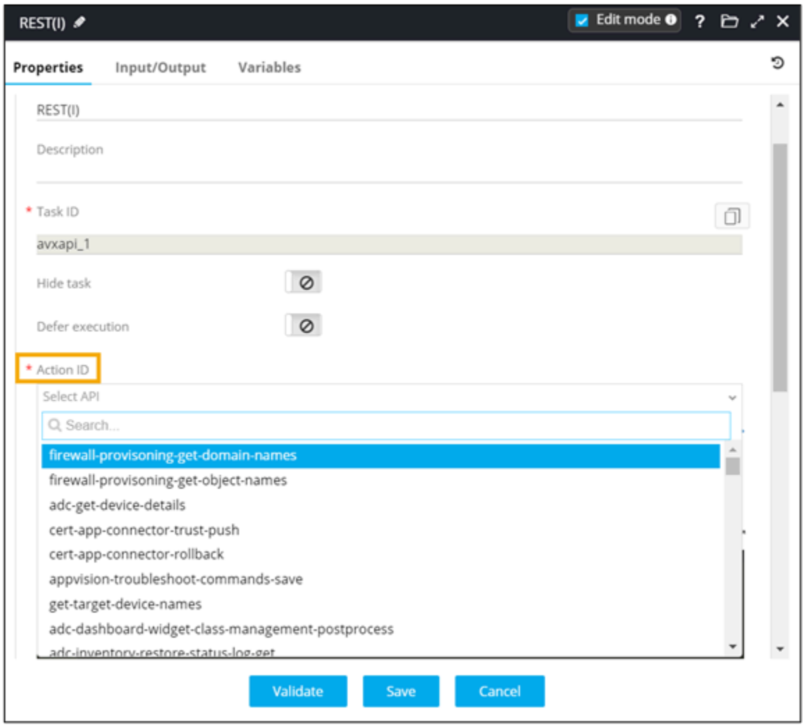


4. Click **Save**.

REST(I)

REST (I) is an API Task used for internal communication within AppViewX modules such as Device Management, Dashboard, Certificate, Firewall, Statistics, Accounts and so on. This involves using in-house AppViewX APIs.

- Uses the session of the logged in user to make internal API calls
- Option to define Action ID - an internal API that has to be defined in the database collection
- Provision to pass the required payload.
- Examples include:
  - Internal communication using ‘Add-device-API’ in order to add a new device through workflow
  - Internal communication using ‘Get unused Objects’ in order to retrieve unused object details from Insight through workflow.

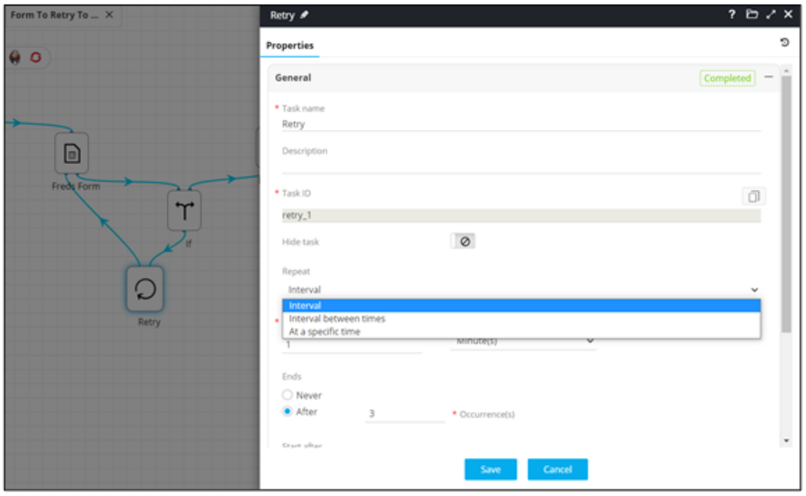


Note: Relevant APIs have to be added into the database collection (visualworkflow_avxapi) in order to be consumed as Action ID via the REST (I) task.

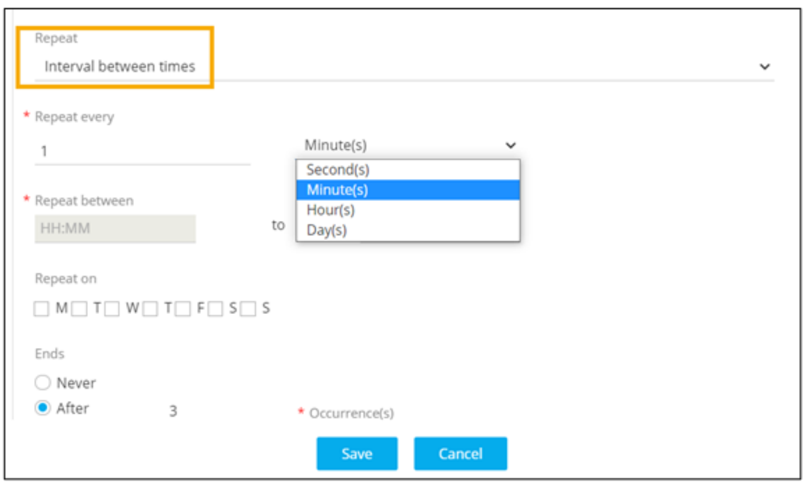
Retry

Retry task allows you to define custom retry intervals between workflow task(s) during automation.

- **Interval**: Provision to retry for a certain time interval (seconds, minutes, hours, days) before exit



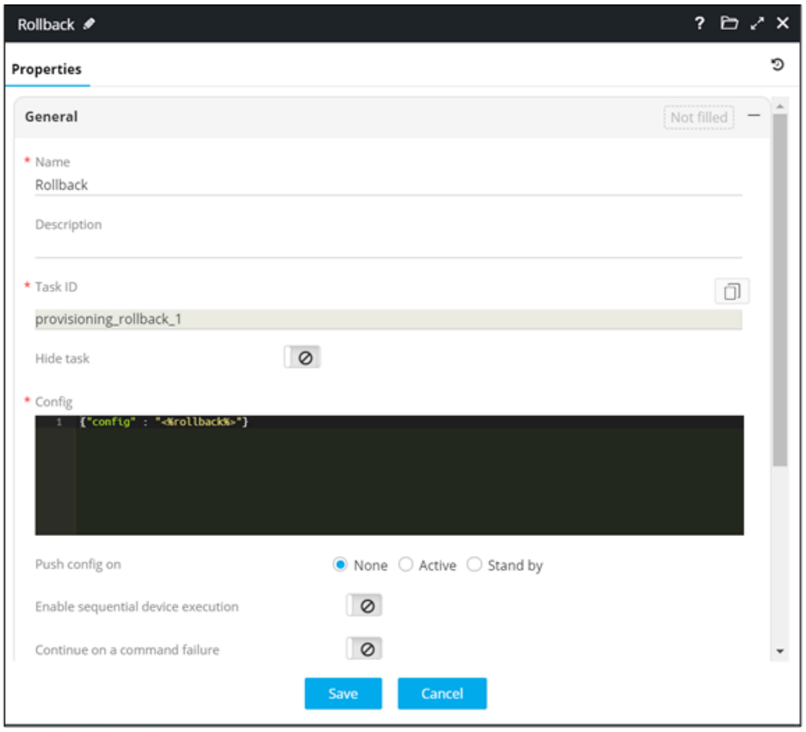
• **Interval between times**: Provision to retry for a time between specific time intervals before exit



• **At a specific time**: Provision to retry only at a specific time, day before exit.

Rollback

This task allows you to rollback implemented changes.



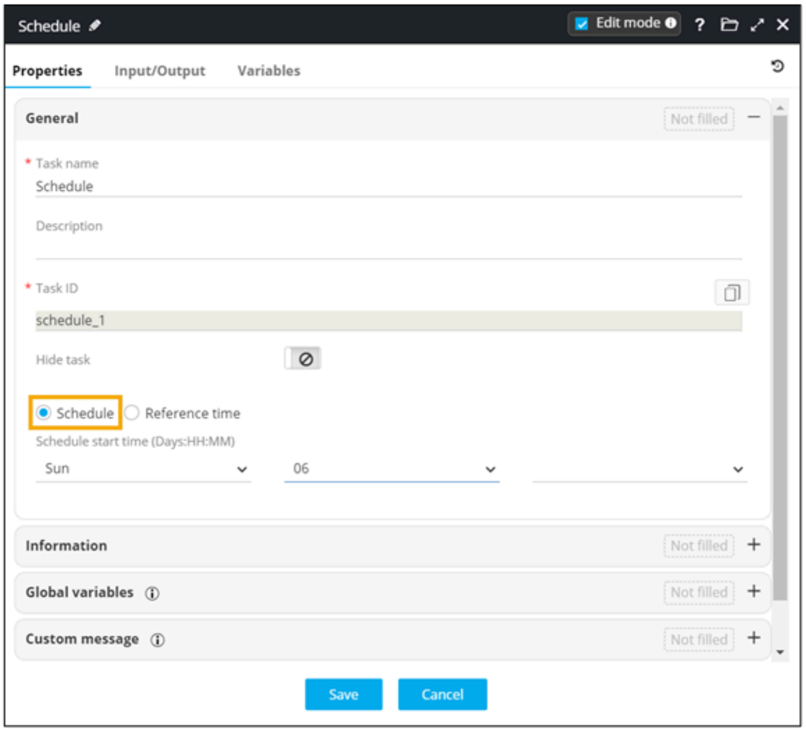
Note: For more information, refer to the section on Rollback Workflow.

Schedule

The scheduled task allows you to schedule a specific task in the workflow sequence.

- Provision to add a scheduler job between different workflow tasks
- Provision to schedule the job by a specific day of the week, hour and minute
- Provision to pass schedule date, time as variable across workflow tasks.

For example, in order to auto implement or schedule the implementation of a work order, you can define a Schedule task after a Review task.



Script

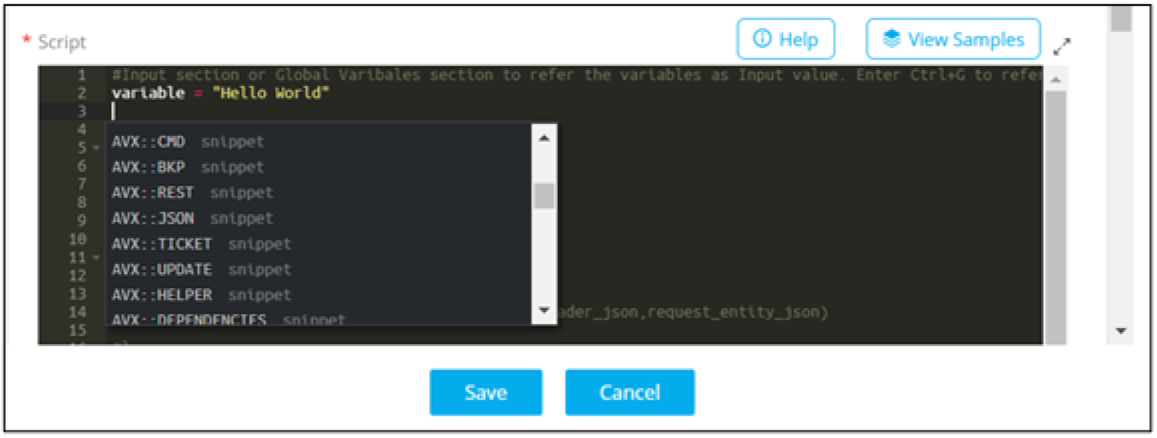
Script task allows for defining of custom scripts (using python), business logic as part of the workflow automation process.

- Supported versions: Python 3.x
- Actual version: Python 3.5
- Provision to debug script with variables
- Allows for comprehensive debugging of the scripts, syntax, and error handling as part of the workflow automation process
- Script tasks can be used with two connections - success and failover
- Provision to select a list of custom commands from the script editor such as AVX::CMD, AVX::OUTPUT, AVX::LOG, AVX::CONFIG and AVX::REST
- Provision to reference global variables.

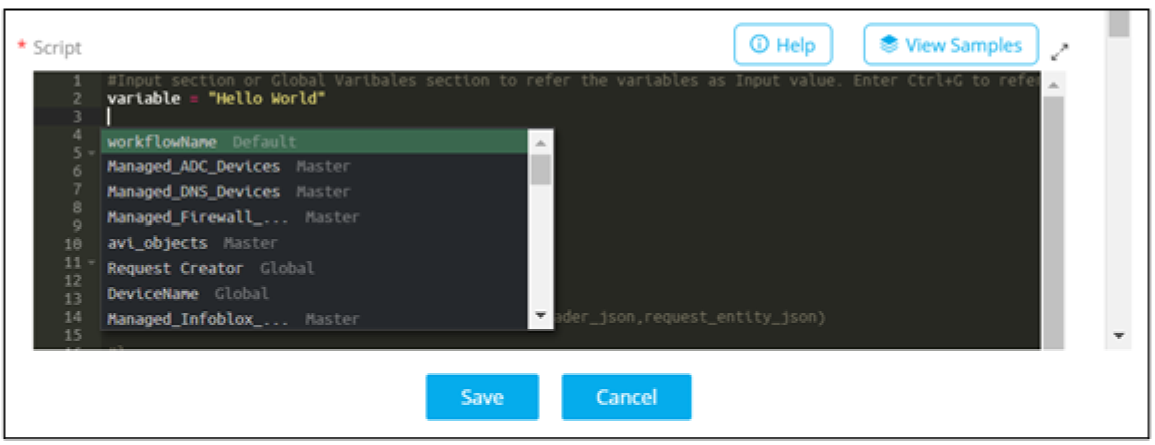
- Using Keyboard Shortcuts
- Custom Commands

Using Keyboard Shortcuts

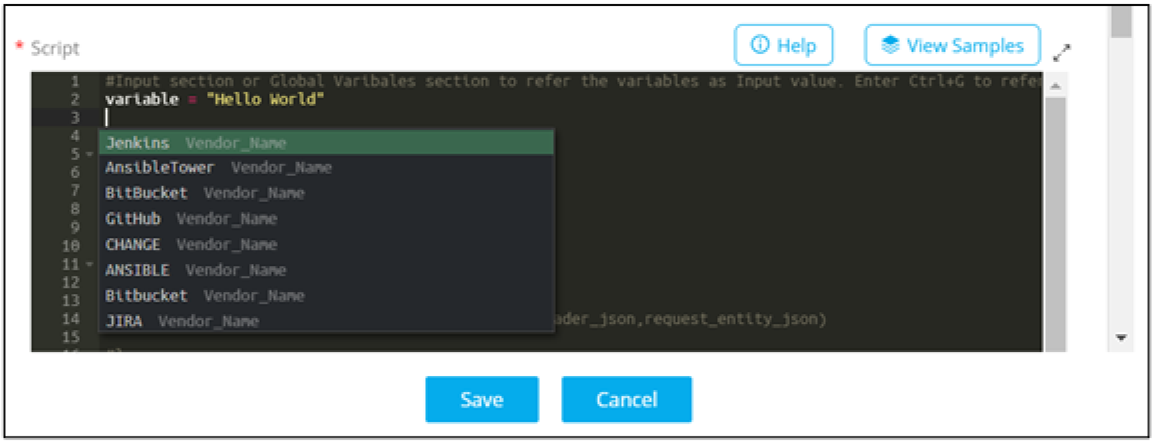
- Provision to use keyboard shortcut **Ctrl+Space** to get a list of custom commands.



- Provision to use the keyboard shortcut **Ctrl+G** to refer global variable(s).



- Provision to use the keyboard shortcut **Ctrl+K** to load integration variables.



Custom Commands

The workflow design process allows you to natively use a few custom commands within Tasks in order to cater to specific use cases. This is an extension to the custom commands used natively in APS such as AVX::CMD, AVX::REST, and AVX::JSON etc.

Following are the command generating functions available in the Workflow Studio:

- AVX::CONFIG
- AVX::PUSH and AVX::JSONCONVERTER
- AVX::LOG
- AVX::CMD
- AVX::REST and AVX::JSON
- AVX::PASSWORD
- AVX::UPDATE
- AVX::FILEDOWNLOAD
- AVX::OUTPUT
- AVX::MERGECONFIG

AVX::CONFIG

This command is used when other commands like AVX::CMD, AVX::REST and AVX::JSON are used within a workflow task such as Script. AVX::CONFIG is used to assign the commands generated script under a specific key in a python dictionary or a variable. In a workflow, the commands to be generated can be assigned using AVX::CONFIG (within a single script) and can be referred anywhere across the workflow.

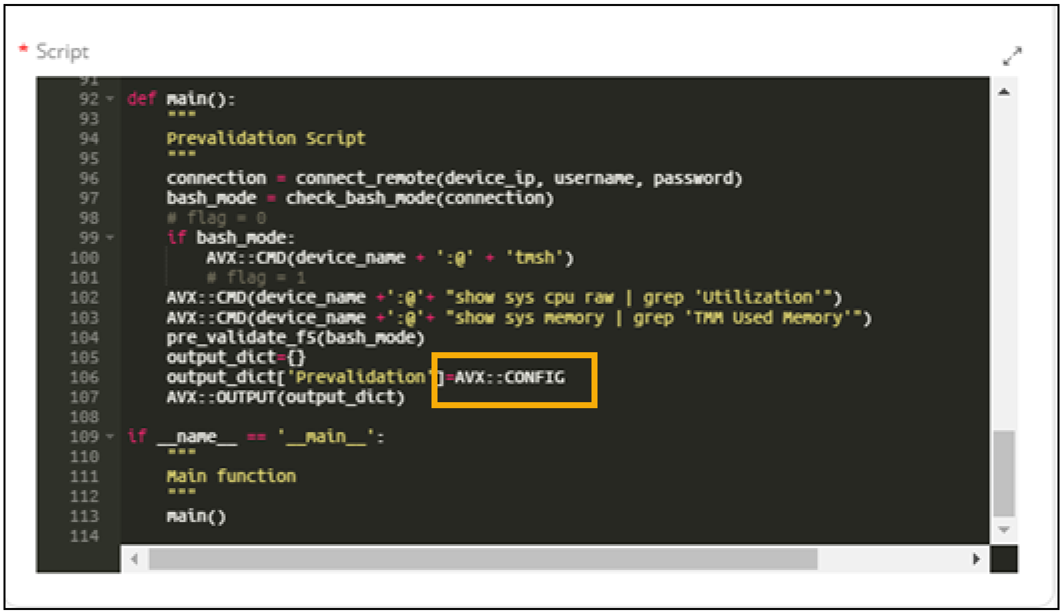
Usage:

`AVX::CMD("testdevice"+':@'+"implementation command 1")`
`AVX::CMD("testdevice"+':@'+"implementation command 2")`
`commands ={}`
`commands["implementation"] = AVX::CONFIG`
	• Commands 1 and 2 will get added to the implementation key.
	• This will add up the commands generated in the script to the provided key.

Example: Generating F5 VIP prevalidation commands

```
AVX::CMD(device + ':@' + 'tmsh')
AVX::CMD(device + ':@' + 'list ltm pool ' + pool_name)
AVX::CMD(device + ':@' + 'list ltm virtual ' + vip_name)
AVX::CMD(device + ':@' + 'quit')

config = {}
config['prevalidation_config'] = AVX::CONFIG
AVX::LOG('Configurations generated successfully.')
```

• AVX::CONFIG within script



Create F5 Sample Config:

```
import sys

import json

sys.path.insert(0,AVX::DEPENDENCIES)

sys.path.insert(0,AVX::HELPER)

import os

import f5

import appviewx

import Decrypt_Python3 as Decrypt

from f5 import *

from Decrypt_Python3 import getpassword, getencoded

import paramiko

null=None

connect_db=appviewx.db_connection()

db = connect_db.appviewx

vendor = 'F5'

device_name = '<%Device%>'

fqdn = '<%AppName%>'.lstrip('www.').split('.')[0]

vip_ip = '<%IP%>'

vip_port = '<%Port%>'

persistence = '<%Persistence%>'

create_http_profile = '<%CreateHTTPProfile%>'

create_http_monitor = '<%CreateHTTPMonitor%>'
```



```
lb_method = '<%LoadBalancingMethod%>'

defaults_from = '<%DefaultsFrom%>'

ratio = '<%Ratio%>'

interval = '<%Interval%>'

timeout = '<%Timeout%>'

send_string = '<%SendString%>'

receive_string = '<%ReceiveString%>'

existing_monitor = '<%ExistingMonitor%>'

irulecheck = '<%irulecheck%>'

irulefile = '<%uploadedfile%>'

pool_members = <%PoolMembers%>

partition='Common'

output_dict = {}

cmd2 = []

deviceInfo = db.device

devVersion = deviceInfo.find_one({'name':device_name})['version']

device_ip = deviceInfo.find_one({'name':device_name} , {"ip": 1})['ip']

username, password = Decrypt.getpassword(device_name)

try:

    tmos_version =deviceInfo.find_one({"name": device_name}, {'detailedVersion': 1})['detailedVersion'].split(" ")[0]

except  Exception:

    tmos_version = None

def connect_remote(IP,username,password):

    try:

        remote_conn=paramiko.SSHClient()

        remote_conn.set_missing_host_key_policy(paramiko.AutoAddPolicy())

        remote_conn.connect(IP,username=username,password=password,look_for_keys=False,allow_agent=False)

        return remote_conn

    except Exception as e:

        print (str(e))

def check_bash_mode(connection) :

    stdin, stdout, err=connection.exec_command('echo "testing"')

    err = err.read().decode()

    if "Error" not in err:

        return True

    else:

        return False
```

```
def create_f5():
    '''
    Create F5 vip
    '''
    monitors = []
    cmd = []
    connection = connect_remote(device_ip, username, password)
    bash_mode = check_bash_mode(connection)
    if bash_mode:
        cmd.append('tmsh')
        cmd2.append('tmsh')
    if tmos_version and tmos_version not in ('11.4.1','11.4.3'):
        cmd.append("create cli transaction")
        cmd2.append("create cli transaction")
    if devVersion in ['v11', 'v12']:
        cmd.append('cd /{0}'.format(partition))
        cmd2.append('cd /{0}'.format(partition))
    else:
        cmd.append('modify cli admin-partitions update-partition {0}'.format(partition))
        cmd2.append('modify cli admin-partitions update-partition {0}'.format(partition))
    if create_http_monitor == 'Yes':
        http_mon = F5Monitor(**{'type': 'http',
                    'app_name': fqdn,
                    'vip_port': vip_port,
                    'interval': interval,
                    'timeout': timeout,
                    'send': send_string,
                    'recv': receive_string})
        cmd.append(http_mon.create)
        cmd2.append(http_mon.delete)
        mon_index = len(cmd2) - 1
        monitors.append(http_mon)
        output_dict['monitor'] = http_mon.name
    if existing_monitor != 'None':
        ex_mon = F5Monitor(**{'type': existing_monitor,
                       'associate': True})
        monitors.append(ex_mon)
```

```
    output_dict['monitor'] = ex_mon.name
profile_list = []
if create_http_profile == 'Yes':
    prof = F5Profile(**{'type': 'http',
                'app_name': fqdn,
                'vip_port': vip_port,
                'defaults-from': defaults_from})
    if defaults_from and defaults_from.lower() != "select" and defaults_from.lower() != "none":
        prof_create_cmd = prof.create
    else:
        prof_create_cmd = prof.create.split("defaults-from")[0]
    cmd.append(prof_create_cmd)
    cmd2.append(prof.delete)
    profile_list.append(prof)
    output_dict['profile'] = prof.name
else:
    output_dict['profile'] = ''
pool = F5Pool(**{'lb_mode': lb_method,
              'app_name': fqdn,
              'port': vip_port,
              'ratio': ratio,
              'monitors': monitors,
              'pool_members': {'keys': ['PoolMemberIP', 'PoolMemberPort', 'Ratio'],
                               'values': pool_members}})
cmd.append(pool.create)
cmd2.append(pool.delete)
output_dict['pool'] = pool.name
per = F5Persistence(**{'persistence': persistence})
output_dict['persistance'] = per.name
vip = F5VirtualServer(**{'app_name': fqdn,
             'vip_ip': vip_ip,
             'vip_port': vip_port,
             'pool': pool,
             'profiles': profile_list,
             'persistence': per})
cmd.append(vip.create)
cmd2.append(vip.delete)
```

```
    vip_index = len(cmd2) - 1
    try:
        cmd2[vip_index], cmd2[mon_index] = cmd2[mon_index], cmd2[vip_index]
    except:
        pass
    output_dict['vip'] = vip.name
    if irulecheck == "Yes":
        # AVX::LOG(str(type(irulefile))+" "+str(irulefile))
        fileName = irulefile['fileName']
        #filename = 'irulee1.txt'
        try:
            # file_location = '/home/'+username+os.sep
            file_location = '/config/'
            destination_json = AVX::JSONCONVERTER(device_name,'$$username$$', '$$password$$', file_location)
            AVX::PUSH('N/A','None', destination_json,'sftp', irulefile)
            # AVX::LOG("iRule Successfully pushed to device")
        except:
            pass
    if tmos_version and tmos_version not in ('11.4.1','11.4.3'):
        cmd.append('submit cli transaction')
        cmd2.append('submit cli transaction')
    if irulecheck == 'Yes':
        # fileName = irulefile['fileName']
        # file_location = '/home/'+username+os.sep
        cmd.append("load sys config file "+file_location+fileName+" merge")
    cmd.extend(['save sys config', 'quit'])
    cmd2.extend(['save sys config', 'quit'])
    for c in cmd:
        AVX::CMD(device_name+':@'+c)
def main():
    '''
    Main method
    '''
            if vendor == 'F5':
        create_f5()


    output_dict['Implementation']=AVX::CONFIG
```

```
    for c in cmd2:
        AVX::CMD(device_name+':@'+c)

    output_dict['Rollback']=AVX::CONFIG

    AVX::OUTPUT(output_dict)

if __name__ == '__main__':
    '''
    Main function
    '''
main()
```

AVX::PUSH and AVX::JSONCONVERTER

These commands are used to push file(s) between destinations - local to remote, remote to remote.

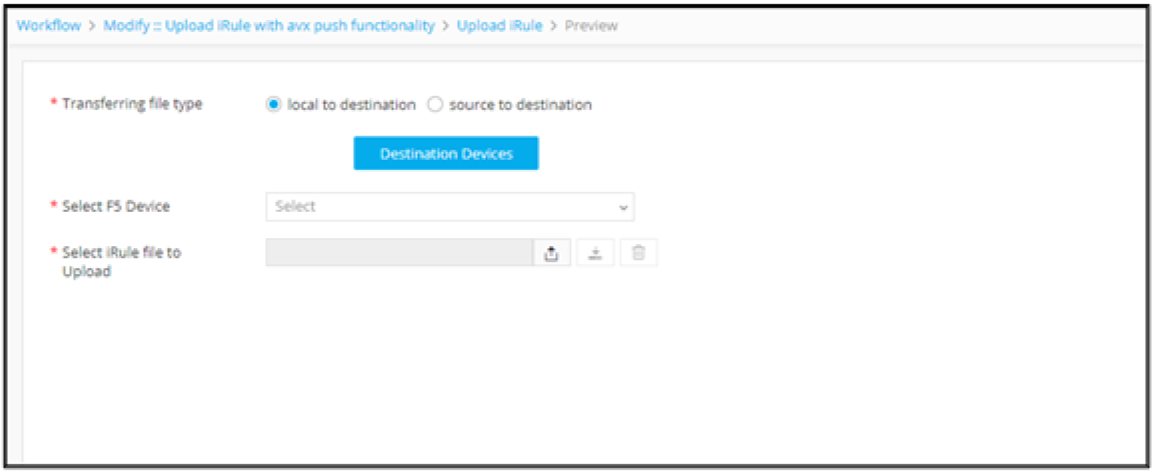
Usage:

```
AVX::PUSH ('device_name',source_json, destination_json,protocol, fieldId , port_number, recursive, seq_no=0, properties={'sleep':0})
```

- Device_name: Name of the device from where the scp/sftp command to be executed.
- Source_json: AVX::JSONCONVERTER('ipAddress', 'username', 'password', 'location').
- Destination_json: AVX::JSONCONVERTER('ipAddress', 'username', 'password', 'location').
- Protocol: “scp” or “sftp”.
- fieldId: Field Id of the uploaded file in the form.
- Port_number: SCP port (Default is 22).
- Recursive: “True” for folders.
- Seq_no and other properties remains the same as AVX::CMD.

To use these commands:

1. Design a workflow.
2. To allow users to select the source of the destination device and upload file, drag and drop a **Form** task.
3. Configure the form fields as shown.



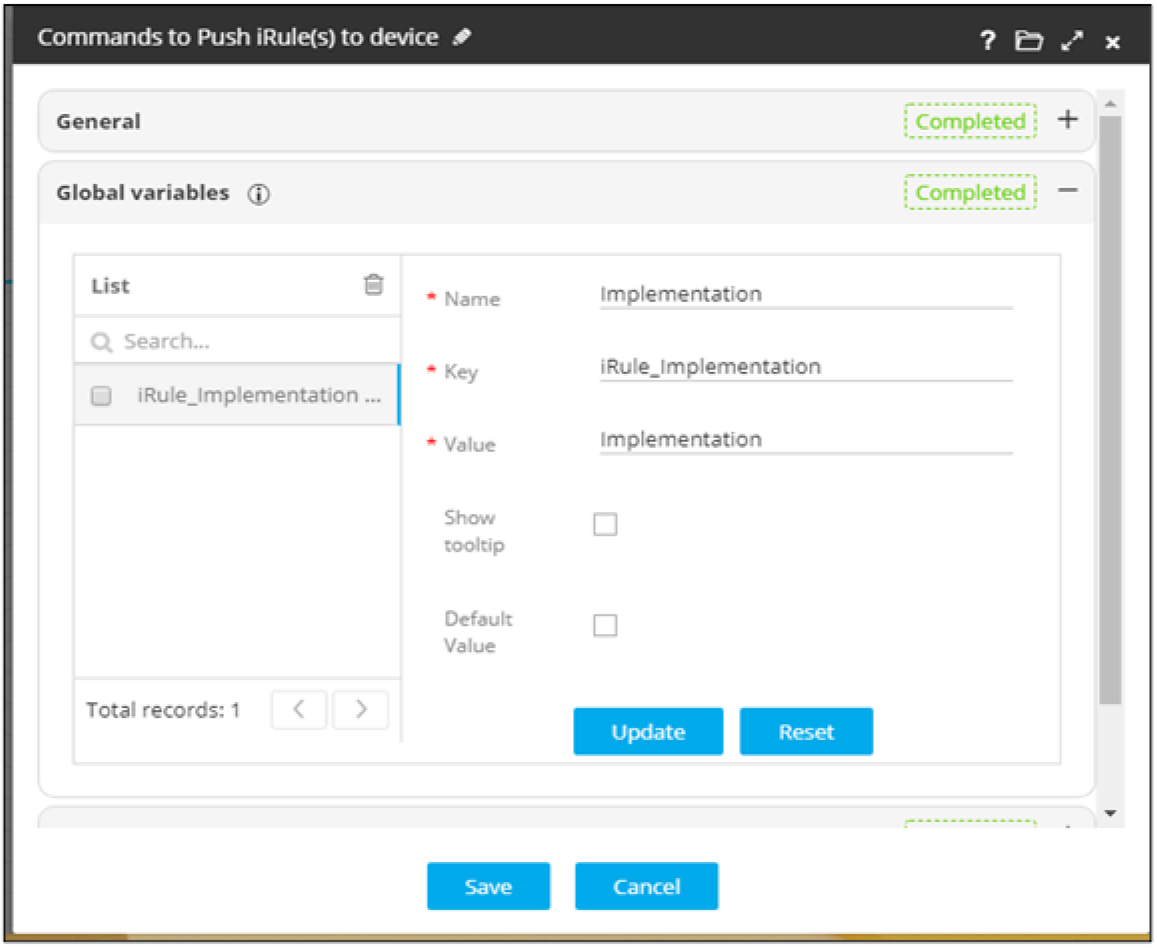
4. To get inputs from the form and push file(s) to a destination (AVX::PUSH logic), drag and drop a **Script** task and define the script as shown below.

```
#load sys config file <irulepath> verify merge


import sys
import os
sys.path.insert(0,AVX::HELPER)
sys.path.insert(0,AVX::DEPENDENCIES)
import paramiko
import appviewx
import Decrypt_Python3 as Decrypt
connection = appviewx.db_connection()
db = connection.appviewx
device_name = "<%device%>"
deviceInfo = db.device
transfer_type = '<%transfer_type%>'


AVX::LOG(transfer_type)
if transfer_type == "local to destination":
    try:
        #device_ip,username,password = get_device_details(device_name)
        irulefile = '<%uploadedfile%>'
        fileName = irulefile['fileName']
        file_location = '<%destination_path%>'
        destination_json = AVX::JSONCONVERTER(device_name,'$$username$$', '$$password$$', file_location)
        AVX::PUSH('N/A','None', destination_json,'sftp', irulefile)
```

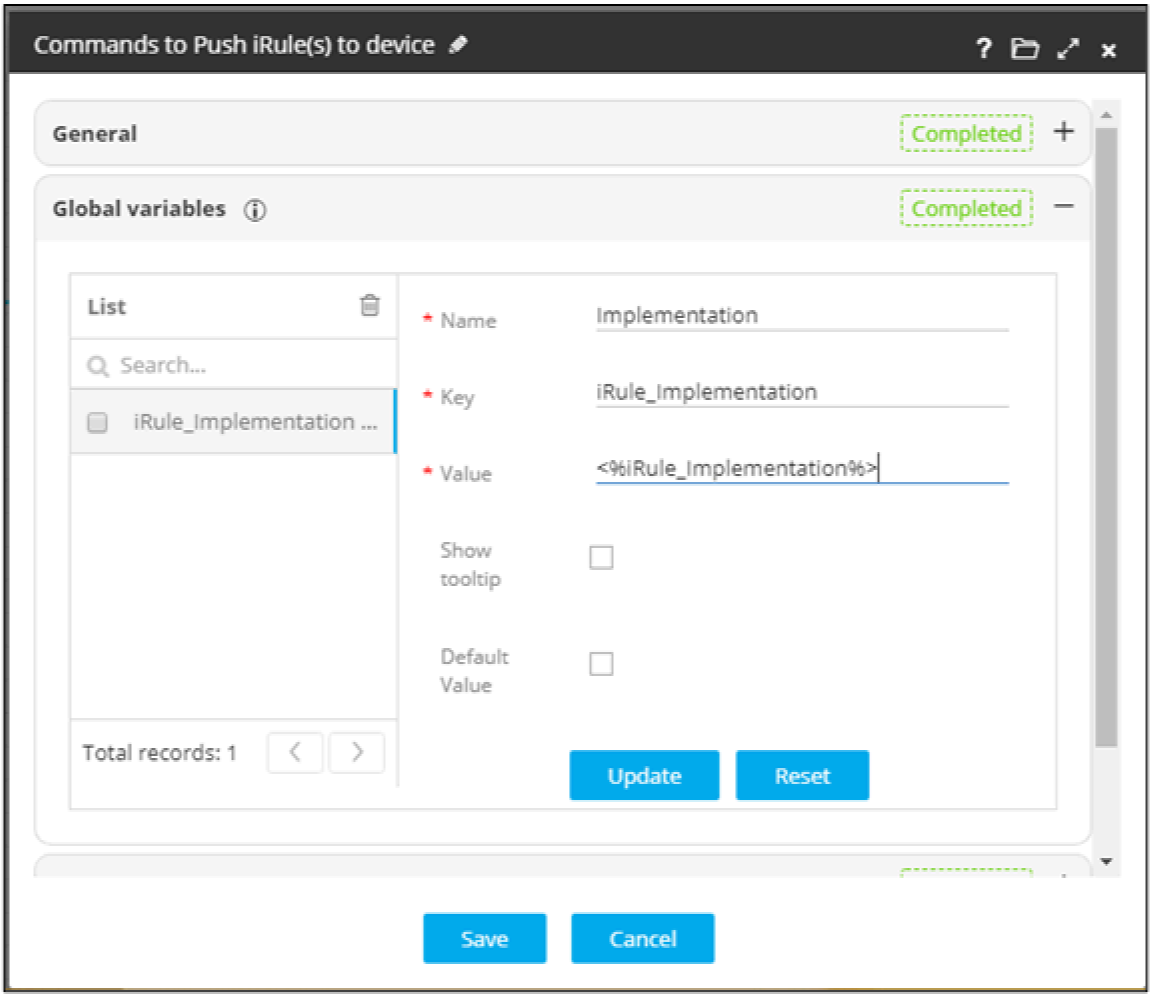


```
        AVX::LOG("iRule Successfully pushed to device")

        AVX::CMD(device_name+":@tmsh")

        AVX::CMD(device_name+":@load sys config file "+file_location+fileName+" merge")

        AVX::CMD(device_name+":@quit")

    except Exception as e:

        AVX::LOG(str(e))

else:

    try:

        #AVX::PUSH('device_name',source_json, destination_json,protocol, fieldId , port_number, recursive, seq_no=0, properties={'sleep':0})

        source_path = '<%source_path%>'

        source_device = "<%source_device%>"

        source_json = AVX::JSONCONVERTER(source_device,'$$username$$', '$$password$$', source_path)

        file_location = '<%destination_path%>'

        file_name = source_path.split('/')[-1]

        destination_json = AVX::JSONCONVERTER(device_name,'$$username$$', '$$password$$', file_location)

        AVX::PUSH(source_device,source_json, destination_json,'scp')

        AVX::LOG("iRule Successfully pushed to device")

        AVX::CMD(device_name+":@tmsh")

        AVX::CMD(device_name+":@load sys config file "+file_location+source_path.split('/')[-1]+" merge")

        AVX::CMD(device_name+":@quit")

    except Exception as e:

        AVX::LOG(str(e))


config = {}

config["Implementation"] = AVX::CONFIG


AVX::OUTPUT(config)
```

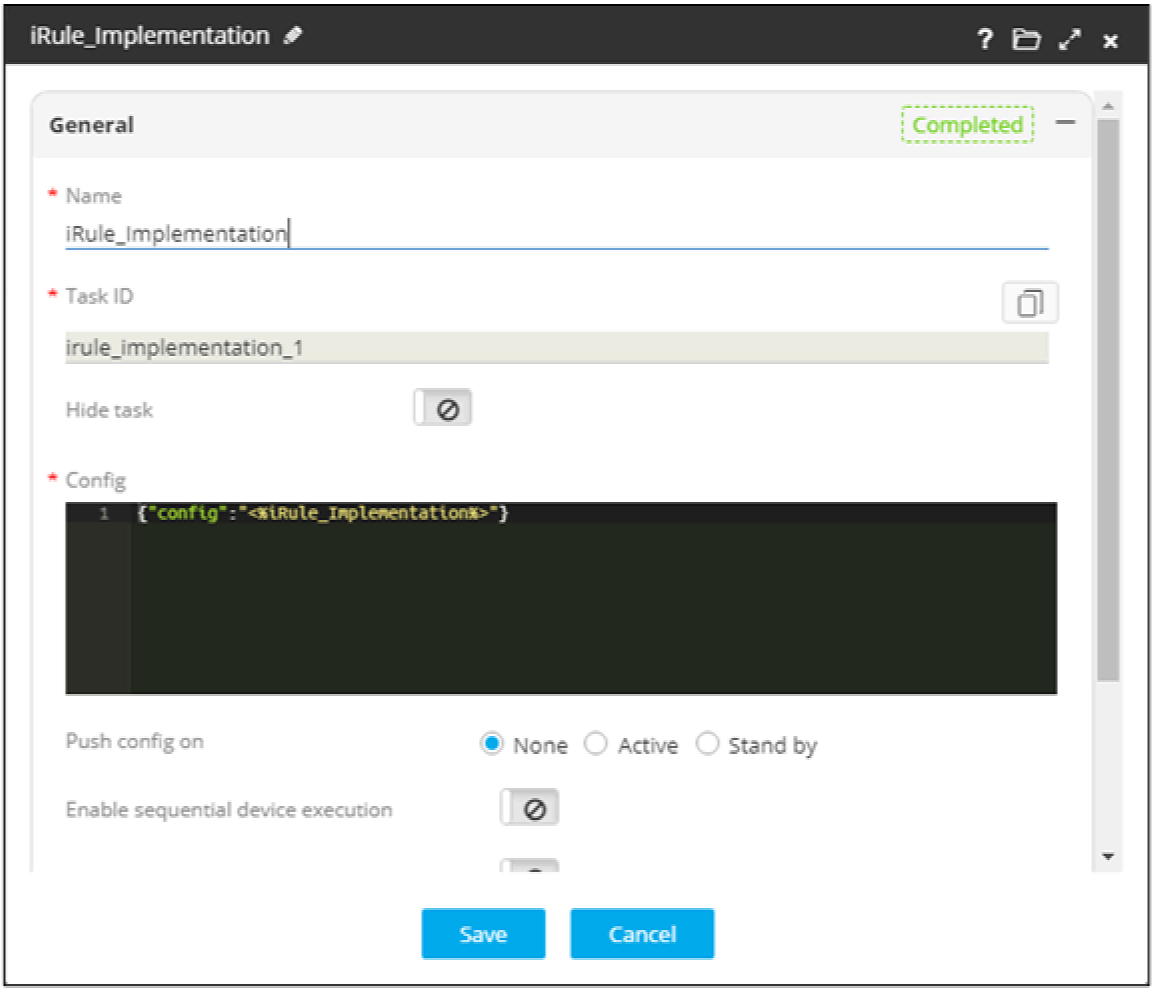
5. Define the configuration to be generated from the script as a global variable value.



6. To display the configurations to be pushed to a device, drag and drop a **Review** task.

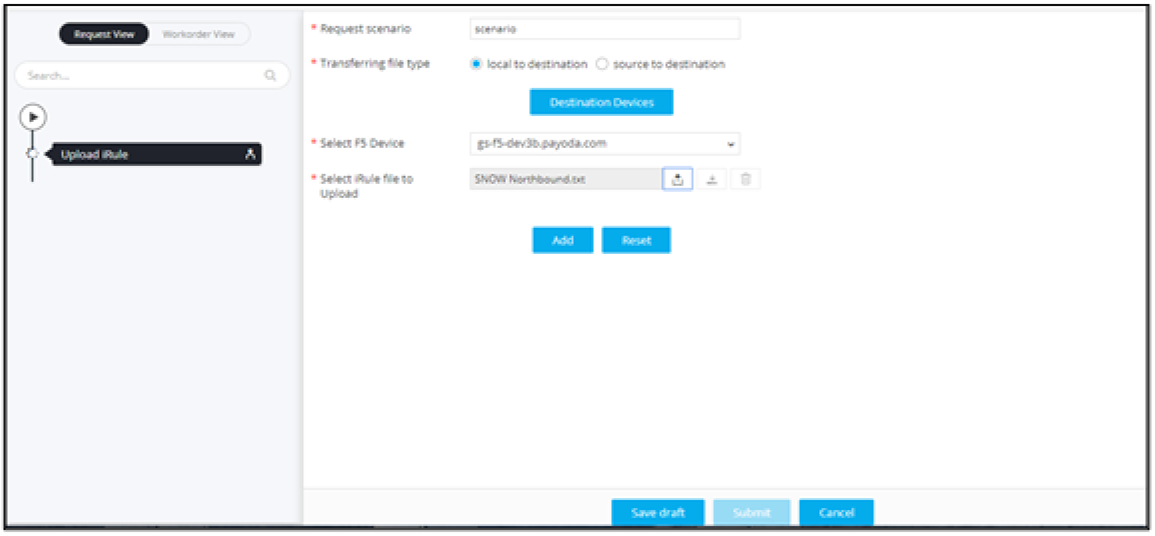


7. To push the configuration on the end device, drag and drop the **Implementation** task.

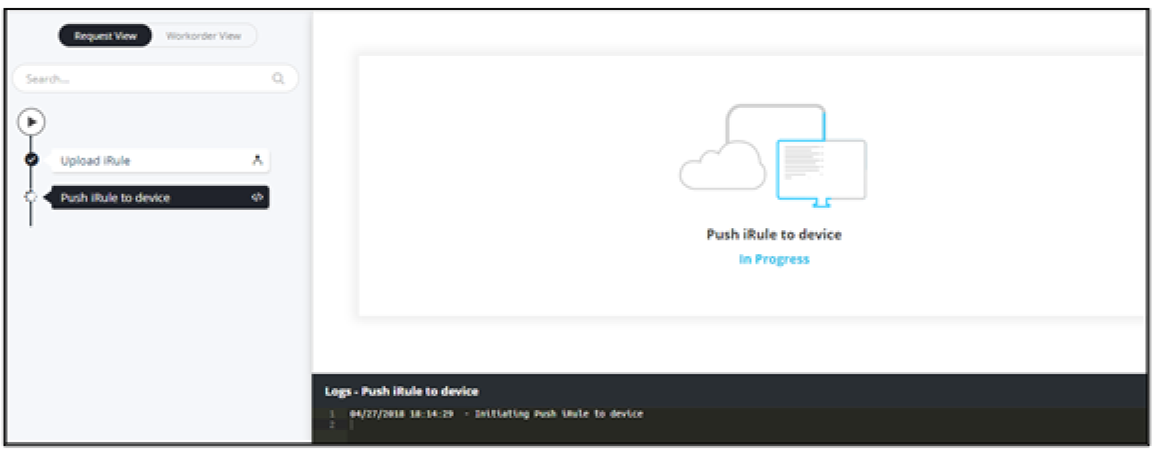


8. Refer the global variable from the Script task.
9. Enable and trigger the workflow.

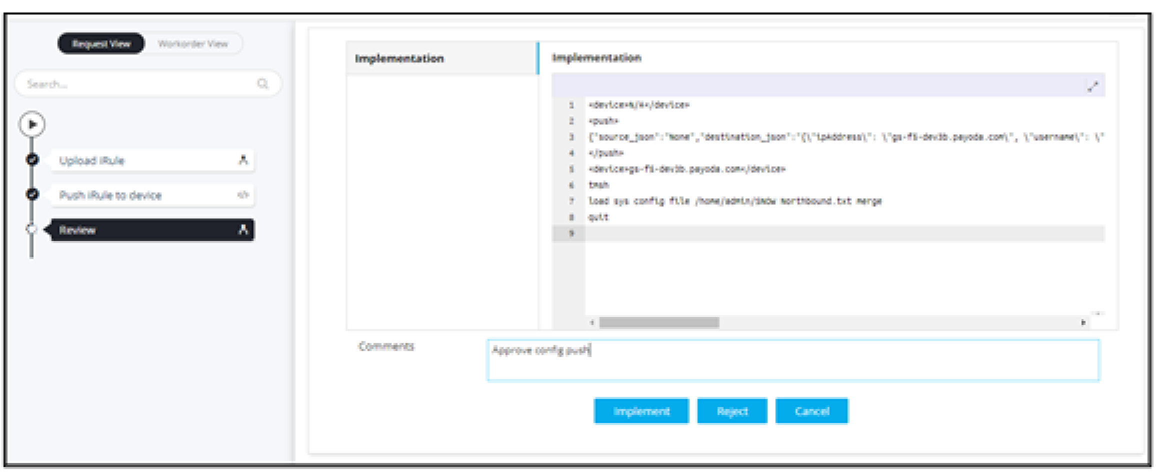
   The workflow is executed with the user inputs requested at the first stage.
10. Select the destination device and upload the file.



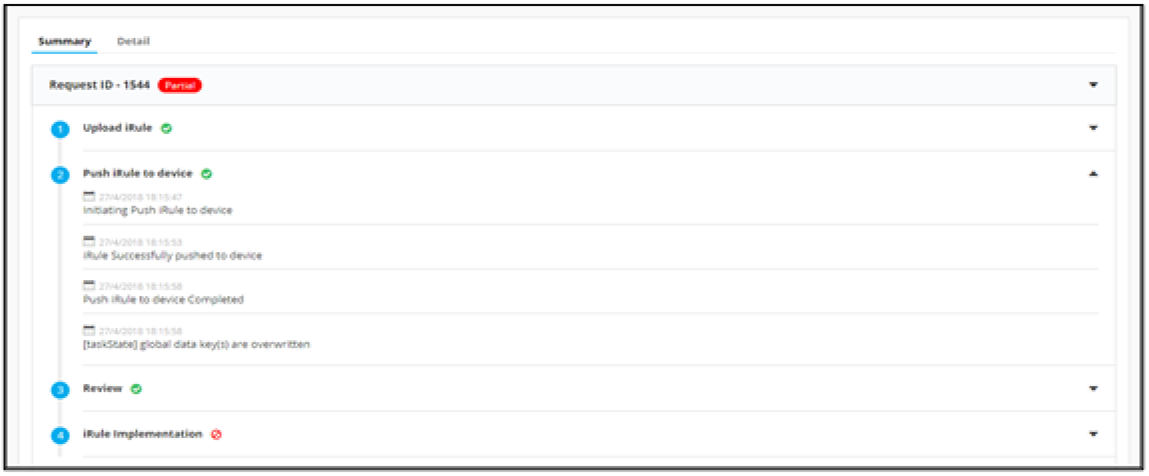
• Push iRule to device task is in progress.



• Implementation stage.



- Summary of workflow execution after file push.



AVX::LOG

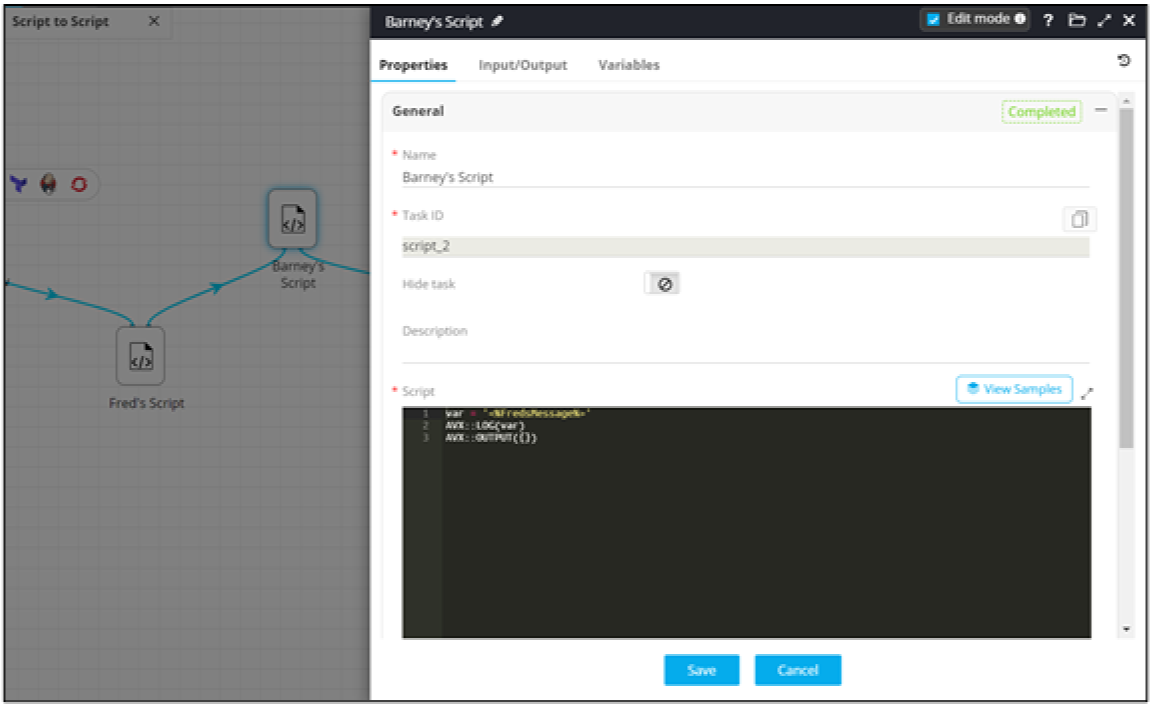
This command allows you to record and print custom messages in the logs (in the request stage view). AVX::LOG takes a string value (i.e. the message to be logged) as an argument and prints it in the log. AVX::OUTPUT collates all the logs and finally prints them.

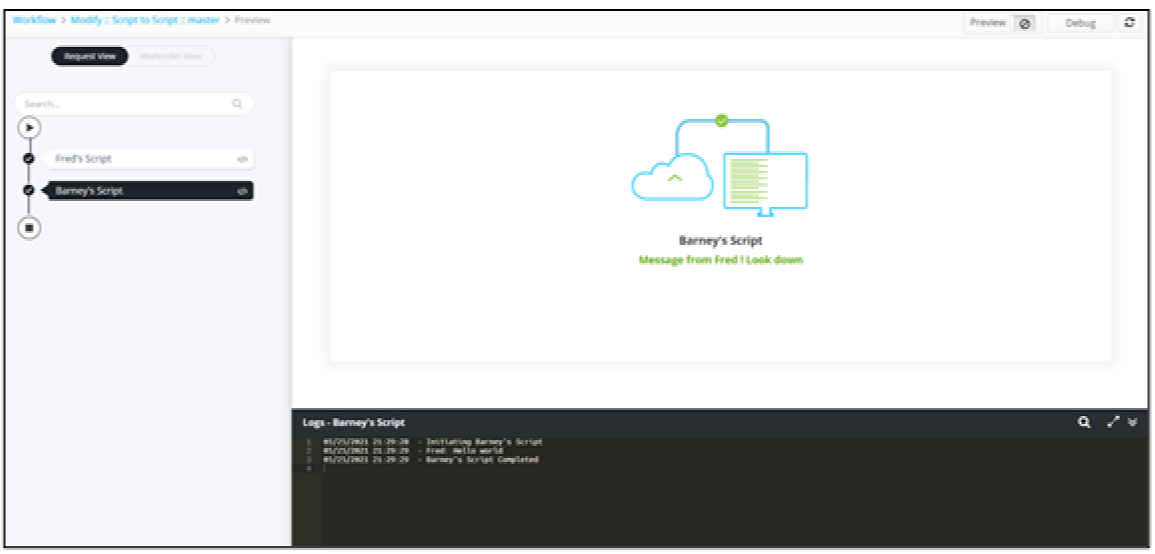
Usage:

```
AVX::LOG(“Print a simple log message”)

AVX::OUTPUT({})
```

Example 1:

```
message_from_fred = '<%FredsMessage%>'

AVX::LOG("Fred's Message is "+message_from_fred)

AVX::OUTPUT({})
```





Example 2:

```
AVX::LOG("Creation of Virtual Machine Completed Successfully")

cpu_mem = int(stdout.read())

 outjson={"cpu_mem":cpu_mem , "ci" : "F5_192_41_101","description":"CPU utilization exceeded 40 % | Current cpu utilization is {} %".format(cpu_mem) }

AVX::LOG("Cpu Memory : "+str(cpu_mem))
```

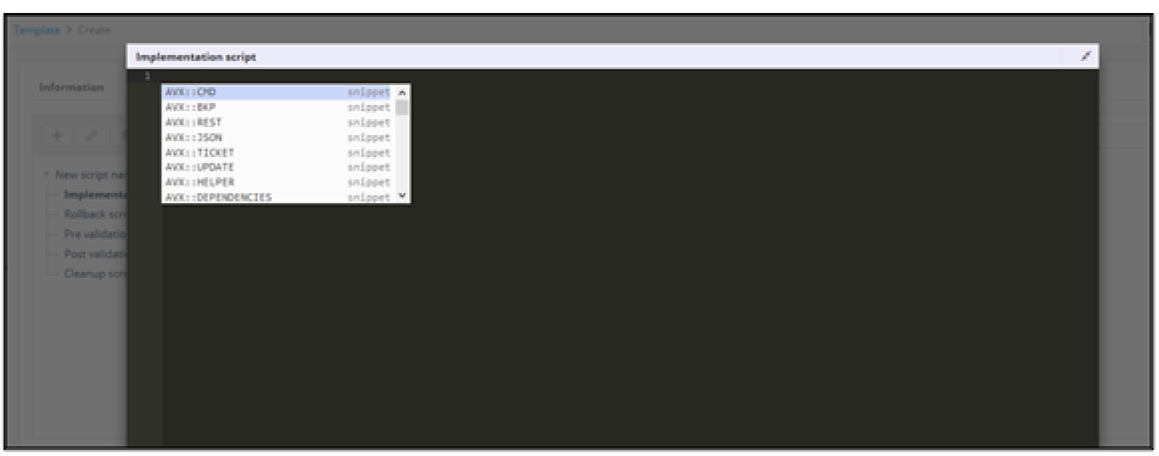

AVX::CMD

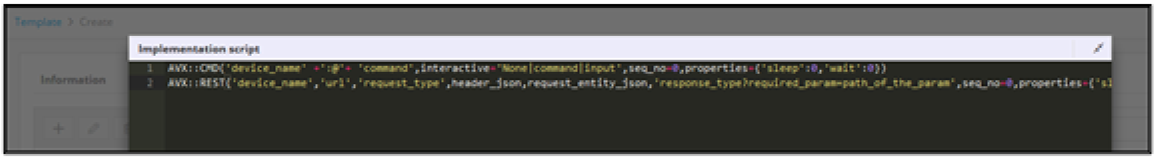
```
AVX::CMD('device_name' +':@'+ 'command',interactive='None|command|input',seq_no=0,properties={'sleep':0,'wait':0})
```

String	Description
device name	Name of the device on which the command needs to be executed.

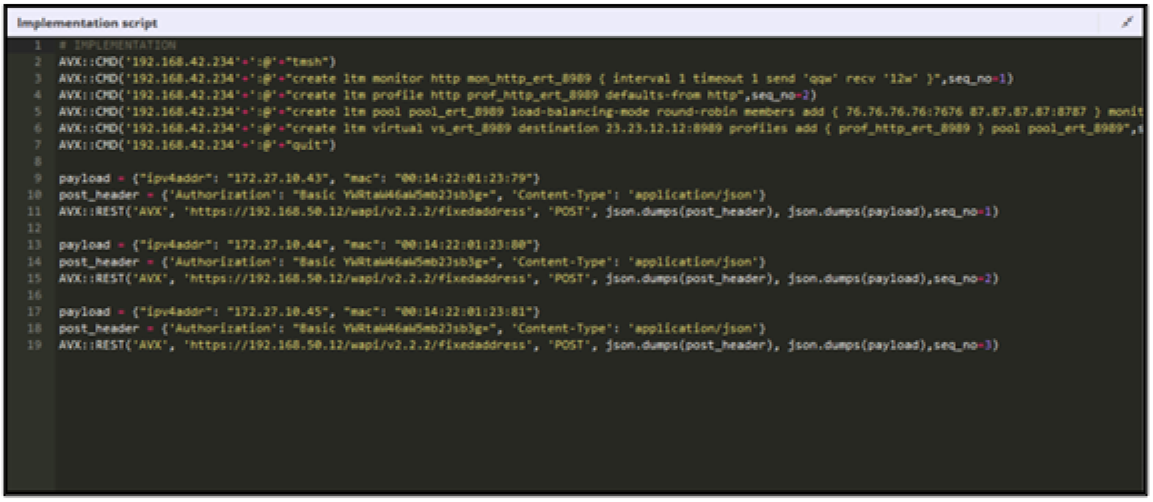
String	Description
command	Actual command to be executed on the device.
interactive	• Used if the command needs to be executed where devices require interactive prompts (such as A10) • Similar to providing a password for SSH.
Seq_no	• Provision to define sequence numbers/tags against configurations (Implementation, Rollback, Prevalidation, Postvalidation, Cleanup) • Sequence number ensures a one on one mapping of the Implementation versus Rollback configurations • Used as a provision to rollback ONLY specific configuration(s) that get implemented.
Sleep	• Provision to define command specific properties • Specifying ‘sleep duration’ will ensure an explicit wait (for the specified time) after a specific command is executed.
Wait	• Provision to define command specific properties • Wait for the specified time to get the response for the implemented command • Should there be no response, it will timeout.

Syntax to define Sequence number/Tags for SSH and REST:



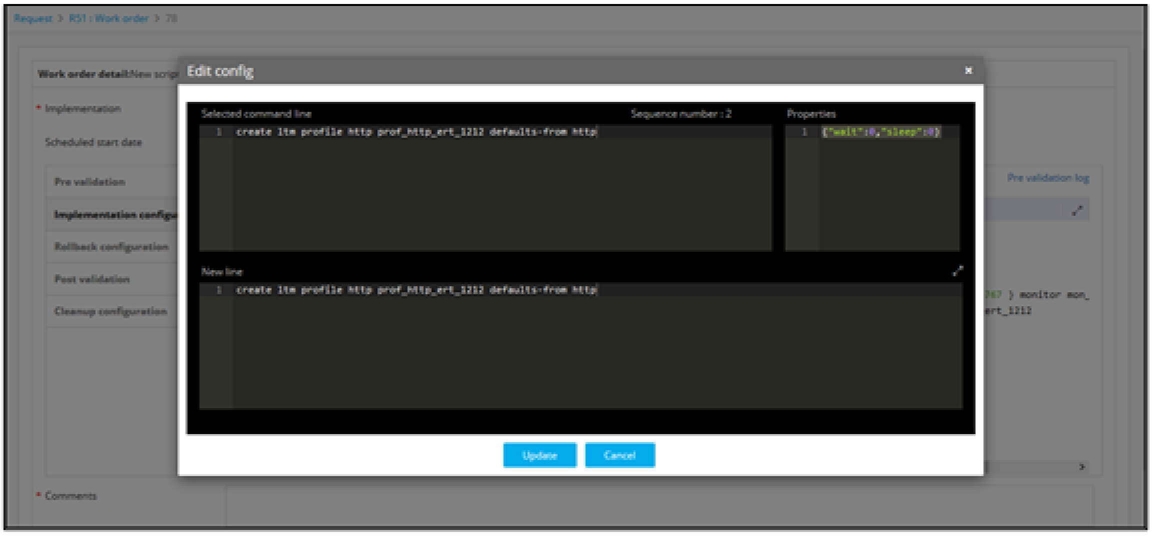


Script with Sequence number/Tags - Implementation Config:

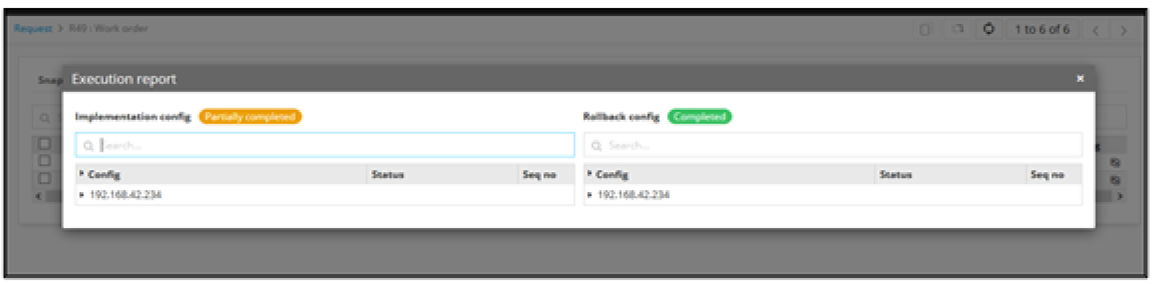


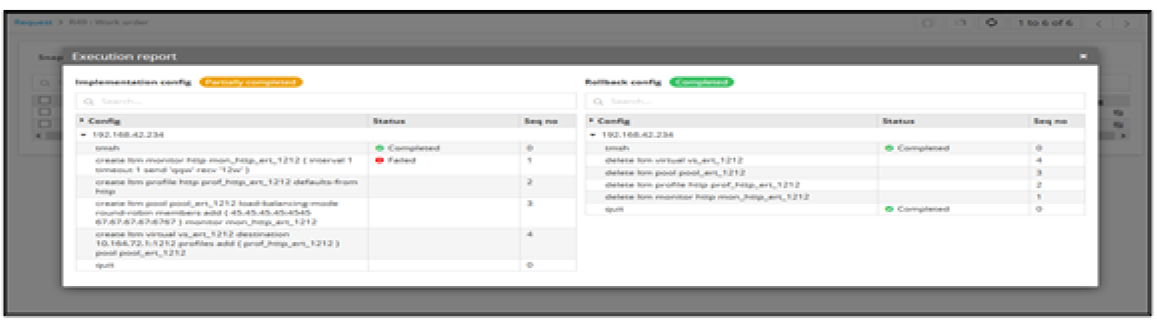
Script with Sequence number/Tags - Rollback Config:

Work order Review pane with option to edit command and properties:

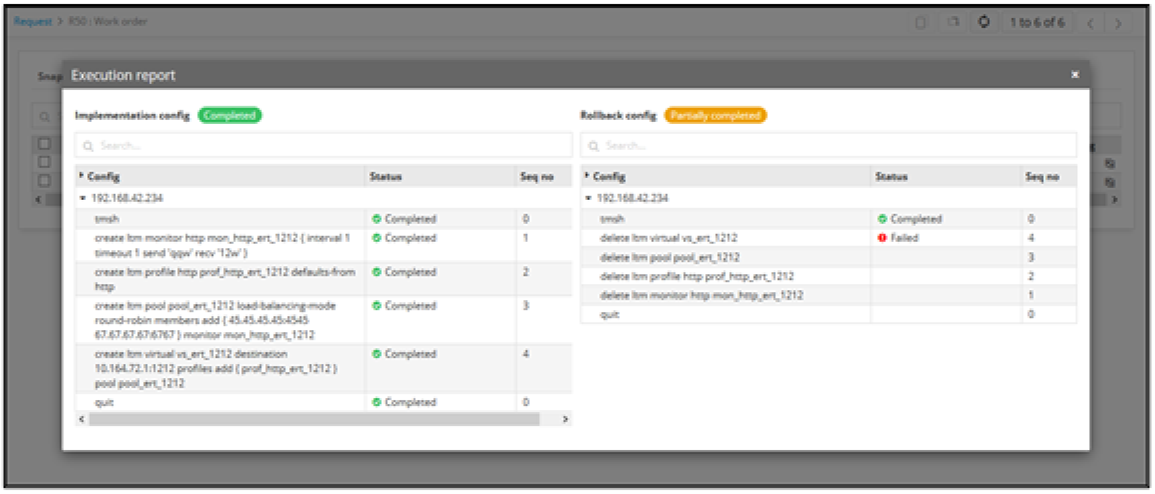


Execution Report - Rollback completion for Partial Implementation:

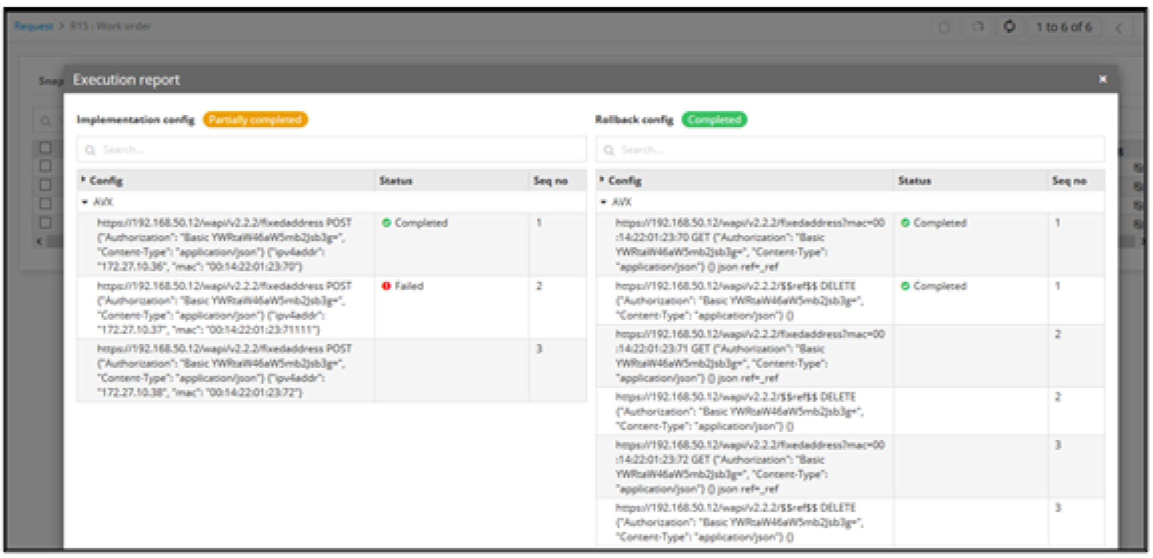




Execution Report for Commands:



Execution Report - REST:



AVX::REST and AVX::JSON

These commands are used via scripts in order to communicate and push files to any REST endpoint.

URL	Endpoint URL to hit the REST
Request_type	“GET”, “PUT”, “POST” or “DELETE”
Header_json	AVX::JSON(json_data,seq_no=0,properties={‘sleep’:0,’wait’:0}) Json_data is the json value to be passed to REST

URL	Endpoint URL to hit the REST
request_entity_json	AVX::JSON(json_data,seq_no=0,properties={‘sleep’:0,’wait’:0})
response_type? required_param	Path_of_the_param is the query and other params which can be provided as string itself

To pass variables when using AVX::REST:

1. Execute REST call.
2. Store the value called “Signature”.

```
auth_url  = "https://{}/axapi/v3/auth".format(deviceip)

    headerjson_auth = {"Content-Type": "application/json"}

    bodyjson_auth = {"credentials": {"username": username ,"password": '@$avx_password@$' }}

    AVX::REST(dev_name,auth_url,'POST',json.dumps(headerjson_auth),json.dumps(bodyjson_auth),'json?sig=authresponse/signature')
```

```
{

  "authresponse" : {

     "signature":"6181d42578cc03dd1088d58884637e",

     "description":"the signature should be set in Authorization header for following request."

   }

}
```

The value is stored in ’json?sig=authresponse/signature’. In the above, json signature lies under authresponse, hence the response_type?required_param will be json?sig=authresponse/signature.

3. The variable signature (sig) is reused in the following REST call’s header.

```
url = 'https://{}/axapi/v3/slb/service-group/{}'.format(deviceip,value)

headerjson = {"Authorization": "A10 $$sig$$", "Content-Type": "application/json"}

bodyjson = {"service-group": { "member-list":  state_pool_dict[value] }}

AVX::REST(dev_name,url,'POST',json.dumps(headerjson),json.dumps(bodyjson))
```

AVX::PASSWORD

This command is used in the event of having to hide any password credentials from being displayed to the user. A key can be used to map the password and the password value can be retrieved with the key.

```
password = 'ninjaturtle@hello'

bodyjson_auth = {"credentials": {"username": username ,"password": '@$avx_password@$' }}
```

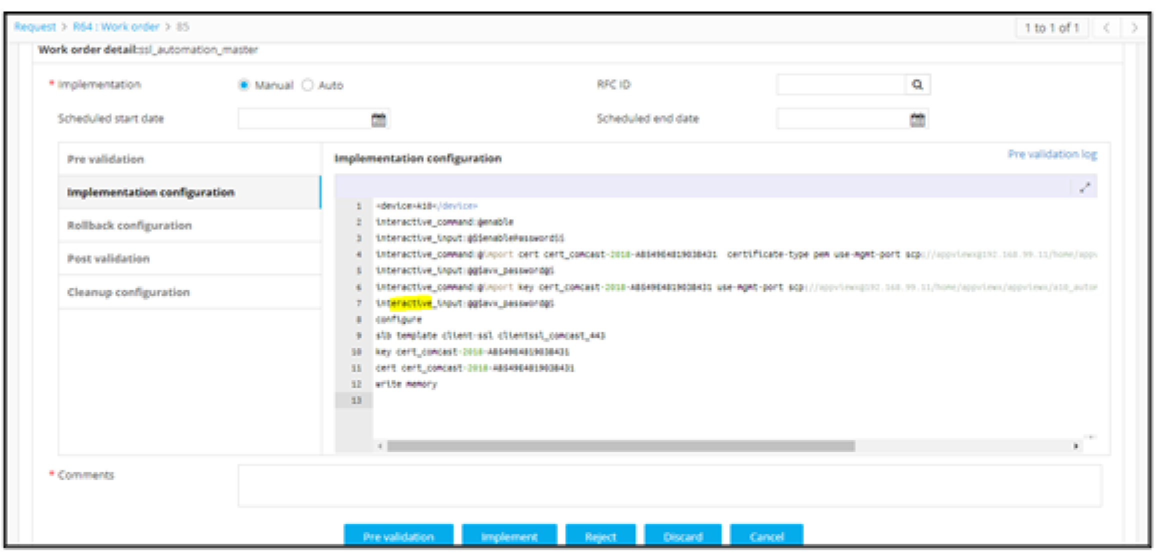
```
AVX::PASSWORD('@$avx_password@$',password)
```

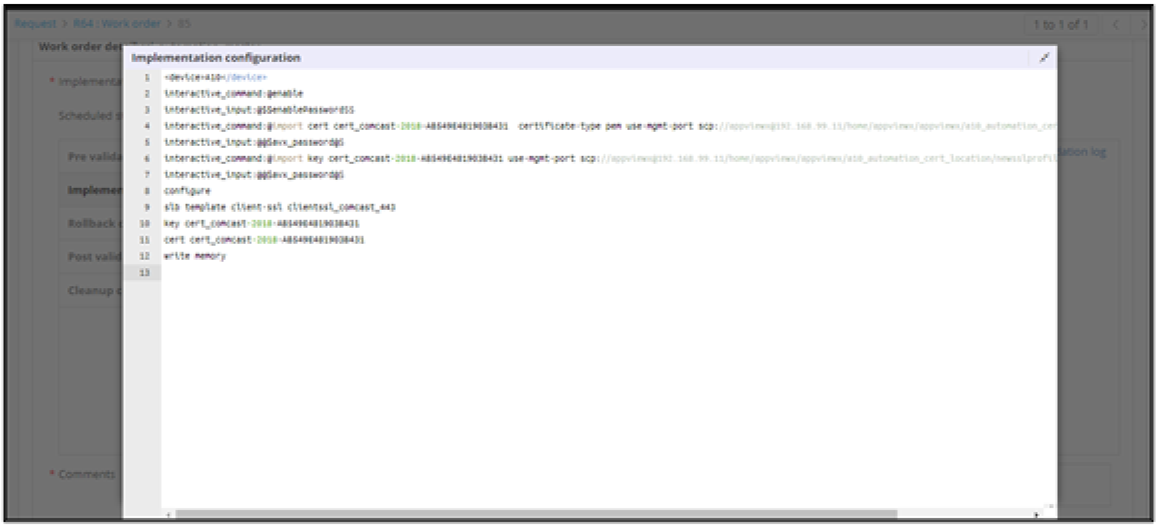
String	Description
password	The variable which holds the password. For example: ninjaturtle@hello
@$avx_password@$	This is to mask the declared password.
AVX::PASSWORD('@$avx_password@$',password)	Mapping of the masked password

AVX::PASSWORD usage with configuration sample:

```
<device>A10</device>

<rest>

<url>https://172.16.30.110/axapi/v3/auth</url>

<type>POST</type>

<header>{"Content-Type": "application/json"}</header>

<request_entity>{"credentials": {"password": "@$avx_password@$", "username": "admin"}}</request_entity>

<response_parse_type>json</response_parse_type>

<response_param>sig=authresponse/signature</response_param>

</rest>
```

Masking of password from users during Review:





AVX::UPDATE

This command is used to update specific parameters within Collections. It is used to filter out the documents from the collection named ‘Collections’.

```
AVX::UPDATE (query_dict,'key','value','action')
```

String	Description
query_dict	The dictionary value to be provided as a querying parameter in the Mongo DB.
key	Key for which the value needs to be updated in the queried document.
value	The value which needs to be modified for the provided key.
action	It can be either: • Add - To add the provided value in the array of values • Remove - To remove the provided value from the array of values.

AVX::FILEDOWNLOAD

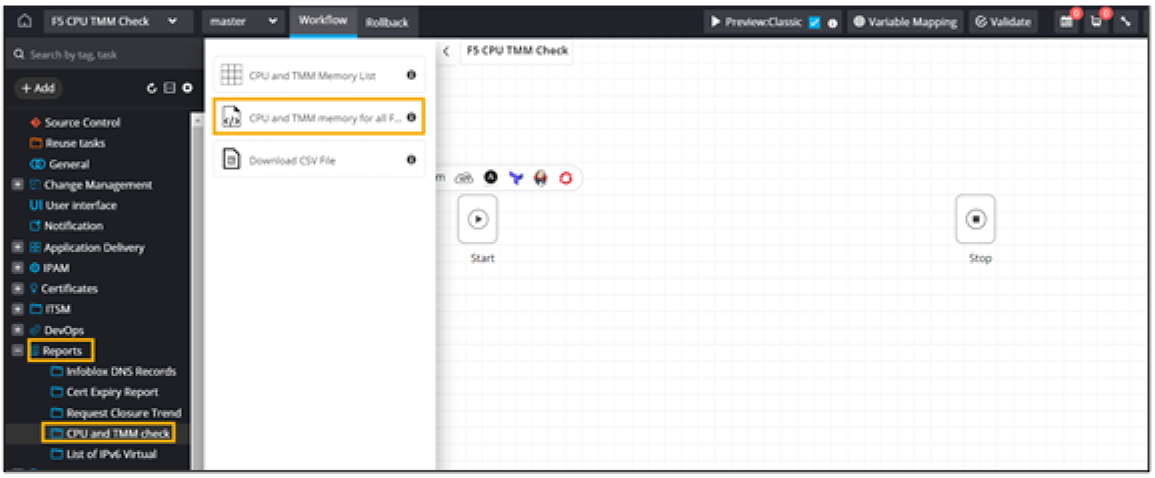
This command is used to generate a .csv file on runtime based on a dictionary of values and download the file output.

Syntax:


```
output = {"AVX::FileDownload" :{"linkData": <dictionary value> ,"fileName":'<excel file name>', "columnHeaders":<list of column headers to be added>}}
```

To use this command:

1. Design a new workflow.
2. In the deisgn space, from the left menu, under **Reports** > **CPU and TMM** folder, drag and drop the **CPU and TMM Memory** task. This script will scan all the F5 devices managed for the CPU and TMM memory.

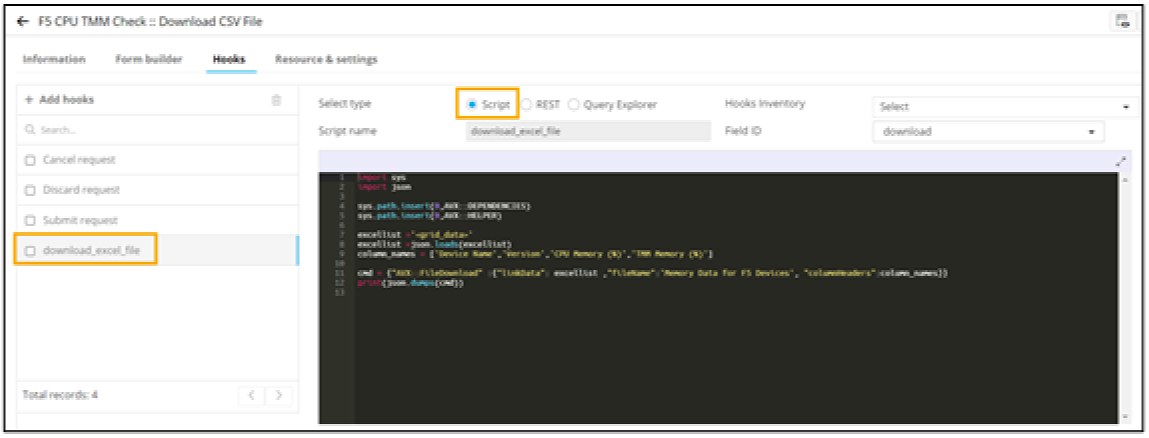


3. From the same folder, drag and drop the **CPU and TMM memory List** task. This task will display the Device name, CPU (%), and TMM (%) in a grid view.

> **Note:** Assign RBAC to users who can access this task.

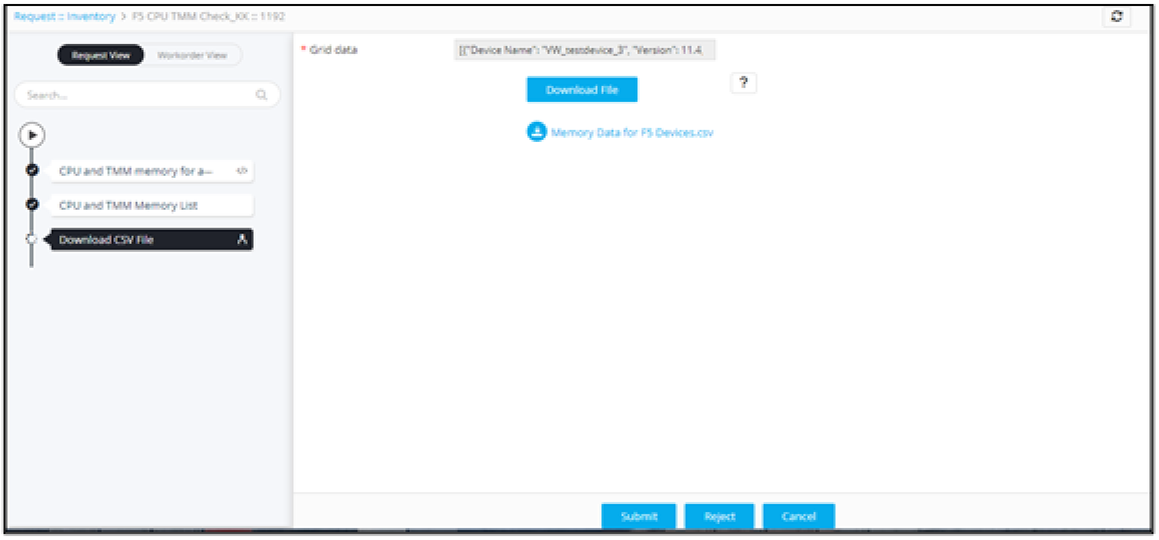
4. From the same folder, drag and drop the **Download CSV file** task. This task contains the device and CPU/TMM details from the Grid task in a .csv file.

5. In the **Download CSV file** task window, define the file download logic to generate a <.csv> file on run time.




```
import sys
import json

sys.path.insert(0,AVX::DEPENDENCIES)
sys.path.insert(0,AVX::HELPER)

excellist ='<grid_data>'
excellist =json.loads(excellist)
column_names = ['Device Name','Version','CPU Memory (%)','TMM Memory (%)']

cmd = {"AVX::FileDownload" :{"linkData": excellist ,"fileName":'Memory Data for F5 Devices', "columnHeaders":column_names}}
print(json.dumps(cmd))
```

6. Connect and enable the workflow.
7. Trigger the workflow.
8. At the Download CSV file stage of workflow execution, click **Download File** to generate the .csv file.



The file is downloaded to your machine.

AVX::OUTPUT

This task is used within the Script task to record the output of the script execution logic.

- AVX::OUTPUT takes the following arguments - dictionary, a string
- Another usage takes an additional ‘integer’ argument which represents the execution state (1-Success, 2-Failed).

Usage:

```
AVX::OUTPUT(config_dict) / AVX::OUTPUT(error_data, 2)
```

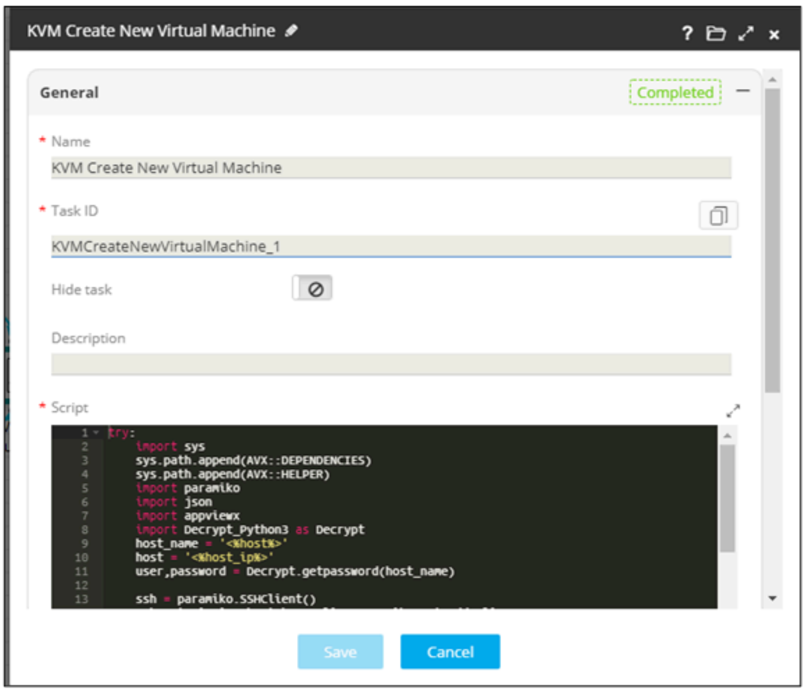
```
mgmt_interface=mgmt_interface,add_interface=add_interface)


  #print(cmd)
  AVX::LOG(cmd)
  stdin_vm, stdout_vm, stderr_vm = ssh.exec_command(cmd)
  outJson = {"mgmt_interface":mgmt_interface,'add_interface':add_interface}
  if (stdout_vm.channel.recv_exit_status() == 0):
    AVX::LOG("Creation of Virtual Machine Completed Successfully")
    AVX::OUTPUT(outJson)


  else:
```

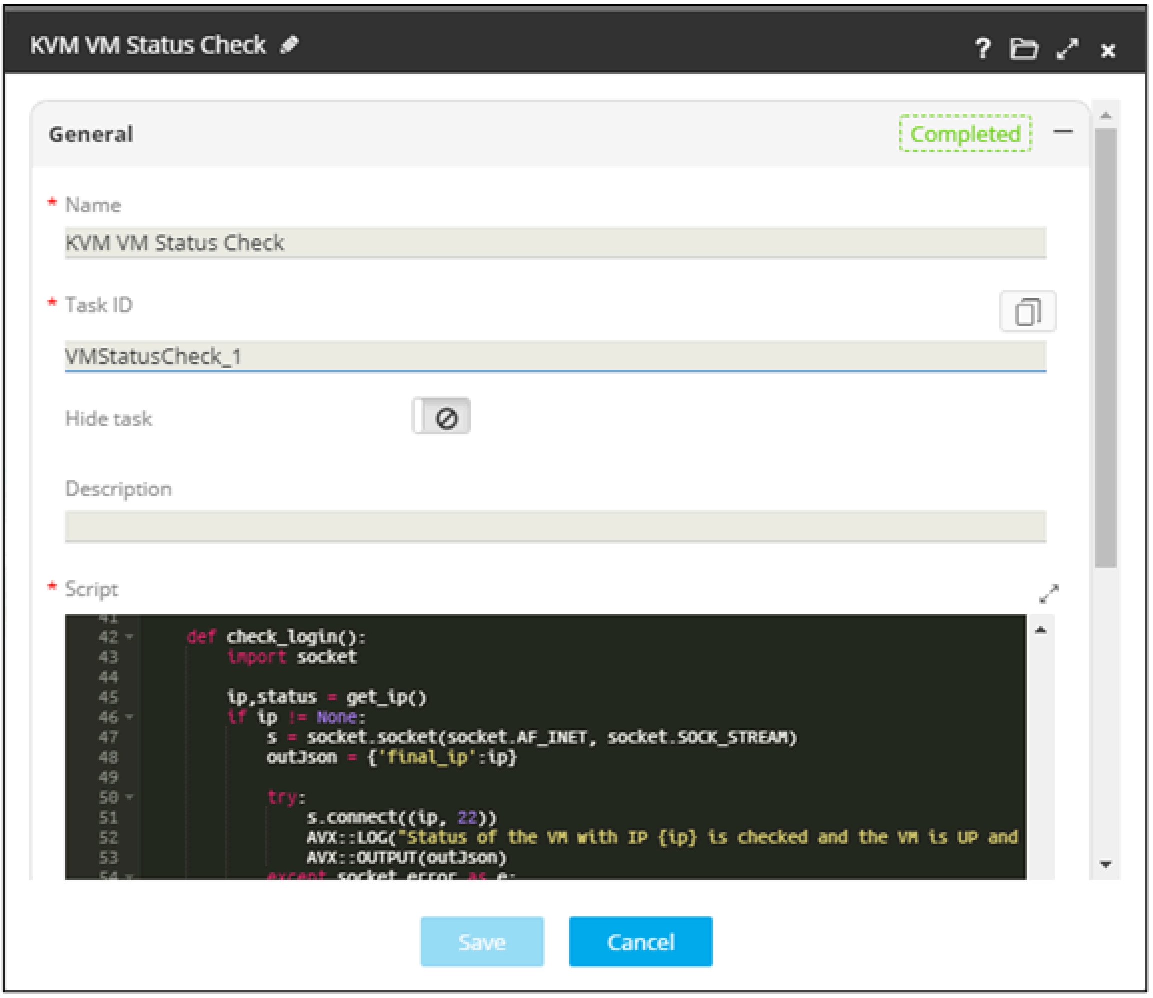


```
        AVX::LOG("Disk and VM Creation Failed"+str(stderr_vm.read().decode()))

        AVX::OUTPUT(outJson,2)

    ssh.close()
```

```
if __name__ == '__main__':

    ip,cmd=execute_command()

    outJson = {'ip':ip,'prov_modules':prov_modules,'commands':cmd,'host':host_name,'timezone':time_zone,'vlan':vlan_name}

    AVX::LOG("Device has been provisioned with the information given as the input")

    AVX::OUTPUT(outJson)
```

AVX::OUTPUT usage with failover options:





Script with 'try-except':

```
def check_login():
    import socket
    s = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
    outJson={'Success':'Success'}
    try:
        s.connect((ip, 22))
        #print "Port 22 reachable"
        #print json.dumps({"status":"success", "conditions":{ "login_success": "login_success"}})
        # dataJson = {}
        # dataJson["data"] = {"outputData":"Success"}
        # dataJson["status"] = "success"
        # dataJson['state'] = 1
        # dataJson["logs"] = [{"message": "Status of the VM with IP {ip} is checked and the VM is UP and RUNNING".format(ip=ip)}]
```



```
    # print(json.dumps(dataJson))
    AVX::LOG("Status of the VM with IP {ip} is checked and the VM is UP and RUNNING".format(ip=ip))
    AVX::OUTPUT(outJson)
  except socket.error as e:
    #print json.dumps({"status":"failure", "conditions":{ "failure": "failure"}})
    # dataJson = {}
    # dataJson["data"] = {"outputData":"Failure"}
    # dataJson["status"] = "failure"
    # dataJson['state'] = 2
    # dataJson["logs"] = [{"message": "Status of the VM with IP {ip} is checked and the VM is still not UP".format(ip=ip)}]
    # print(json.dumps(dataJson))
    AVX::LOG( "Status of the VM with IP {ip} is checked and the VM is still not UP".format(ip=ip))
    AVX::OUTPUT(outJson,2)
  s.close()
```

Note: In case of using ‘try-except’ in the python script, you must include ‘AVX::OUTPUT with state 2’ to record script failure actions. Try-except is used to handle errors and exceptions.

AVX::MERGECONFIG

This command is used to merge multiple configurations in a workflow to be generated as a single entity. The parameters must have the config object generated using the AVX custom commands only. It is useful in cases where a few commands have to be added at runtime prior to implementation depending on the outcome of certain prevalidation commands.

- Initial Implementation config generated for device version v11 scheduled for 10th March.
- Run pre-validation on 9th March to check for device versions.
- If there is a change in device version, generate and add additional commands, and merge Implementation configuration into one.

Usage:

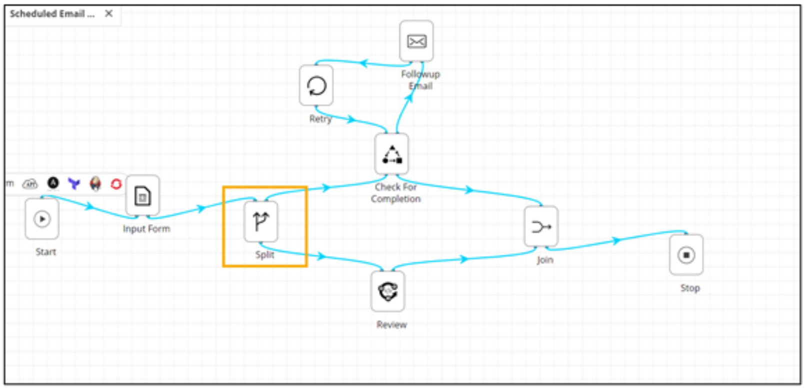
```
Conf1 = <%Script_1.output %>
Conf2 = <%Script_3.output %>
Conf3 = <%Script_4.output %>
AVX::OUTPUT (AVS::MERGECONFIG(config, [conf1, conf2, conf3]))
```


OR

```
Merged_config = AVX::MERGECONFIG(config,[conf1, conf2, conf3]))
```

Split

Split task is used when parallel flows have to be spawned as part of a parent workflow.

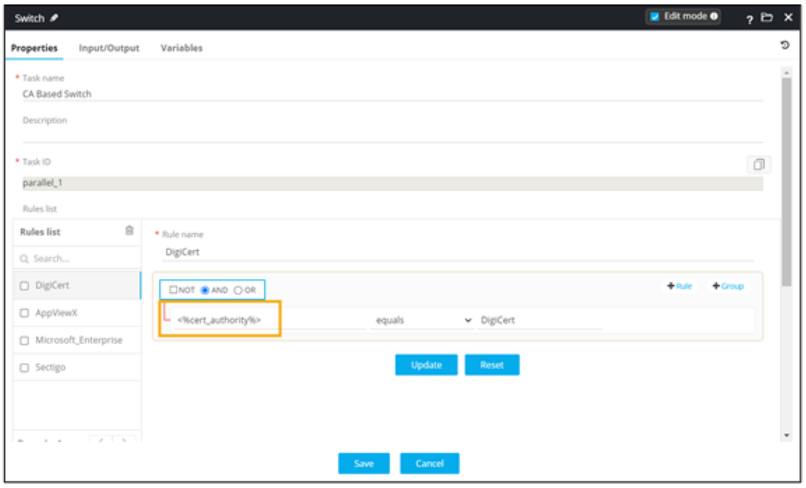


Note: The Join task must be used as a common connector when a Split task is used.

Switch

The Switch task allows users to configure multiple conditional rules in order to take decisions as part of the workflow automation process.

- Provision to define rules with multiple decisions based on which workflow can be routed.
- Provision to reference variables from a previous task.
- Provision to define multi-boolean conditional rule logic as a truth table.

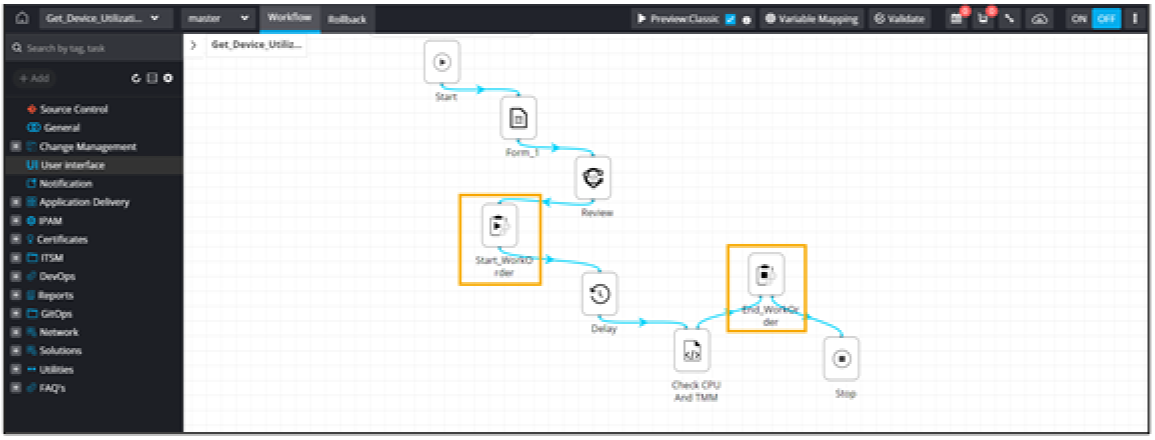


WorkOrder

This task allows you to generate a work order (aka Child request) under a main workflow request.

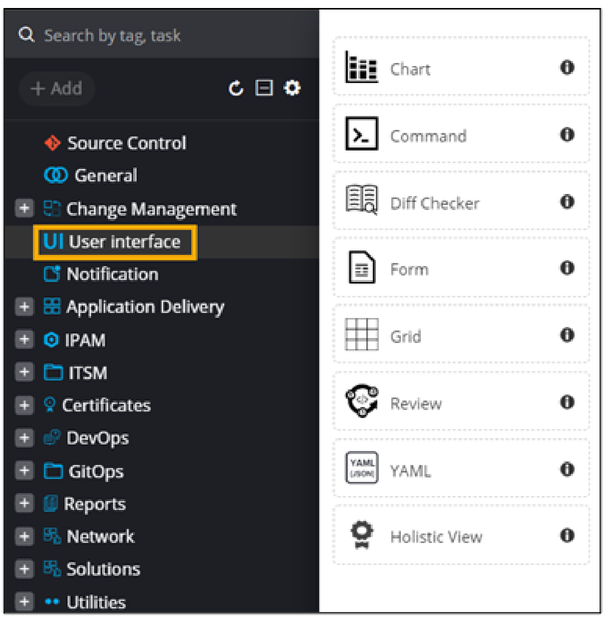
- Work order is an optional task
- Work order is akin to a mini workflow under a parent workflow
- Work order can be used to mimic the existing APS (Application Provisioning System) capability where different configurations can be tracked as part of a work order under a single workflow request (For example, Create VIP, Create DNS, and New CSR)
- On workflow execution, the work order task renders the work order review page for the user
- Each work order triggers a unique work order ID that can be used for audit
- A work order can have a specific Rollback option defined and associated.

For example, you are executing two implementations, one for a firewall device and the other for a certificate. Now if we just want to rollback the firewall because of some issue in its implementation, then a work order can be used in this situation.



Task Category - User Interface

This category comprises tasks that require a certain amount of user intervention.



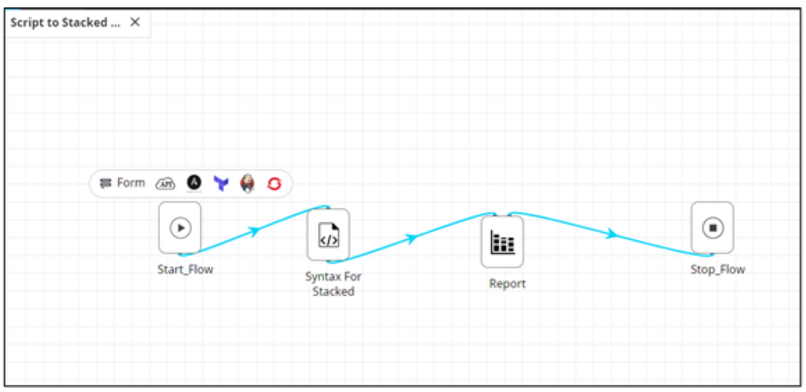
- Chart
- Command
- Diff Checker
- Form
- Grid
- Review

- YAML
- Holistic View

Chart

Chart Task allows for designing and visualizing data in the form of pie charts, bar charts or stacked bar charts.

- Provision to represent data using pie, bar and stacked charts
- Provision to define charts manually or by passing data from a previous source (such as script, form) using global variables
- Provision to define custom data set for the chart - sequence number, labels and values
- Provision to dynamically configure the chart elements by passing the global variable defined in the script in the “source” field
- Provision to assign user role (RBAC) access to the chart task in the request stage
- Provision to preview chart types with sample data.

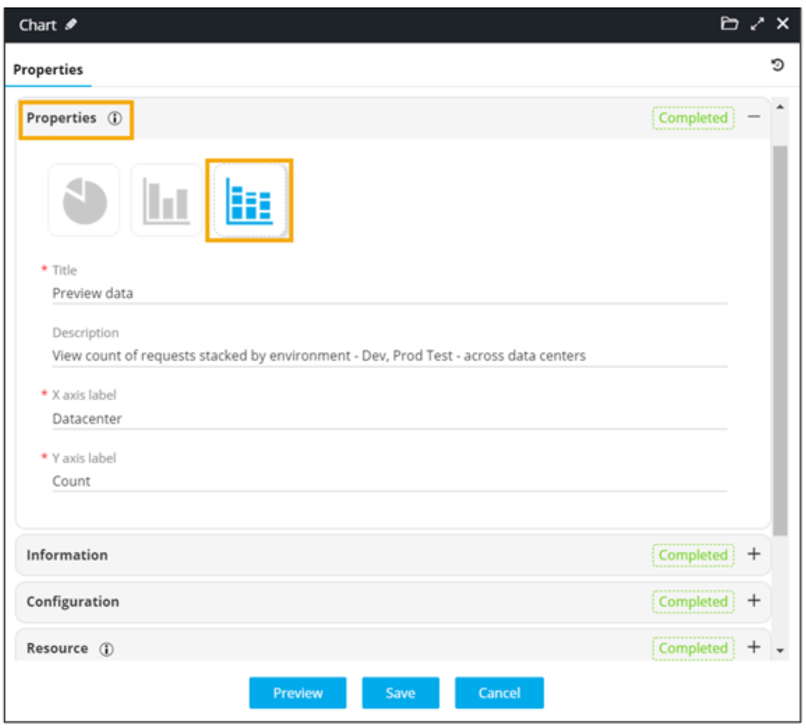


- Designing a Chart with Static Data
- Designing a Chart with Dynamic Values

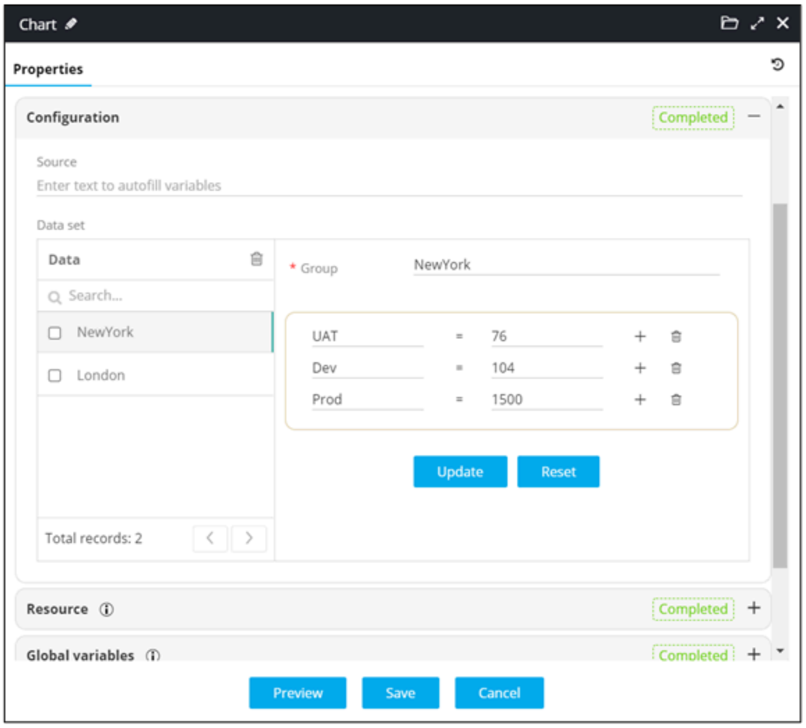
Designing a Chart with Static Data

To design a stacked chart using static data:

1. Design a workflow.
2. From the User Interface section, drag and drop the **Chart** task.
3. In the **Chart** task window, under **Properties**, select the chart type as stacked chart  .
4. In the **Chart** task window, under **Properties**, enter values for chart **Title** and X-axis, Y-axis labels.



5. In the **Chart** task window, under **Configuration**, define the static data to stack values in the chart.

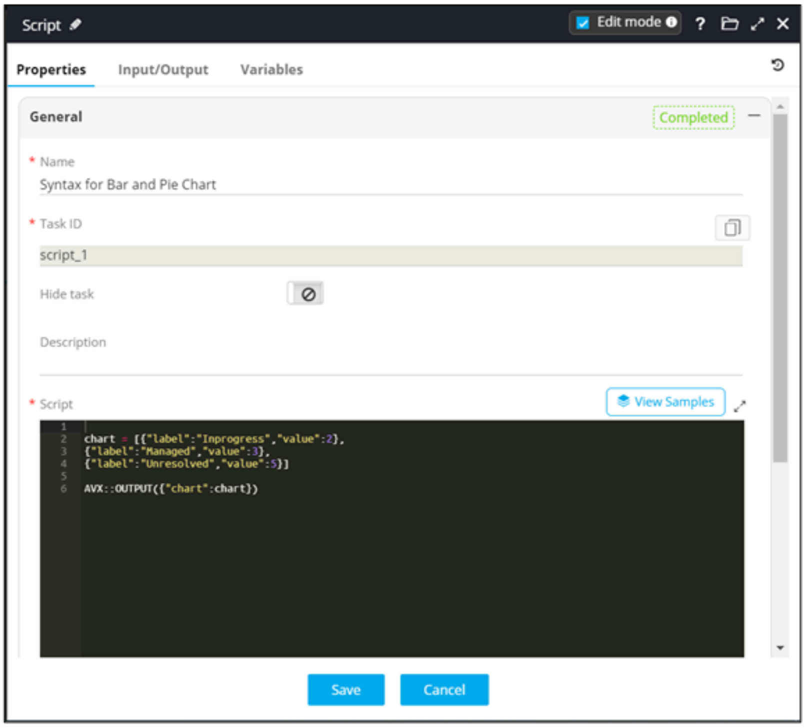


6. Click **Save**.
7. Connect the workflow tasks.
8. From the command bar on the top, click **Preview**.

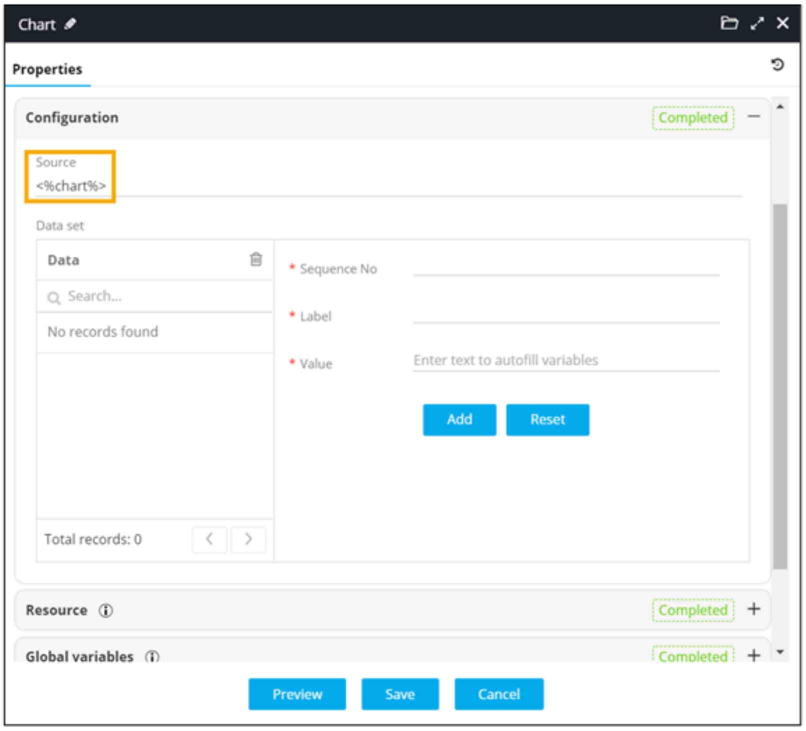
   Stacked chart with static data is displayed.

Designing a Chart with Dynamic Values

1. Design a workflow.
2. From the **General** section, drag and drop a **Script** task.
3. In the **Script** task window, under **Properties**, define a **Script** to get a list of values for the chart.

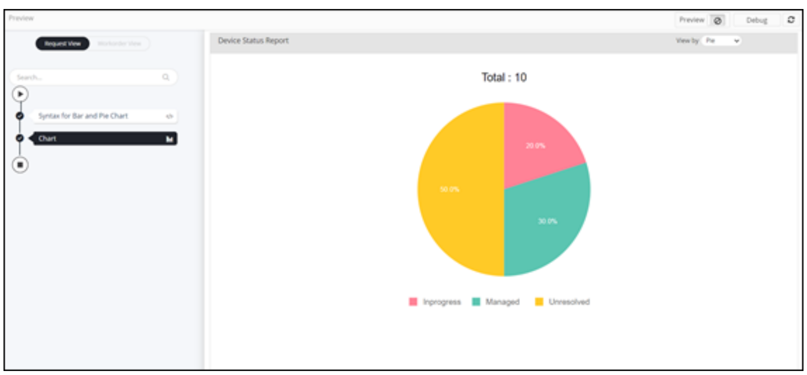


4. From the **User Interface** section, drag and drop a **Chart** task.
5. In the **Chart** task window, under **Properties**, select the chart type as Pie  and enter or select the field information.
6. Enter the chart **Title** (mandatory field) and **Description** and select the **Legend position**.
7. Under **Configuration**, refer the global variable to render data.



8. [Connect](#) the workflow tasks and from the menu on the top, click **Preview**.

   Chart with dynamic values is displayed.



> **Note:** You can view this chart as a **Donut** chart by selecting the option from the **View by** dropdown.

Command

The Command task allows you to execute command configurations. You can either define a YAML structure or use prebuilt Command samples. The Command task simplifies configuration automation using SSH and CLI.

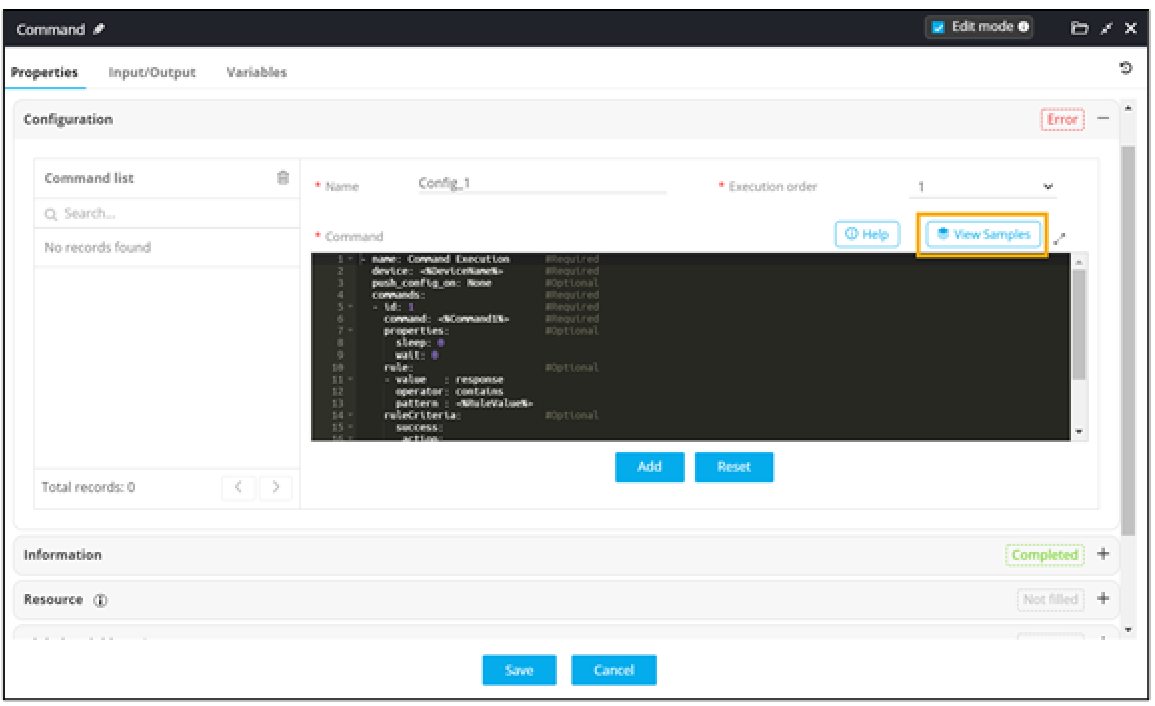
- YAML - declarative command execution
- Parallel, sequential configuration execution
- Intelligent command execution with in-built rules (Multi-boolean conditions operators with shortcuts)
- Event handling and dynamic variable support
- Modify configurations and reviews
- Pre-validation and post-validation of configurations
- Automate configuration automation for ADC devices, routers, and switches

To use a Command task within your workflow:

- Using Command Task in the Workflow
- Using Command task to pass the task response as a Global Variable

Using Command Task in the Workflow

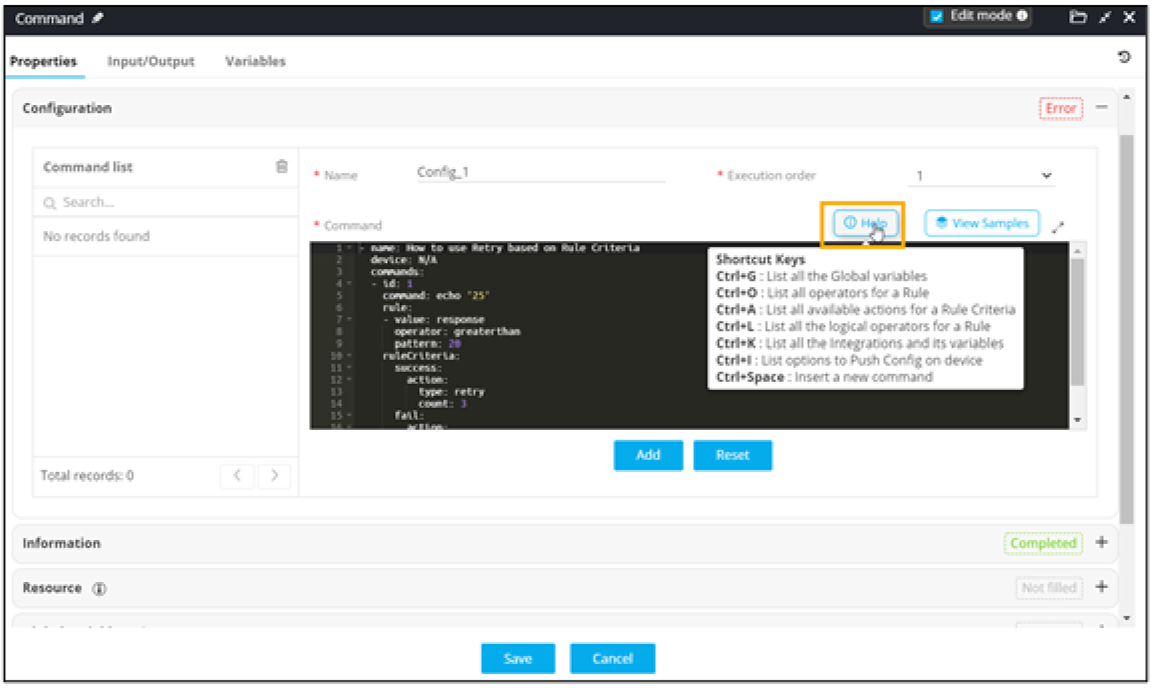
1. Design a workflow.
2. From the **User Interface** section in the left menu, drag and drop the **Command** task.
3. To use a prebuilt command sample, in the **Command** task window, under **Properties**, in the **Configuration** section, click **View Samples**.



4. In the **Command Samples** window, click **View** next to the Command Sample.
5. To use this configuration, click **Use Sample**.

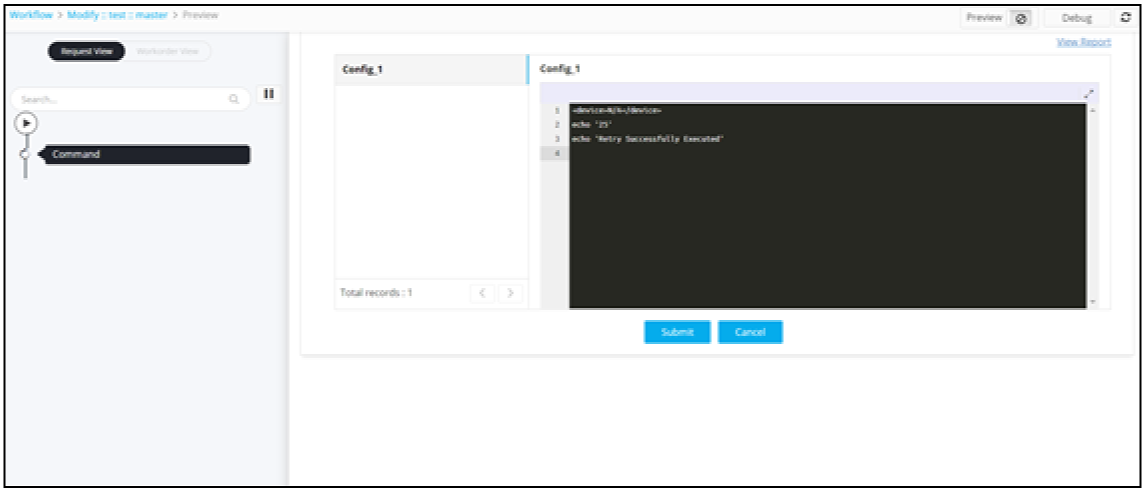
6. To save the configuration settings, in the **Command** task window, under **Properties**, in the **Configuration** section, click **Add**.

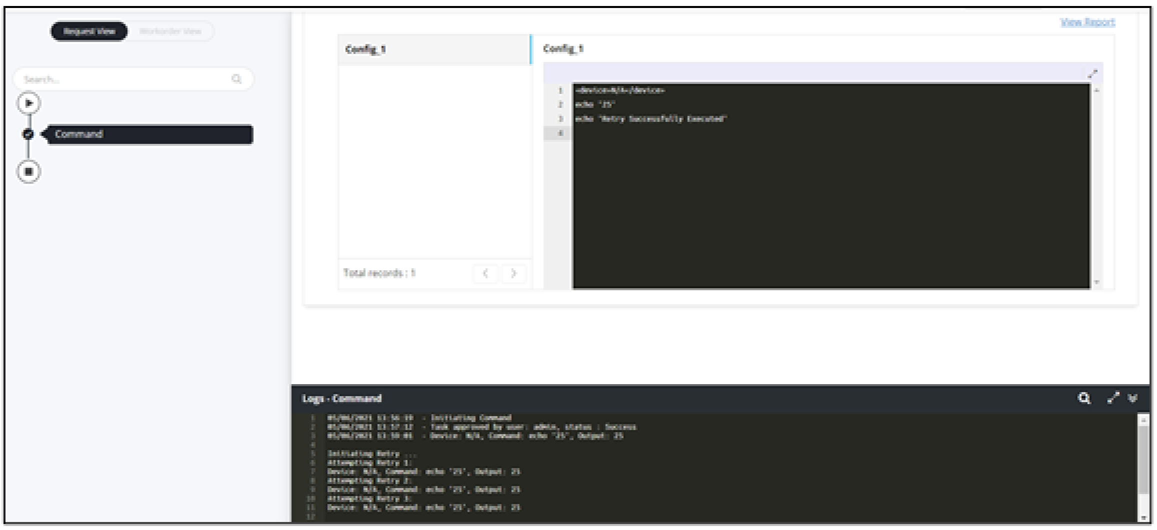
> **Tip:** To see a list of keyboard shortcuts, hover your mouse over **Help**.



7. Click **Save**.
8. Connect the workflow tasks and from the menu on the top, click **Preview**.

   Command task is executed.





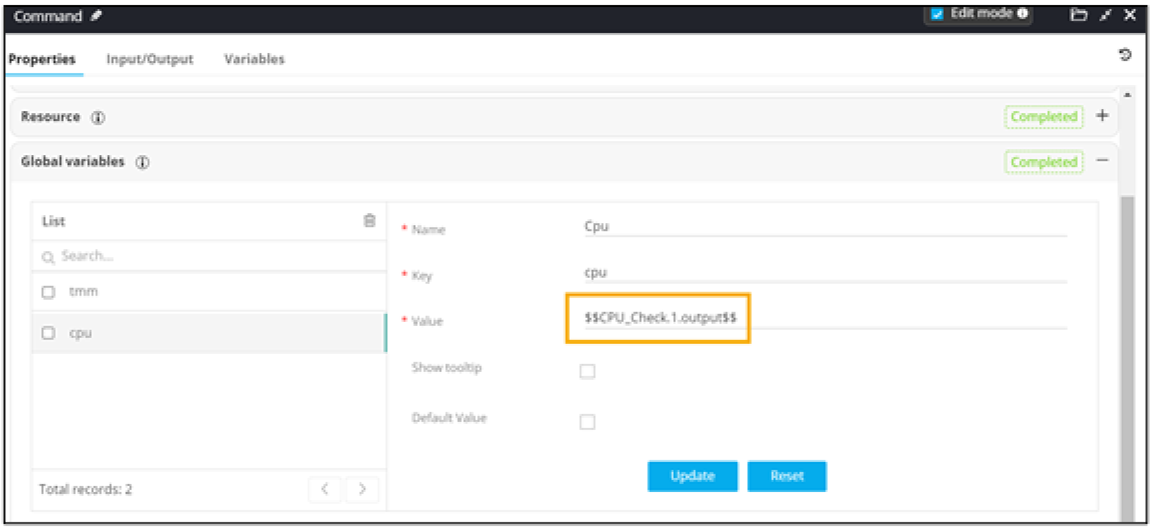
Using Command task to pass the task response as a Global Variable

The Command task can be used to pass the response of a task as a global variable for the next task in the workflow. The global variable can be declared in the following ways:

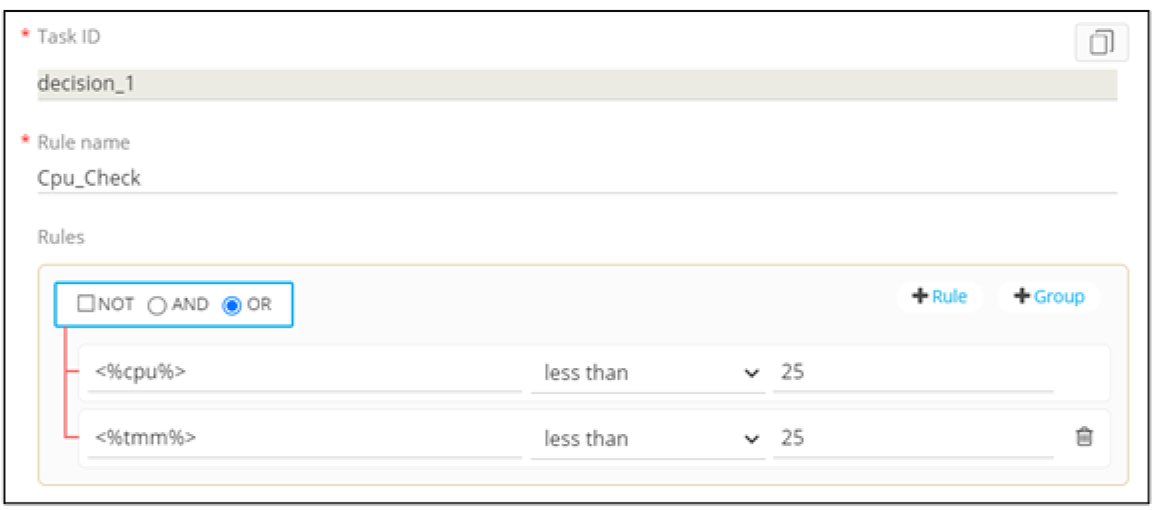
- `$$<Config Name>$$` - This returns the value of all the Commands with the Device Name and Response.
- `$$<Config Name>.<Command ID>$$` - This returns the value of the specific Command with the Device Name and Response.
- `$$<Config Name>.<Command ID>.output$$` - This returns the response of the Command ID.
- `$$<Config Name>.<Command ID>.command$$` - This returns the value of the command of the provided Command ID.
- `$$<Config Name>.<Command ID>.deviceName$$` - This returns the Device Name of the provided Command ID.

To use the output from the Command task to perform a CPU/TMM check and create a virtual LTM server on a F5 BIG-IP device:

1. Design a workflow.
2. To pass the output of the Command task as a global variable, in the **Command** task window, under **Properties**, in the **Global variables** section, define the value to be declared as a global variables using the following syntax: `$$Cpu.1.output$$` `$$Cpu.2.output$$`

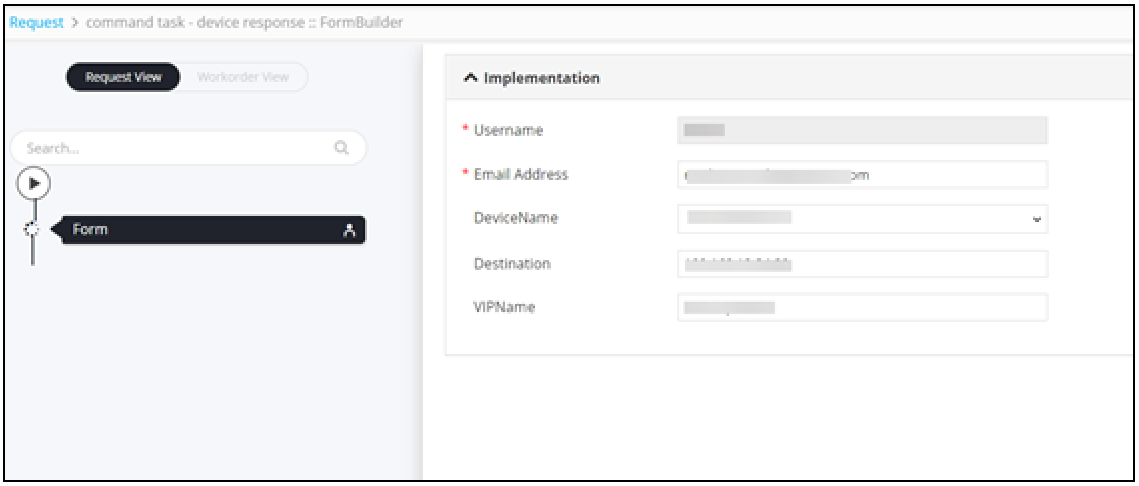


3. In the **If** task window, define the rules for executing the next task in the workflow using the Global variables defined in the Command task.
   For example, create a device if the CPU/TMM values are less than 25.

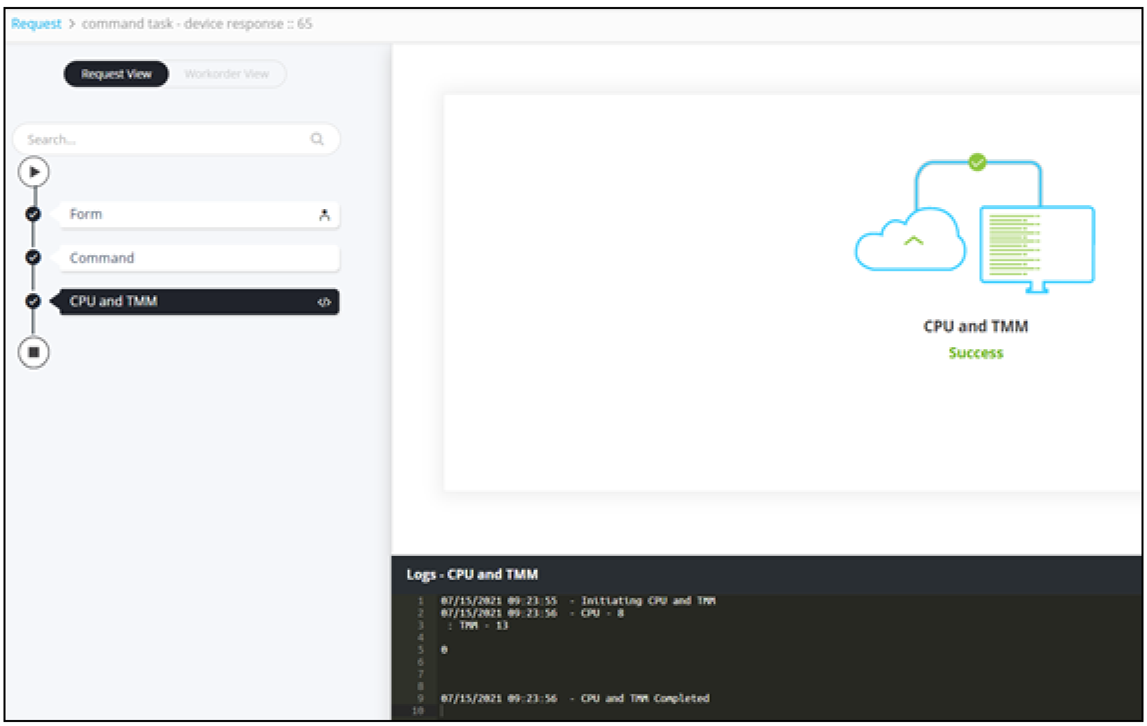


4. Connect the workflow tasks and enable the workflow.
5. Trigger the workflow.
   The workflow execution page is displayed with the workflow inputs requested at the first stage.

6. Enter the Destination and VIP Name in the Input form.



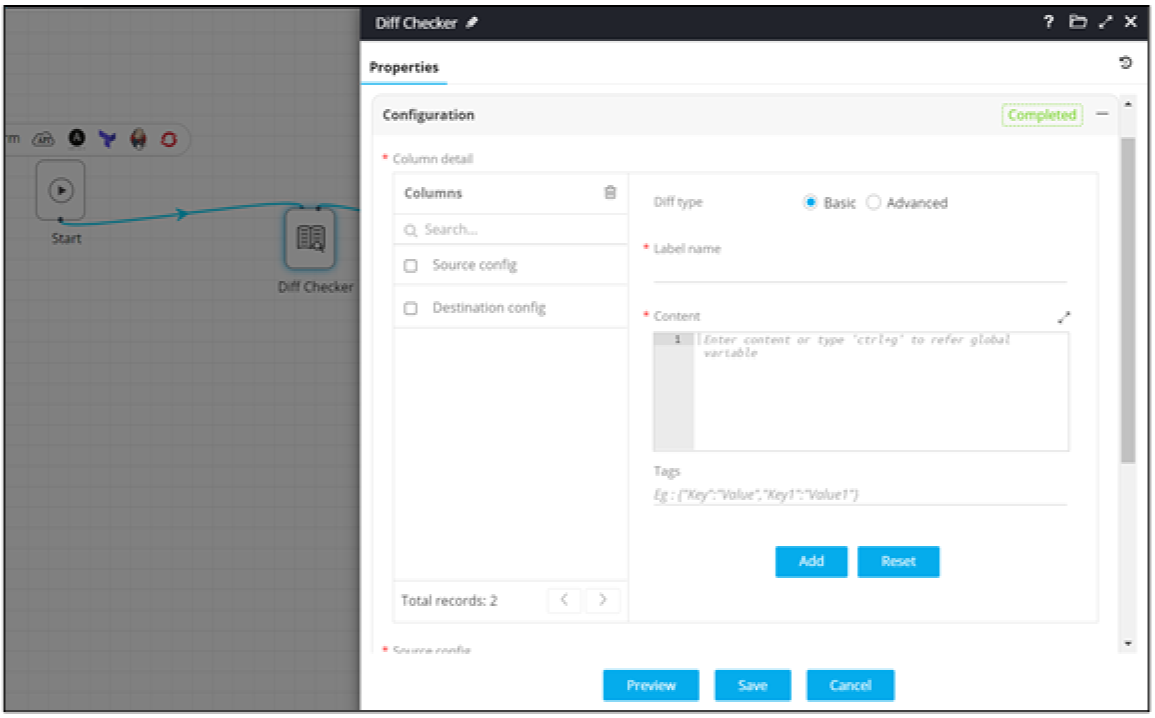
Command task initiated and CPU and TMM check completed.



Diff Checker

The Diff Checker task allows you to perform a side-by-side comparison of configuration files. You can automate network configuration compliance and ensure that network device configurations adhere to best practices and standards, thus avoiding manual configuration errors.

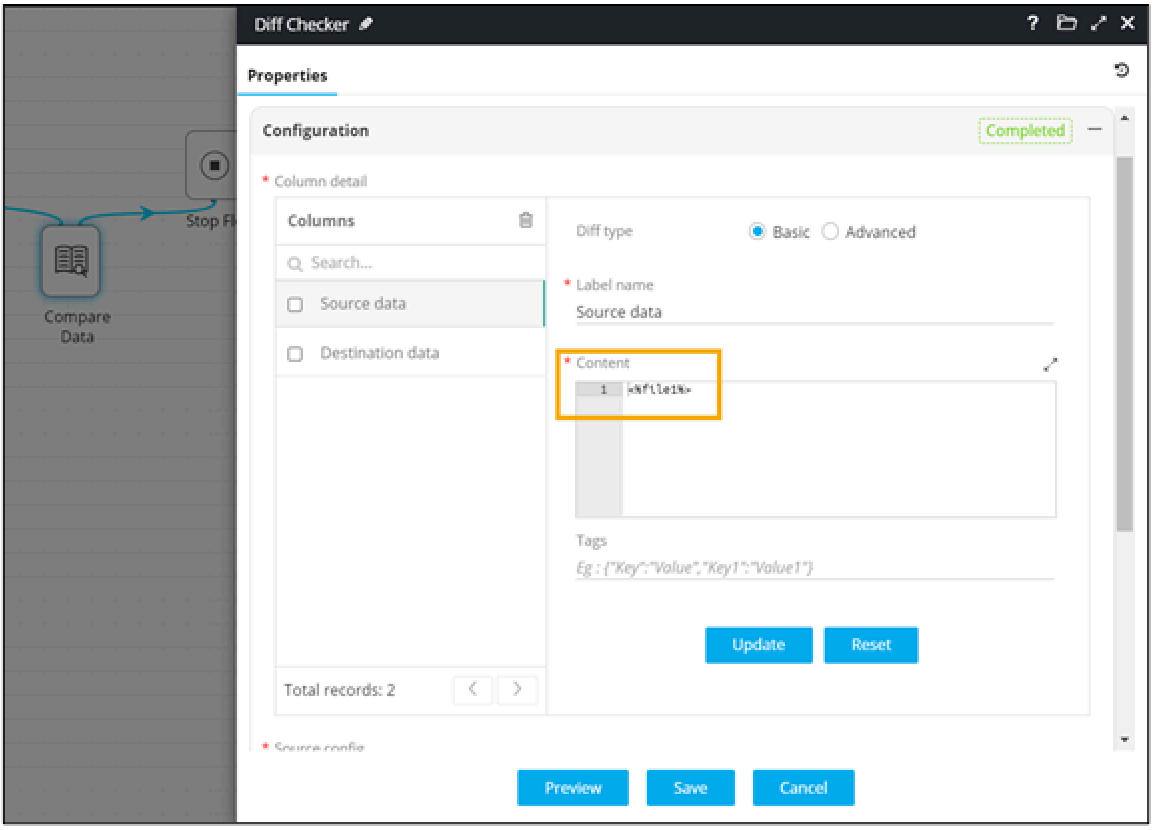
- Easy to build low-code diff checker task.
- Perform static and dynamic comparison of configurations across two to five devices/objects.
- Provision for basic and advanced diff checker options.
- Provision to reference variables from any workflow task for comparison.
- Provision to define custom hooks (scripts, REST API) within the advanced diff checker.
- Provision to compare differences in configuration before pushing to the end device.
- Provision to compare changes in peer review and approval process.



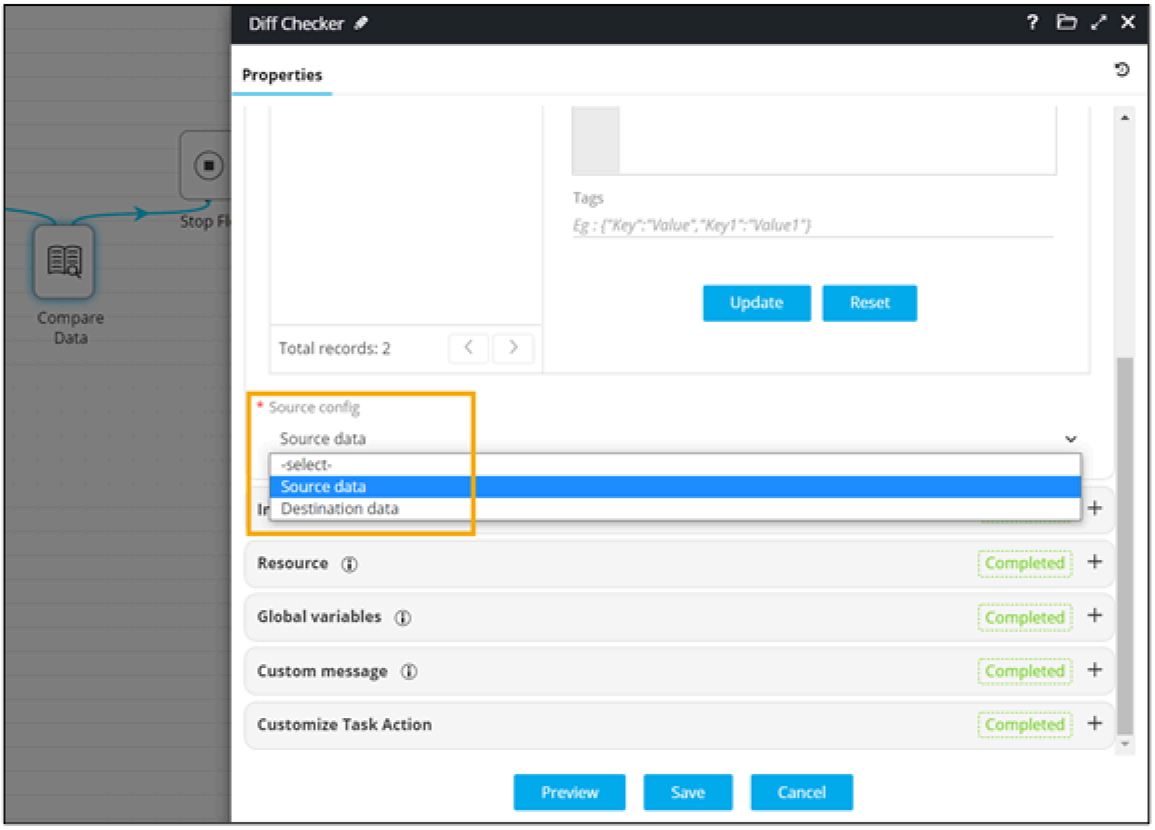
- Configuring a Basic Diff Checker
- Configuring an Advanced Diff Checker

Configuring a Basic Diff Checker

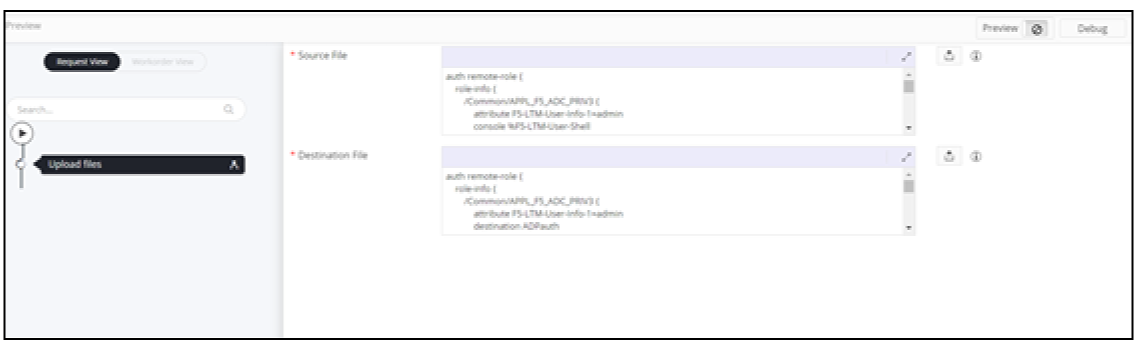
1. Design a workflow.
2. In the design workspace, from the User Interface section, drag and drop the **Diff Checker** task.
3. In the **Diff Checker** task window, under **Properties**, in the **Configuration**section, select **Diff type** as **Basic**.
4. Enter the **Content** to be compared or reference content from any previous workflow task as a variable.



5. Under **Source config**, select the source column against which the diff checker comparison must be made.



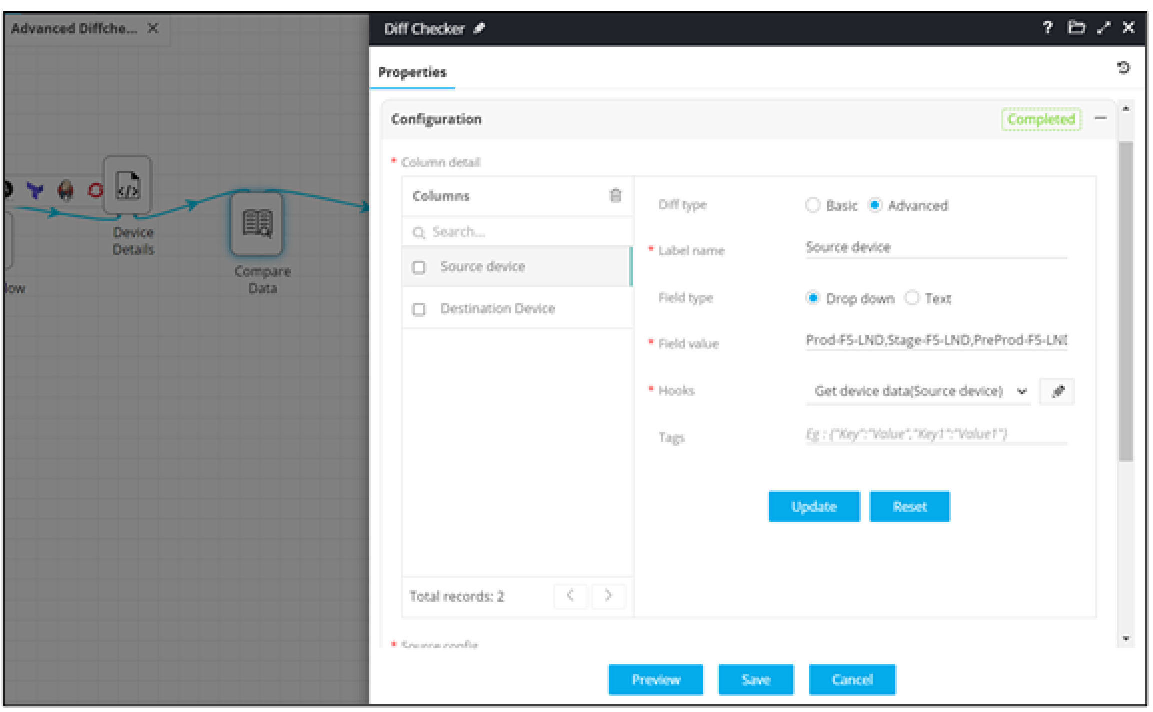
6. Connect the workflow tasks.
7. From the command bar on the top, click **Preview**.
8. Upload the source files.



Basic Diff Checker allows you to compare the data.

Configuring an Advanced Diff Checker

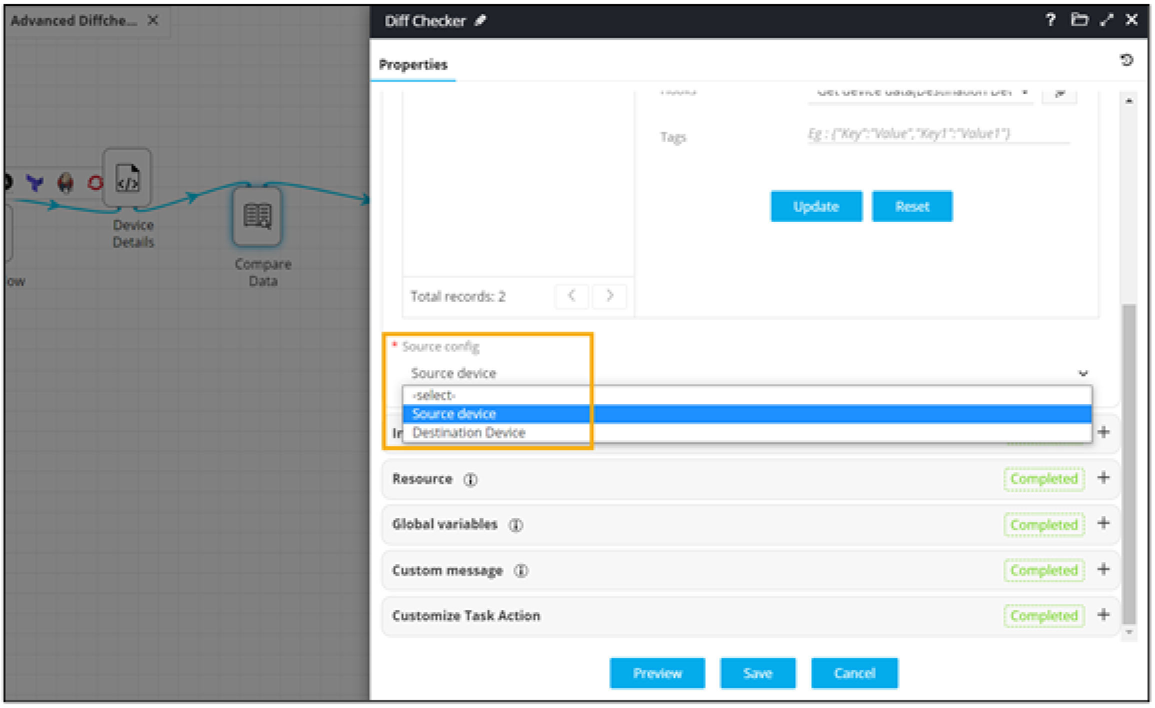
1. Design a workflow.
2. From the User Interface section, drag and drop the **Diff Checker** task.
3. In the **Diff Checker** task window, under **Properties**, in the **Configuration**section, select **Diff type** as **Advanced**.
4. Enter or select the field information as required.



Field descriptions for Advanced section

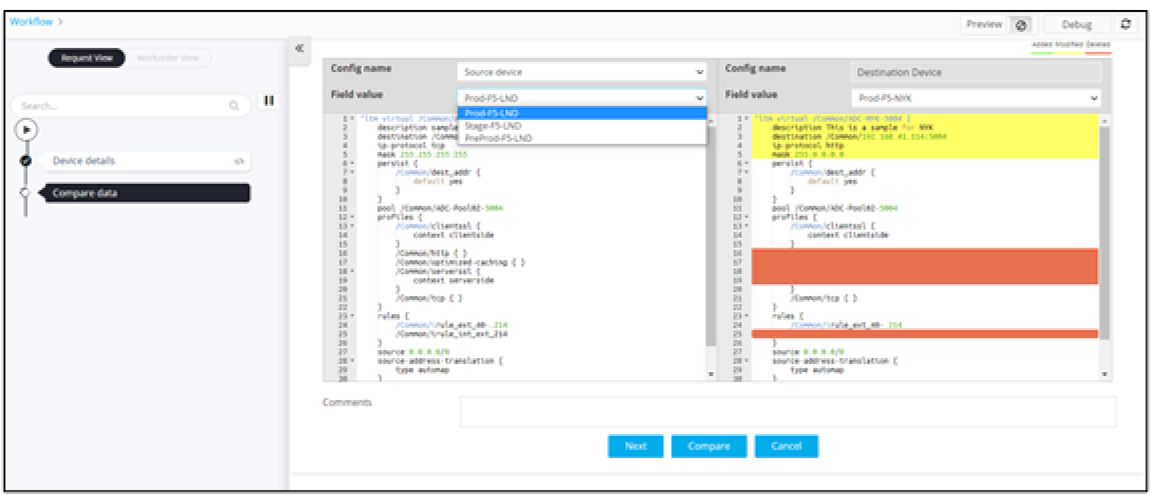
Field	Description
***Label Name**	Define custom label names as headings for every diff checker column.
Field Type	Allows you to define the input required to execute a hook dynamically during the diff checker view in real time. It allows for either providing the inputs dynamically as: • **Text**: Values can be entered within a text box which will be taken as an input variable to execute the hook. • **Dropdown**: Inputs to execute the hooks can be defined as dropdown values allowing hooks to trigger dynamically.
***Field Value**	Define either a static or dynamic value through a variable to be displayed in the dropdown field.
***Hooks**	Select an existing hook from the Hooks Inventory or define a custom hook specific to the diff checker task.
Note: The asterisk (*) symbol indicates mandatory fields.	

5. Under **Source config**, select the source column against which the diff checker comparison must be made.



6. Connect the workflow tasks.
7. From the command bar on the top, click **Preview**.

   Advanced Diff Checker allows you to compare the data.



Form

Forms enable easy self-servicing capabilities to the Line of Business (Application owners, NetOps and so on) via an intuitive GUI-based interface, thus abstracting the end user from underlying infrastructure configurations.

The form can either be used out of the box form from the inventory for quick self-servicing or customized using the Form builder based on your automation requirements to build simple and complex workflows.

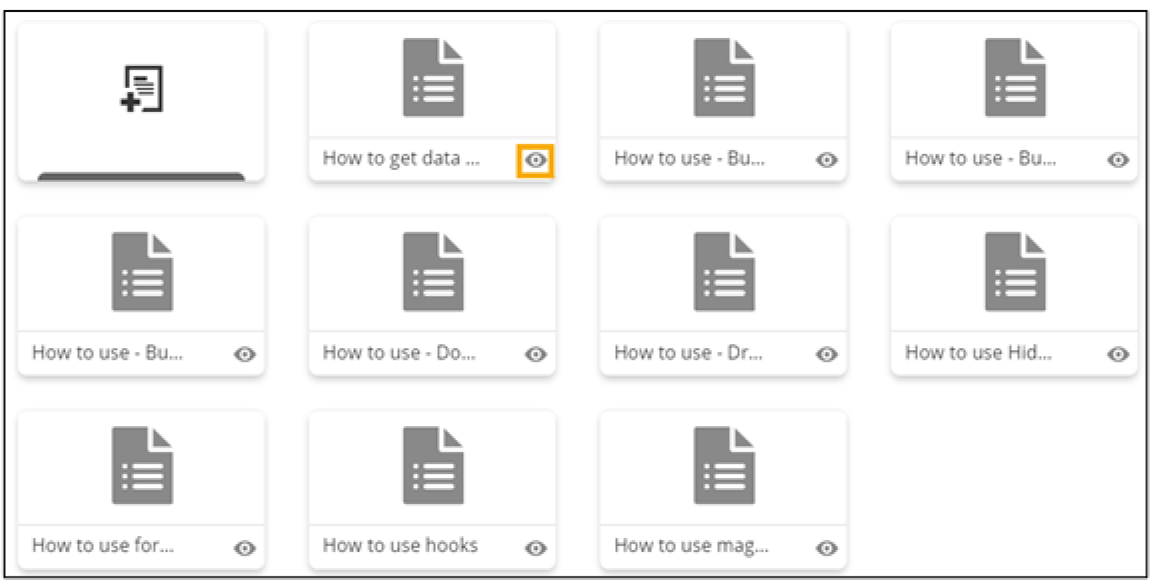
- Out of the box Forms
- Designing a Custom Form
- Adding Form Fields in the Form Builder
- Using Hooks in a Form
- Defining Resource & Settings

Out of the box Forms

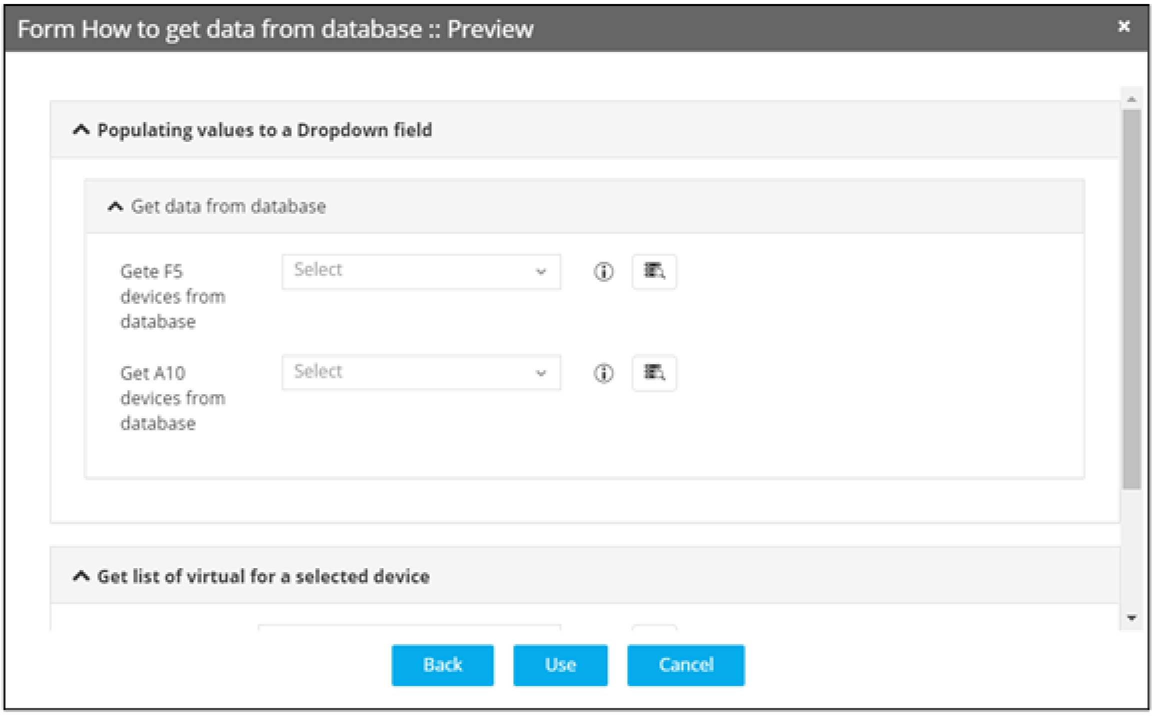
You can customize out of the box (OOB) forms to build workflows.

To use an OOB form:

1. Design a workflow.
2. From the **User Interface** section, drag and drop the **Form** task.
3. In the **Gallery** window, to preview the form, click **(View)** icon on any of the forms.



A preview of the form is displayed.



4. To use this form, click **Use**.
5. To go back and preview another form, click **Back**.

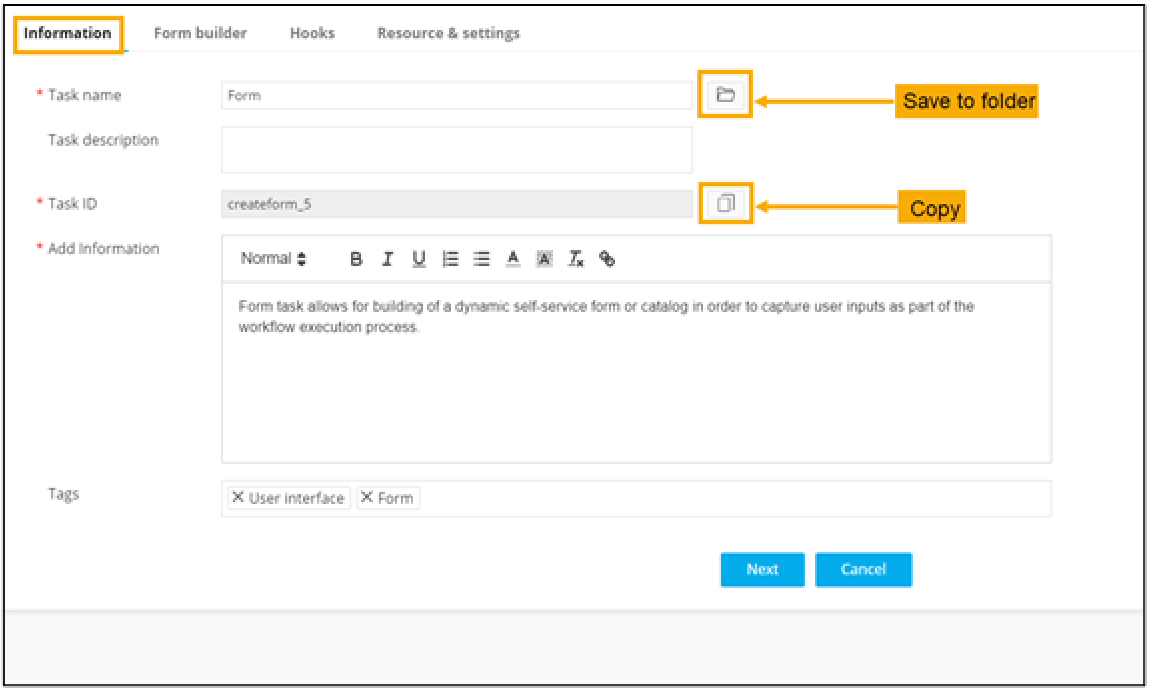
Designing a Custom Form

You can build a dynamic self-service form or catalog using the form builder GUI in order to capture user inputs as part of the workflow execution process. The Form builder allows you to:

- Build a custom form.
- Associate custom script(s) within the form builder.
- Assign form access to specific user role(s).
- Declare one or more form field values as ‘global variables’.

To design a custom form:

1. Design a workflow.
2. From the **User Interface** section, drag and drop the **Form** task.
3. To design a new form, in the **Gallery** window, click **(Create)** icon.
4. Under **Information**, enter the necessary field values.

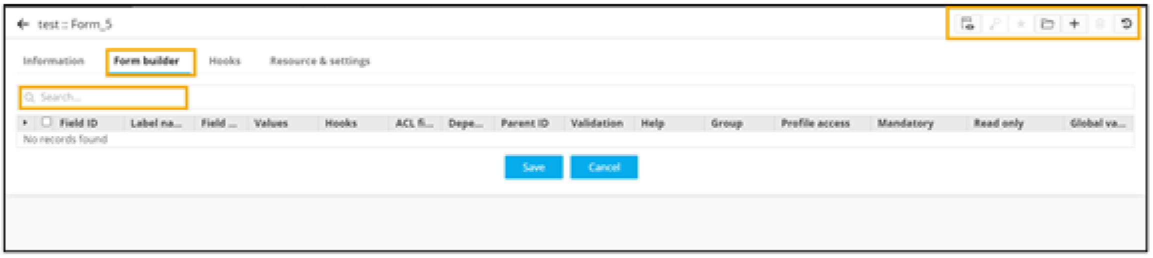


Field descriptions for Information

Field	Description
***Task name**	Provide a unique task name.
	Provision to bookmark the task within a custom folder in the workflow studio.
Task description	Enter a valid task description.
***Task ID**	Enter a unique task ID of the form task.
	Provision to copy the task ID and use it across the workflow.
***Add Information**	Provide information about the form task.

Note: The asterisk (*) symbol indicates mandatory fields.

5. Under **Form builder**, add the form fields.

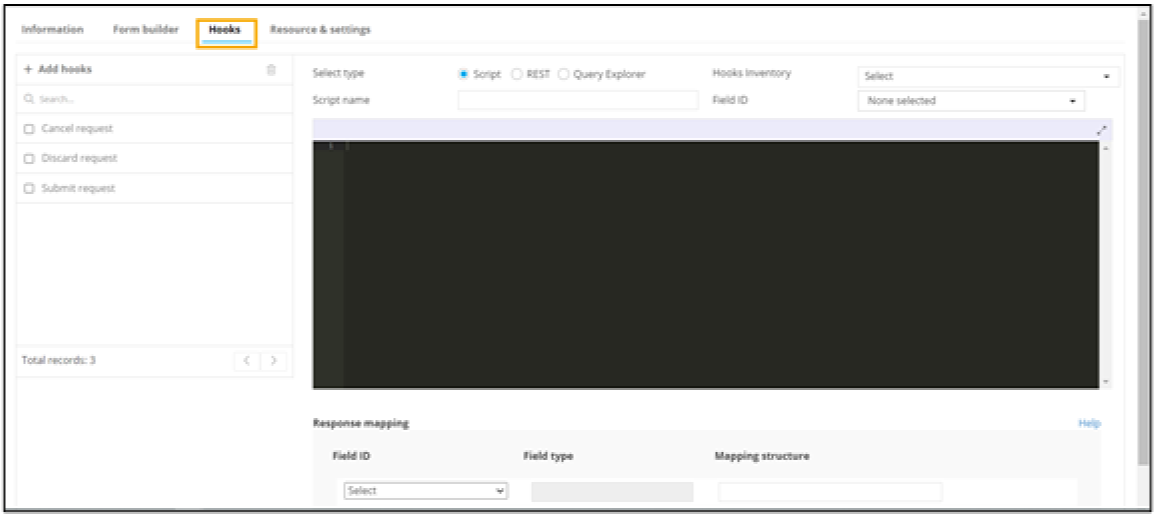


Field descriptions for Form builder

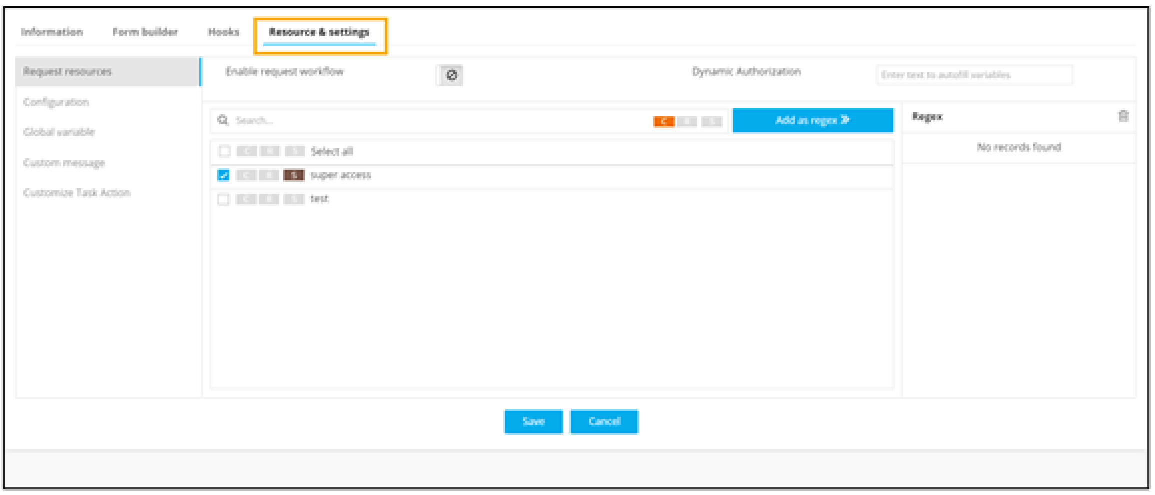
Option	Description
Search	Allows you to search for a specific form field.
	Allows you to preview the form.
	Allows you to select one or more form fields and assign to a user profile.
	Allows you to select one or more form fields and mark them as mandatory or otherwise.
	Allows you to select multiple form fields and group them logically.
	Allows you to add multiple form fields.
	Allows you to select one or more form fields and delete them.
	Allows you to view form history.

Note: For more information on adding form fields, click here.

6. To query different data sources such as database, device, or external systems and aid in mapping the response values to a form field, click **Hooks**.



7. To assign access to specific roles(s), click **Resource & Settings**.



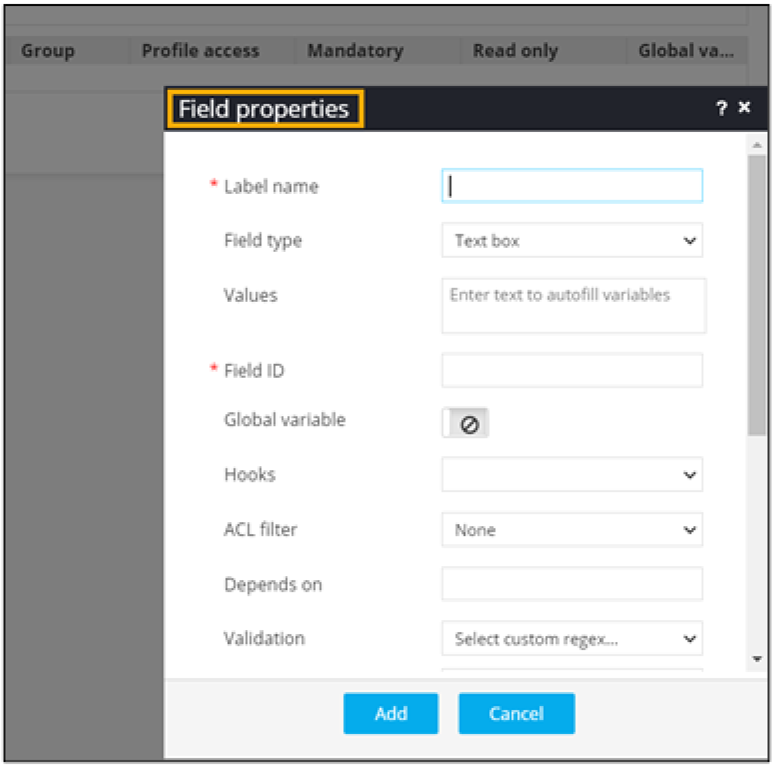
When trying to add the regex "*", a validation error will be triggered, resulting in the disabling of the "Add as regex" option.

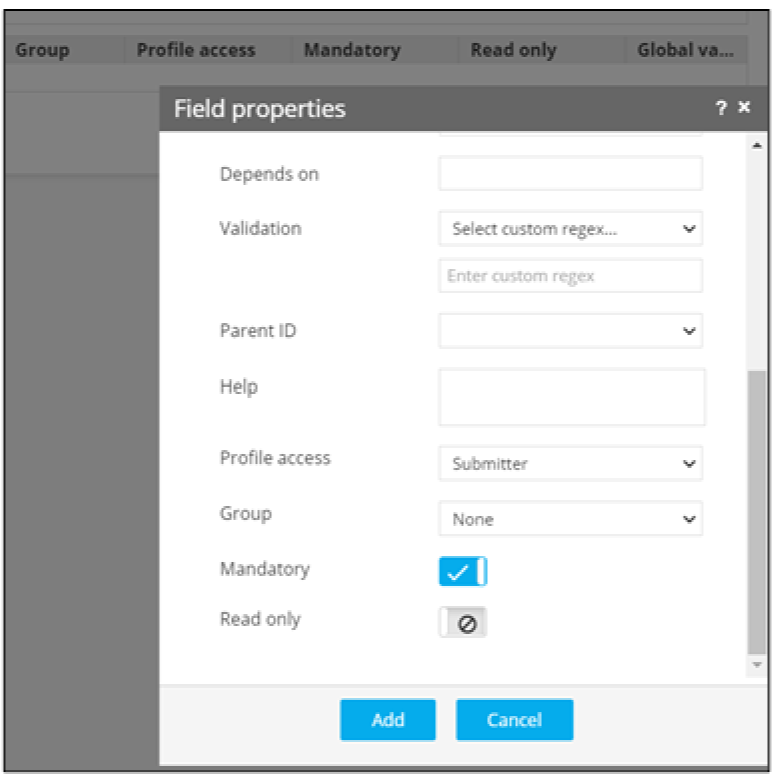
Adding Form Fields in the Form Builder

To add a form field:

1. Design a workflow.
2. From the **User Interface** section, drag and drop the **Form** task.
3. To design a new form, in the **Gallery** window, click **(Create)** icon.
4. Under the **Form Builder** section, from the upper right corner of the screen, click **(Add)** icon.

5. In the **Field properties** window, enter or select the field information.





Field descriptions for Field properties

Field	Description
***Label name**	• Enter a valid label or field name such as Username, Enter IP address, Environment Type, Environment Info and so on. • Char Length: Min (2), Max (120).
Field type	Select field type from the dropdown.
Values	• Define field values relevant to a form field. • Values can be delimited by a comma. • Chars not allowed: ‘&’ and ‘|’.
***Field ID**	• Enter a unique field ID to reference the field name. • Allowed characters: ‘-‘ and ‘_’ • Character Length: Min (1), Max (50).
Global variable	Assign the field as a global variable.
Hooks	• Select a hook that is relevant to the from field. • Hooks can be either Script or REST API. • Hooks are used to fetch details against a specific form field from the device and/ or database and map them to the respective field on the user interface.
ACL filter	• This field is used to validate, filter information displayed in the provisioning request page based on RBAC defined through ‘authorized control’ under account management. • Options: Device, Device Objects, Cert Groups, and None (default).
Depends on	• Define dependencies between form fields based on ‘Field ID’ and ‘Values’. • Operators used: ‘&’ and ‘|’ • Max character: 550 • Format: [Field Id1: value1 & Field Id2: value1 & Field Id3: value3], [Field Id1: value1 | Field Id2: value1].
Parent ID	• Select the relevant tabular component (Field ID) against which the form field must be mapped to. • Used only when defining a tabular element in the form.
Help	Enter the help instructions that need to be defined against the field for the user’s reference.

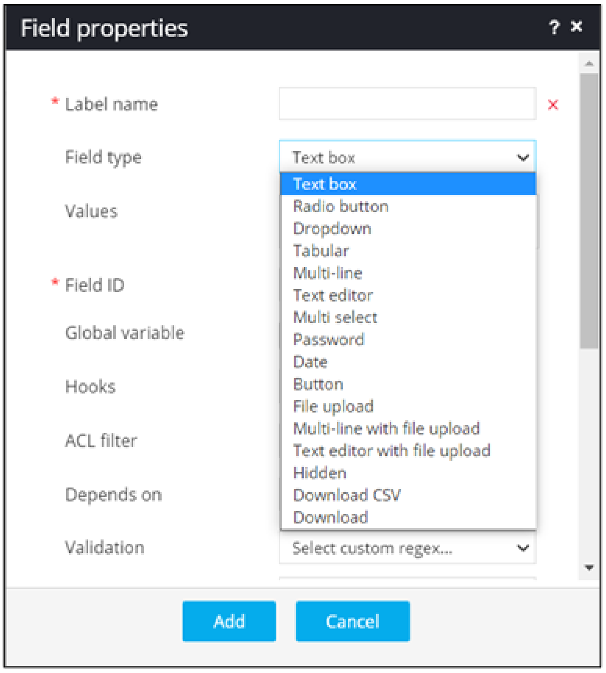
Field	Description
Profile access	Select the profile from the dropdown to restrict form field display within a template based on the relevant profile.
Group	• Group fields logically within the form. • Associate an existing group or create a new group.
Mandatory	Provision to mark the field as mandatory or otherwise.
Read only	Provision to mark the field as read only or otherwise.

Note: The asterisk (*) symbol indicates mandatory fields.

- Field Types
- Validating Form Fields
- Defining Dependencies between Form Fields
- Defining Tasks as Read Only
- Grouping Multiple Form Fields

Field Types

The form builder allows for specific customization of the form fields using various field types.

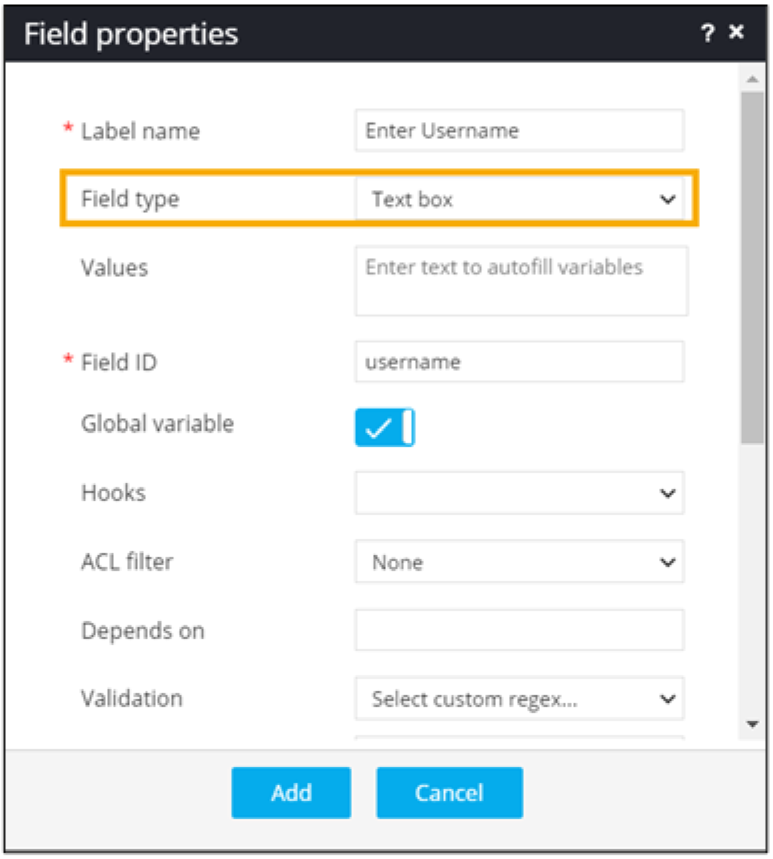


- Field Type - Text Box
- Field Type - Radio button
- Field Type - Dropdown
- Field Type - Tabular
- Field Type - Multi-line
- Field Type - Text editor
- Field Type - Multi select
- Field Type - Password
- Field Type - Date
- Field Type - Button
- Field Type - File upload
- Field Type - Multi-line with file upload
- Field Type - Text editor with file upload
- Field Type - Hidden
- Field Type - Download CSV
- Field Type - Download

Field Type - Text Box

You can define a text box element on the form.

1. To add a **Text box** to the form, under **Form builder**, from the top right corner of the screen, click **(Add)** icon in the command bar.
2. In the **Field properties** window, enter or select the field information as shown.

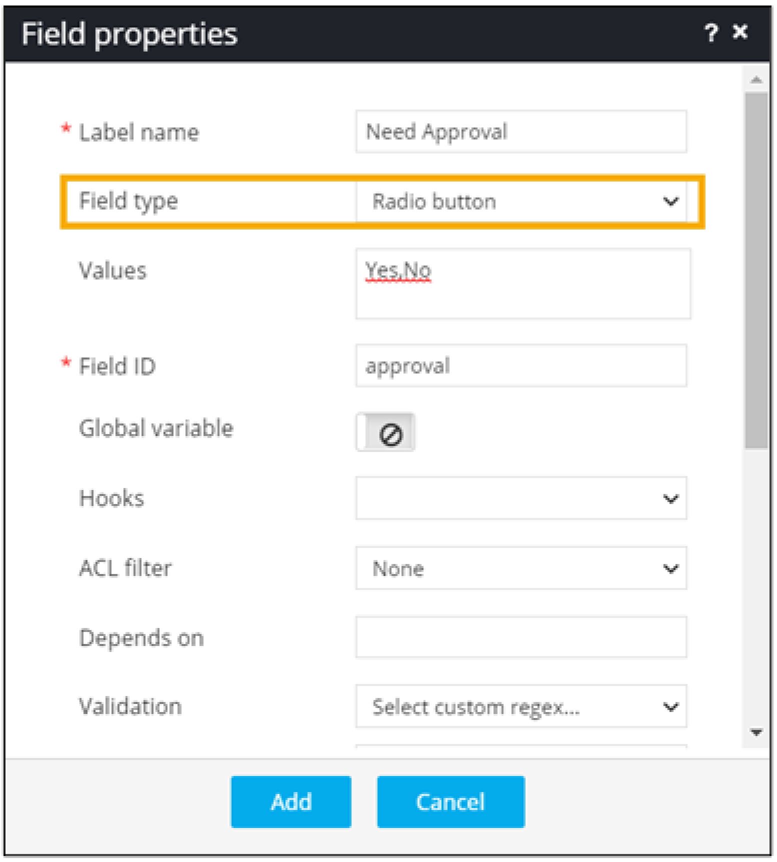


3. To add this field to the form, click **Add**.
4. To see the text box added to the form, from the top right corner of the screen, click **(View)** icon.

Field Type - Radio button

You can define a multi-select option using a radio button on the form.

1. To add a **Radio button** to the form, under **Form builder**, click **(Add)** icon in the command bar.
2. In the **Field properties** window, enter or select the field information as shown.



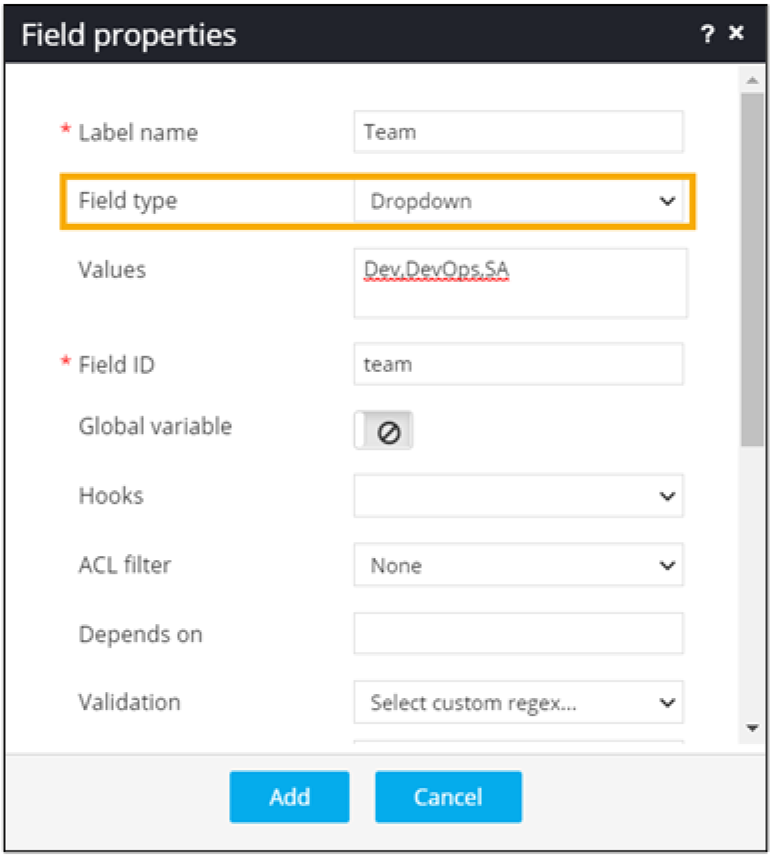
3. To add this field to the form, click **Add**.

4. To see the Radio button added to the form, from the top right corner of the screen, click **(View)** icon.

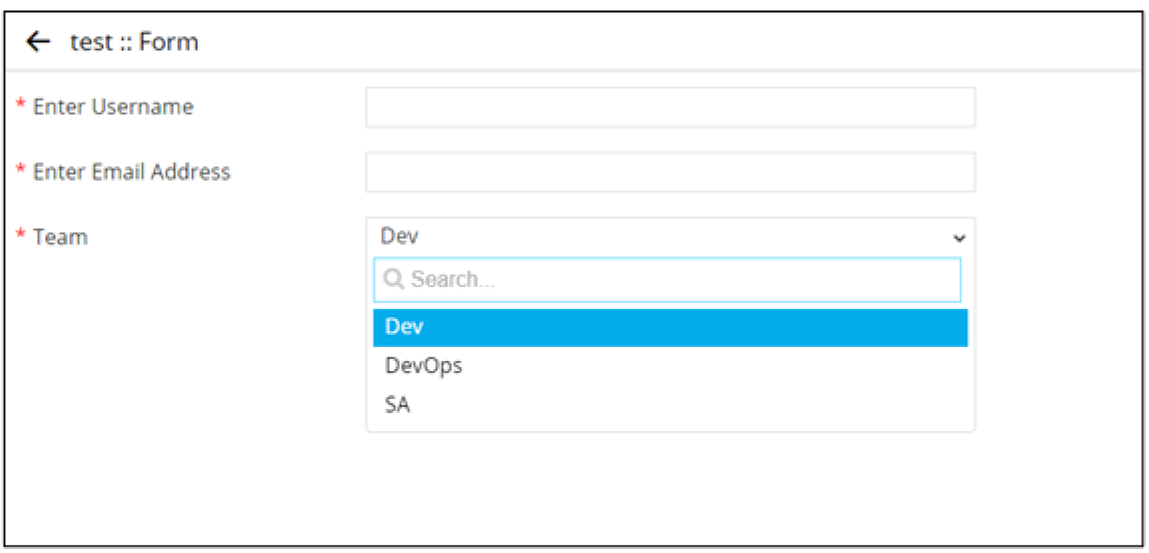
Field Type - Dropdown

You can define a dropdown element with multiple values to select from in the form.

1. To add a **Dropdown** button to the form, under **Form builder**, click **(Add)** icon in the command bar.
2. In the **Field properties** window, enter or select the information as shown.



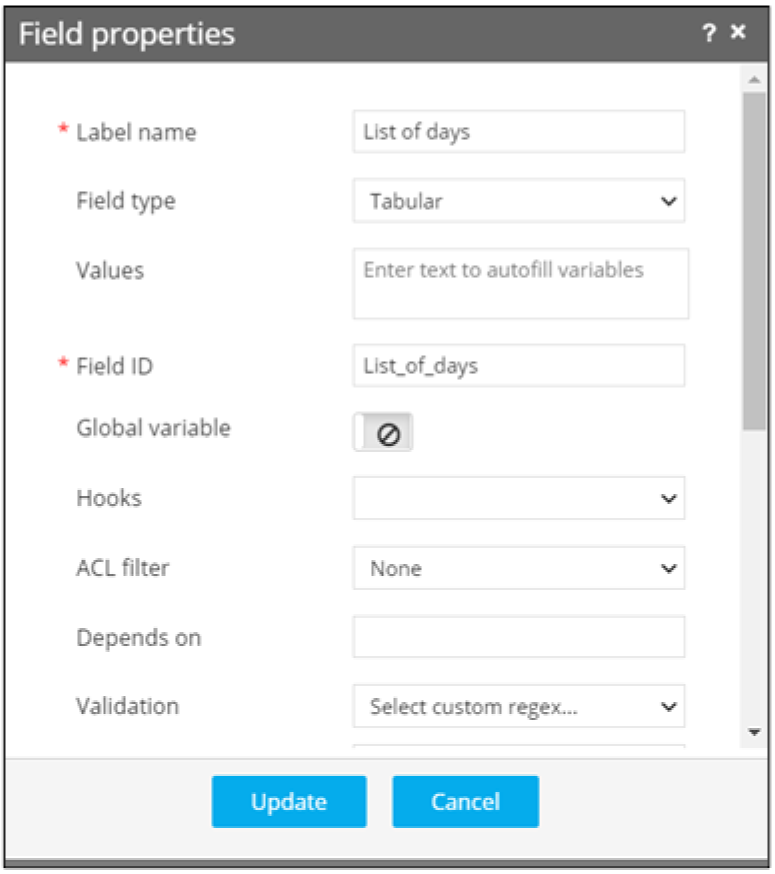
3. To add this field to the form, click **Add**.
4. To see the Dropdown field added to the form, from the top right corner of the screen, click **(View)** icon.



Field Type - Tabular

This field type in the form builder allows you to define a tabular component (grid) and add,modify, delete data from the grid.

1. To add a **Tabular** field to the form, under **Form builder**, click **(Add)** icon in the command bar.
2. In the **Field properties** window, define a form field with field type as **Tabular**, with the respective Field ID.

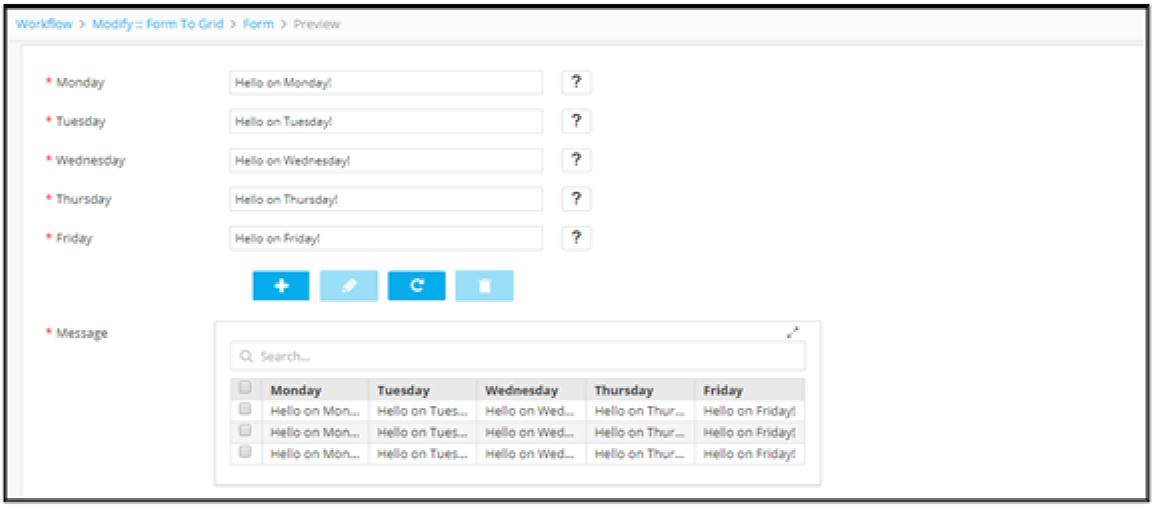


3. Define other form fields that are required for the tabular component.
   For example: Label name: Monday, Field type: Text box, Values: Hello on Monday.
4. Link the Parent ID (the field ID of the parent tabular element) against these fields.

Label name	Field type	Values	Field ID	Parent ID
Monday	Textbox	Hello on Monday	Monday	List_of_days
Tuesday	Textbox	Hello on Tuesday	Tuesday	List_of_days
Wednesday	Textbox	Hello on Wednesday	Wednesday	List_of_days

Label name	Field type	Values	Field ID	Parent ID
List of days	Tabular		List_of_days	

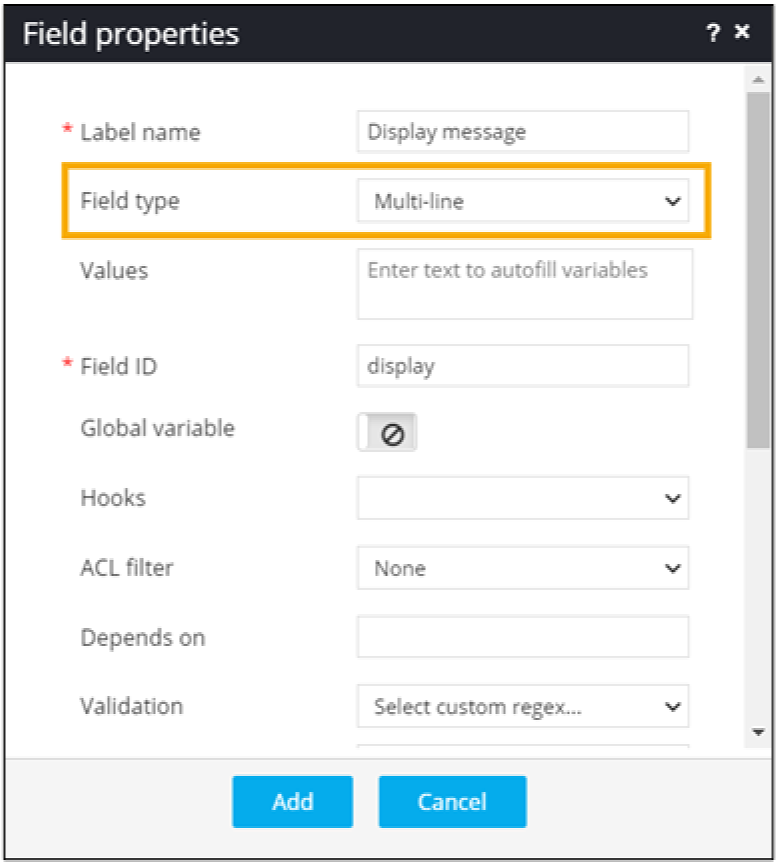
5. To see the Tabular field added to the form, click **(View)** icon.



Field Type - Multi-line

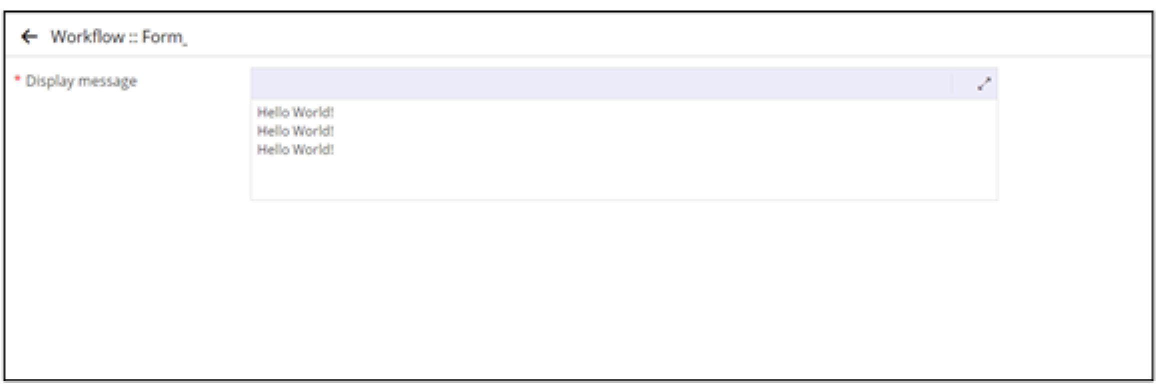
This field type in the form builder allows you to capture multiple lines of data within a form.

1. To add a **Multi-line** field to the form, under **Form builder**, click **(Add)** icon in the command bar.
2. In the **Field properties** window, enter or select the field information as shown.



3. To add the Multi-line field to the form, click **Add**.

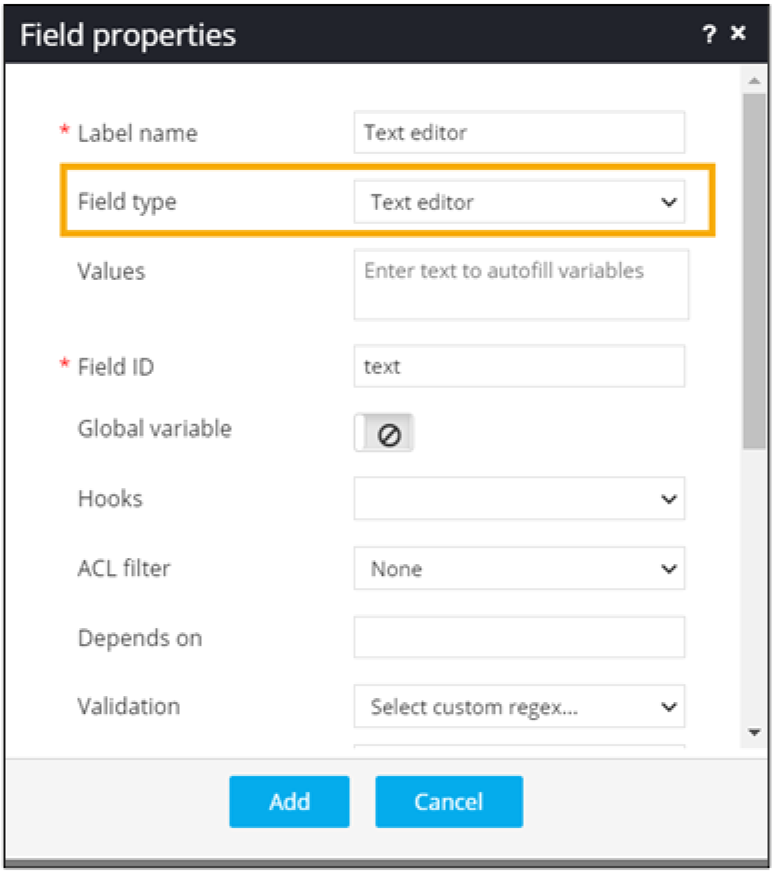
4. To see this field added to the form, from the top right corner of the screen, click **(View)** icon.



Field Type - Text editor

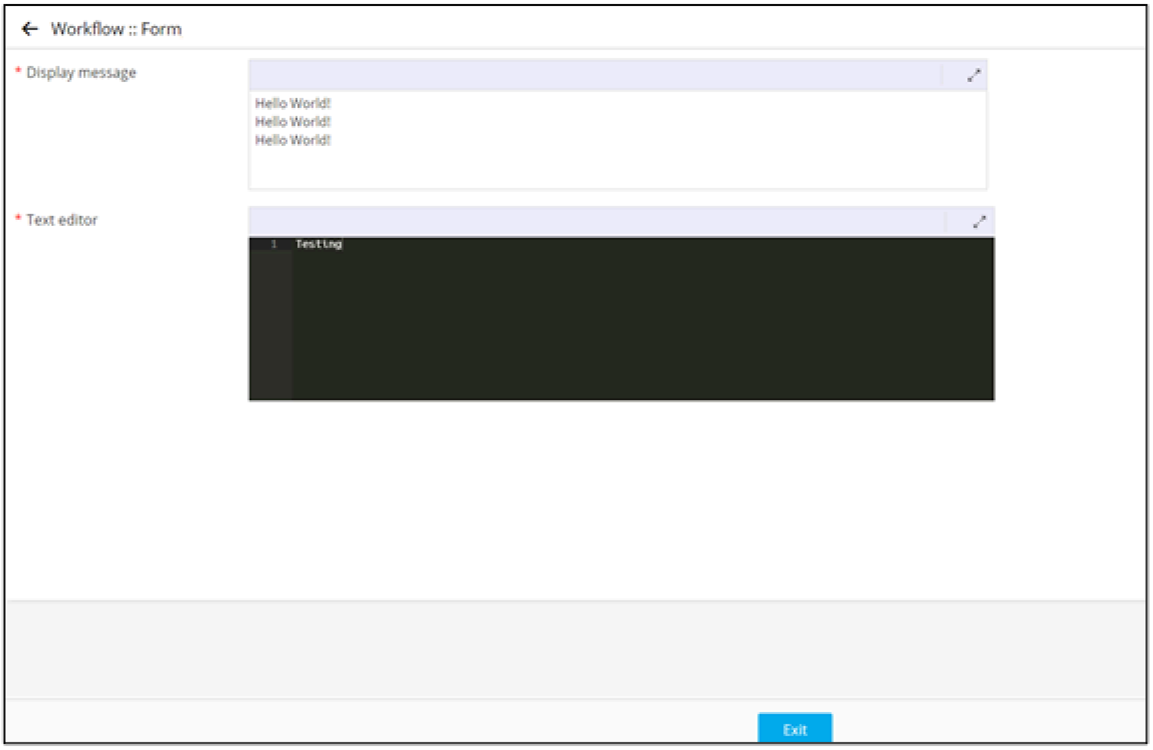
This field type allows you to add a text editor to their form.

1. To add a **Text editor** field to the form, under **Form builder**, click + **(Add)** icon in the command bar.
2. In the **Field properties** window, enter or select information as shown.



3. To add the form field, click **Add**.

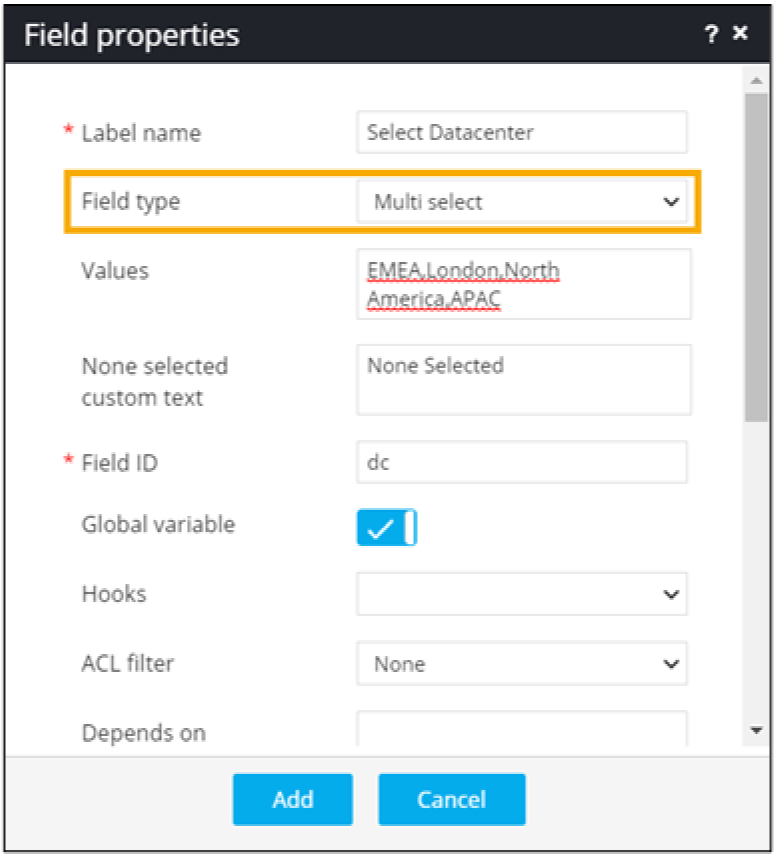
4. To see the field added to the form, click **(View)** icon.



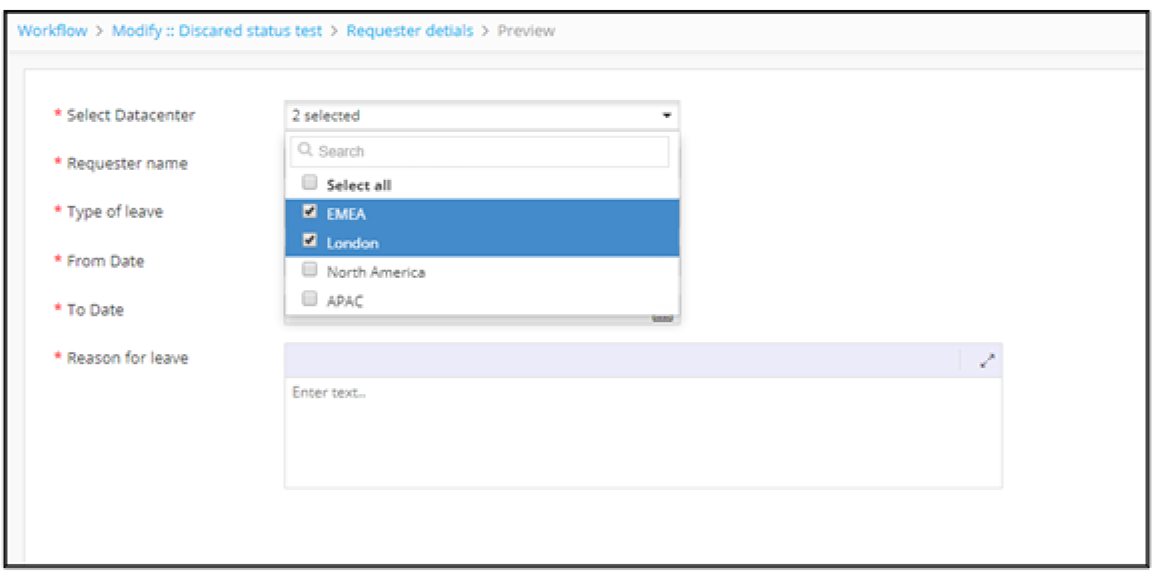
Field Type - Multi select

This field type in the form builder allows you to search and select multiple values from a specific dropdown form field within the request form.

1. To add a **Multi-select** field to the form, under **Form builder**, click **(Add)** icon in the command bar.
2. In the **Field properties** window, enter or select field information as shown here.



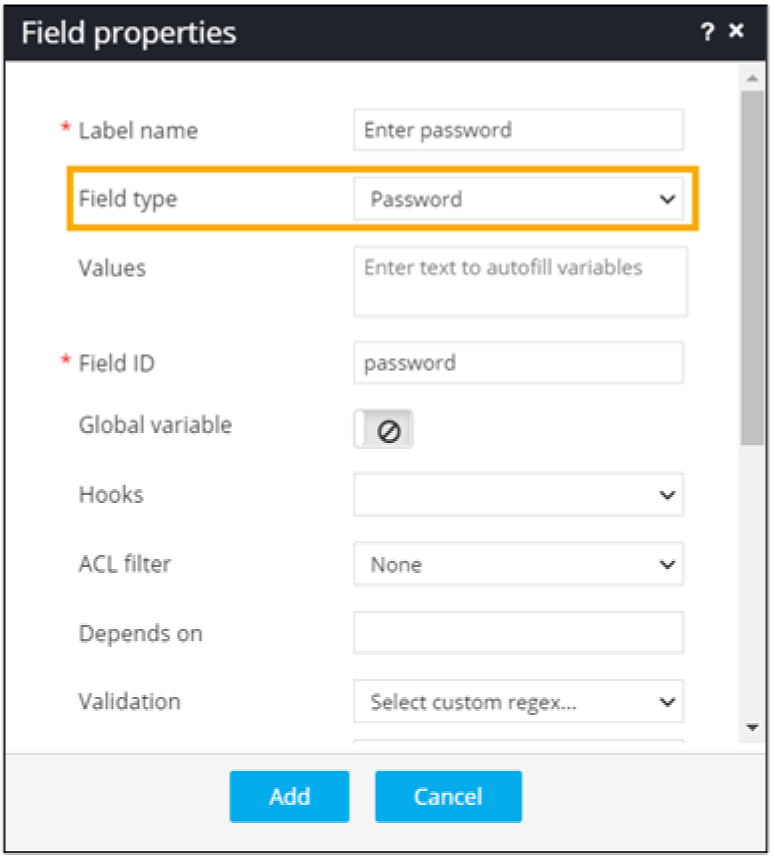
3. To add this field to the form, click **Add**.
4. To see the field added to the form, from the top right corner of the screen, click **(View)** icon.



Field Type - Password

This field type in the form builder allows you to enter passwords if necessary.

1. To add a **Password** field to the form, under **Form builder**, click **(Add)** icon in the command bar.
2. In the **Field properties** window, enter or select the field information as shown here.



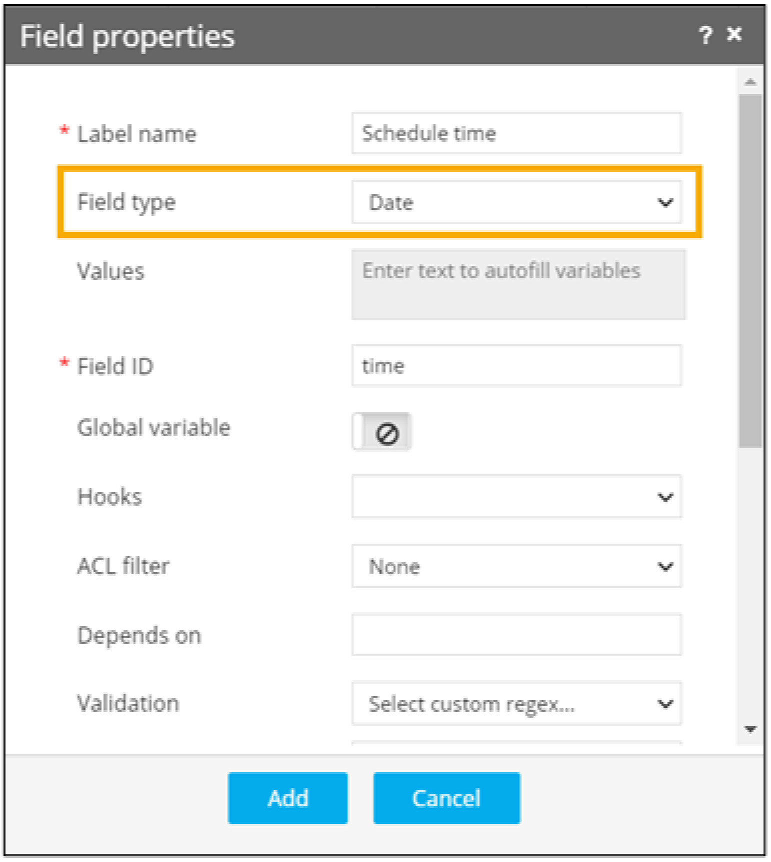
3. To add this field to the form, click **Add**.

4. To see the field added to the form, from the top right corner of the screen, click **(View)** icon.

Field Type - Date

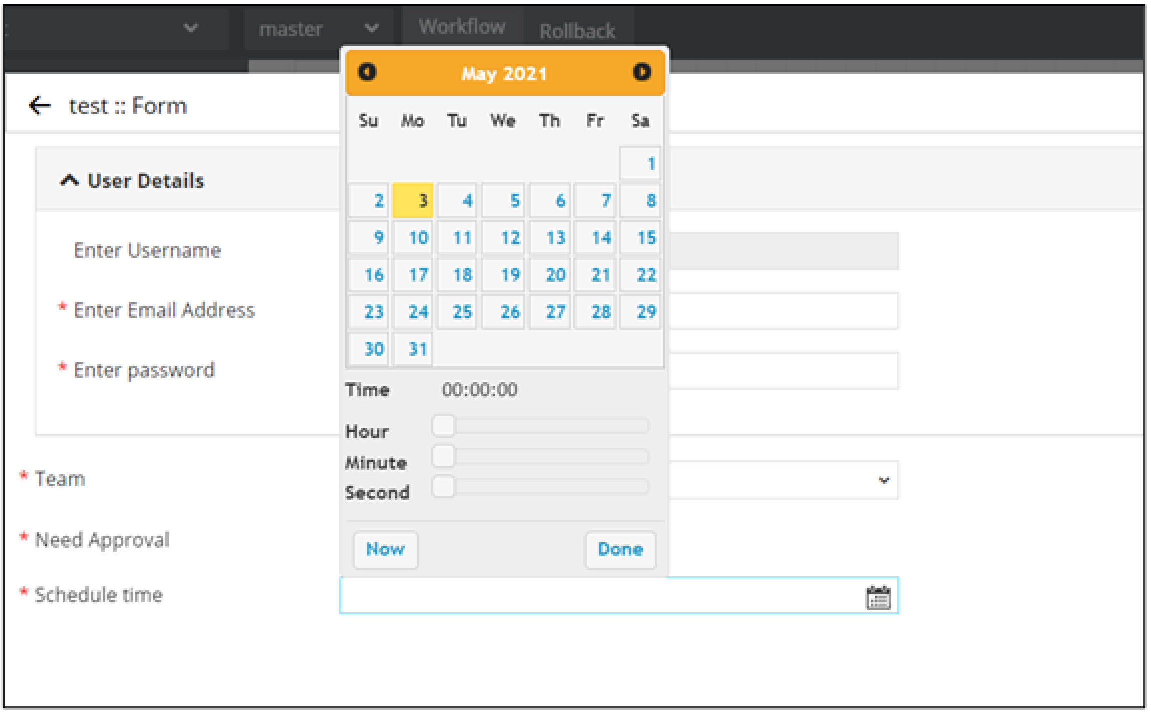
You can add this field type to your form to choose a specific date and time for scheduling tasks within the form.

1. To add a **Date** field to the form, under **Form builder**, click **(Add)** icon in the command bar.
2. In the **Field properties** window, enter or select the field information as shown here.


Field properties
* Label name
Schedule time
Field type
Date
Values
Enter text to autofill variables
* Field ID
time
Global variable
Hooks
ACL filter
None
Depends on
Validation
Select custom regex...
Add
Cancel

3. Click **Add**.

4. To see the field added to the form, click **(View)** icon.

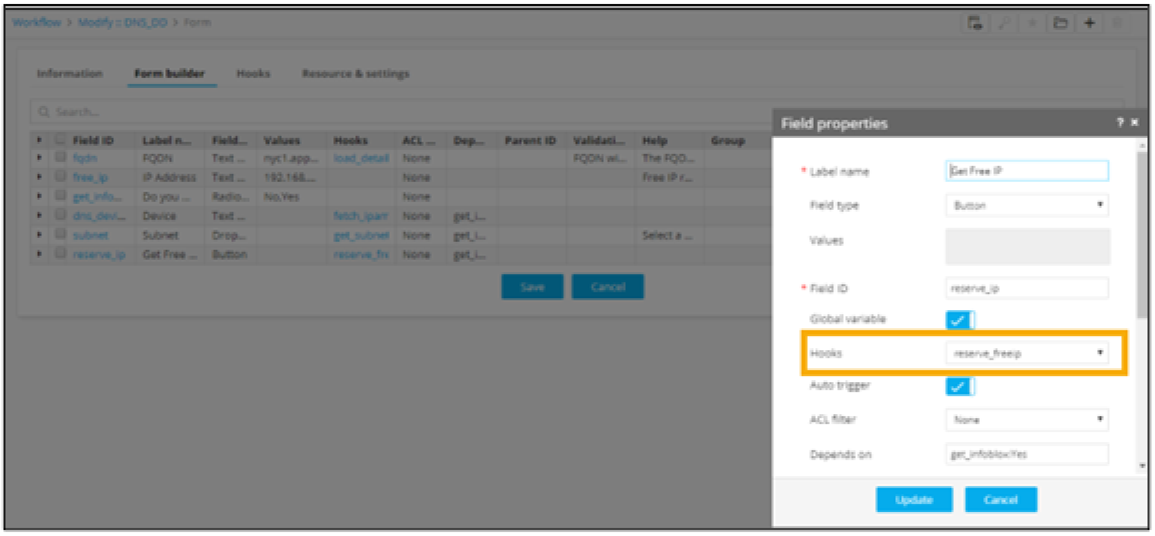


Field Type - Button

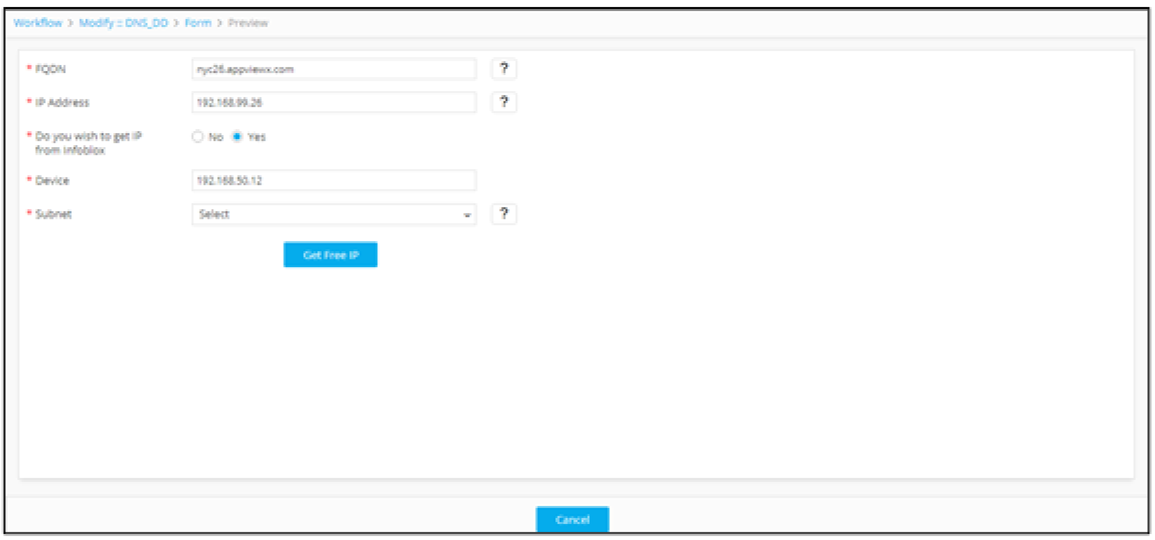
You can define custom buttons with label names using the button field. Button fields can be used to retrieve specific details from the device, database or used to perform an action by communicating to an external system via API.

A relevant associate script must be mapped against the button field for it to perform a specific operation or redirect the output to a dependent field.

1. To add a **Button** to the form, under **Form builder**, click **(Add)** icon in the command bar.
2. In the **Field properties** window, define a form field with field type as **Button** with the respective Field ID.
3. Map the relevant associate script that needs to be invoked on clicking the button.



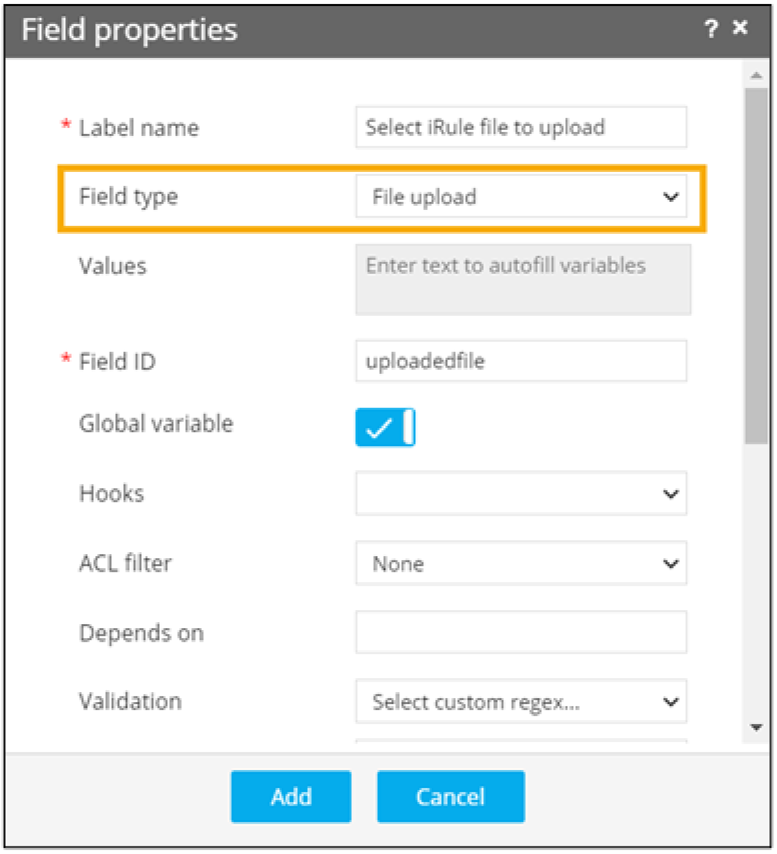
4. To see the button added to the form, from the top right corner of the screen, click **(View)** icon.



Field Type - File upload

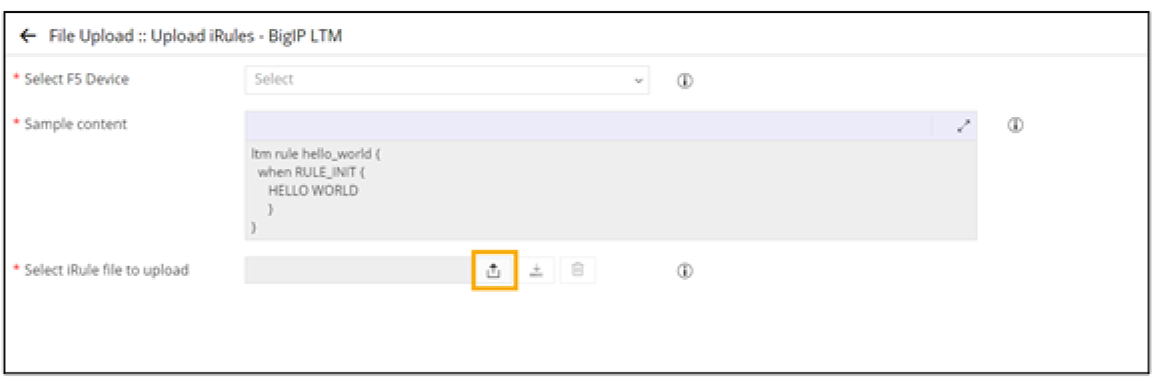
This form field allows you to add a file upload element to your form. You can upload files of maximum 2 MB in .txt, .pem, .crt, .csv formats.

1. To add a **File Upload** field to the form, under **Form builder**, click **(Add)** icon in the command bar.
2. In the **Field properties** window, enter or select the field information as shown here.



3. To add this field to the form, click **Add**.

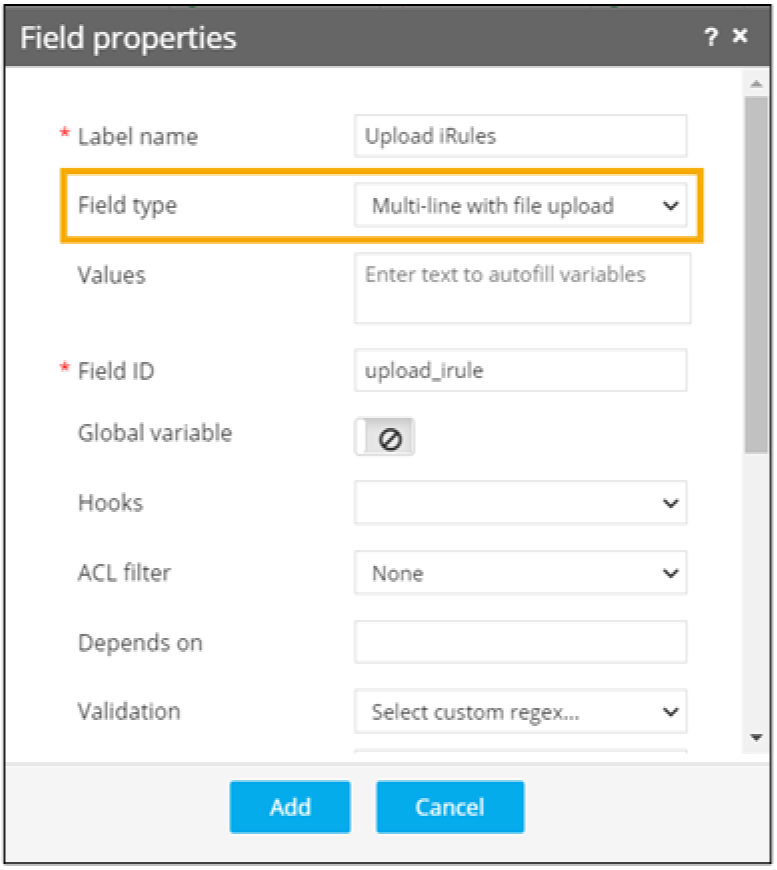
4. To see the field added to the form, from the top right corner of the screen, click **(View)** icon.



Field Type - Multi-line with file upload

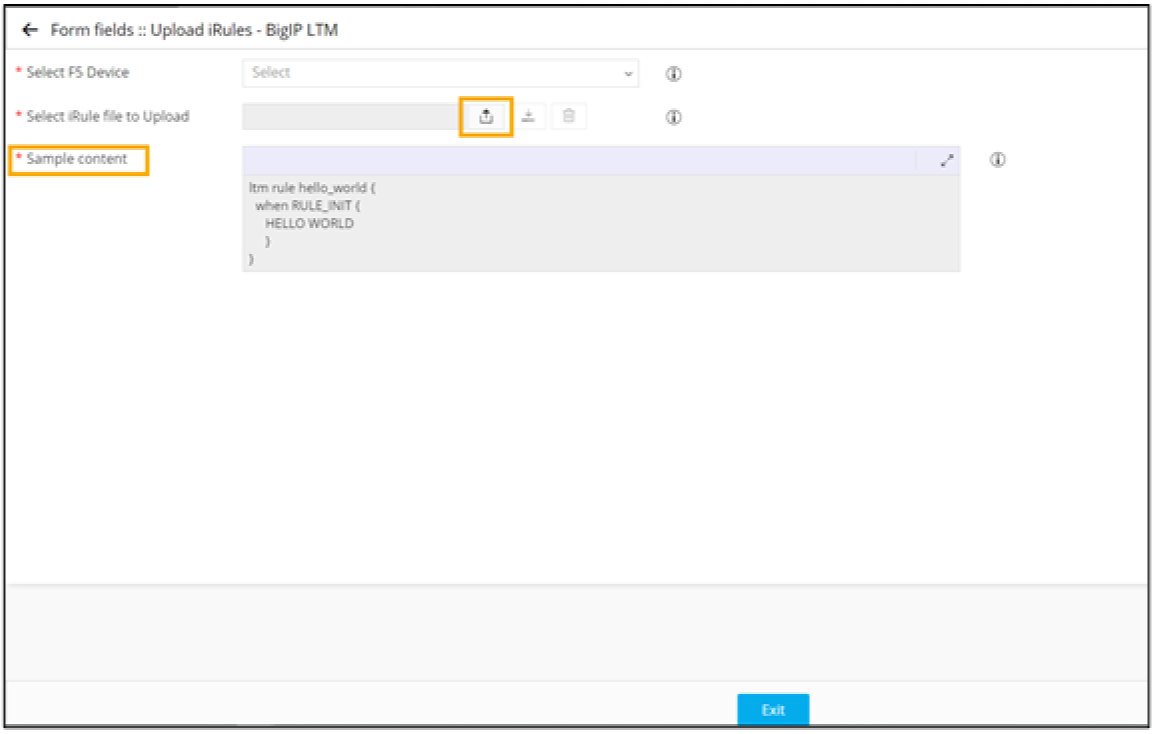
This form field allows you to use the ‘file upload’ element with a multi-line element and convert them into configurations in order to push to the end device. Used to push file(s) between destinations – local to remote, remote to remote. It supports .txt, .pem, .crt, .csv files and allows maximum file size of 2 MB.

1. To add a **Multi-line with file upload** field to the form, under **Form builder**, click + **(Add)** icon in the command bar.
2. In the **Field properties** window, enter or select the field information as shown.



3. To add this field to the form, click **Add**.

4. To see the field added to the form, from the top right corner of the screen, click **(View)** icon.

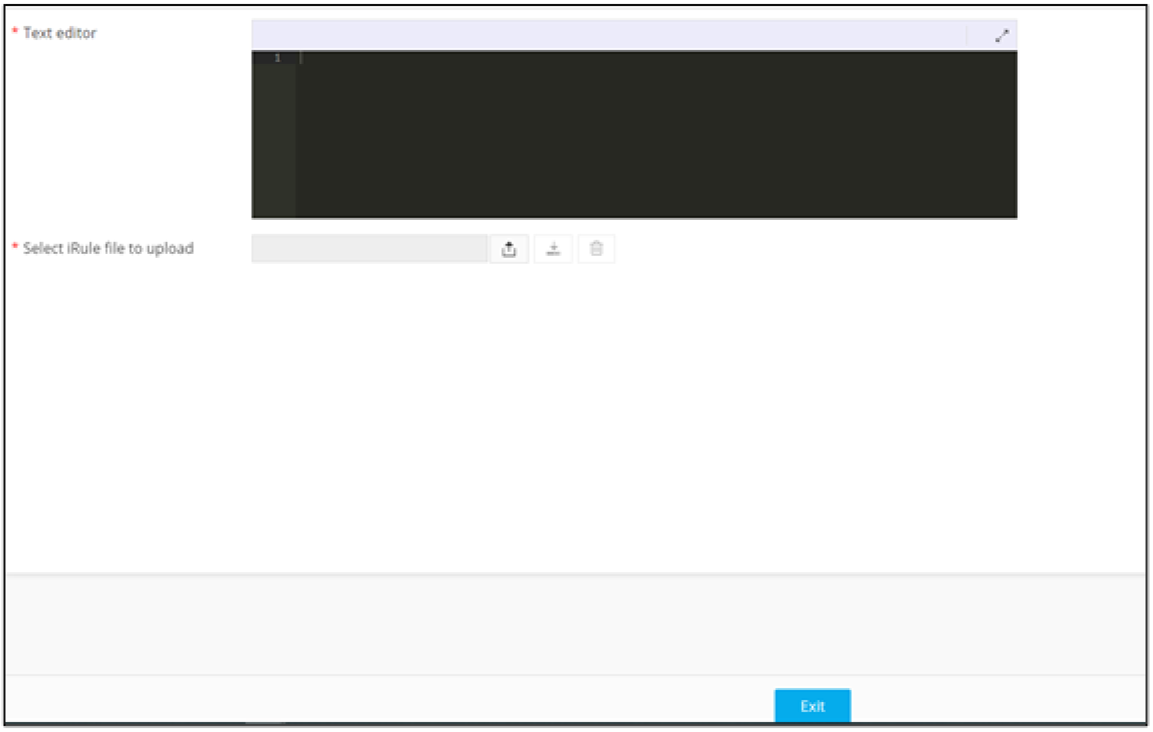


Field Type - Text editor with file upload

This form field allows you to use the file upload element with a text editor element and convert them into configurations in order to push to the end device. Used to push file(s) between destinations – local to remote, remote to remote. It supports .txt, .pem, .crt, .csv files and allows maximum file size of 2 MB.

1. To add a **Text editor with file upload** field to the form, under **Form builder**, click **(Add)** icon in the command bar.
2. In the **Field properties** window, add the **Text Editor** form field.
3. To add the form field, click **Add**.
4. Add the **File upload** form field.

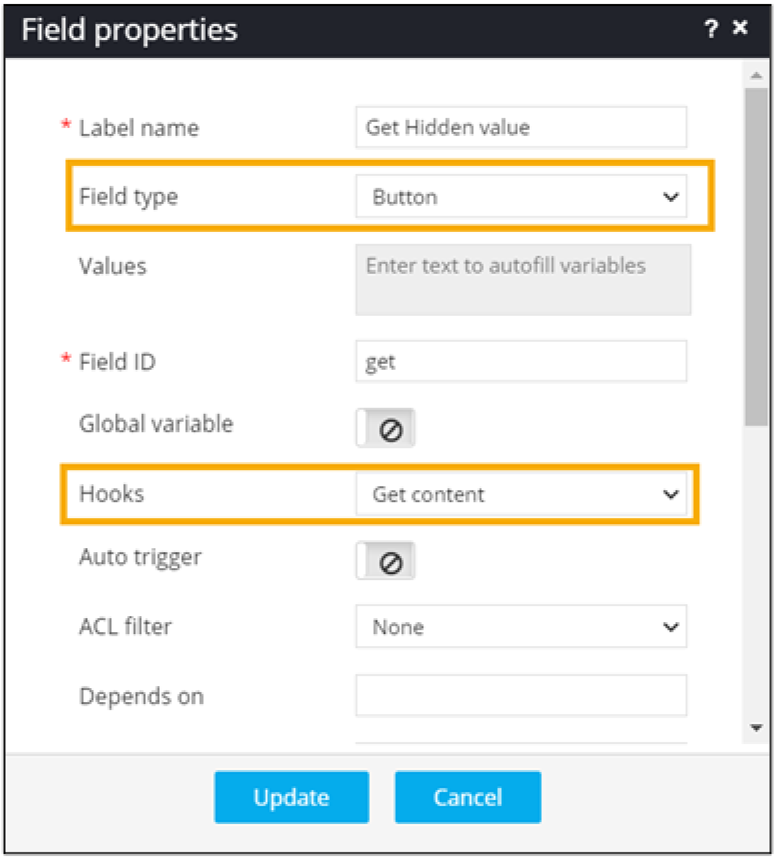
5. To see the field added to the form, click **(View)** icon.



Field Type - Hidden

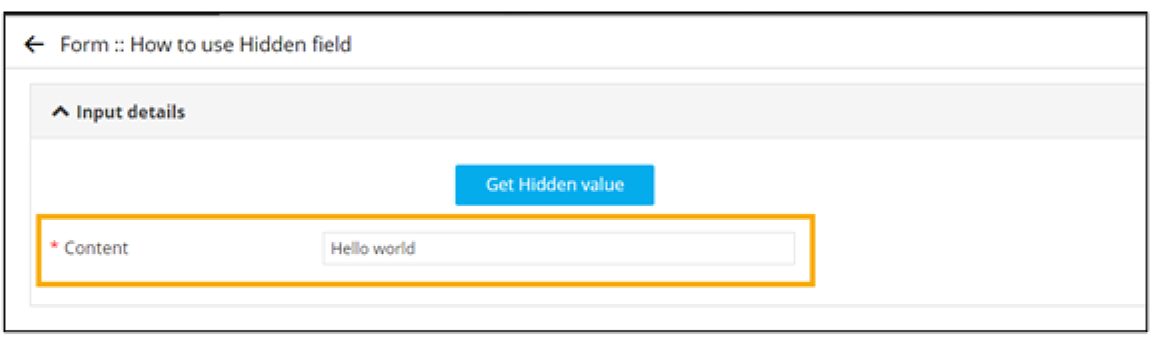
You can use the Hidden field to hide certain values or content in the form and fetch it only when required.

1. To add a **Hidden** field to the form, under **Form builder**, click **(Add)** icon in the command bar.
2. In the **Field properties** window, enter/select the field information as shown here.
3. To fetch hidden data, add the **Button** field and define the relevant hook.



4. Add a **Text box** field.
5. To see the field added to the form, click **(View)** icon.
6. In the form preview, click **Get Hidden value**.

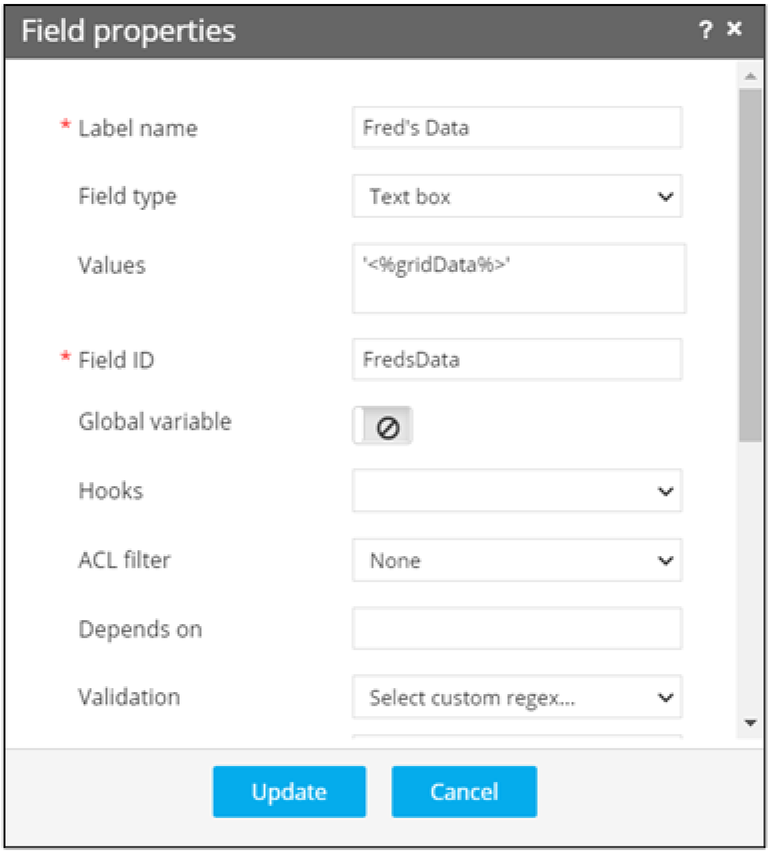
   Hidden value is displayed.



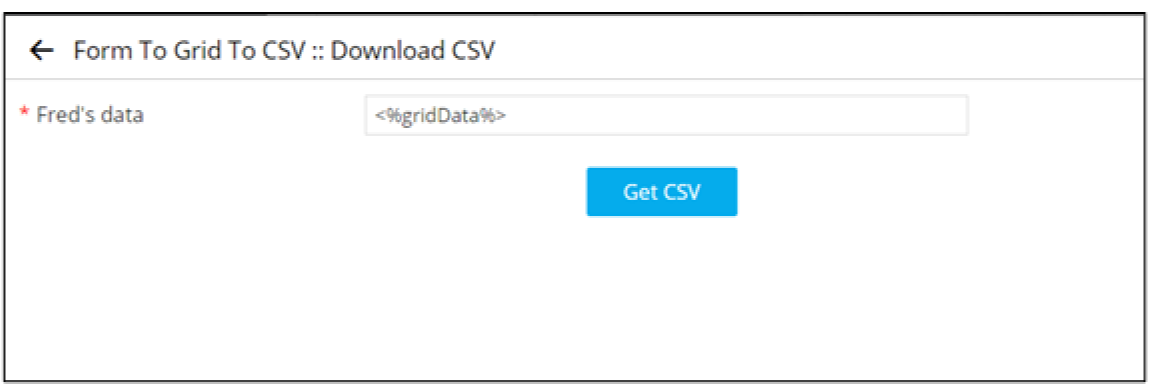
Field Type - Download CSV

This form field allows you to download the data displayed in a grid to your machine as a .csv file.

1. To add a **Download CSV** button to the form, under **Form builder**, click **(Add)** icon in the command bar.
2. In the **Field properties** window, enter or select the information as shown.



3. Add another form field (Get CSV) with field type as **Button**.
4. To see the field added to the form, click **(View)** icon.



Field Type - Download

This form field allows you to download the data from the file to your machine.

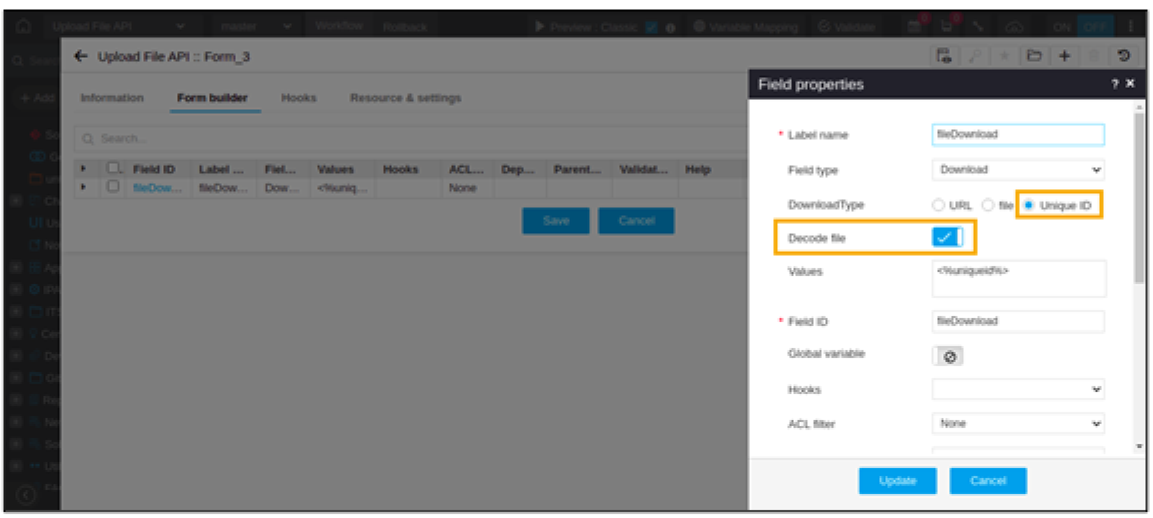
- Provision to mention the unique file ID that is used to retrieve files that are stored within the database as part of visual workflow upload file API.
- Provision to choose the Decode file option to retrieve the saved encoded file data.

1. To add a **Download** field, under **Form builder**, click **(Add)** icon in the command bar.
2. In the **Field properties** window, add a **Download** field.

   The **Download Type** field displays three options:
   - **URL**
   - **File**
   - **Unique ID**.
3. Select the **Unique ID** option to display the **Decode** file field.

> **Note:** The Decode file field is an optional field and is disabled by default.

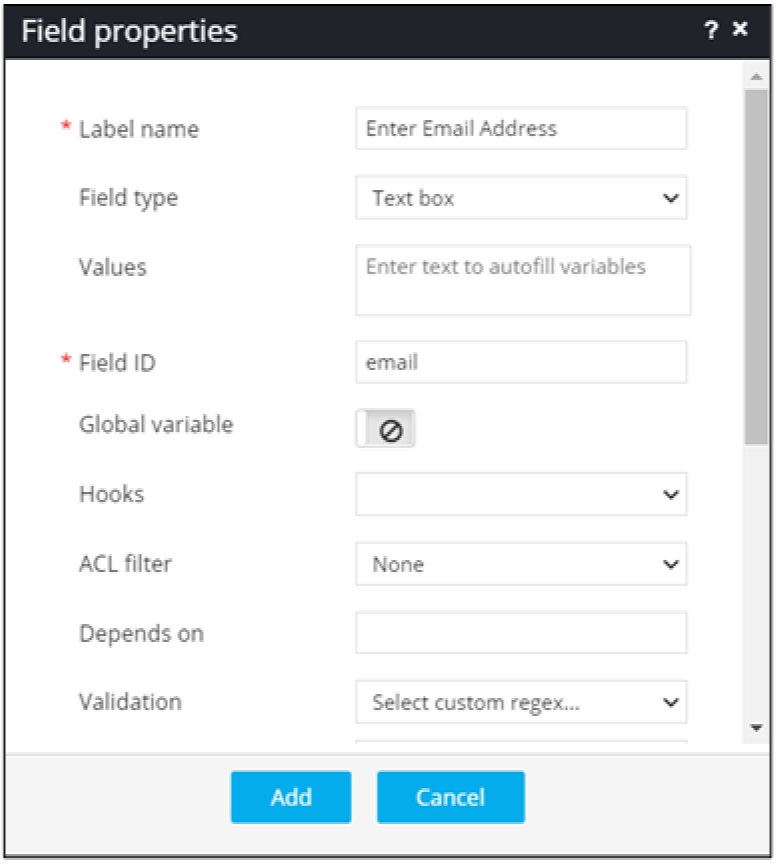
4. To enable the **Decode file** option, turn on the toggle.



Validating Form Fields

You can define custom regex validation for certain form fields. Regex validations can be referenced against a form field with specific notifications to the user. You can add or edit any regex in the inventory.

1. Design a workflow.
2. From the **User Interface** section, drag and drop a **Form** task.
3. To add a form field, in the **Form builder** section, click **(Add)** icon in the command bar.
4. In the **Field properties** window, enter or select the field information.



5. In the Validation field, select **Email**.

Note: For more information, refer to the section on Regex Library.

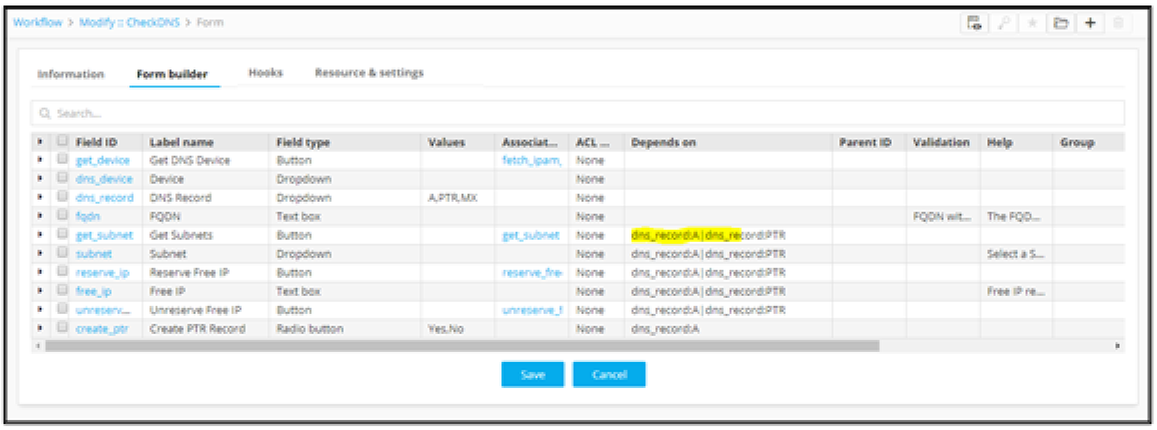
6. To add this form field to the form, click **Add**.
7. To see the field added to the form, from the top right corner of the screen, click **(View)** icon. Only values that match the defined regex are accepted.

Note: A is displayed next to the form field if the entered value does not match the defined regex.

Defining Dependencies between Form Fields

In order to make the request form dynamic, the form builder allows you to define dependencies between multiple form fields using conditional operators. This can be achieved using the ‘Depends On’ field through a combination of ‘Field ID’ and ‘Values’ parameters in tandem with either the ‘logical AND’ (&); and ‘logical OR’ (|) operators.

- Operators supported: ‘&’ (AND); and ‘|’ (OR)
- Defining dependency based on single field and value: [Field ID: Value]
- Defining dependency based on multiple fields and values: [Field ID1: Value1 & Field ID2: Value1], [Field ID1: Value1 | Field ID2: Value1]

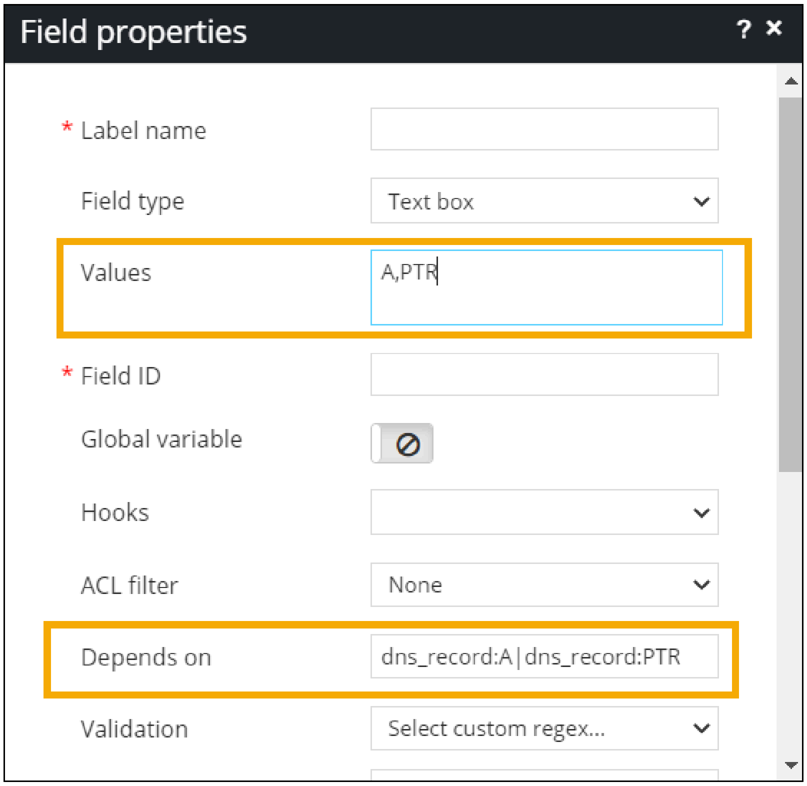


To be able to display the Create PTR Record field based on the type of selection of the DNS Record:

1. Design a workflow.
2. From the **User Interface** section, drag and drop a **Form** task.
3. To add form field(s), in the **Form builder** section, click + **(Add)** icon in the command bar.
4. In the **Field properties** window, define a field called DNS Record with a dropdown that shows two values to select from - A and PTR.
5. Define the parameters for the conditional form field - Create PTR Record based on which this field must be shown.

   dns_record:A|dns_record:PTR

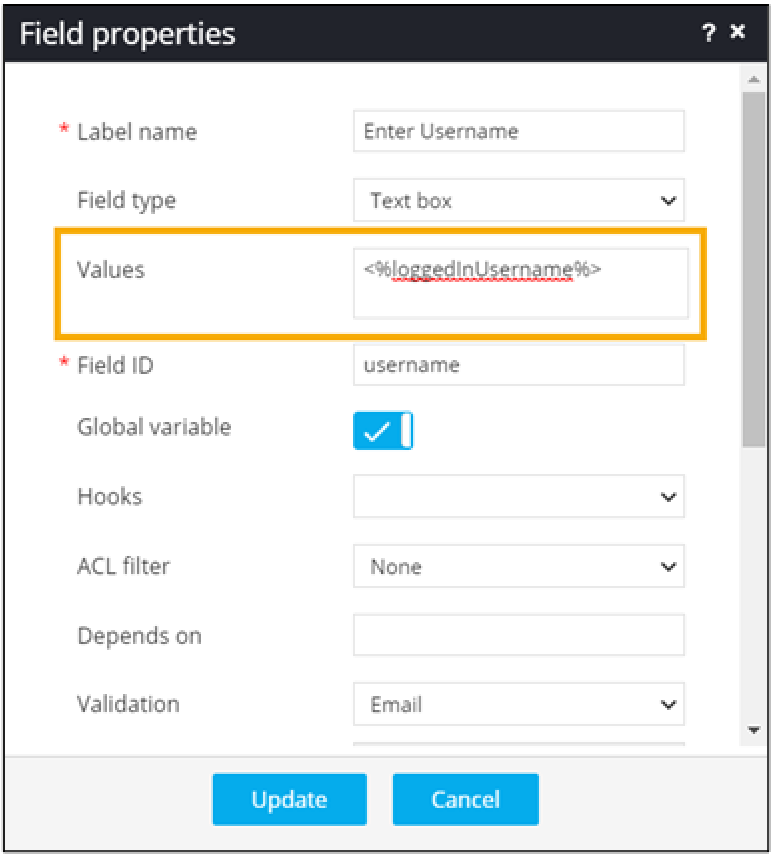
   The field Get Subnets will be displayed according to the selection of the DNS record as A or PTR.



6. Add other form fields and define dependencies as required.
7. To see the form fields added to the form, from the top right corner of the screen, click **(View)** icon.

Defining Tasks as Read Only

1. Design a workflow.
2. From the **User Interface** section, drag and drop a **Form** task.
3. To add form fields, click **(Add)** icon in the command bar.
4. In the **Field properties** window, enter or select the field information to define the form field.

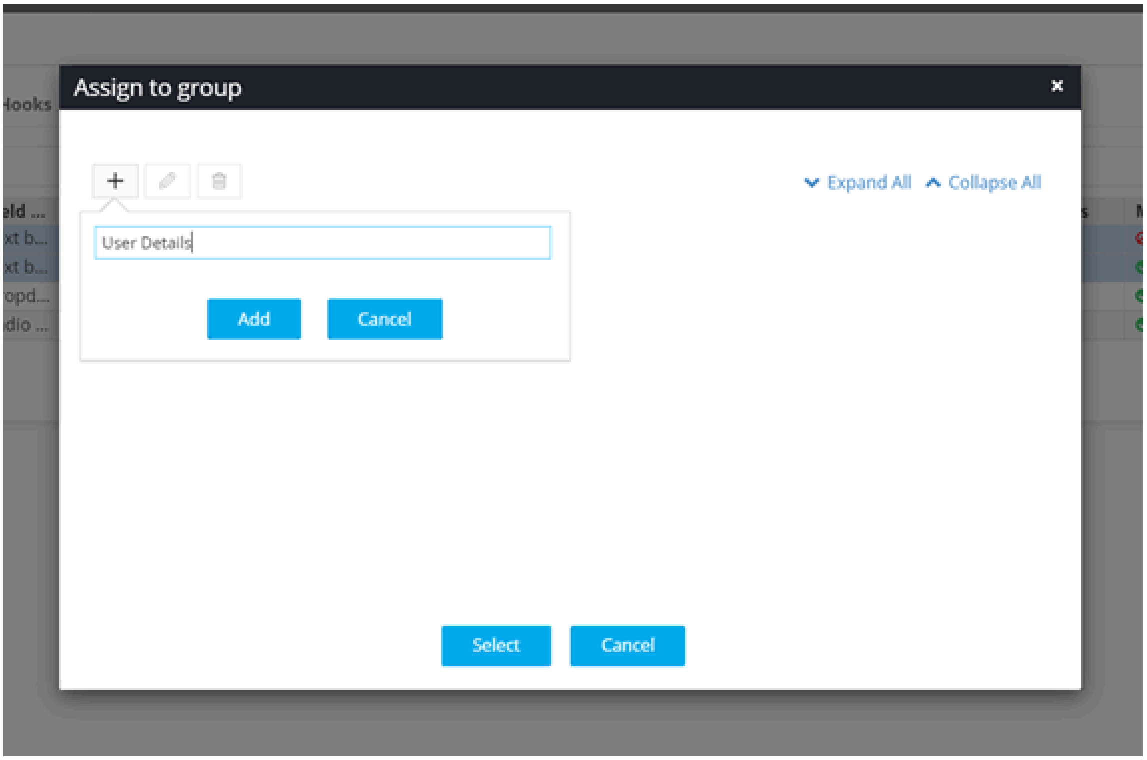


5. To define the field as **Read only**, turn on the toggle.
6. To see the Read only username field added to the form, from the top right corner of the screen, click **(View)** icon.

Grouping Multiple Form Fields

You can group multiple fields logically within the form.

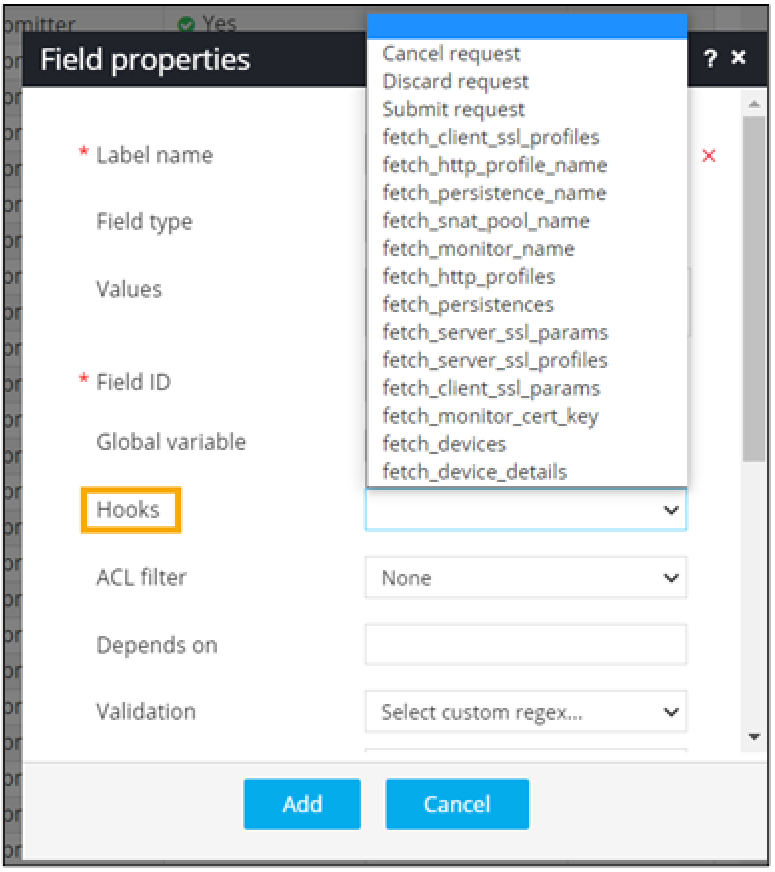
1. Design a workflow.
2. From the **User Interface** section, drag and drop a **Form** task.
3. Under **Form builder**, select the form fields to be grouped.
4. To group the selected form fields, from the top right corner of the screen, click **(Folder)** icon.
5. To create a new group, in the **Assign to group** window, click **(Add)** icon and enter the group name.



6. To see the form fields listed under the assigned group, from the top right corner of the scree, click **(View)** icon.

Using Hooks in a Form

You can fetch values dynamically from a data source such as a database, device or external systems by using hooks. You can either select hooks from the existing list or create a new hook.



Hooks are of three types:

- **Script**: The form builder allows you to define one or more ‘associate scripts’ and map them against the relevant form field. The associate script acts as an abstract layer which enables linking of the data retrieved from the query and maps them against appropriate fields within the form.
- **REST API** (Internal & External API): REST API allows querying the database, device or external systems
- **Query Explorer**.

- Using Scripts with Forms
- Auto-trigger Scripts
- Default Script for Cancel and Discard Actions
- Define Custom Notification Messages using Scripts
- Defining Query Explorer to Build Forms

Using Scripts with Forms

You can define scripts within a form and leverage them to retrieve values and populate form fields or print output.

- Triggering Script with Button
- Triggering Script to Get Device List
- Triggering Script to Get Device List by Vendor
- Triggering a Script to Print Custom Message
- Using Auto-trigger to Invoke the Script Automatically based on Input
- Triggering a Script Directly from a Form Field to Map its Values to Destination Fields

Triggering Script with Button

To associate a script to a ‘button’ element, trigger and retrieve values and populate them to a form field:

1. Design a form using the following form fields:

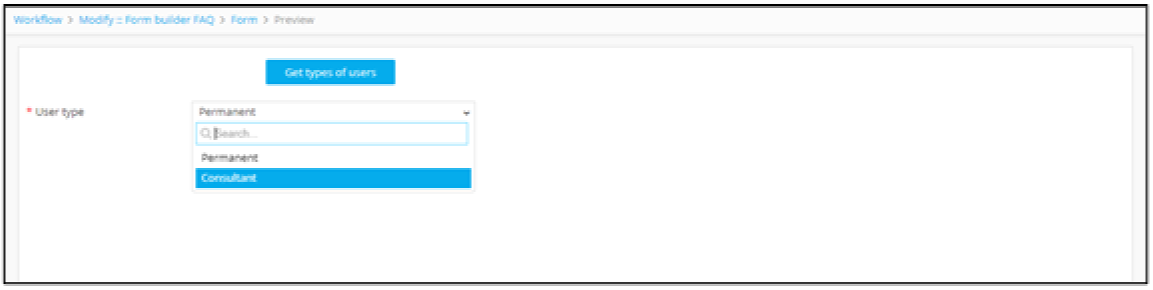
Field Name	Field ID	Type
Get type of users	get	Button
User type	type	Dropdown

2. Define an associate script to get the **User type** and map the script to field **Get types of users**.

Script	Description
`sers=[{“type”:”Permanent”},` `{“type”:”Consultant”}]print(json.dumps(users))`	Where **type** is the Field ID to which the retrieved values are appended to on the form.

3. From the menu on the top, click **(View)** icon.

Clicking the **Get type of users** button triggers the script and populated the **User type** field.



Triggering Script to Get Device List

To query the database by defining an associate script and get a list of F5 devices:

1. Design a form using the following form fields:

Field Name	Field ID	Type
Get device list	get	Button
Select the device	device	Dropdown

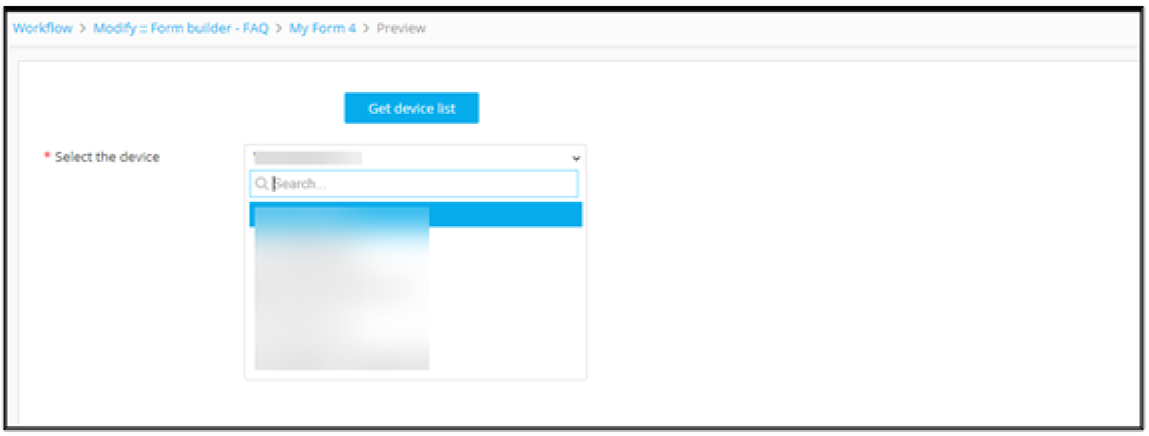
2. Define an associate script to get the **devices** and map the script to field **Get**.

Script	Description
### To make use of the helper script ### *sys.path.insert(0,AVX::HELPER)* *sys.path.insert(0,AVX::DEPENDENCIES)* ### Importing the appviewx helper ### import appviewx ### calling the inbuilt function the helper script to connect with the DB ### `db_connection = appviewx.db_connection() device_collection = db_connection.appviewx.device`	Where **device** is the Field ID to which the retrieved devices are mapped to on the form

Script	Description

###Creating a list of dictionaries, where each dictionary will have the key as 'device'. Where device is the form field id to which we need to populate the values###
`device_list = [{"device":value['name']} for value in device_collection.find({"vendor":"F5","category":"ADC"})]`
### Giving an output as part of completion of the script ### 	
`print(json.dumps(device_list))`	

3. From the menu on the top, click **(View)** icon.



Clicking the **Get device list** button triggers the script and populates the **Select the device** field.

Triggering Script to Get Device List by Vendor

To query the database by defining an associate script and get a list of devices by vendor:

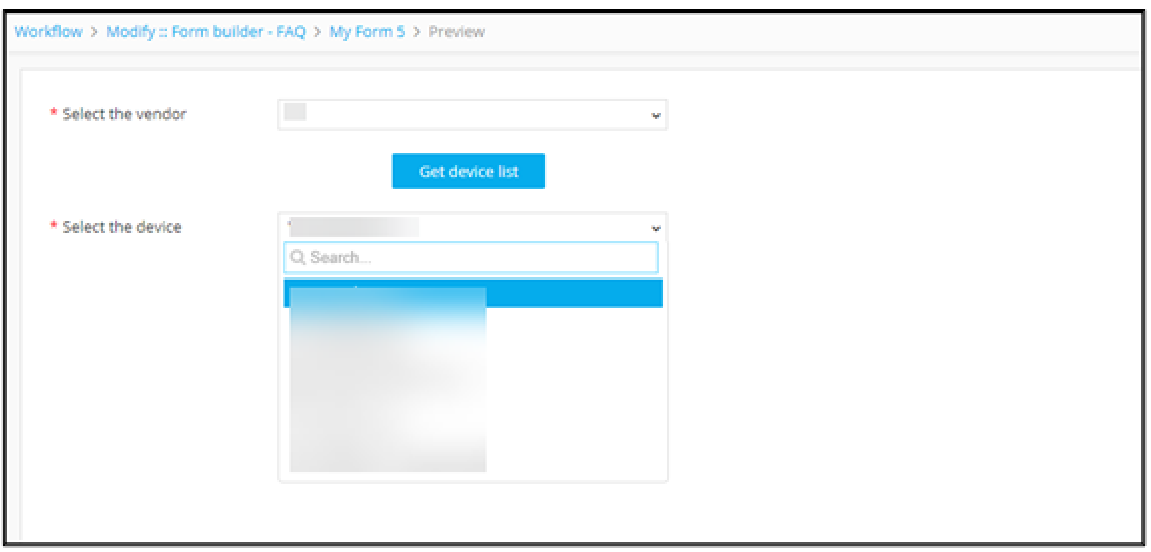
1. Design a form using the following form fields:

Field Name	Field ID	Type
Select vendor	type	Dropdown
Get device list	get	Button
Select the device	device	Dropdown

2. Define an associate script to get the **devices** and map the script to field **Get**.

Script	Description
sys.path.insert(0,AVX::HELPER) sys.path.insert(0,AVX::DEPENDENCIES) import appviewx db_connection = appviewx.db_connection() device_collection = db_connection.appviewx.device ***###Syntax to read any of the other form fields in the associate script -- '<field id>'##*** vendor = '<type>' device_list = [{"device":value['name']} for value in device_collection.find ({"vendor":vendor,"category":"ADC"})] print(json.dumps(device_list))	• Where type is input value from the form based on which the device(s) will be retrieved • device is the destination Field ID to which the list of devices are mapped back on the form.

3. From the menu on the top, click **(View)** icon.
   The script is triggered to get the device list by vendor.

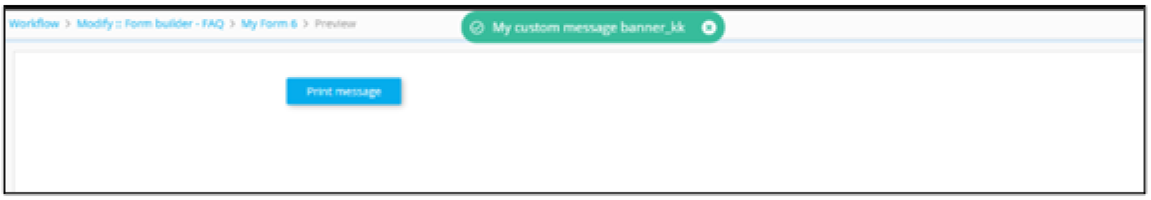


Triggering a Script to Print Custom Message

To print a custom banner message by defining an associate script:

1. Define a form field with Field Name as Print Message, Field ID as print, and Type as Button.
2. Define associate Script and map it to the field ID. The following syntax can be used to print the message by adding the string to the key "message": `print(json.dumps({"message":"My custom message banner"}))`
3. From the menu on the top, click **(View)** icon.

   Clicking the **Print message** button triggers the script to display the custom message.

Using Auto-trigger to Invoke the Script Automatically based on Input

To be able to retrieve the IP address and LB method for a selected device:

1. Design a form using the following form fields:

Field Name	Field ID	Type
Select device	device	Dropdown
Enter Virtual Name	name	Text Box
Get Virtual Details	get_virt	Button

2. Define an associate script and script logic.
3. Map the associate script against the relevant form field ID.

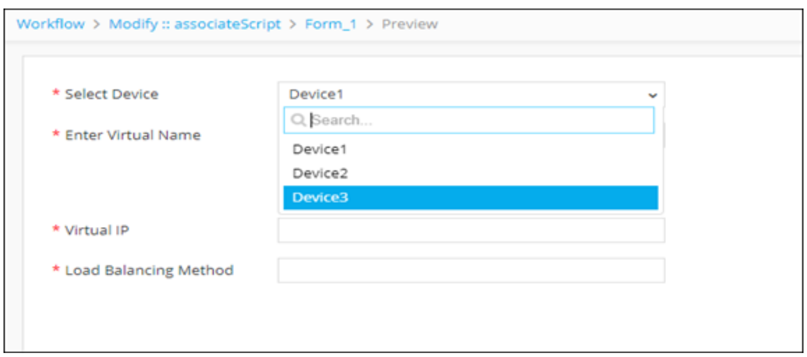
   Associate Script logic

```
import sys
import json
device = '<device>'---------------------------------------------------------------------------------Input field
used to query
virtual_name = '<virtual_name>'
def get_virtual_details():
    if device == 'Device1':
        ip = '192.168.41.237'
        lb_method = 'ratio'--------------------------------------------------------------------------Logic to get IP
and lb method for a device
    if device == 'Device2':
        ip = '192.168.41.199'
        lb_method = 'round-robin'
        print(json.dumps([{'virtual_ip':ip},{'lb_method':lb_method}])) -----------------Return values
mapped to the destination field
get_virtual_details()
```

4. Define two more form fields to show the Virtual IP and Load balancing method.

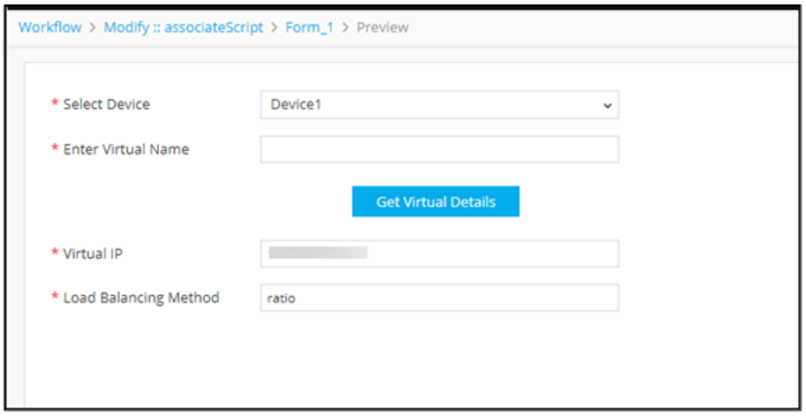
Field Name	Field ID	Type
Virtual IP	virtual_ip	Text Box
Load Balancing Method	lb_method	Text Box

5. From the menu on the top, click **(View)** icon.
6. In the **Select Device** field, select the device as **Device1**.



7. Enter the virtual name and click the **Get Virtual Details** button.

The response from the query is mapped to the destination fields: Virtual IP and Load Balancing Method.



Triggering a Script Directly from a Form Field to Map its Values to Destination Fields

To map destination field values by triggering a script from form fields:

1. Design a form using the following form fields:

Field Name	Field ID	Type
Get F5 LTM Device List	DeviceList	Button
F5 LTM Device	Device	Dropdown

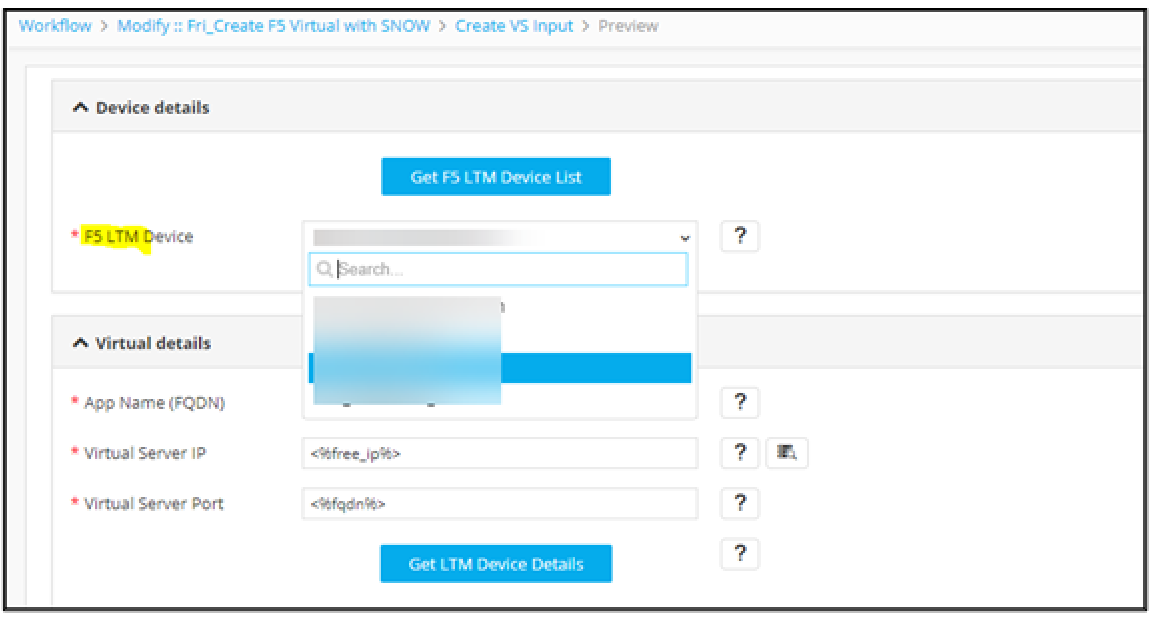
2. Define custom script logic. Use ‘Ctrl + Space’ in the script editor to list all the pre-defined helper scripts and embed the appropriate helper script(s) within an ‘associate script’.
Associate Script

Associate Script (Get device List)	Mapping Details

*import sys
import json
sys.path.insert(0, AVX::DEPENDENCIES)
sys.path.insert(0, AVX::HELPER)
import appviewx
# reload(appviewx) 	
def device_list():
    connection = appviewx.db_connection()
    collection = connection.appviewx.device
    dev_list = [value['name'] for value in collection.find({"vendor":"F5", "category" : "ADC"})]
    dev_list = sorted(filter(None,dev_list), key = lambda s: s.lower())
    print (json.dumps([{'Device': name} for name in dev_list]) if dev_list else (json.dumps({'error':"No Device to list"}))
device_list()*
	• The script queries the database and retrieves the list of F5 devices
	• Field marked in yellow is the destination field on the form where the queried values from the script are mapped to

3. From the menu on the top, click **(View)** icon.
4. To execute the associate script on the form, click **Get F5 LTM Device List**.

   The **F5 LTM Device** field is populated with the device names based on the script.



Auto-trigger Scripts

Associate scripts in the form can be triggered either manually or automatically. Auto trigger eliminates manual intervention in the form. The input from the form is used to auto trigger an associate script in the backend.

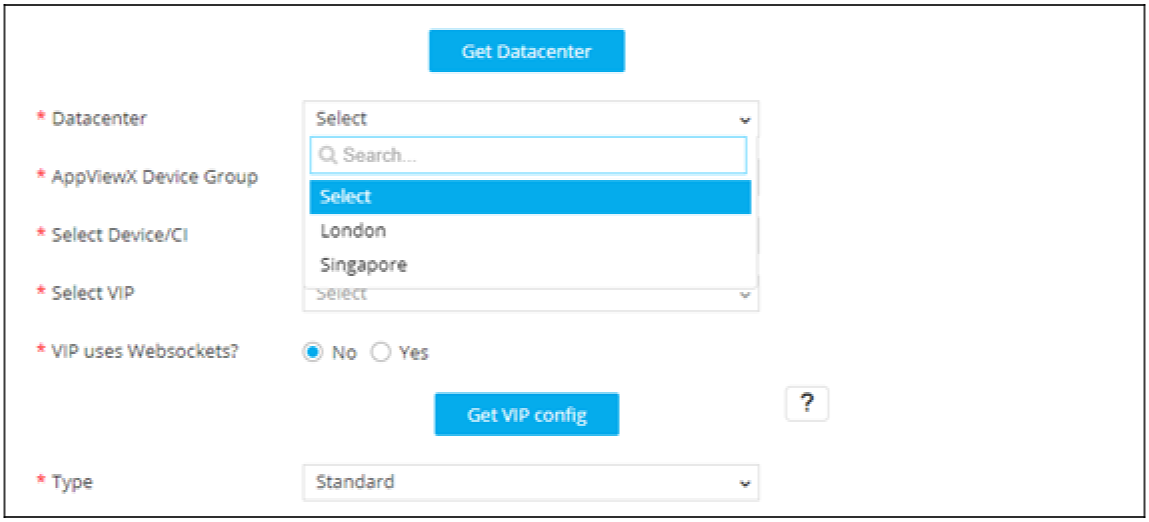
You can choose to enable the auto trigger option against every form field. The scope of auto trigger is limited to the following:

- Auto trigger works for the first form field on loading.
- The input from the form must be from a radio button or dropdown form field.
- Based on the input, auto trigger gets executed ONLY for the next immediate field.

1. Design a form using the following fields and associate scripts:

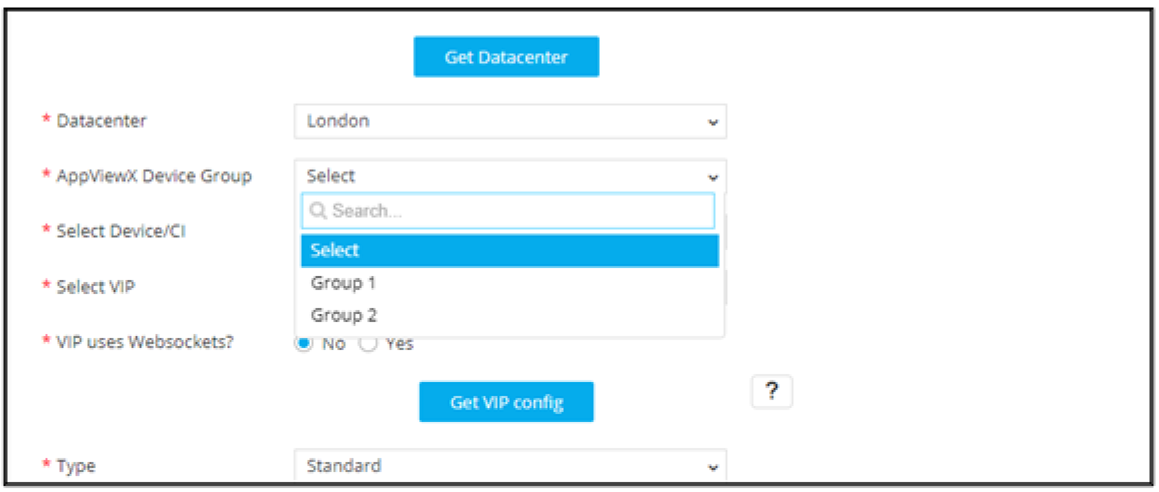
Field Name	Form Input/Response	Autotrigger script to execute
Get Datacenter	Get Datacenter - Button	getdataCenterlist
Datacenter	Response: Display list of datacenters in dropdown	getDeviceGrouplist
AppViewX Device Group	Response: Display List of Device Groups in dropdown	getDevicelist
Select Device/CI	ResponseL Display list of devices in dropdown	getVIPNames
Select VIP	Response: Display List of VIP Names for the device in dropdown	

2. To preview the form, from the menu on the top, click **(View)** icon.
3. In the form preview, click **Get Datacenter**.
   The script is auto triggered to populate the dropdown in the next (Datacenter) field.



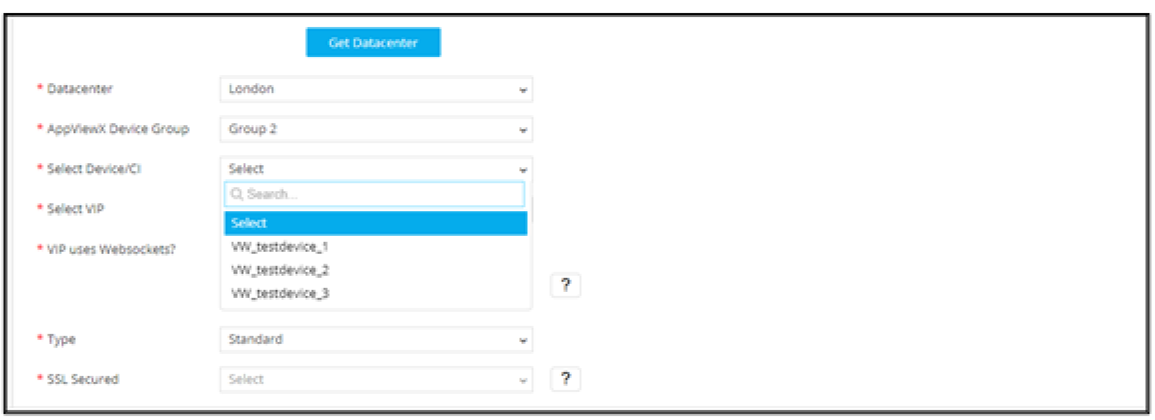
4. From the options displayed in the **Datacenter** dropdown list,select London from the dropdown list.

   The script is auto triggered to populate the next (device group) field.

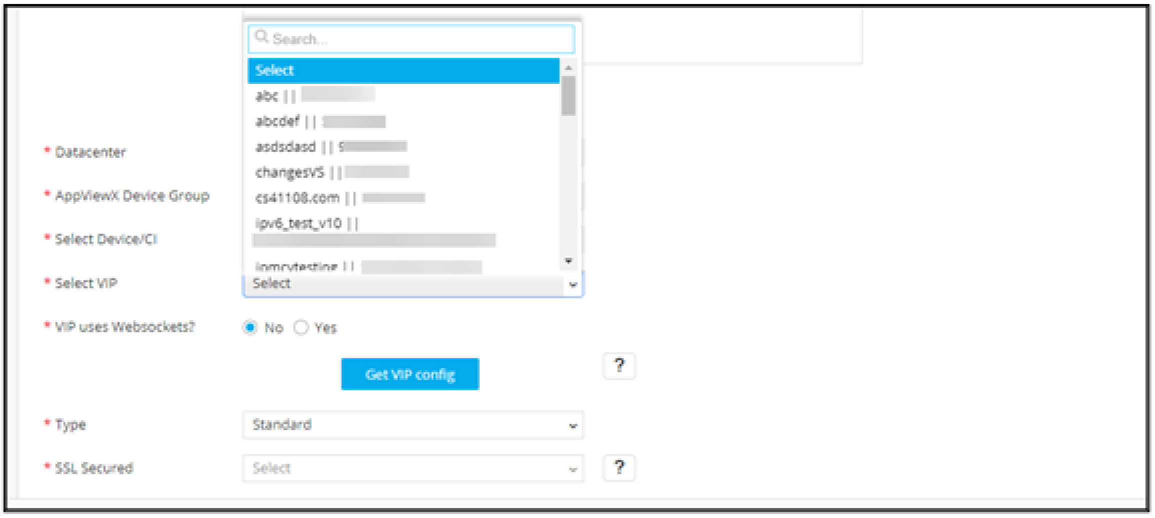


5. Select **Device group** as Group 2.

   The script is auto triggered to populate the next (select device) field.



6. Select the **Device** to auto trigger the script and populate the next (select VIP) field.



Default Script for Cancel and Discard Actions

The request form allows you to either ‘Cancel’ or ‘Discard’ a particular request prior to final submission. Such an event may result in partial transactions which may require a ‘clean-up’.

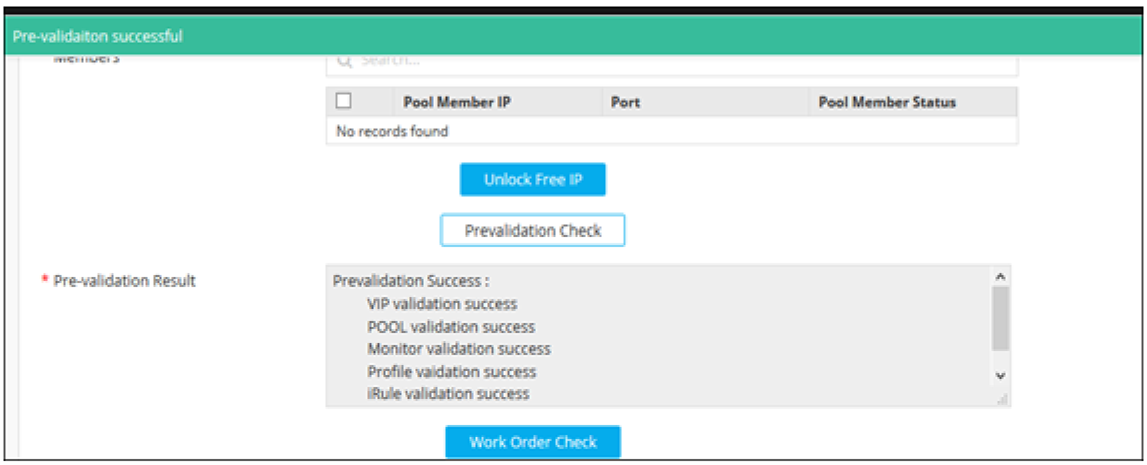
In order to cater to such operations, default associate scripts for cancel and discard actions are provided within the associate script inventory. You can define a custom logic within the default scripts which will be triggered based on the user’s action on the form.

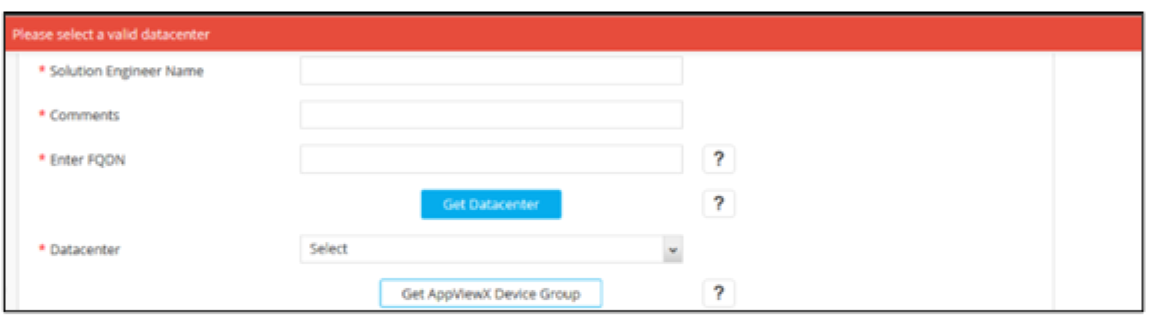
For example, As part of a Create DNS workflow, a user gets a Free IP, reserves it, and then immediately wants to cancel the request transaction. In this case, the reservation of IP and any other transactions will need to be ‘cleaned up’. The Cancel script Logic releases any temporarily locked free IP and releases any reservation on the IPAM in the event of a cancel or discard action on the form.

Note: Default associate scripts for ‘Cancel’ and ‘Discard’ cannot be deleted.

Define Custom Notification Messages using Scripts

For more personalized notifications, you can define custom scripts to display custom notification messages on the request form. Notification messages can be displayed in either ‘red’ or ‘green’ tickers based on the response type - message or error.



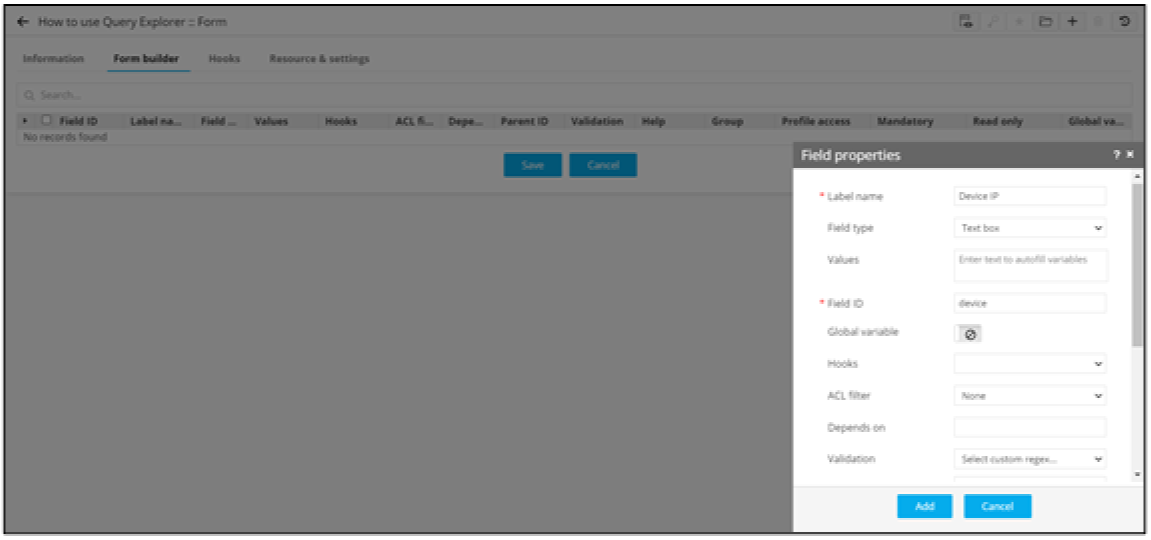


Defining Query Explorer to Build Forms

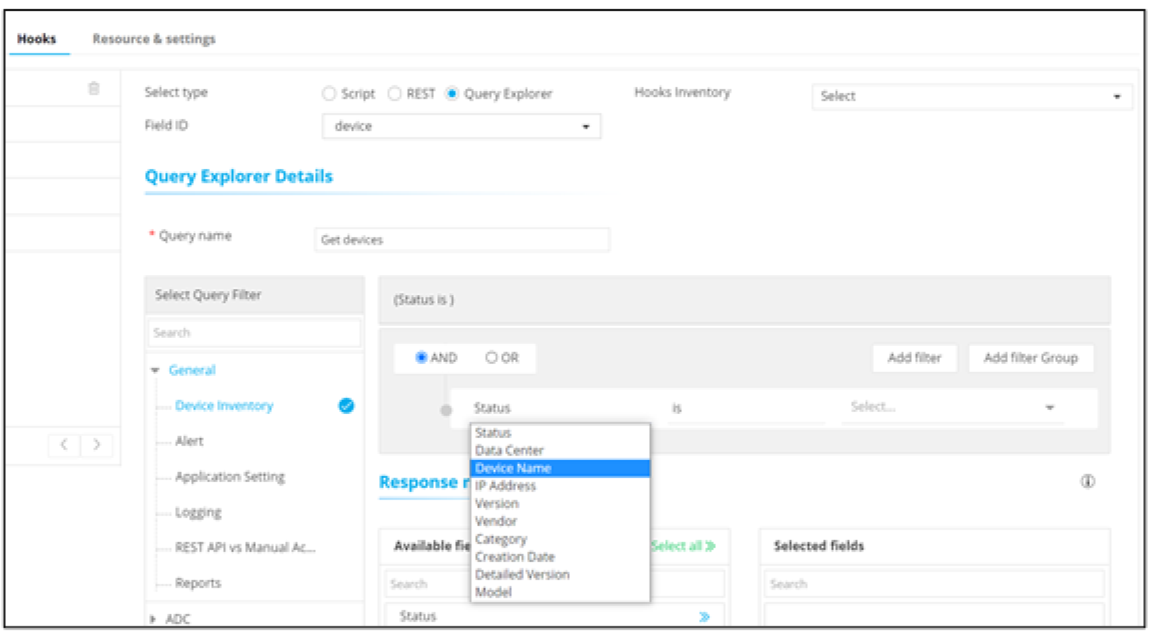
Query explorer allows you to design custom queries, business logic, and build self-service forms quickly.

To design a query to ‘get device details’ using query explorer and build forms quickly:

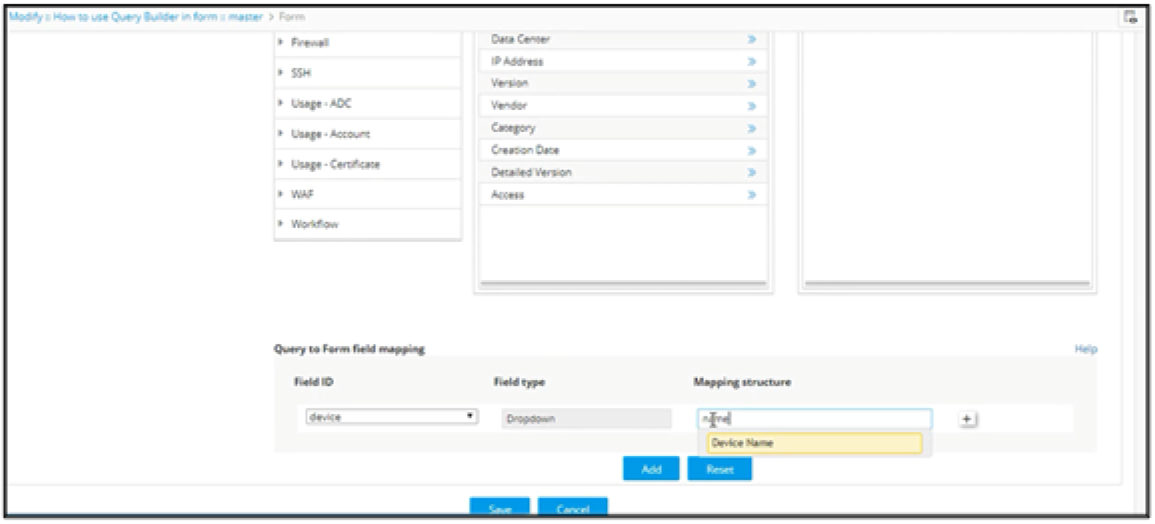
1. Design/Modify a custom workflow.
2. From the **User Interface** section, drag and drop the **Form** task.
3. Under **Form builder**, define the form field to fetch specific details from the database.



4. Under **Hooks**, define the required Hook using Query explorer.
5. Select the required response from the query to be mapped to the form field.



6. Map the query response to the specific form field.

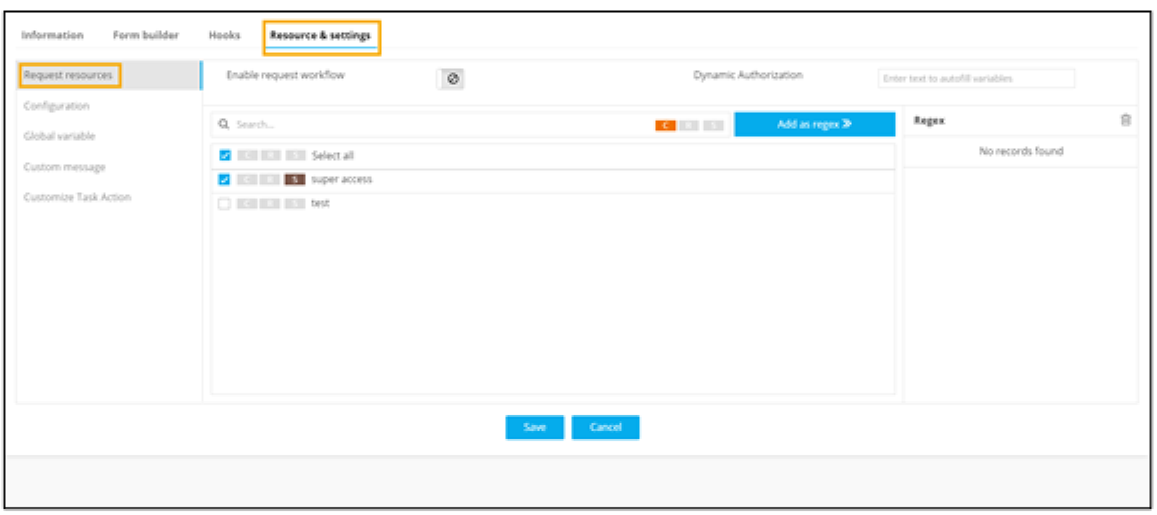


7. To validate the execution, click **(View)** icon.

Defining Resource & Settings

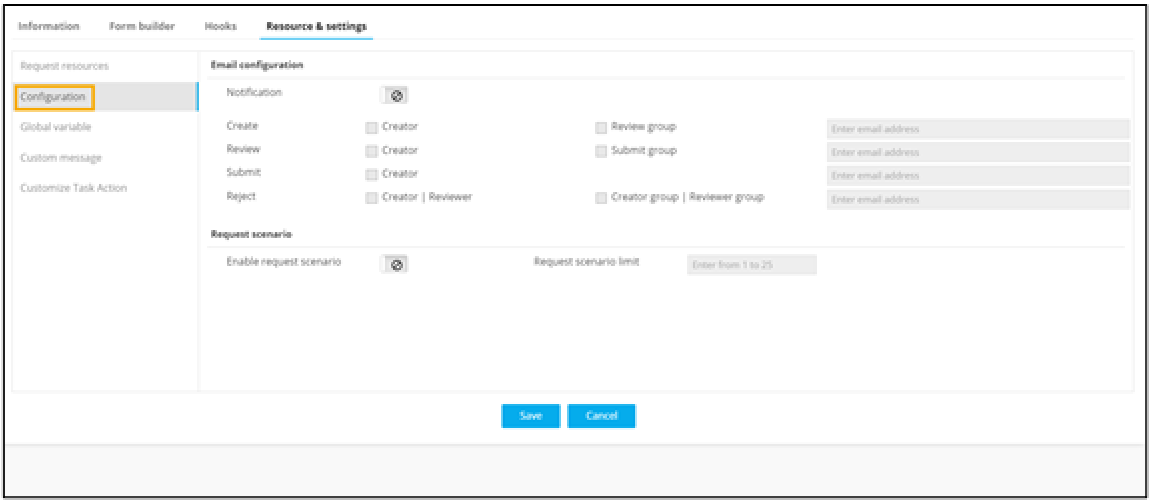
This section supports the following:

- **Request resources**: Provision to assign role based access to the form task.

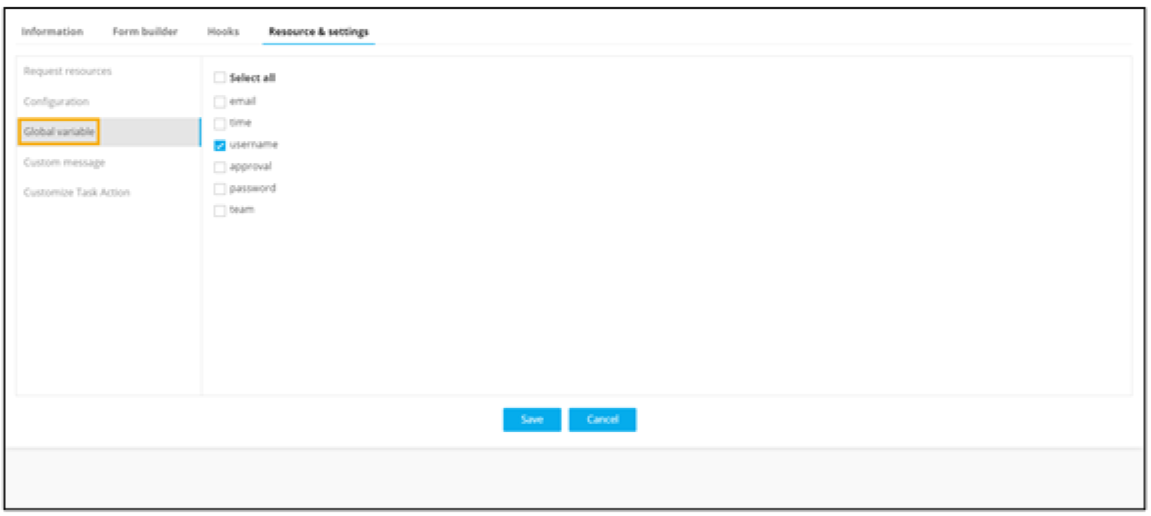


When trying to add the regex "*", a validation error will be triggered, resulting in the disabling of the "Add as regex" option.

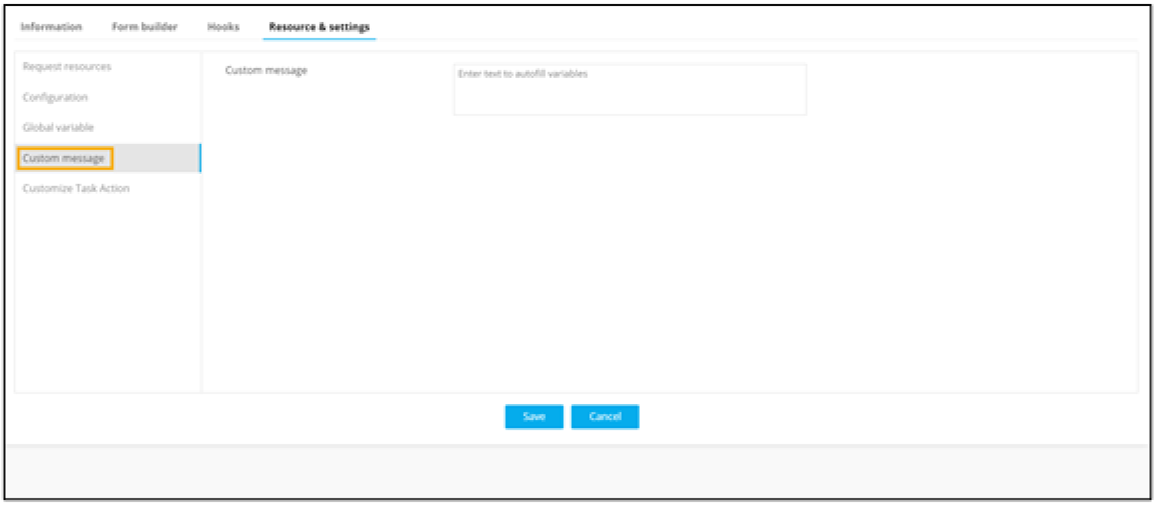
- **Configuration**: Provision to enable multiple approvals of the form task.



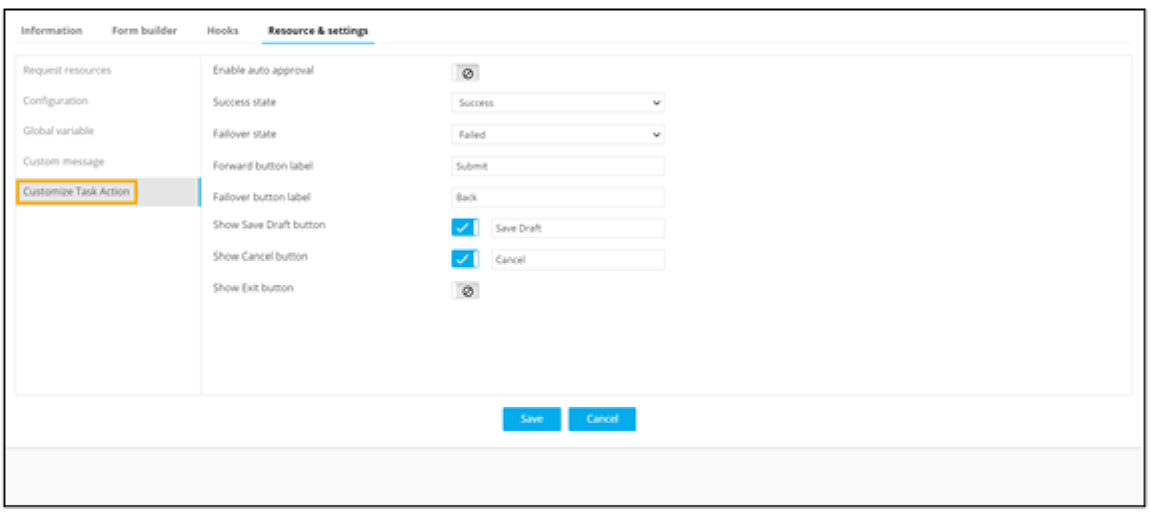
• **Global variable**: Provision to declare one or more form field values as global variables.



• **Custom Message**: Provision to define context based messages for notifications between workflow stages. For example, User access denied.



- **Customize Task Action**: Provision to enable auto approval of the form task, provide an 'alias' of the form actions. For example, the 'Submit' button can be renamed to 'Next', 'Back' button can be renamed to 'Cancel'.

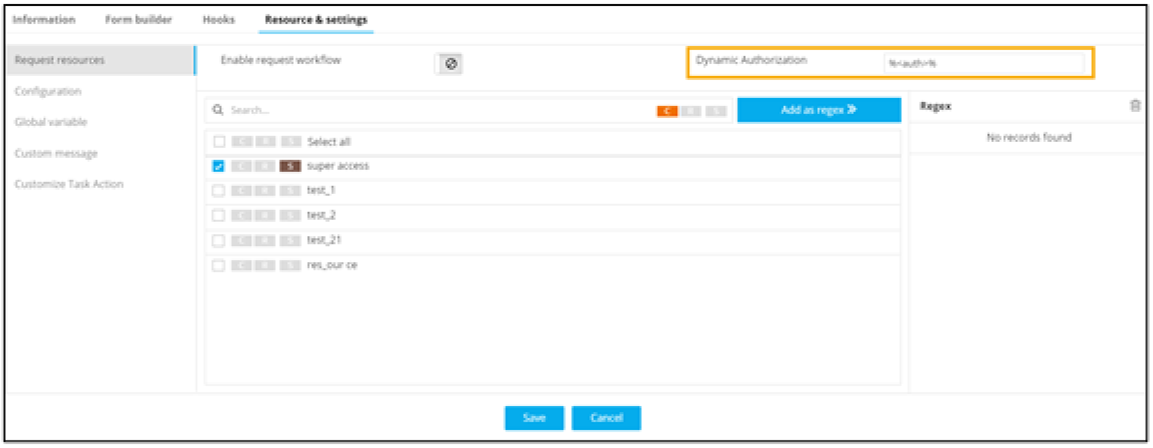


- **Dynamic Authorization**: Provision to enable authorization dynamically for users or user groups by using the global variable `%<auth>%`.

  New users/user groups must be defined in the script logic using the following structure:

```
[{"type":"user","value":["admin","tim"],"access":"R"},
{"type":"role","value":["admin","network-sre"],"access":"RW"},
```

{"type":"usergroup","value":["admin usergroup"],"access":"RW"}]



Note: Dynamic Authorization is not limited to the Form task. It can be enabled for all User Interface tasks.

Grid

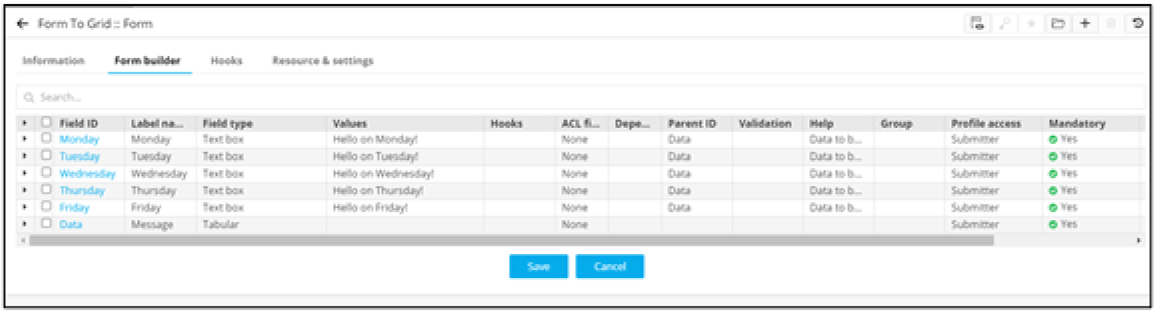
This task allows you to define a custom ‘grid’ component as part of the workflow execution process. The Grid component can be used either to display some data or to gather input from users.

- Provision to define a custom grid component with number of columns.
- Provision to define custom column/label names.
- Provision to dynamically pass data using variables from any stage of the workflow into the grid task.
- Provision to assign user role RBAC access.
- Provision to define custom button/label names in the event of using the grid task as part of any approval process.
- Provision to download the contents from Grid into <.xls> or <.csv> files.
- Provision to dynamically define column/label names from any stage of the workflow into the grid task.
- Provision to define custom messages.
- Provision to declare global variables.

- Downloading the Grid

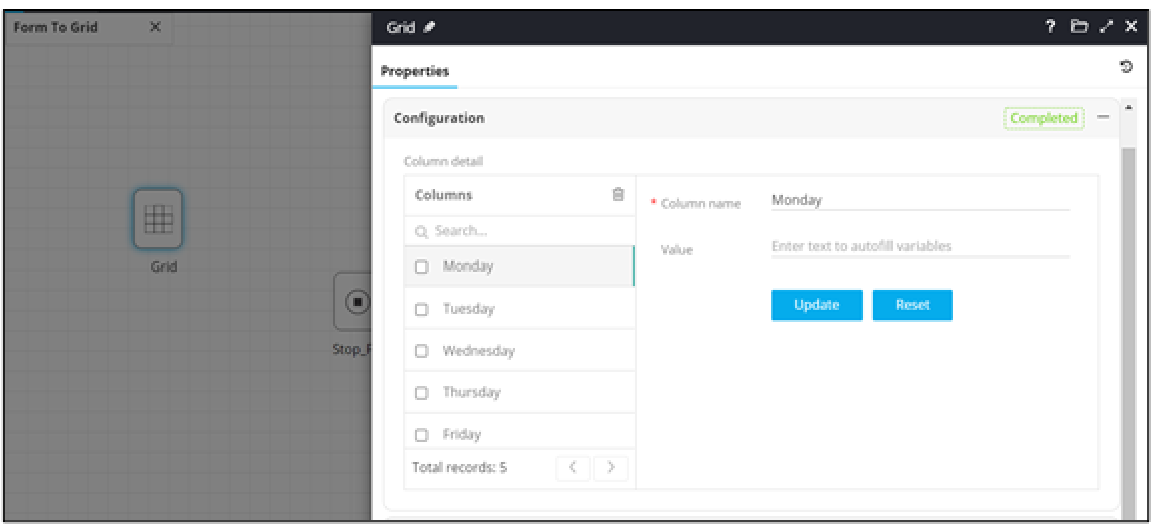
Downloading the Grid

1. Design a new workflow.
2. From the **General** section, drag and drop the **Form** task.
3. Add the required form fields as shown here.



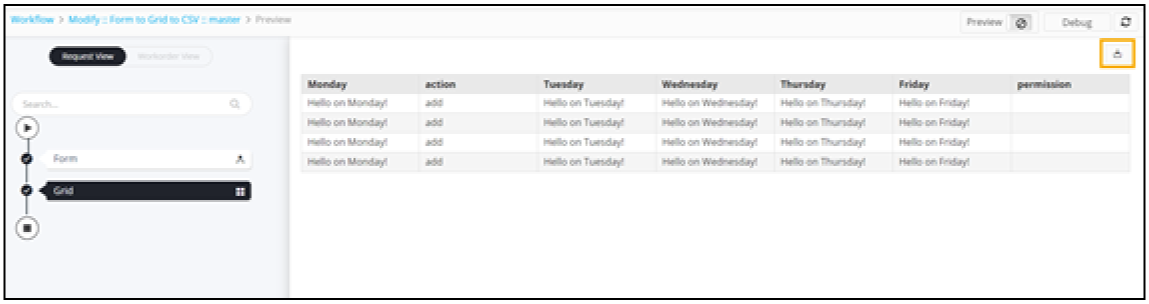
> **Note:** For more information on adding form fields, click here.

4. From the **User Interface** section, drag and drop the **Grid** task.
5. In the **Grid** task window, under **Properties**, in the **Configuration** section, configure the **Columns** for the grid.



6. Connect the workflow tasks and from the menu on the top, click **Preview**.

7. To download the Grid, click **(Download)** icon in the right upper corner of the screen.

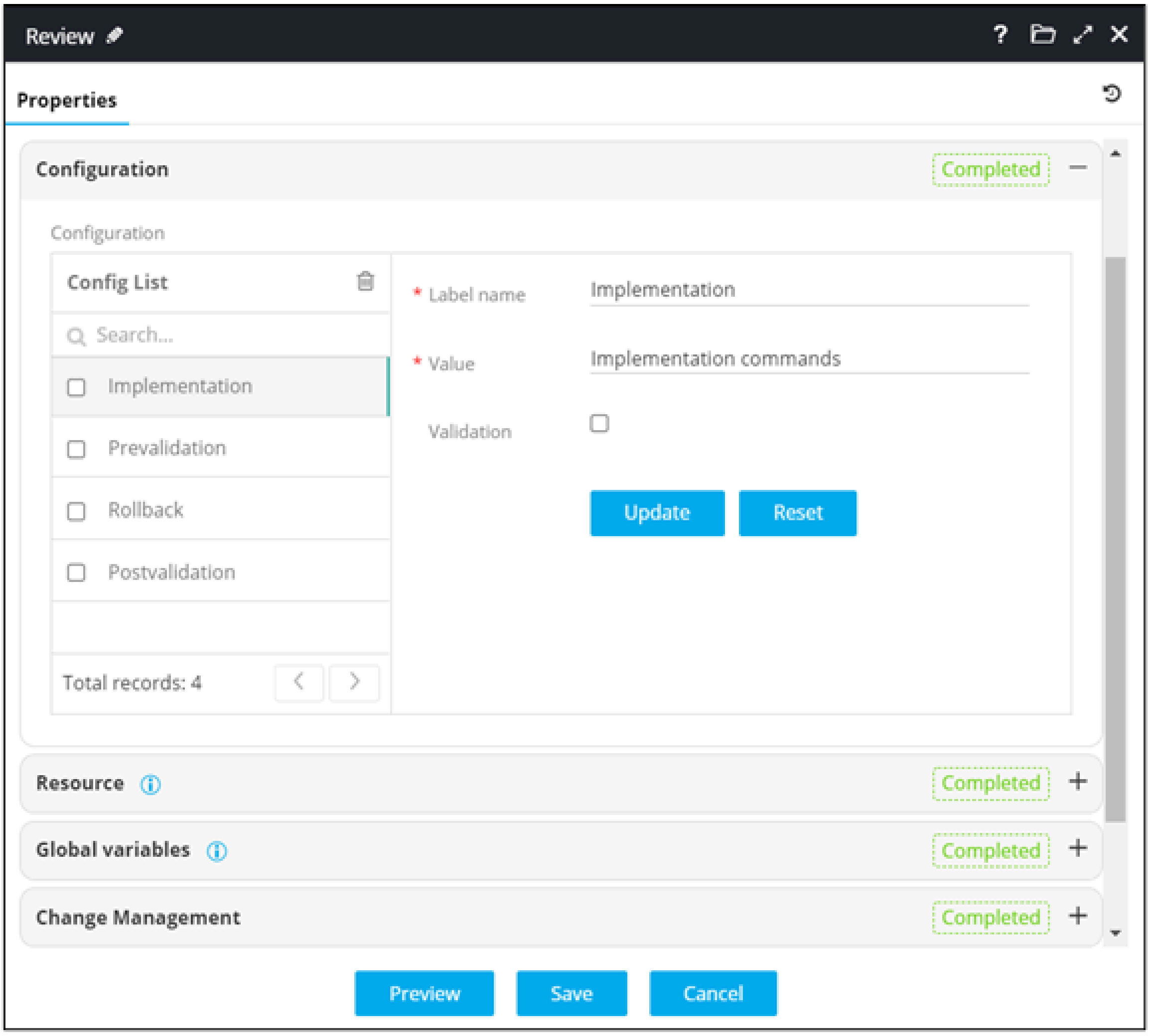


Review

You can define a custom ‘review component’ as part of the workflow automation process. The Review task allows you to enable peer reviews with multiple approval levels. You can review device configurations with Change management integration as part of the approval process.

- Provision to define a custom review or approval component
- Provision to preview the configuration block
- Provision to define custom configuration block and label names in the review component For example, Implementation configuration, Pre-validation configuration, Checkpoint Firewall rule configuration etc.
- Provision to dynamically pass data/value (using variables) from any stage of the workflow into the ‘appropriate configuration block’
- Provision to enable, disable the change management block as part of the request view
- Provision to assign user role (RBAC) access to the review task in the request stage
- Provision to define custom button/label names in the event of using the review task as part of any approval process.

For example, Approve, Implement, Reject, Rollback

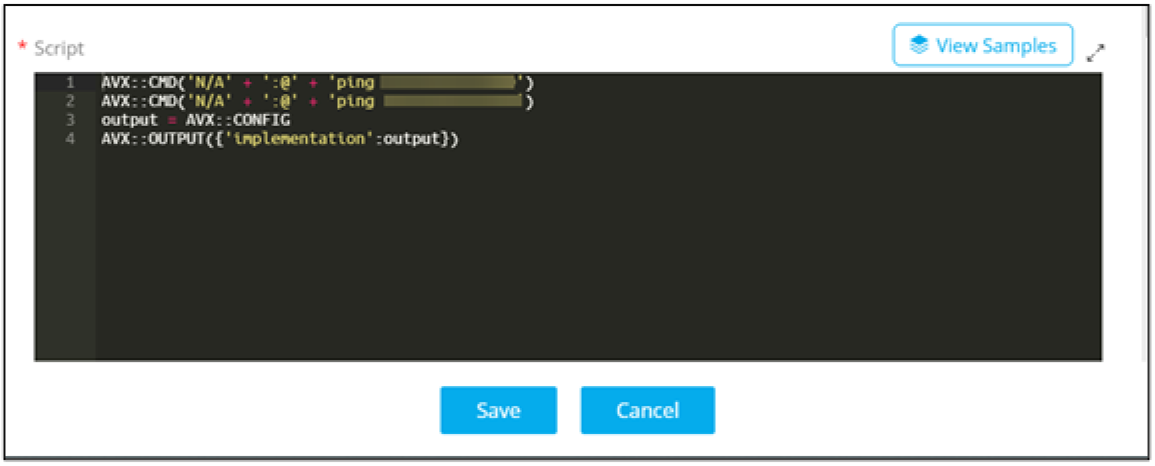


- Defining a Review Task using Variables
- Customizing Configuration Blocks in Review Task

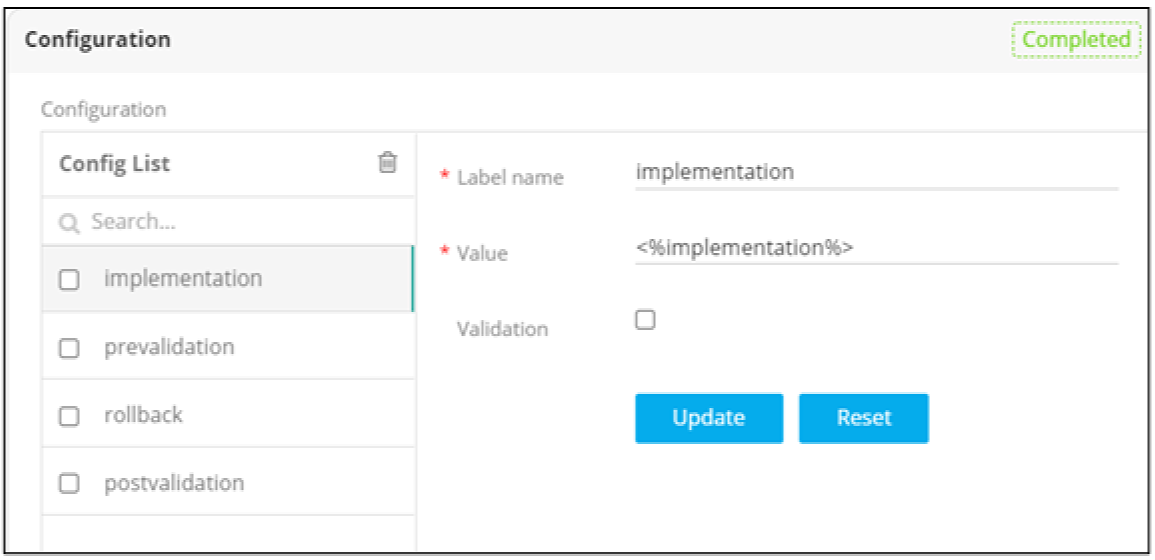
Defining a Review Task using Variables

A Review component can be defined using either static or dynamic values.

1. Design a workflow.
2. In the design space, from the **General** section, drag and drop a **Script** task.
3. In the **Script** task window, under **Properties**, in the **General** section, configure the script to generate commands to ping servers and store the configuration in a variable (AVX::CONFIG).

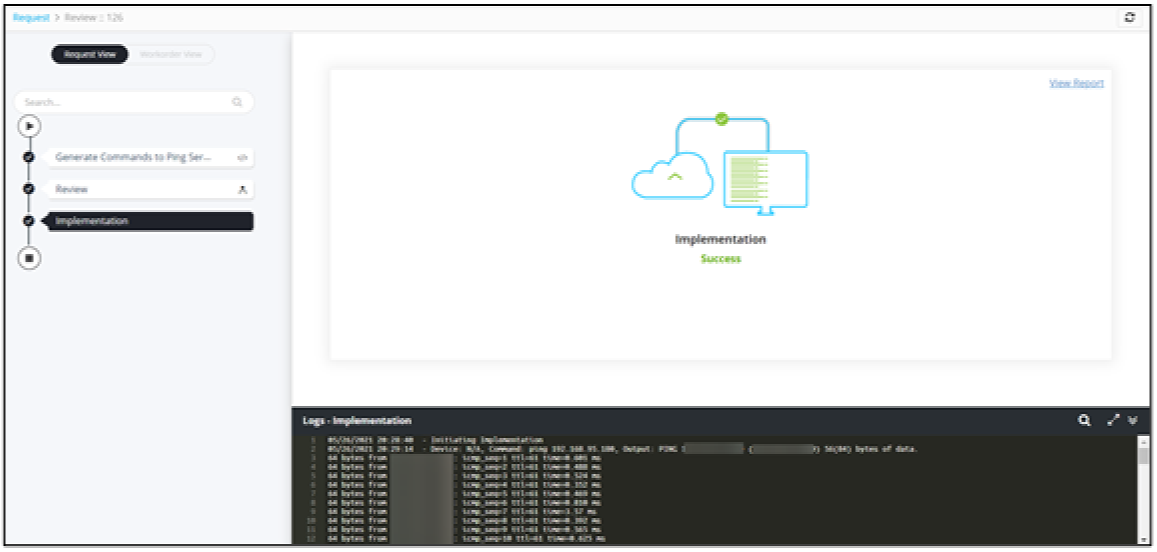


4. From the **User Interface** section, drag and drop the **Review** task.
5. In the **Review** task window, under **Properties**, in the **Configuration** section, define the configuration blocks - Prevalidation, Implementation, Rollback, Postvalidation.
6. Refer the variables from the Script task into the appropriate configuration blocks.



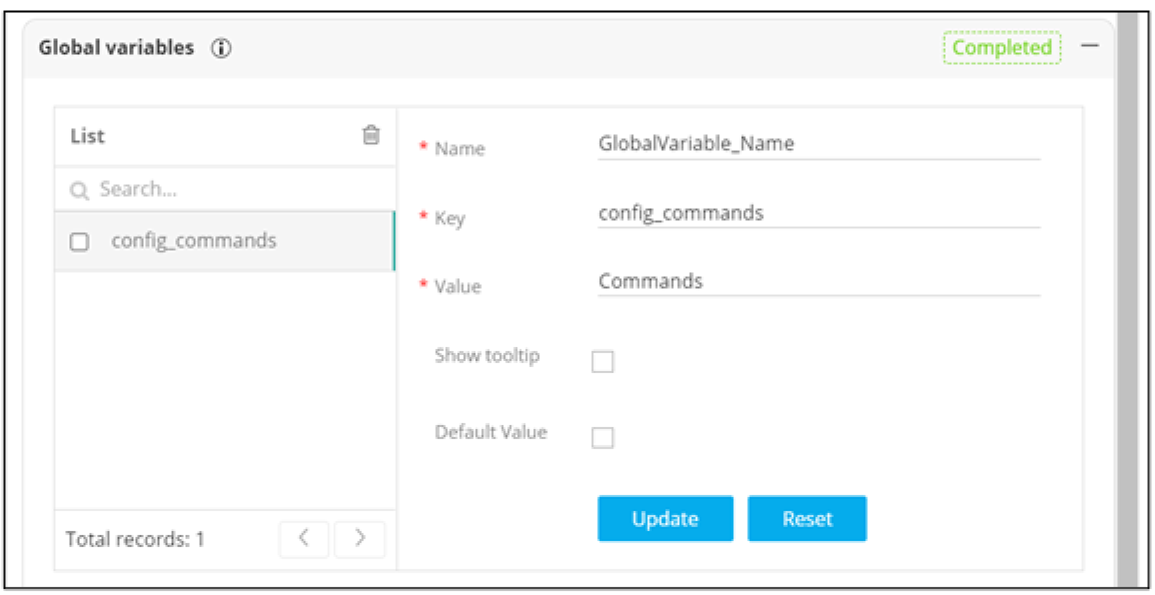
7. From the **General** section, drag and drop the **Implementation** task.
8. Connect and enable the workflow.
9. Trigger the workflow from the Request :: View/Run page.

Workflow is executed successfully with the Review task.



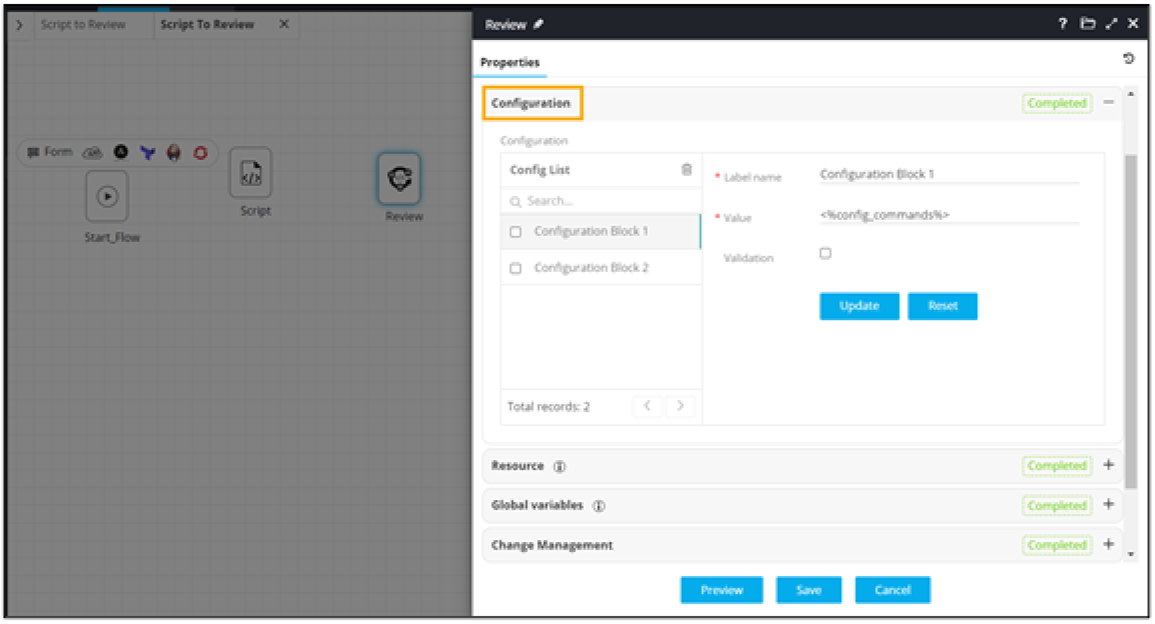
Customizing Configuration Blocks in Review Task

1. Design a workflow.
2. From the **General** section, drag and drop a **Script** task.
3. In the **Script** task window, under **Global variables**, declare the data to be passed as a global variable to be referenced in the Review task.



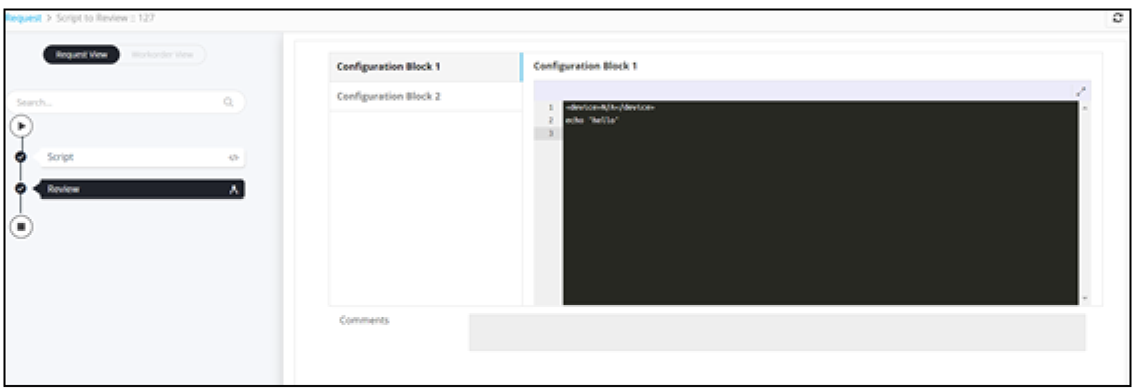
4. From the **User Interface** section, drag and drop the **Review** task.
5. In the **Review** task window, under **Configuration** , define custom configuration blocks.

6. Pass variable(s) from the previous task in the Value field against the appropriate configuration block.



7. Connect and enable the workflow.
8. Trigger the workflow.

   Workflow is executed successfully with the customized configuration blocks.

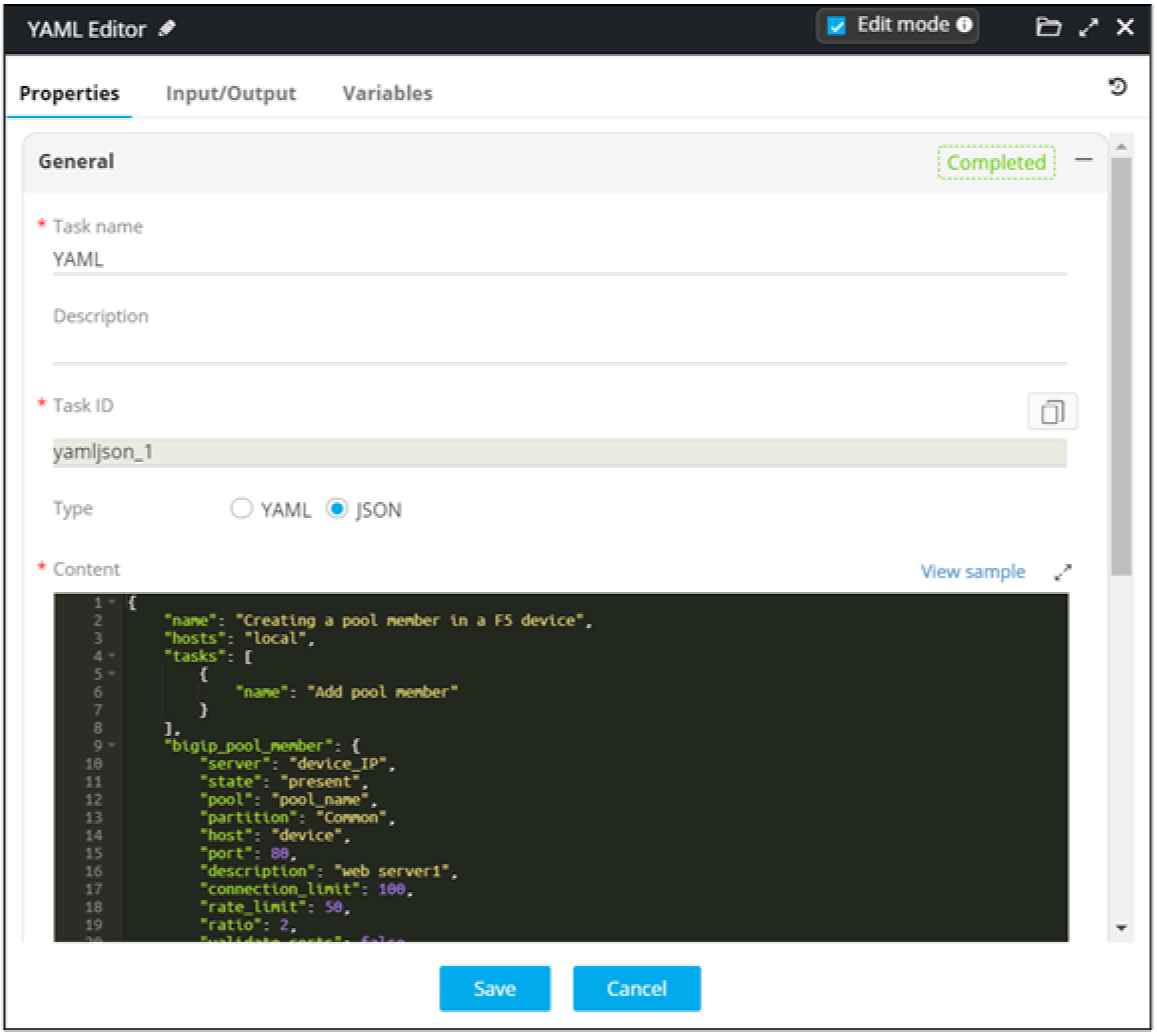


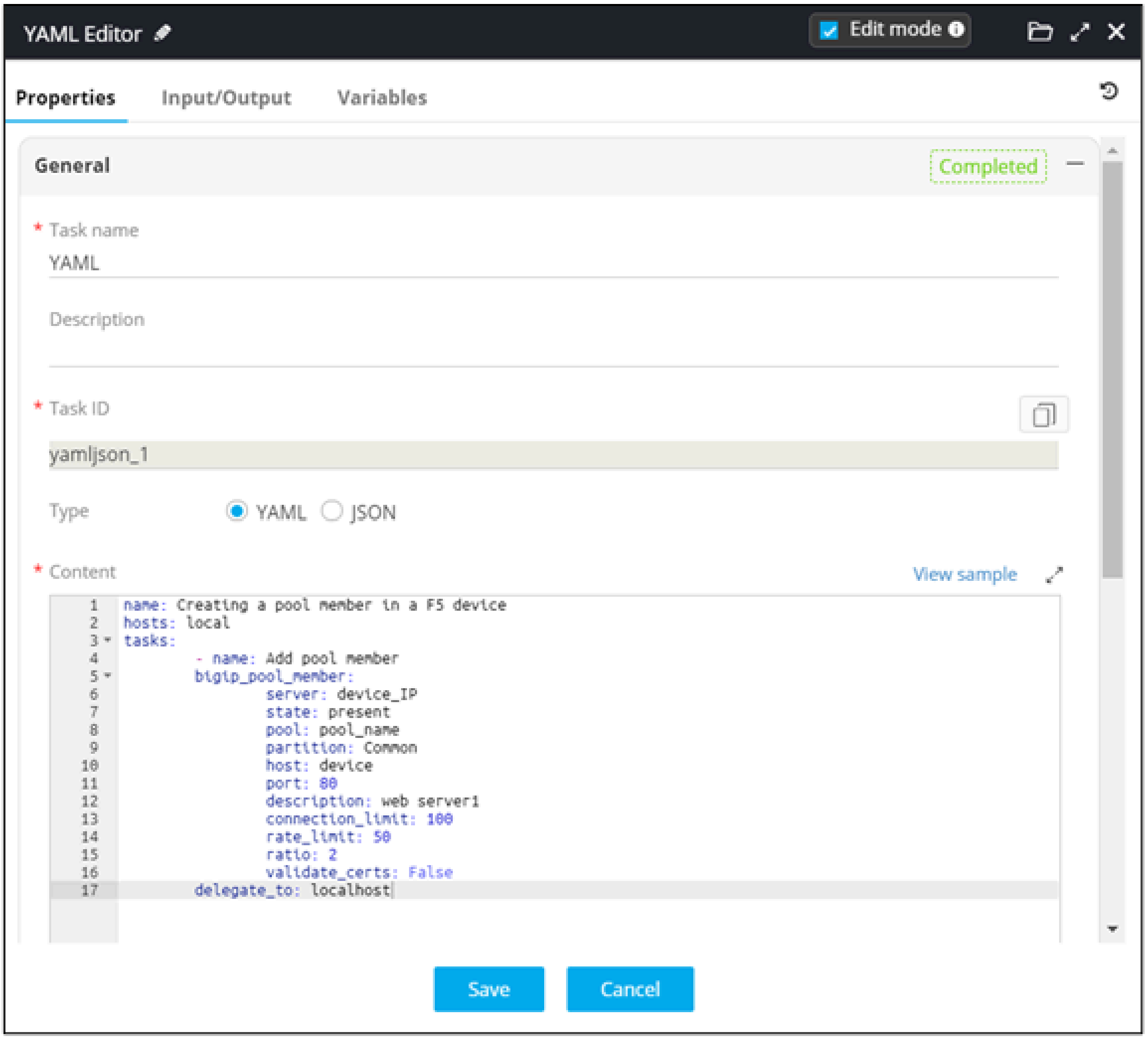
YAML

This task allows you to input the workflow data in the form of YAML or JSON content.

- Provision to input data in either YAML or JSON format
- Provision to refer YAML or JSON samples
- Provision to use the YAML task as either a user interface task or as a service task
- Provision to edit the contents of the task in the request page

- Provision to assign user role (RBAC) access to the task in the request stage
- Provision to skip manual approvals for the task in the workflow request stage
- Provision to retrieve device credentials dynamically.





- Using Variables within a YAML Task
- Getting Device Credentials Dynamically from a YAML Task

Using Variables within a YAML Task

Variables from any workflow task can be referred within the YAML task as part of the automation process. Use the following variable syntax in the YAML task to refer value from a previous task: `<%variable_name%>`

Following is an illustration for referring the ‘fqdn’ and ‘FreeIP’ values from an Infoblox task into a YAML task:

```
--
-
name: Creating a virtual server in a F5 device
```



```
hosts: local
tasks:
    - name: Add virtual server
      bigip_virtual_server:
        server: <%device%>
        user: $$<%device%>.username$$
        password: $$<%device%>.password$$
        state: present
        partition: Common
        name: <%fqdn%>
        destination: <%free_ip%>
        port: 480
        snat: Automap
        description: Test Virtual Server
        validate_certs: no
        all_profiles:
          - http
          - tcp
      delegate_to: localhost
```

Using Variables within a YAML Task

Variables from any workflow task can be referred within the YAML task as part of the automation process. Use the following variable syntax in the YAML task to refer value from a previous task: `<%variable_name%>`

Following is an illustration for referring the ‘fqdn’ and ‘FreeIP’ values from an Infoblox task into a YAML task:

```
--
-
 name: Creating a virtual server in a F5 device
 hosts: local
 tasks:
    - name: Add virtual server
      bigip_virtual_server:
        server: <%device%>
        user: $$<%device%>.username$$
        password: $$<%device%>.password$$
```



```
    state: present
    partition: Common
    name: <%fqdn%>
    destination: <%free_ip%>
    port: 480
    snat: Automap
    description: Test Virtual Server
    validate_certs: no
    all_profiles:
      - http
      - tcp
  delegate_to: localhost
```

Getting Device Credentials Dynamically from a YAML Task

Through the YAML (JSON) task, device credentials can be dynamically retrieved as part of the workflow execution process. The following syntax must be in the task in order to get device credentials:

- Get device detail:

```
$$devicename.username$$
```

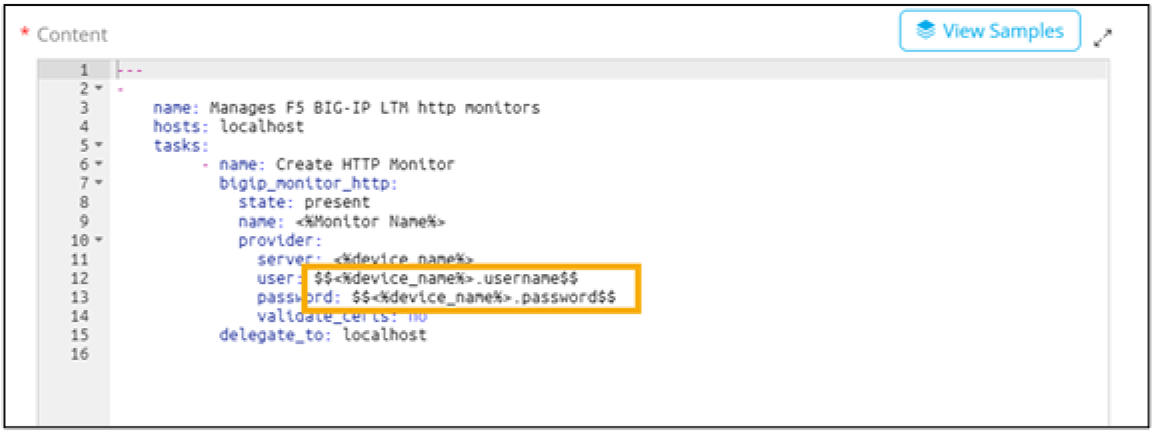
- Get device credential:

```
$$devicename.password$$
```

Note: Device(s) must be added in the Appviewx inventory.

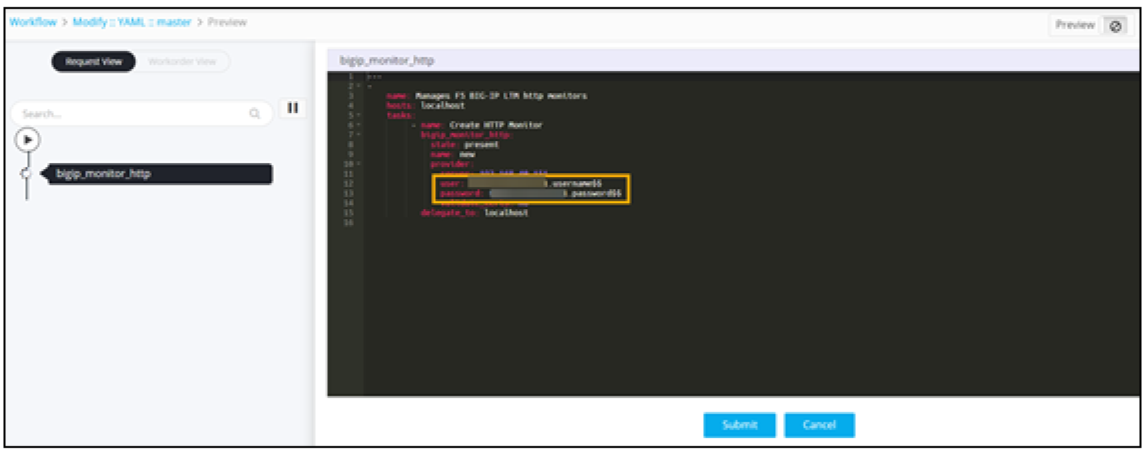
To get the device credentials dynamically:

1. Design a new workflow.
2. From the User Interface section, drag and drop the **YAML** task.
3. Enter the syntax to get device credentials dynamically.



4. Connect the workflow tasks and enable the workflow.
5. Trigger the workflow.

   Device credentials are dynamically pulled from the Inventory.

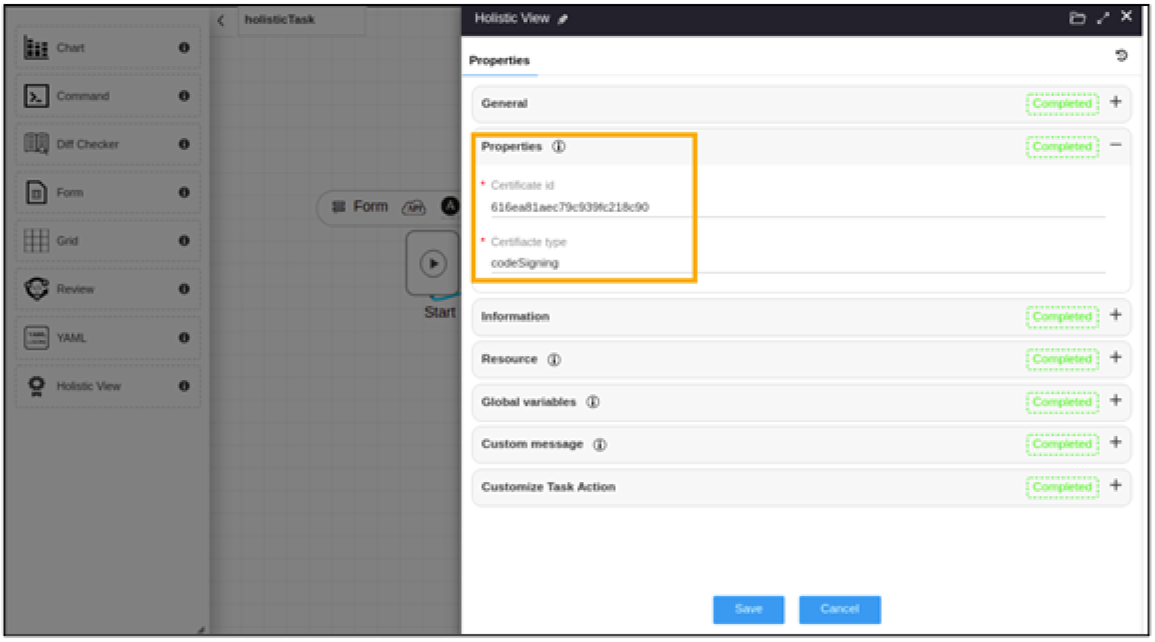


Holistic View

This task allows you to view the certificates in a holistic view in the workflow Studio. You have the option to download the certificate and the certificate key from the Holistic View stage of the workflow.

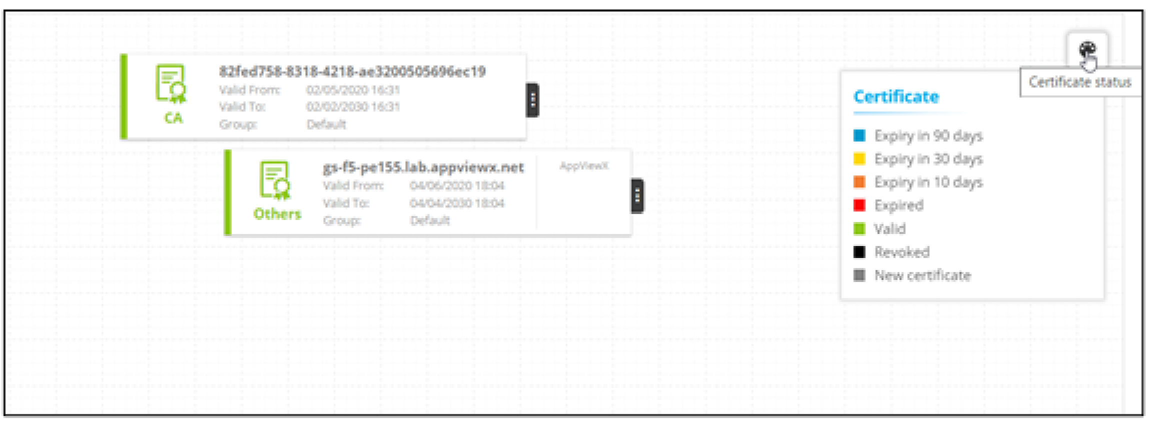
To add Holistic View task to your workflow:

1. Design a workflow.
2. From the User Interface section, drag and drop the **Holistic view** task.
3. In the **Holistic view** task window, under **Properties**, define the **Certificate id** and **Certificate type** either as static values or a global variable.

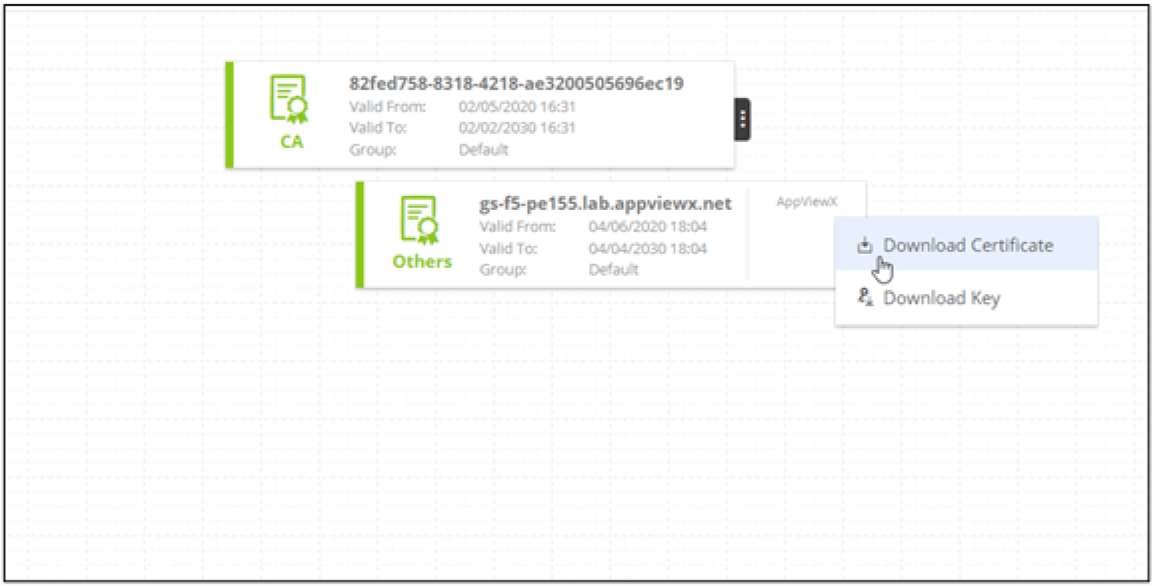


4. Connect the workflow tasks and from the menu on the top, click **Preview**.

   Holistic view is executed. Hover your mouse over **(Certificate status** icon to view the certificate status.



5. To download the certificate and certificate key, click .



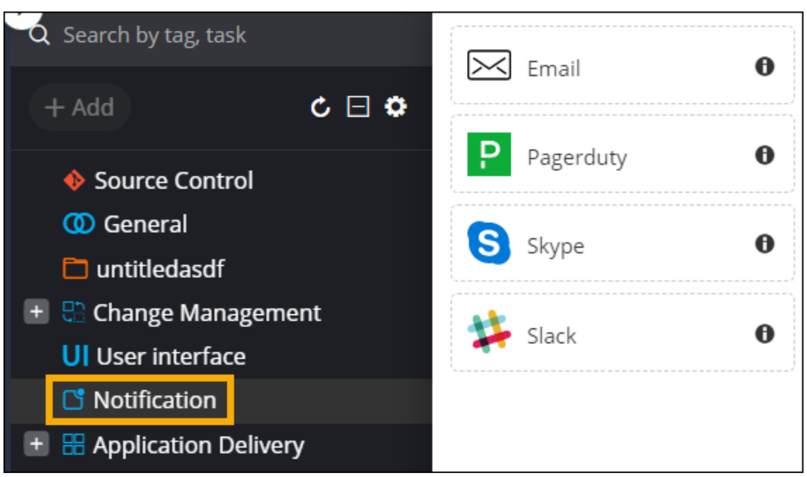
6. Click **Submit**.

Note: You can add multiple Holistic view tasks to the workflow.

Note: For more information on Holistic View, refer to the Cert+ User Guide.

Task Category - ChatOps and Notifications

This category comprises tasks that are used to send out notifications to the user for receiving inputs, approvals/rejections, and task completion.



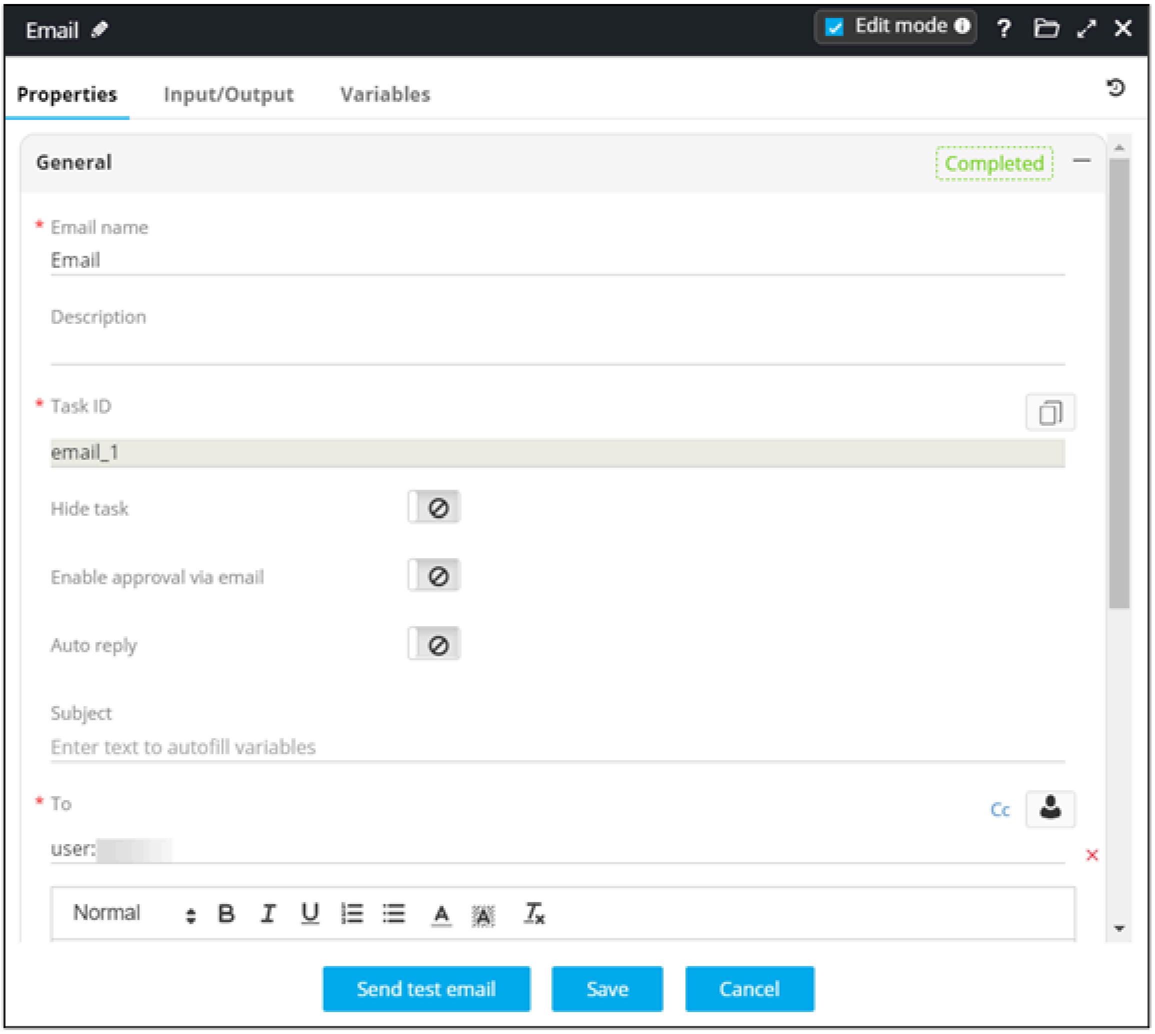
- Email
- Pagerduty
- Slack
- Skype

Email

Email task allows you to configure static or dynamic email notifications as part of the workflow automation process. You can send email attachments in different formats. This task allows you to configure email approvals to automatically trigger the workflow process.

- Provision to select user(s), user role(s), and user group
- Provision to send test email to ensure that the defined content is relevant
- Provision to reuse email tasks from one workflow into another
- Provision to dynamically render emails by referencing data/value (using variables) from any stage of the workflow into the email task
- Variables can be referenced into the following fields – Subject, To, Cc, Content

- Provision to trigger workflow based on email approvals.



- Configuring Dynamic Email Attachments
- Configuring Email Approvals
- Receiving Approvals via Email
- Triggering an Email Notification with Approvals
- Customizing Email Notification

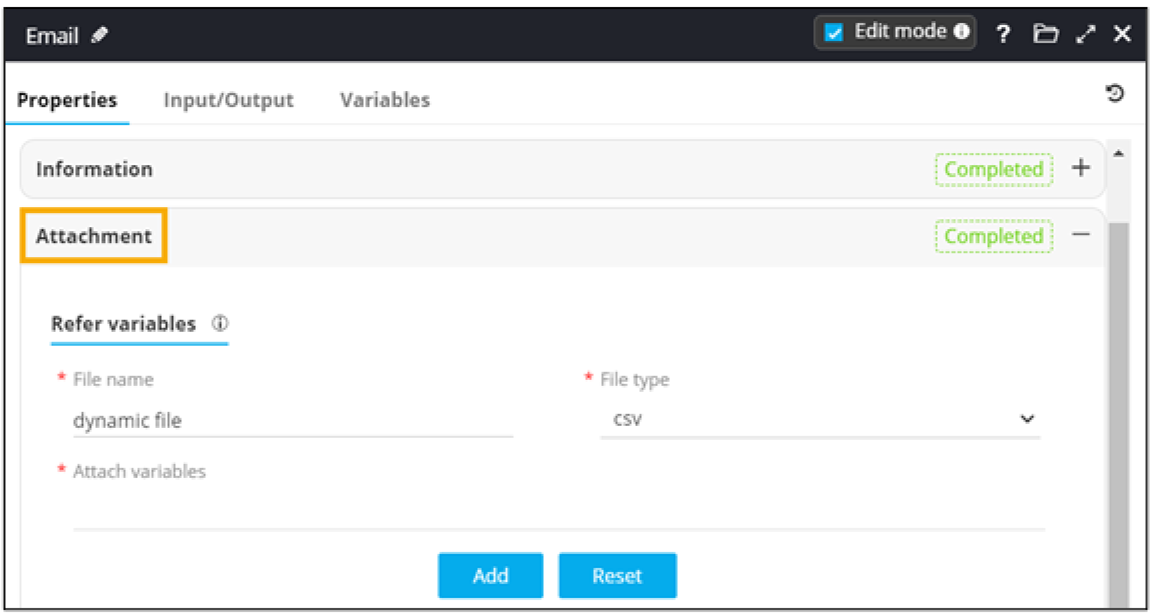
Configuring Dynamic Email Attachments

You can attach static files or dynamically generate file(s) from any workflow stage as part of the email.

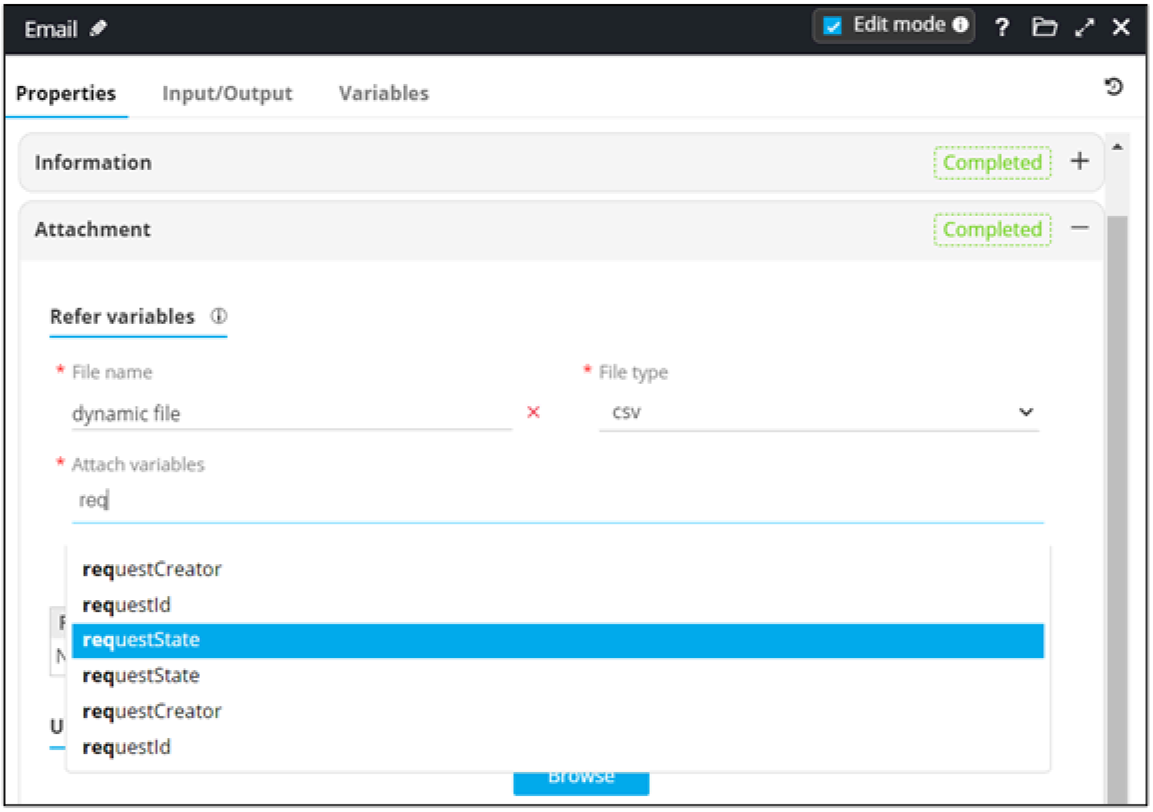
- Provision to refer variable(s) from any stage and dynamically generate file attachments.
- Provision to choose the file type for dynamically generating file attachments (Supported file formats are - pdf, csv, txt).
- Provision to upload/add one or more file attachments from the system with a limit up to 25 MB.
- Provision to delete attachment(s) from the email task.

To send an attachment:

1. Design a workflow.
2. From the Notification section, drag and drop an **Email** task.
3. In the **Email** task window, under **Properties**, in the **Attachment** section, enter or select the field information.



4. Attach variables to add a dynamic email attachment.



5. Click **Add**.

   You can use this task within your workflow to send email attachments dynamically.

Configuring Email Approvals

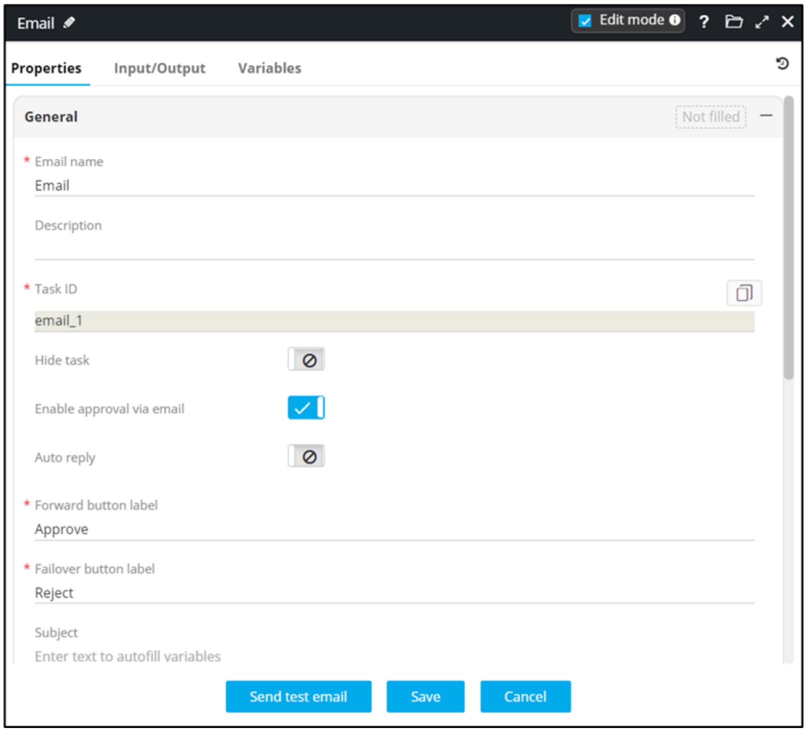
Approvals/peer reviews in the network automation process can be notified and approved directly via email without having to log into AppViewX. You can review config details, and approve or reject based on which the workflow will be triggered further.

- Configure approvals via email and trigger the workflow based on the email event or action.
- Define manual or automatic email approvals.
- Emails include static and dynamic file attachments (pdf, jpeg, doc, xls).

To enable email approval:

1. Design a workflow.
2. From the Notificationsection, drag and drop the **Email** task.

3. In the **Email** task window, under **Properties**, in the **General** section, turn on the **Enable approval via email** toggle.



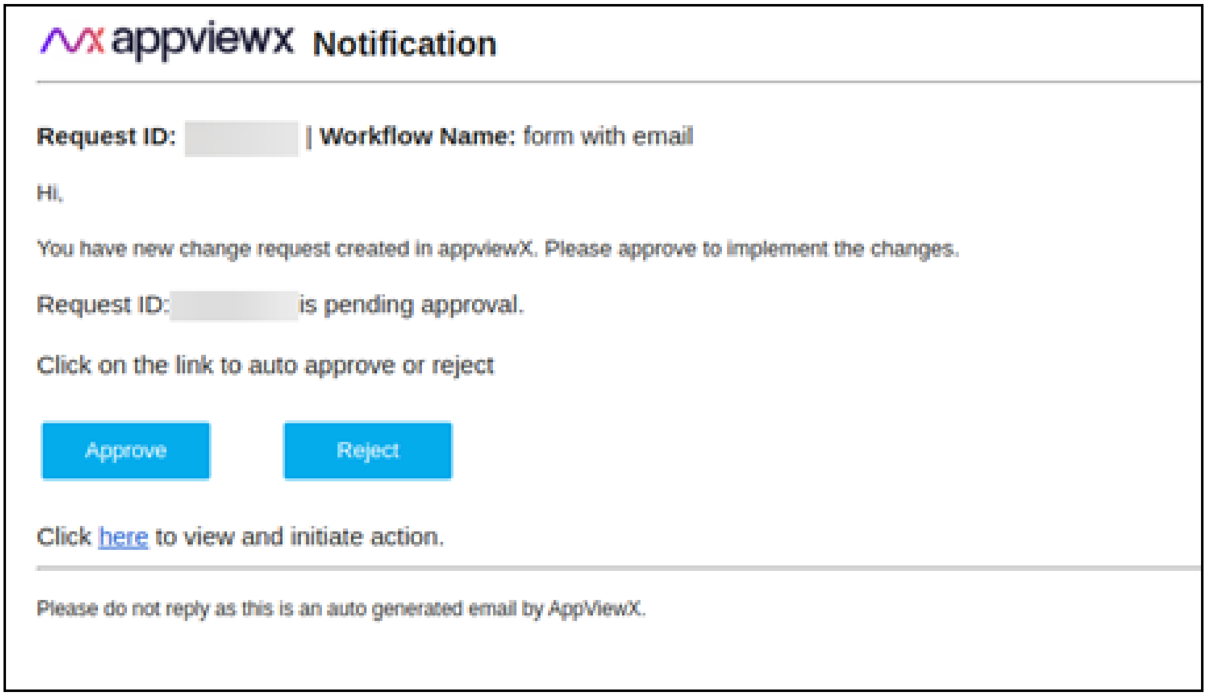
   On enabling the toggle, the following fields are displayed:
   - **Forward button label**
   - **Failover button label**.
4. Enter values for these two fields.
5. Enter custom email content.
6. Click **Save**.

   You can use this task within your workflow to send email approvals.

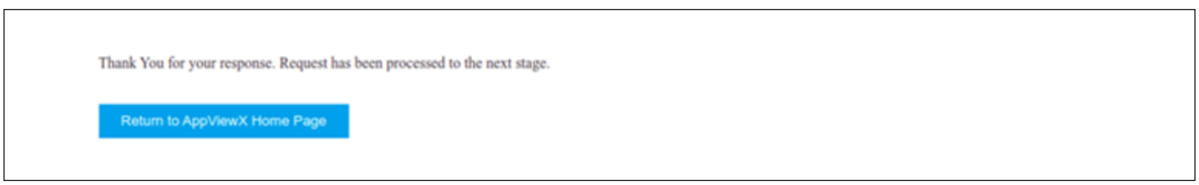
Receiving Approvals via Email

Once the email task to trigger approvals is configured, you can review and approve or reject the workflow automation process directly via email.

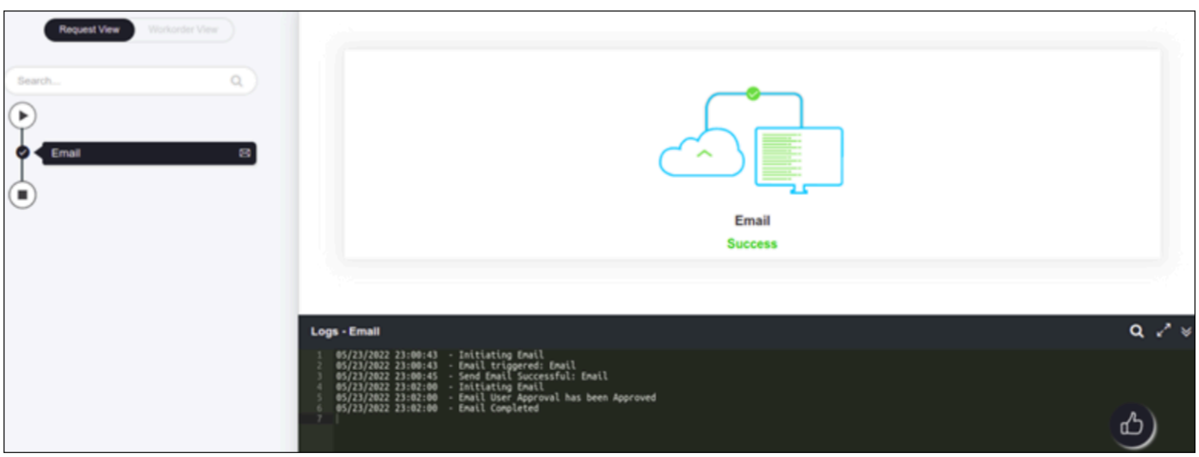
1. Click on the link embedded in the approval email (Approve/Reject) to navigate to AppViewX.



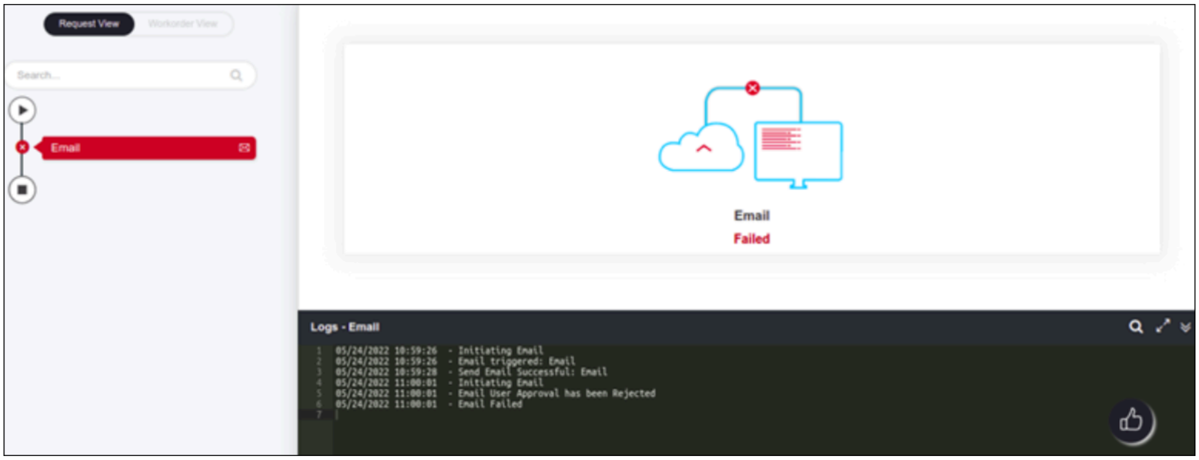
2. Click **Approve** / **Reject**.
   - The following message (HTML text) will be shown in a new tab.



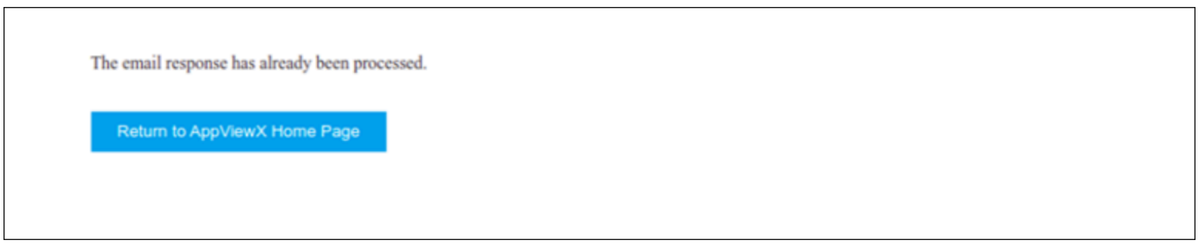
   - If you click **Approve**, the task is displayed as **Success**.



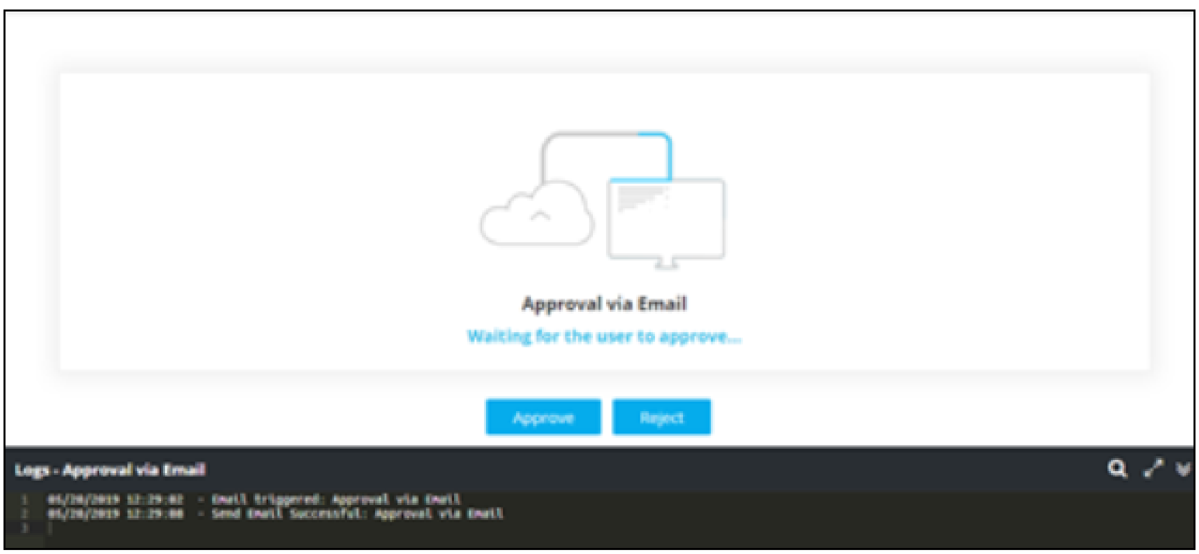
   - If you click **Reject**, the task is displayed as **failed**.



- In the event of clicking the email approval again, the following message indicating that the email response has been processed will be shown in a new tab.



- In the event of not processing the approvals via email on time, users can log into AppViewX, navigate to the pending workflow request and manually approve or reject the workflow task.

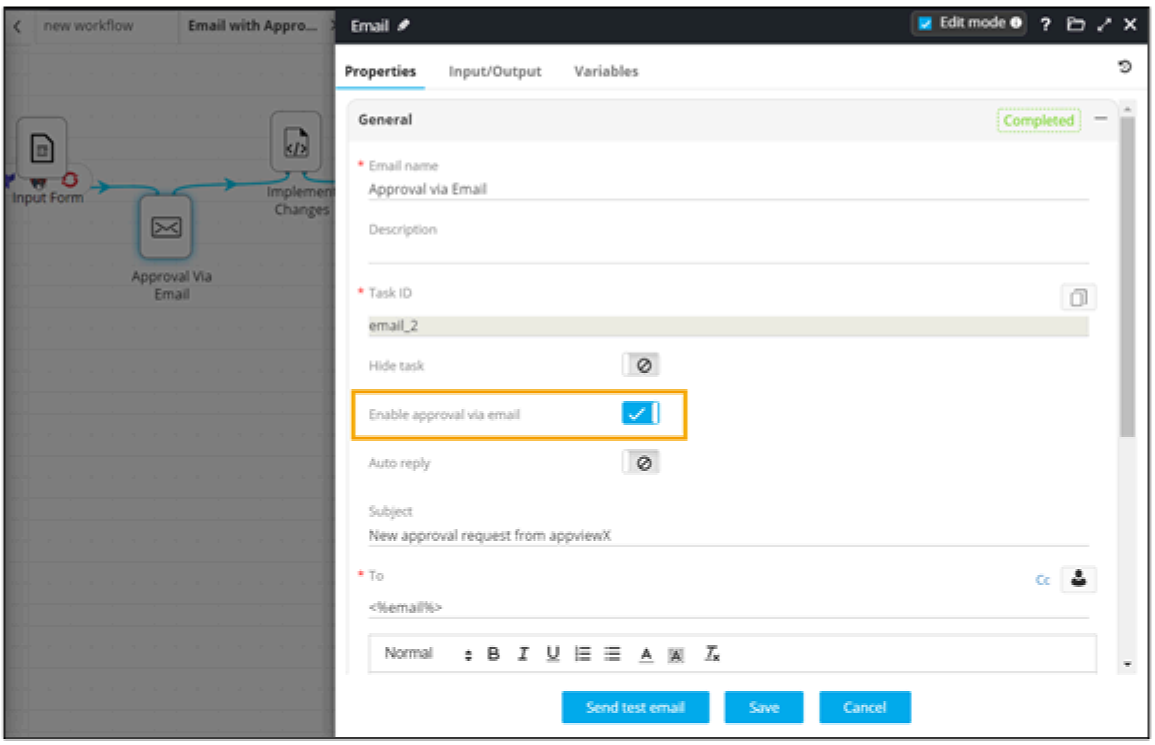


Note: The SMTP settings must be configured prior to triggering any emails. For more information, refer to the Platform User Guide.

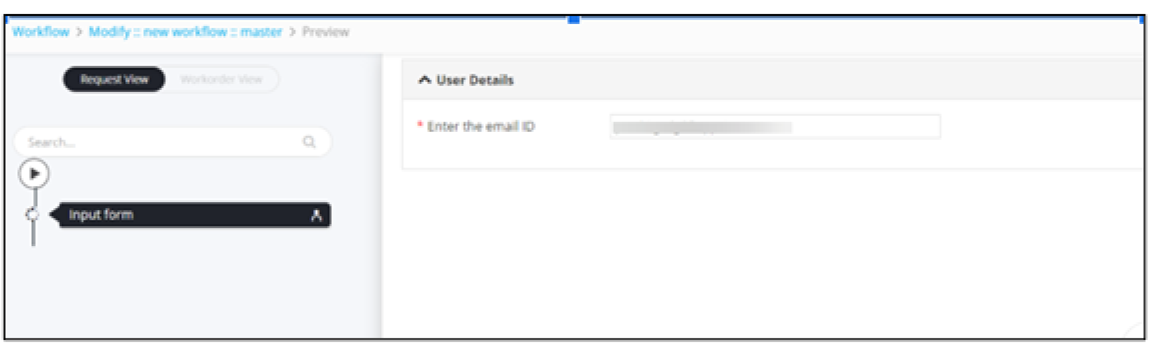
Triggering an Email Notification with Approvals

To generate a simple email notification with an option to approve or reject the workflow:

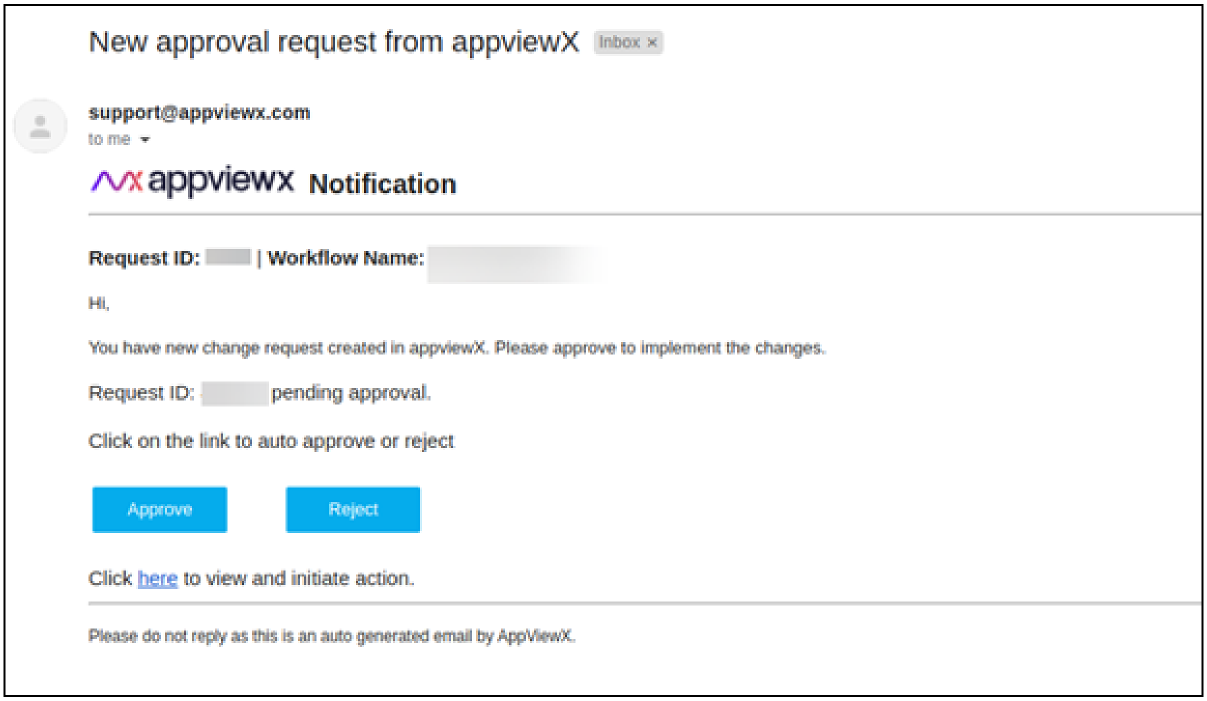
1. Design a new workflow.
2. From the FAQs section drag and drop the > **Email with approval** workflow.
3. Double-click on the workflow to see the tasks within the workflow.
4. Click on the **Approval Via Email** task in the workflow.
5. In the **Email** task window, under **Properties**, in the **General** section, turn on the **Enable approval via email** toggle.



6. Click **Preview**.
7. Enter the email ID in the user input form.



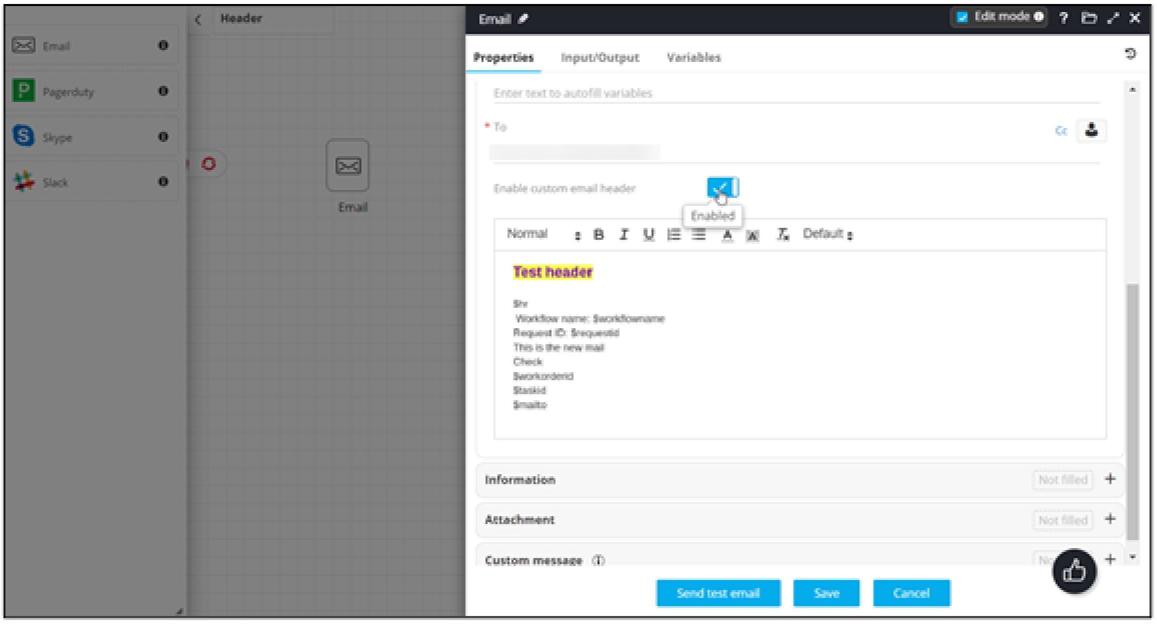
Approval email notification received via email.



Customizing Email Notification

You can customize the email notification by configuring the text in the content box in the email task. You can also customize the email content by including default variables like workflow name, horizontal line and Request ID.

1. Design a workflow.
2. In the **Email** task, to **Enable custom email header**, turn on the toggle.



3. To attach the default variables (Horizontal Line, Workflow Name, Request ID), click **Default** and select the variable from the dropdown.

   Custom email notification received.

Note: When the Enable custom email header toggle is disabled, the email content will display as the header.

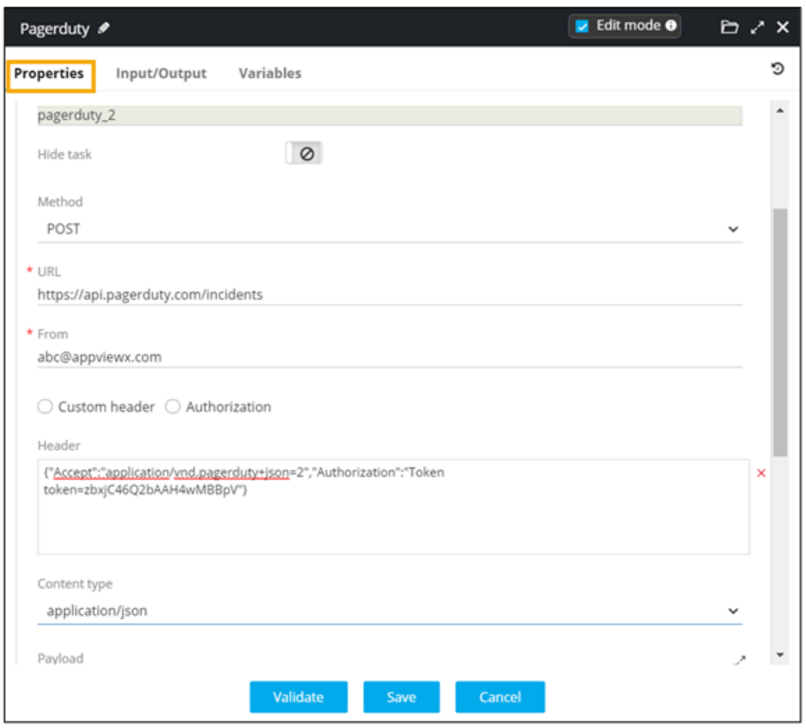
Pagerduty

This task allows you to trigger alarm notifications for quick incident remediation through Pagerduty as part of the workflow automation process.

- Provision to drag and drop the Pagerduty Task from the notification section.
- Provision to define custom webhook/URL, method in order to trigger notification.
- Provision to refer global variables and embed them as part of the payload.

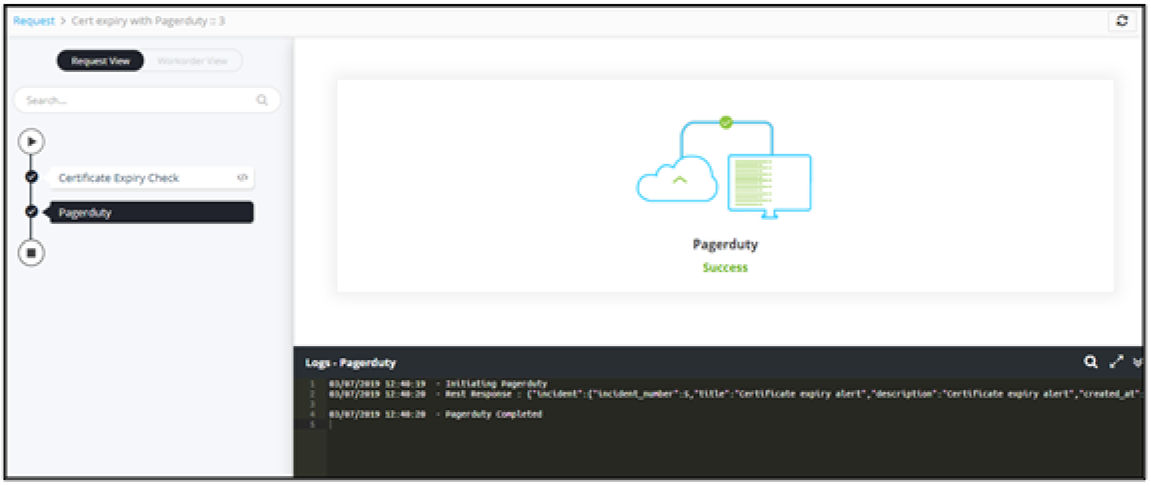
To trigger alarm notifications through Pagerduty:

1. Design a workflow.
2. Drag & drop relevant workflow task(s).
3. Double click on the **Pagerduty** task.
4. In the **Pagerduty** task window, under **Properties**, enter or select the field information as shown.

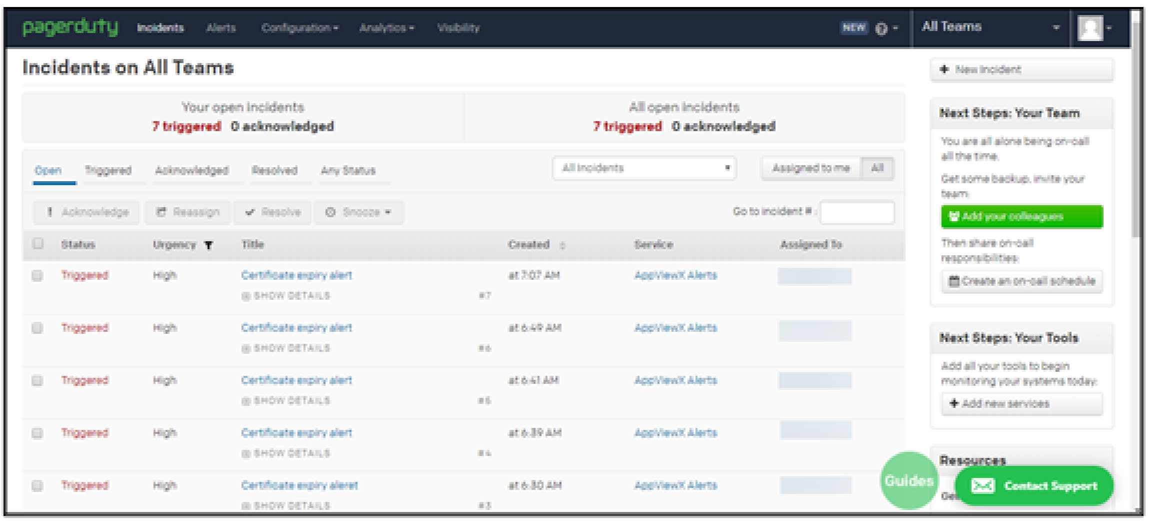


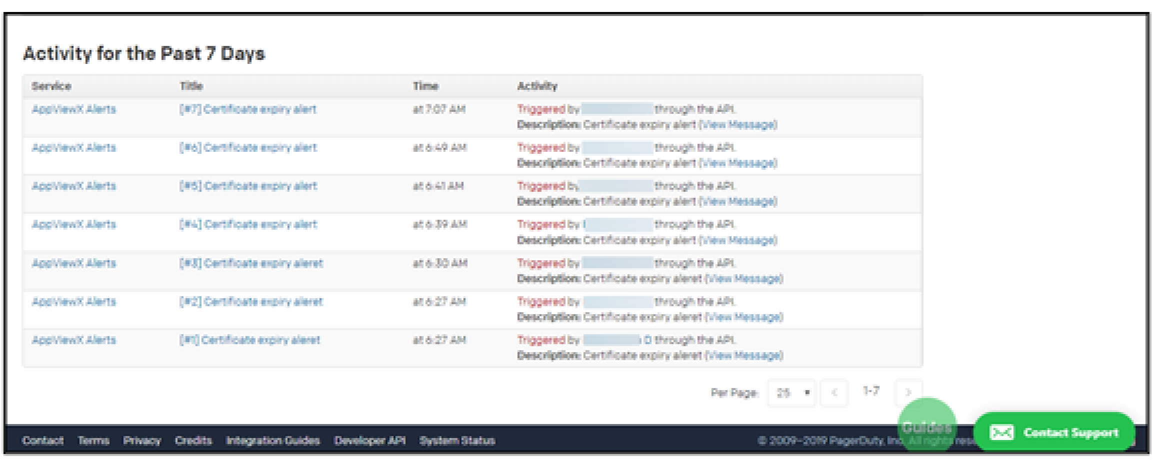
5. Enter the intended payload, with global variable(s) if any.
6. Connect the workflow tasks and enable the workflow.
7. Trigger the workflow from the Request :: View/Run page.

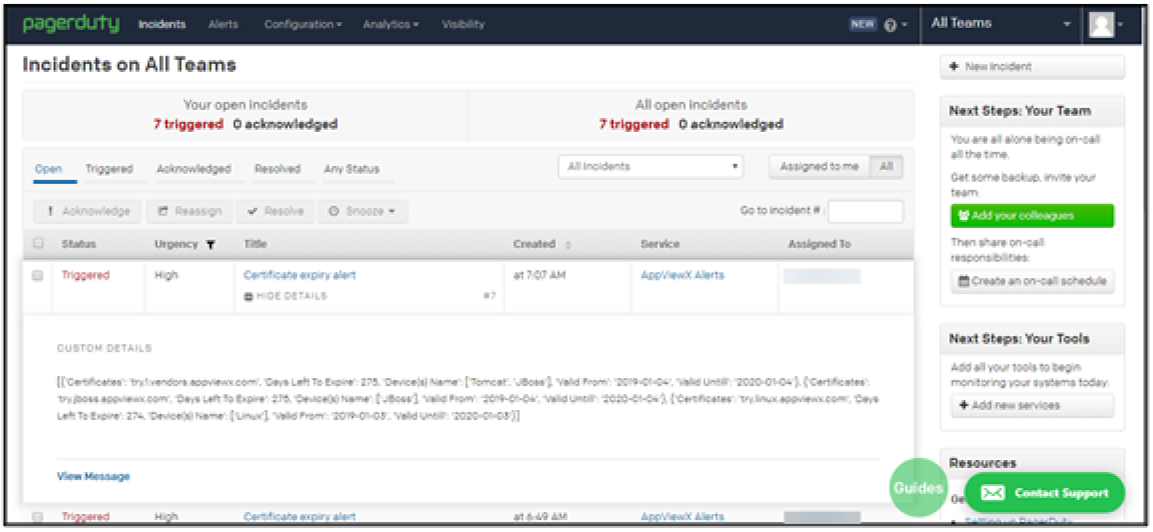
   Workflow is executed successfully.

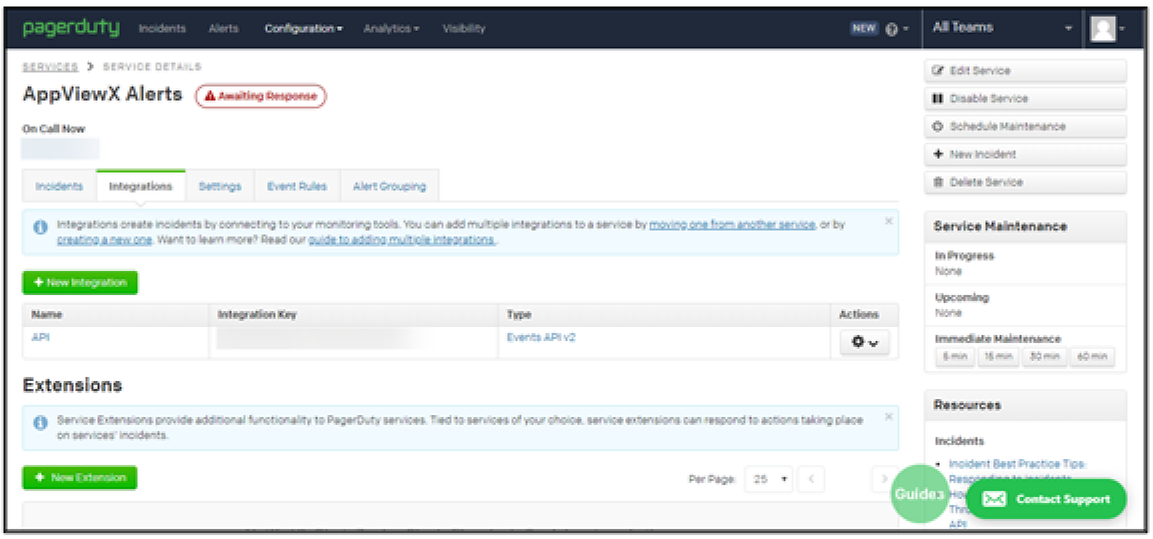


8. To view all Incidents on Pagerduty, log into your account using valid credentials.









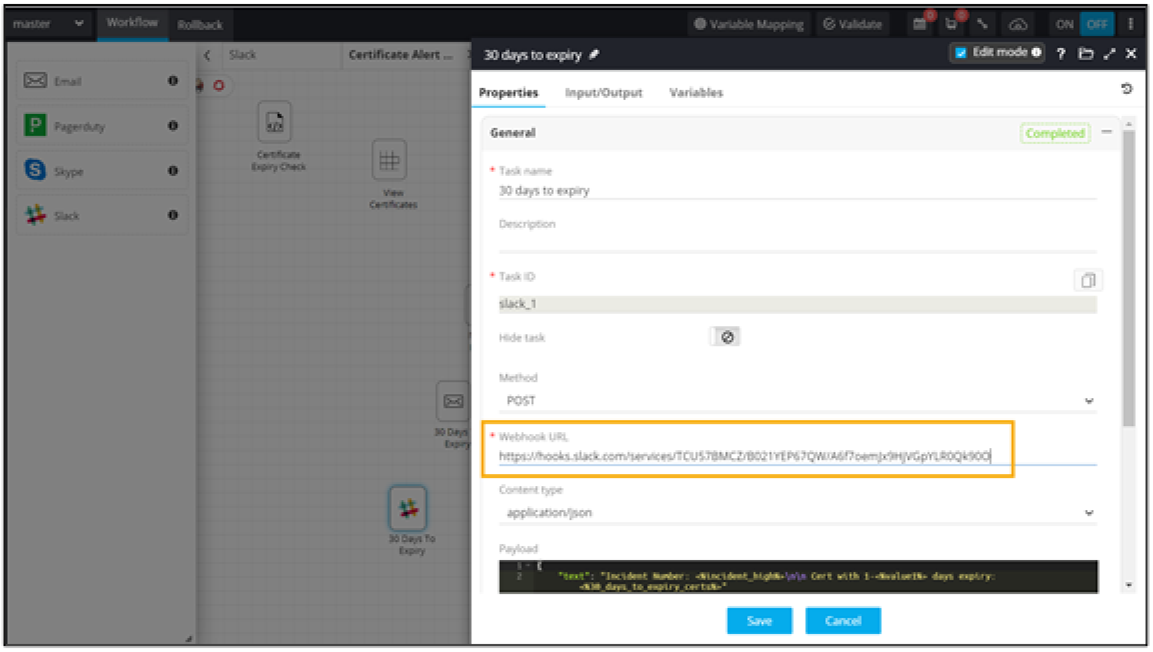
Slack

This task allows you to send notifications on a Slack channel as part of the workflow process.

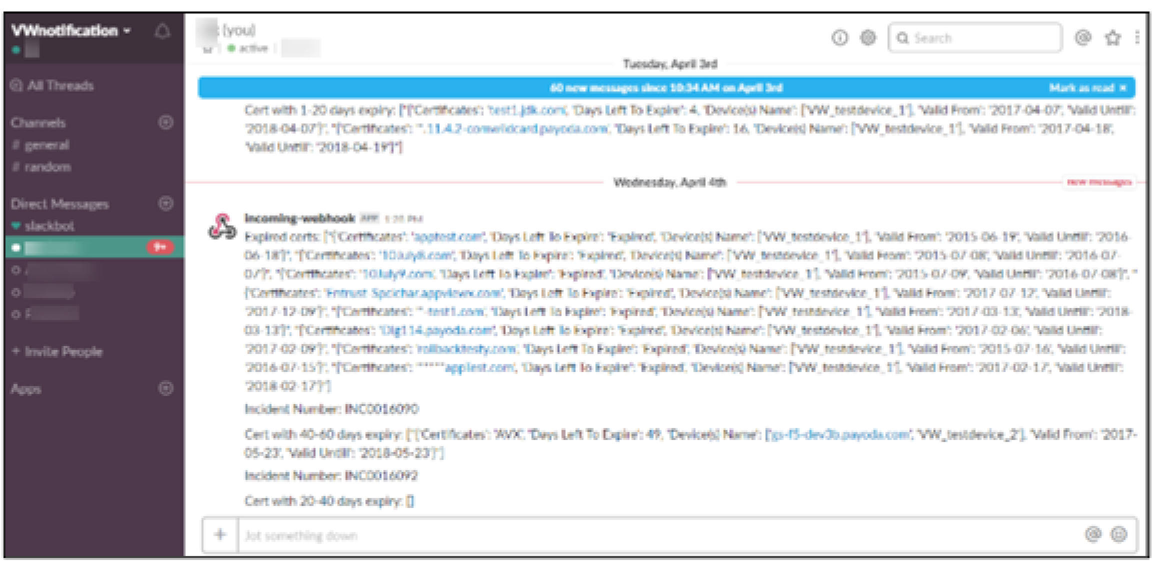
- Provision to drag and drop the slack task from the Notification section.
- Provision to define custom webhook, method in order to trigger slack notification.
- Provision to refer global variables and embed them as part of the payload.

To send a notification through Slack:

1. Design a workflow.
2. Drag and drop relevant workflow tasks.
3. From the Notification section, drag and drop the **Slack** task.
4. Define the appropriate webhook.



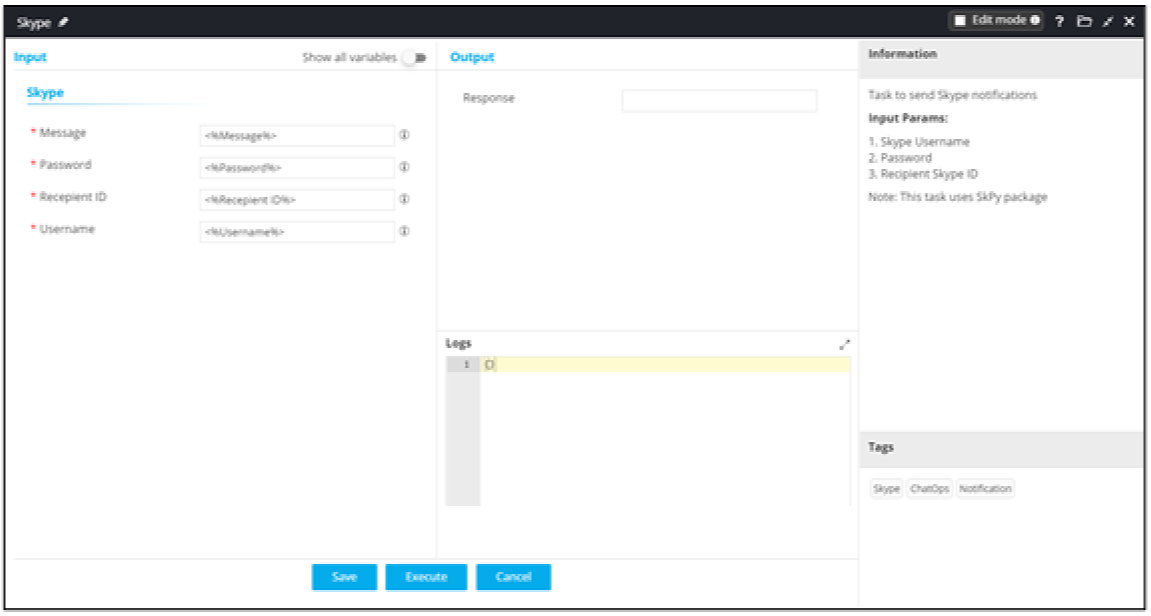
5. Enter the intended payload with global variables, if any.
6. Connect the workflow tasks and enable the workflow.
7. Trigger the workflow.
8. View Slack notifications.



Skype

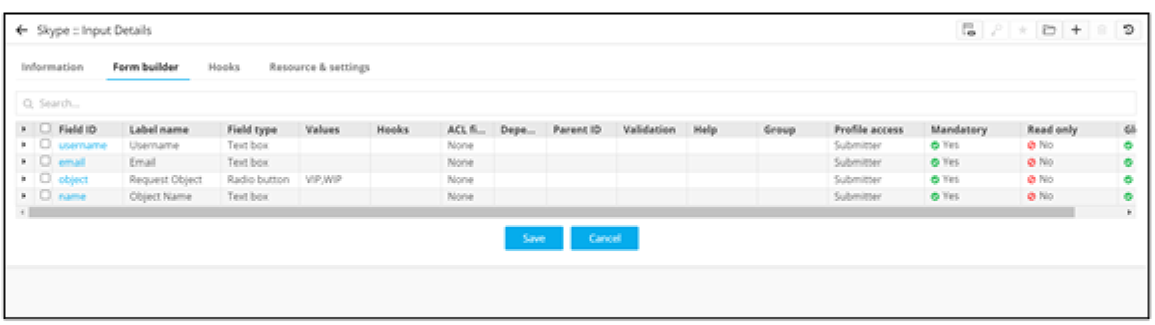
This task allows you to send Skype notifications as part of the workflow process.

- Provision to drag and drop the Skype task from the Notification section.
- Provision to define custom webhook, method in order to trigger Skype notification.
- Provision to refer global variables and embed them as part of the payload.



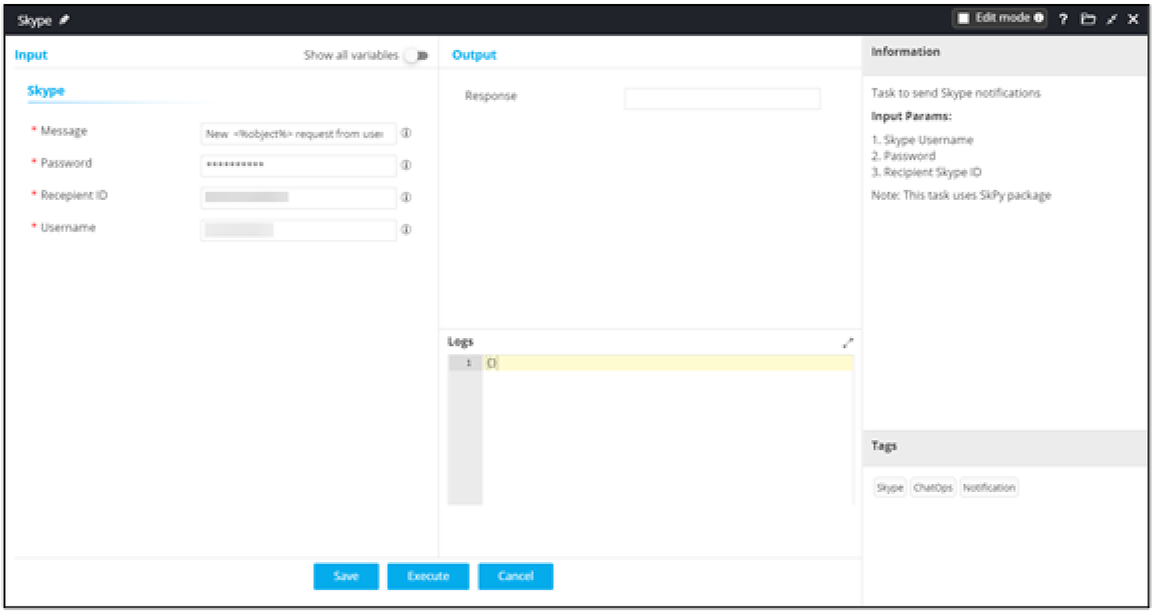
To send a notification through Skype:

1. Design a new workflow.
2. From the User Interface section, drag and drop the **Form** task.
3. Define the relevant form fields as shown here.



Note: For more information on adding form fields, click here.

4. From the Notification section, drag and drop the **Skype** task.
5. In the **Skype** task window, define the input parameters to send Skype notification to the recipient.



6. Connect the workflow tasks and enable the workflow.
7. Trigger the workflow to receive Skype notification.

Task Category - Change Management

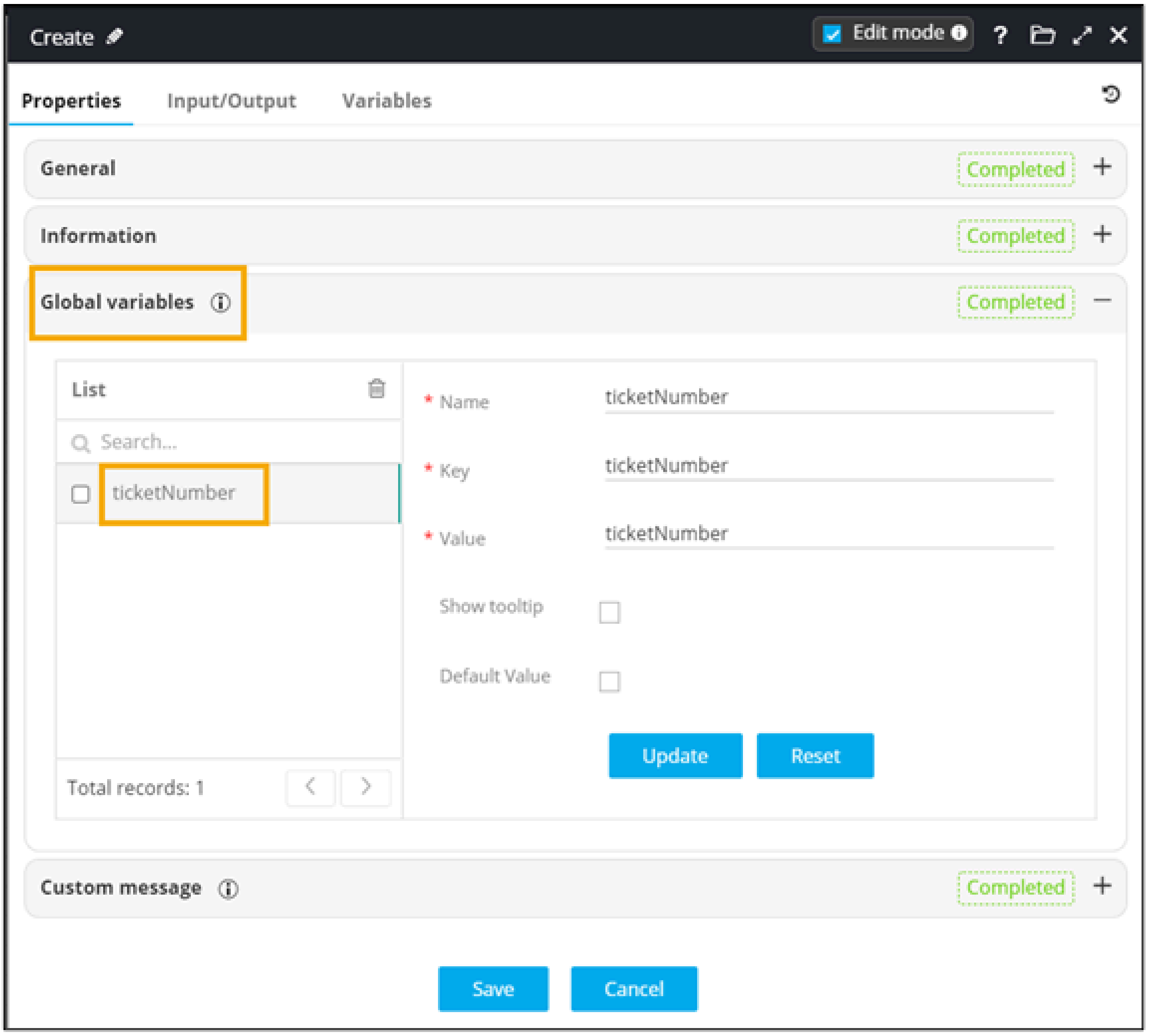
When creating a new workflow, you can leverage the Change management Tasks to ensure change management integration with workflow automation. It allows you to plug and play various components required to integrate a workflow automation flow with Change Management.

To create a workflow with ServiceNow tasks:

1. Design a workflow.
2. In the design workspace, under the **Change Management** folder in the left menu, from the **CHANGE** folder, drag and drop the **Create** task.

> **Note:** Timestamp must be in milliseconds.

3. In the **Create** task window, under **Properties**, in the **Global variables** section, declare the necessary global variables.



4. From the **CHANGE** folder, drag and drop the **Get** task.
5. In the **Get** task window, under **Properties**, pass the global variable declared in the previous task.
6. Connect the workflow tasks and enable the workflow.
7. Trigger the workflow.

   Workflow is executed successfully.

Provisioning for Workflow Tasks

You can define specific configurations for executing the provisioning workflow tasks such as Implementation, Prevalidation, Postvalidation, and Rollback.

To define the configurations:

New Menu	Old Menu
1. Form the main navigation menu, select **Automation**. 2. In the **Automation** module, from the left menu, click **WORKFLOW** > **Provisioning**.	1. Form the main navigation menu, select **Settings**. 2. On the **Settings** module, from the left menu, click **Provisioning**.
Note: For more information on how to switch between menus, click here.	

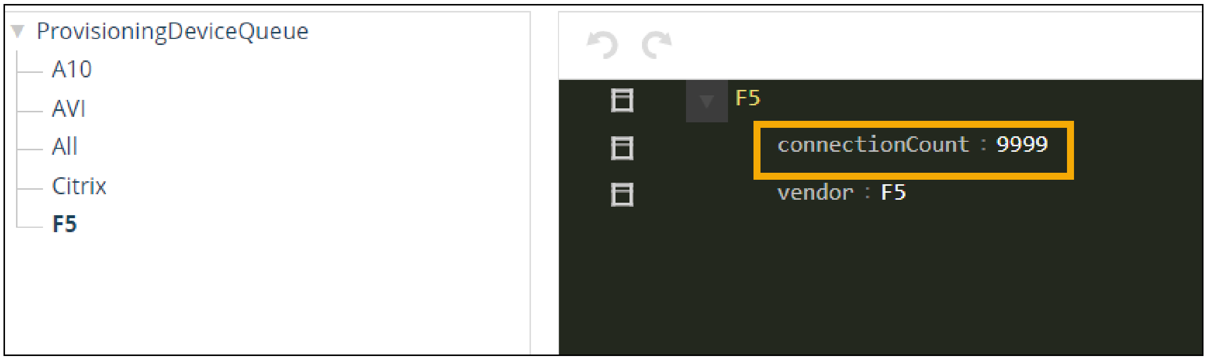
1. Under the **Work order configuration** section, define the configurations for Both or Manual or Auto of the **Implementation type**.
2. In the **Implementation type** field, select the Implementation type from the dropdown.
3. Under the **Command configuration** section, define the configurations for success or failure of the implementation task.
4. In the **Failure messages** field, enter the keywords for error responses to identify the task as failed as shown in the image below.

If the workflow execution displays any of the keywords defined above as error messages, the task will be considered as failed.

5. In the **Excluded failure messages** field, enter the keywords to be excluded for identifying the task as failed.
6. In the **Commands to wait** field, you can enter a set of commands and define a time out for the implementation task. If the task contains the configured commands, then it will time out as per the time out defined. This is useful for tasks or commands that take longer than the default time to execute.
7. In the **Time out (mins)** field, enter the timeout value for the tasks having the configured set of commands.

Note: The default value is 0.5 minutes or 30 seconds.

8. Turn on the **Continue on a command failure** toggle, to proceed in case of a command failure.
9. Turn on the **Enable Device Queue** toggle to enable queuing of device connections as per the defined connection count in the Collection.



10. Turn on the **Enable Device Session Cache** toggle, to check for any existing open sessions for the device and execute other requests in the same session.
11. Turn on the **Rollback work order** toggle to rollback configurations that are partially implemented.

Note: Configurations that were successfully implemented will be excluded from the rollback work order.

12. Under the **Rest configuration** section, define the different success status codes in the **Rest success status code** field.
13. Under the **Collection configuration** section, turn on the **Enable default collection** toggle to create a collection in the name of the device and store all the IPs in the device (using a predefined parsing logic) during the next device parsing.

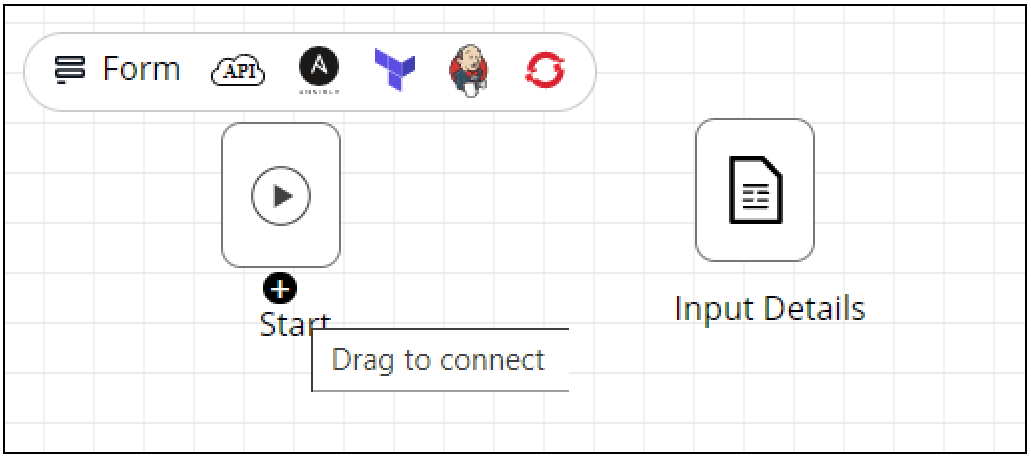
Connecting Workflow Tasks

You cannot trigger a workflow until all the tasks are connected.

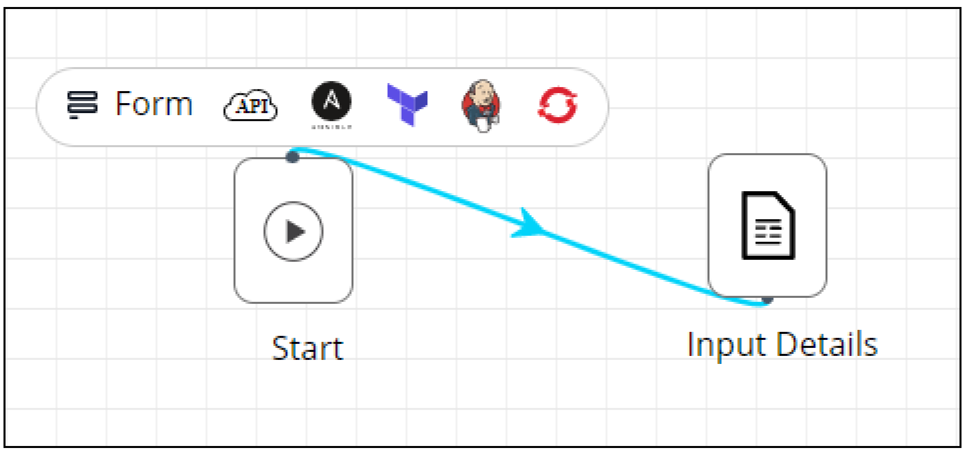
To connect the tasks:

1. Hover your mouse over the first task.

   A  **(Start)** icon displays under it.



2. Place the cursor over the **(Start)** icon and drag your mouse to connect to the next task.
   The tasks are connected.



- Customizable Connectors
- Decisions via Connectors/Links
- Connecting Workflow Tasks with Failover - RGF Flow

Customizable Connectors

Connectors or links allow you to connect one task to the other in order to stitch the workflow. The connector option allows you to customize the connectors depending on the nature of the workflow.

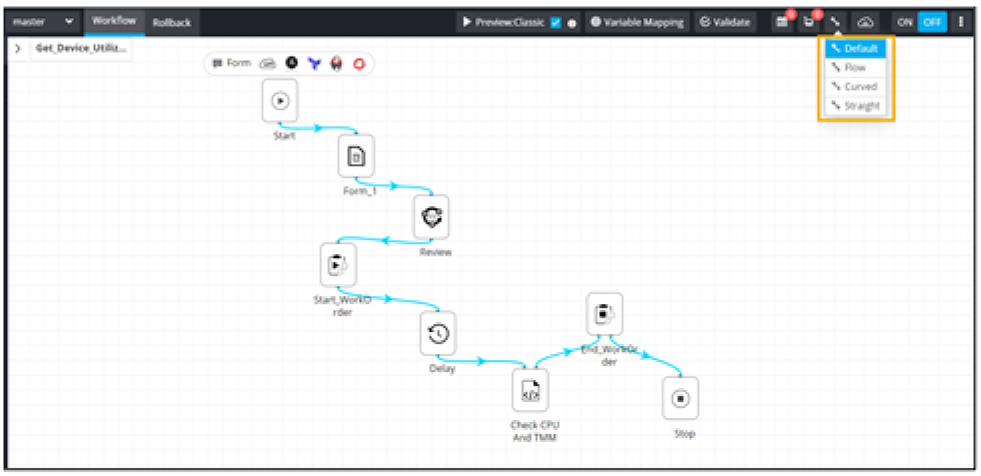
The following types of connectors are available:

- Default
- Flow

- Curved
- Straight.

To customize connectors:

1. Go to the Workflow page.
2. Open an existing workflow.
3. In the design workspace, to view the different connector options, click **(Default)** icon.
4. Select the connector from the available options.

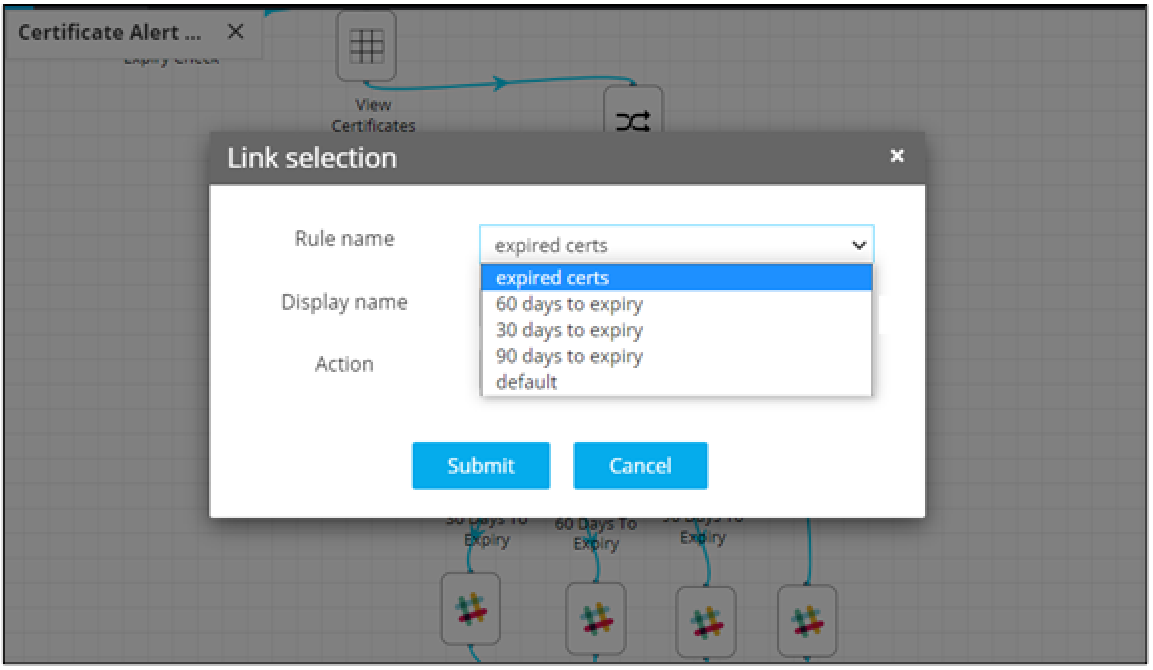


Decisions via Connectors/Links

Connectors or Links allow you to specify decisions (success, failover) based on which a workflow task can be routed intelligently. These links can be updated or deleted as per requirement.

Decisions via links can be used when constructing a workflow with task(s) that allow one or more decisions to be taken.

- **Script**: Allows one or more routes to be taken
- **If/else**: Allows maximum of two decisions to be taken
- **Switch**: Allows more than two decisions to be taken.

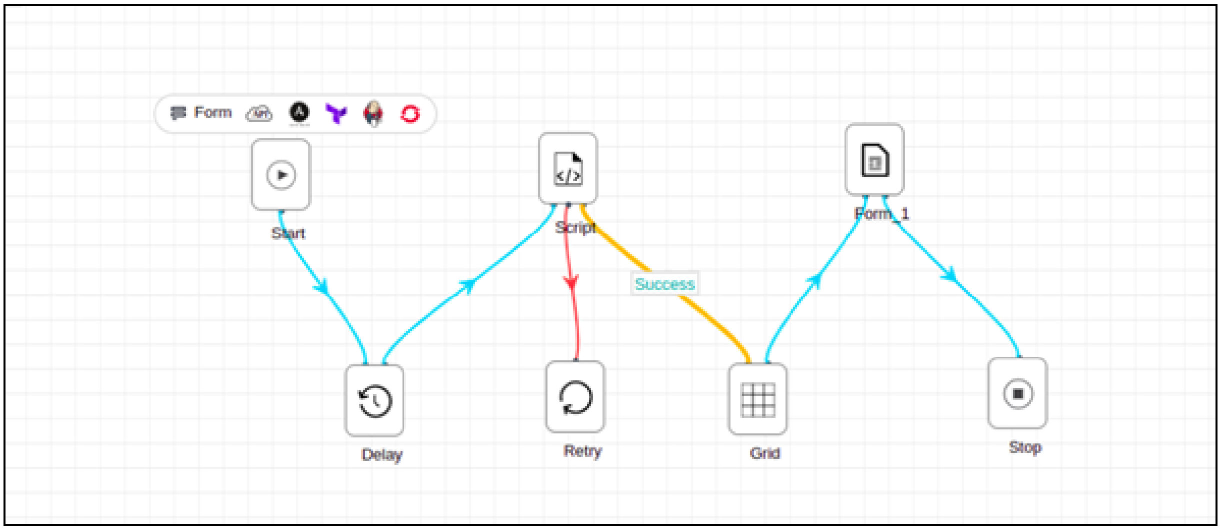


Connecting Workflow Tasks with Failover - RGF Flow

When connecting tasks in a workflow, links can be used to specify decisions (success or failure), based on which a workflow can be routed intelligently. A Script task can be used to make decisions based on the outcome of the script to perform specific workflow actions.

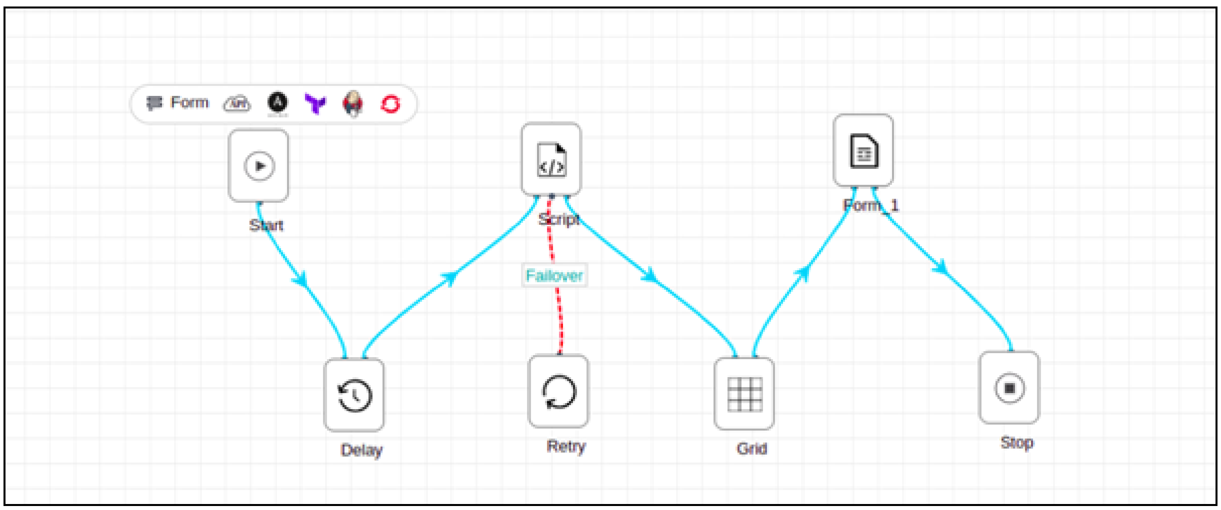
1. Design a new workflow.
2. In the **Script** task, define two outputs for ‘success’ and ‘failover’ outcomes.
3. Based on the script output, connect the links to the relevant workflow tasks.
4. In case of success, proceed to the next task.

   The **Success** link is displayed in blue color.



5. In case of failover, send an email notification and perform a retry.

   The **Failover** link is displayed in red color.



Note: The selected link is displayed in orange color.

Adding Variables to Tasks

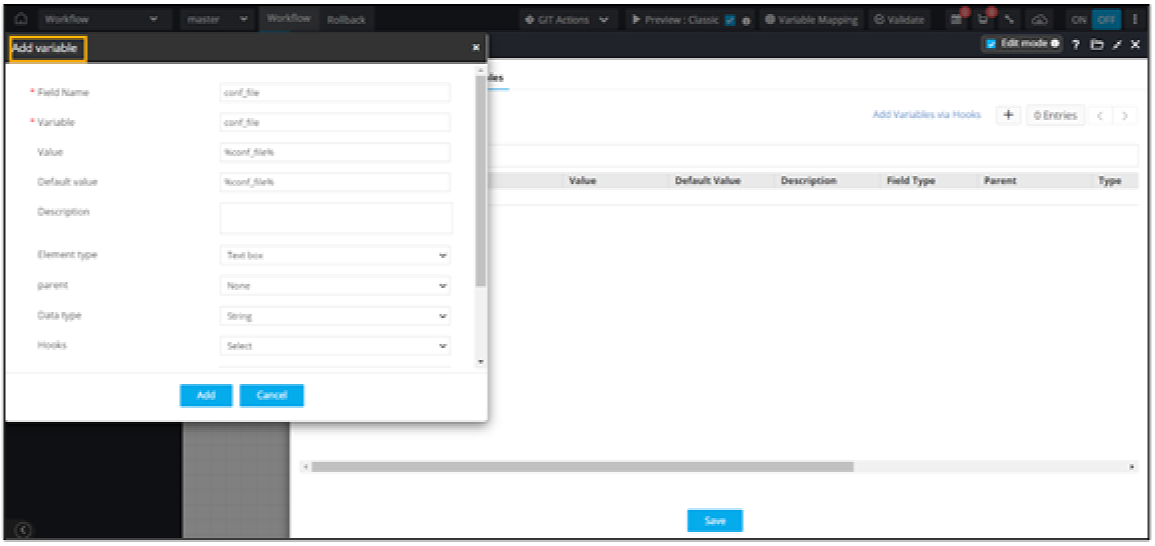
This feature allows you to add variables to tasks to support auto-generation of form in the workflow.

To add variables to a task:

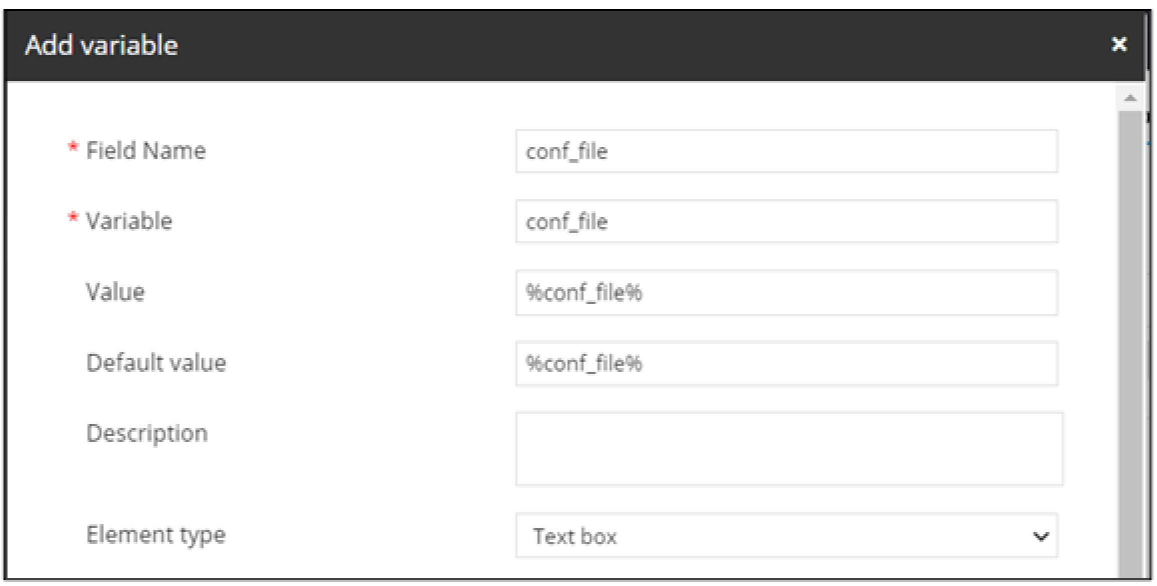
1. Double-click on any task.
2. In the task window, click **Variables**.

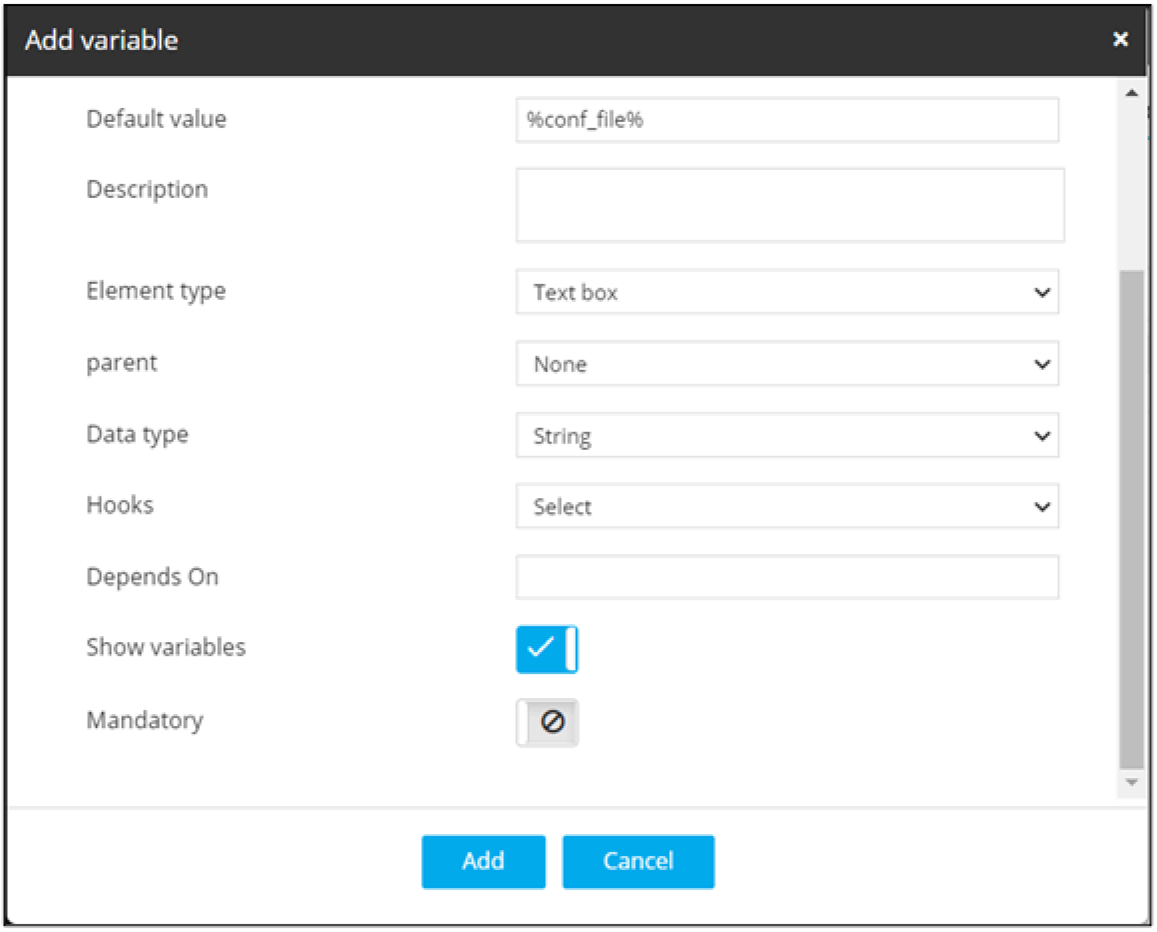
3. Under the **Variables** tab, to add a variable, click + **(Add)** icon.

   The **Add variable** window is displayed.

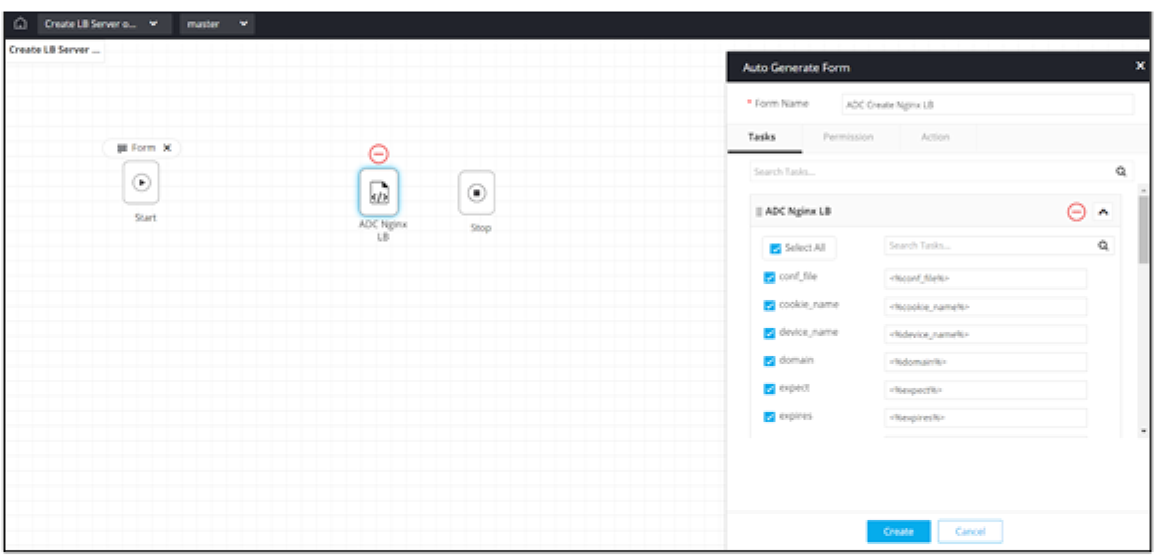


4. In the **Add variable** window, enter or select the field information based on the variable that needs to be added.

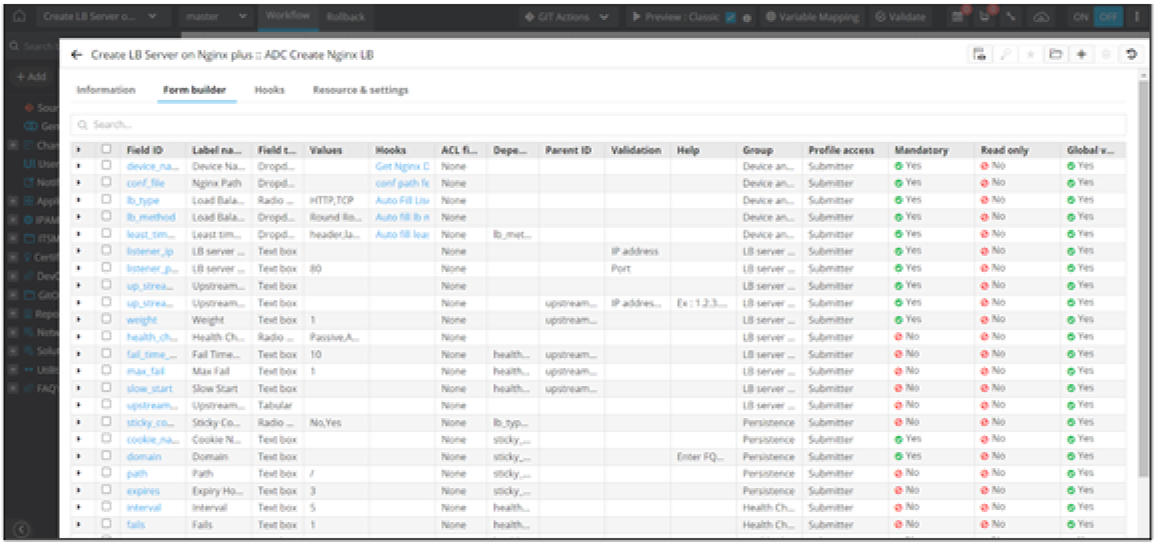




5. Click **Add**.
6. Once you have added all the variables, click **Save**.
7. Auto-generate a self-service form based on the variables defined in the Script task.



The form is auto-generated based on the variables defined in the Script task.



8. Connect the workflow tasks and enable the workflow.
9. Trigger the workflow.

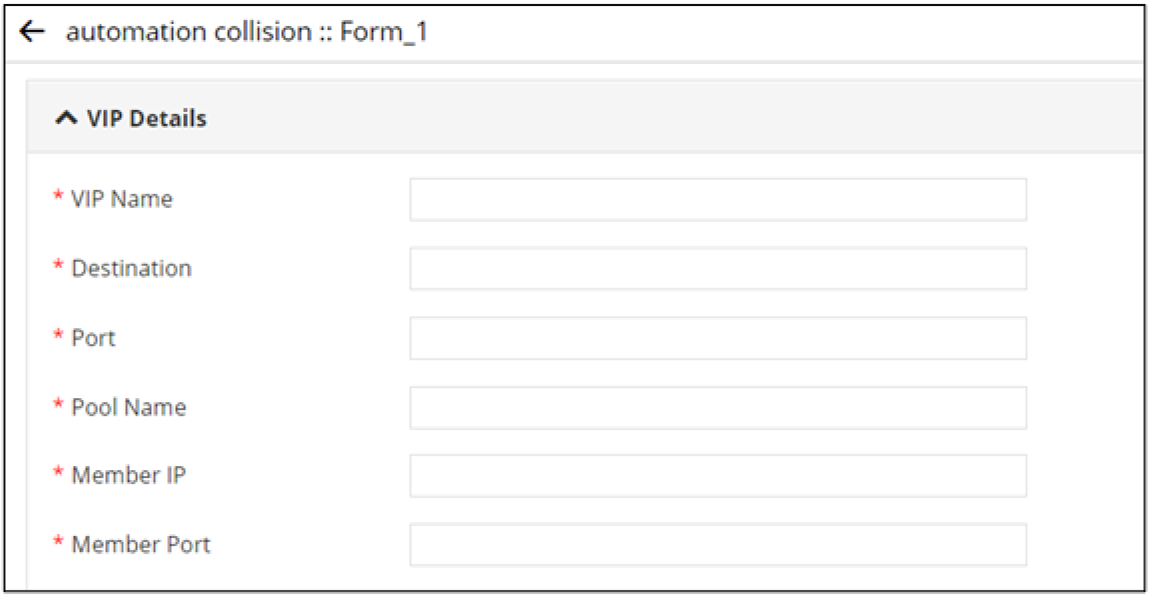
Automation Collision

This feature allows you to check for duplicate configurations across workflow requests.

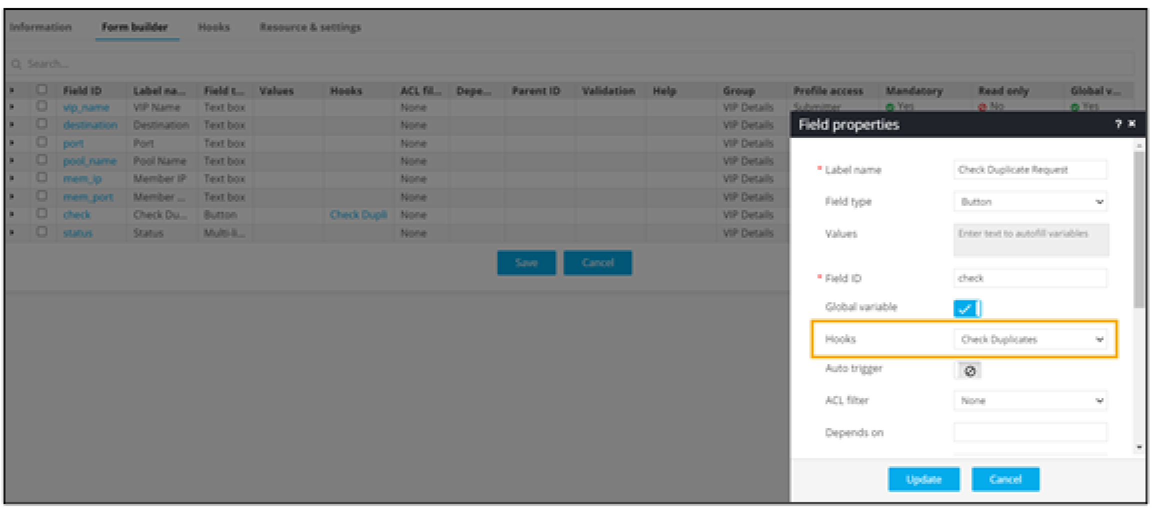
- Check all the pending workflow requests for duplicate object names such as virtual server, pool member, and monitors.
- Check all the pending workflow requests for any variables used within a workflow.
- Approve or reject the workflow requests based on duplicate configuration validation.

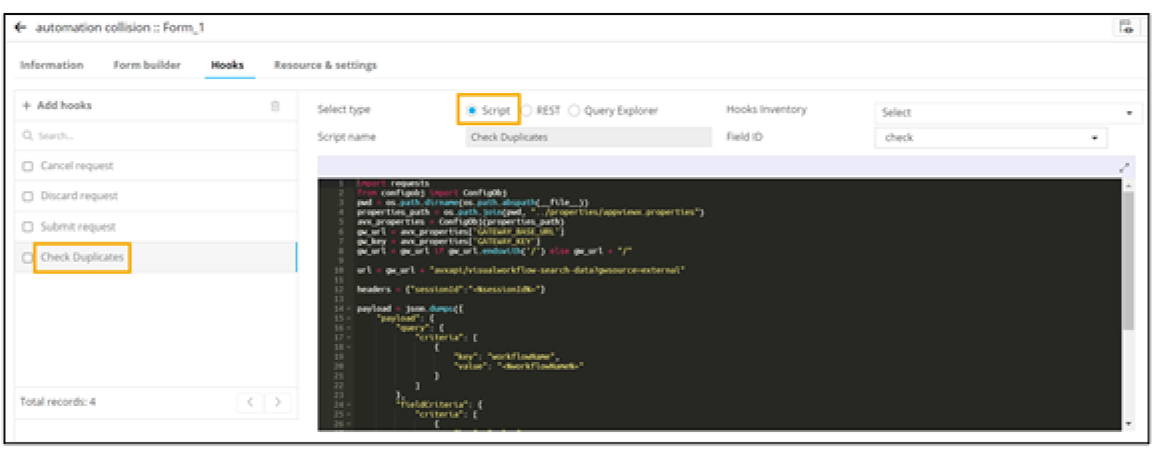
To check for duplicate configurations when designing a workflow to create a virtual server:

1. Design a workflow.
2. From the User Interface section, drag and drop a **Form** task.
3. Add the form fields for creating a virtual server.



4. Add a button field to check for duplicate configurations using a Script logic.





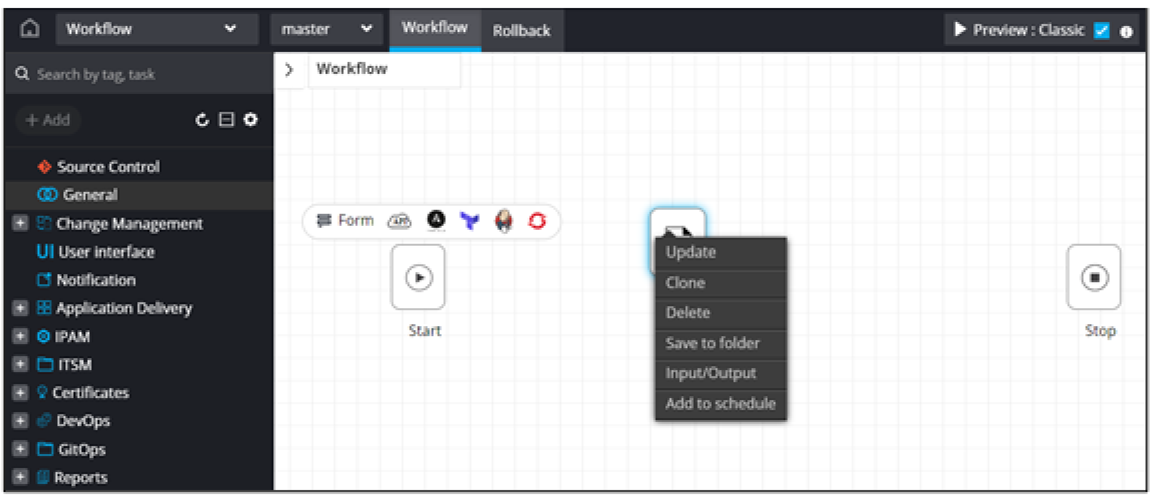
5. Connect the workflow tasks and enable the workflow.
6. Trigger the workflow.

   Workflow is executed successfully.

Workflow Task Actions

You can right-click any workflow task to perform the following actions:

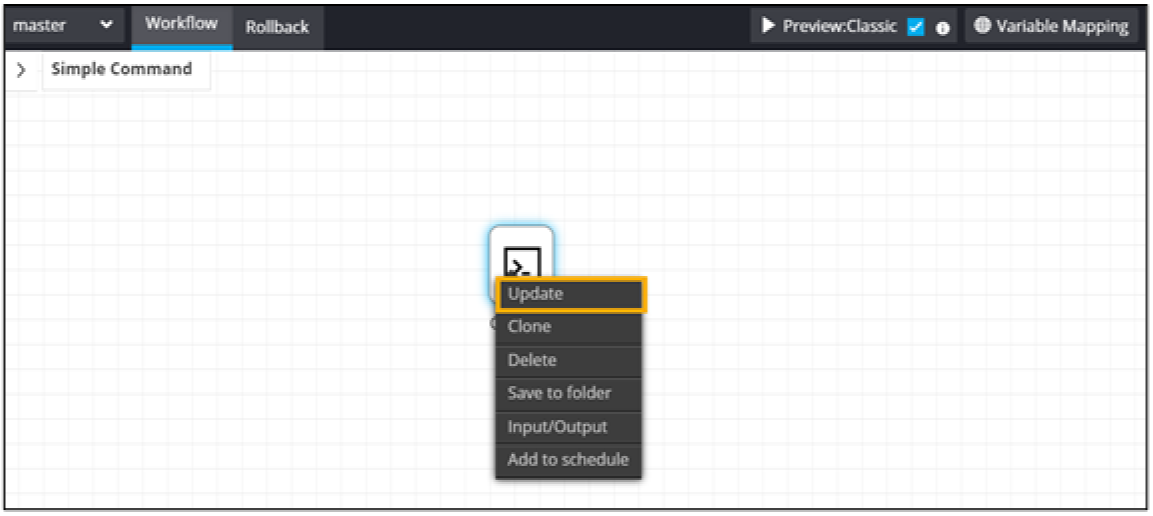
- **Update**: Allows you to update configuration settings within the task.
- **Clone**: Allows you to clone the selected task.
- **Delete**: Allows you to delete the selected task.
- **Save to folder**: Allows you to save the task to an existing or new folder.
- **Input/Output**: Allows quick and easy variable mapping.
- **Add to schedule**: Allows you to add the task as a scheduled job.



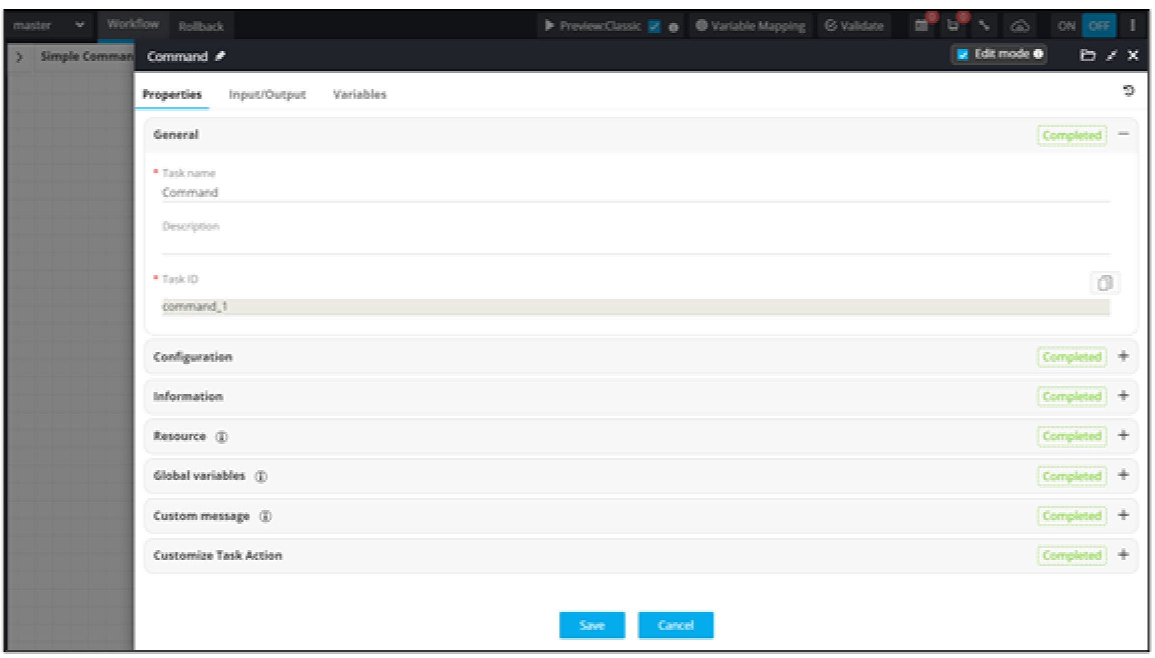
- Updating a Task
- Cloning a Task
- Deleting a Task
- Saving and Reusing a Task
- Input/Output Variable Mapping

Updating a Task

1. Right-click the task to see options.
2. Click **Update**.



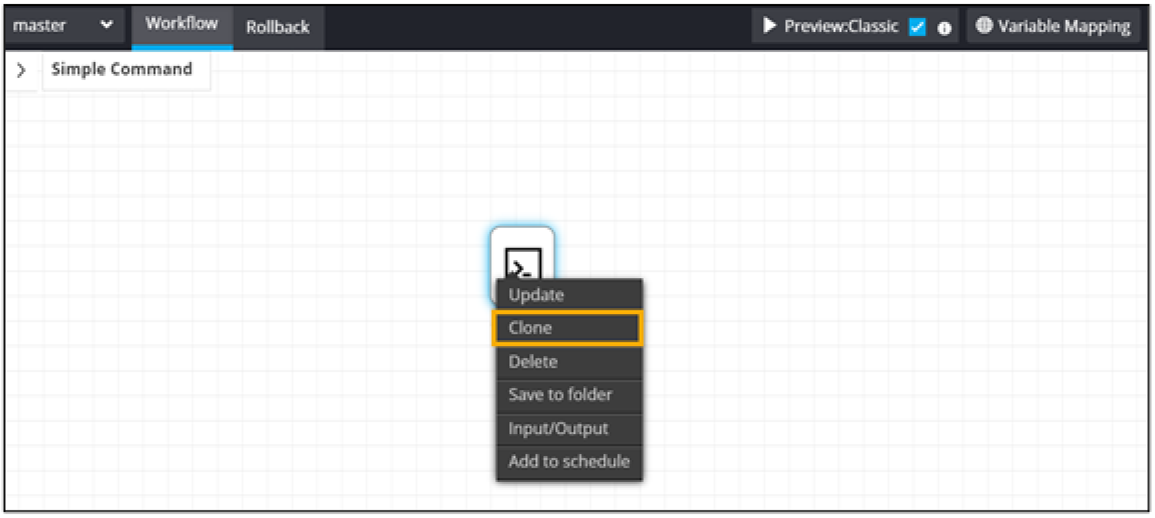
3. Edit the task properties to update the task.



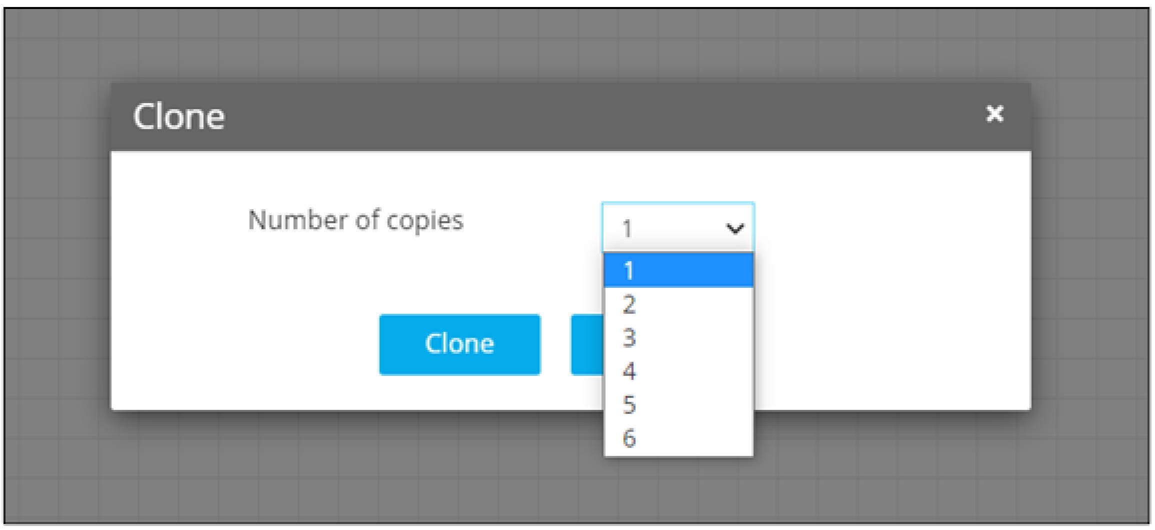
Cloning a Task

You can clone a task in the workspace instead of having to drag and drop the same task from the menu.

1. Right-click a task to see the options.
2. Click **Clone**.

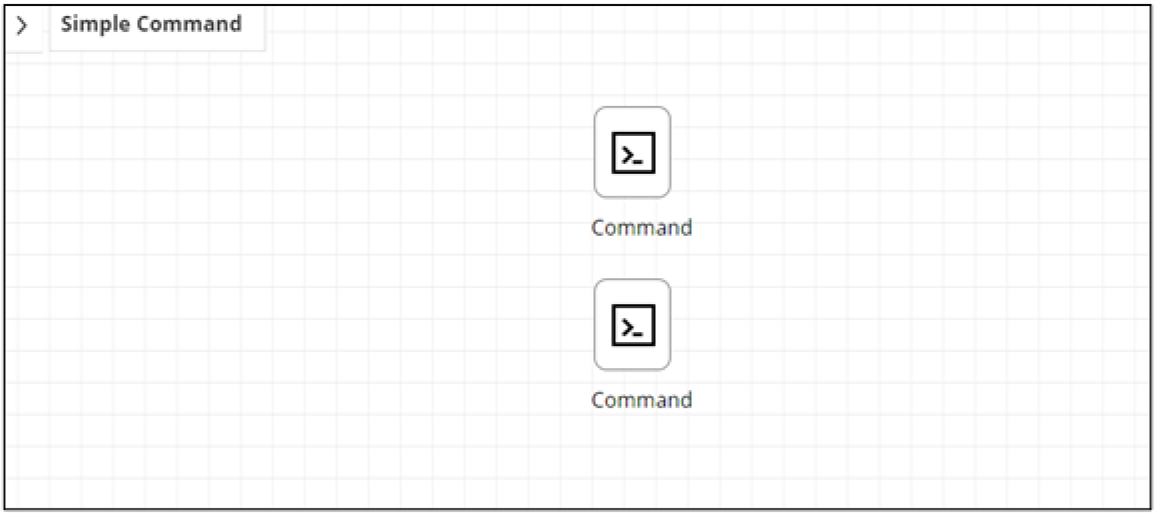


3. In the **Clone** window, select the number of copies from the dropdown list.



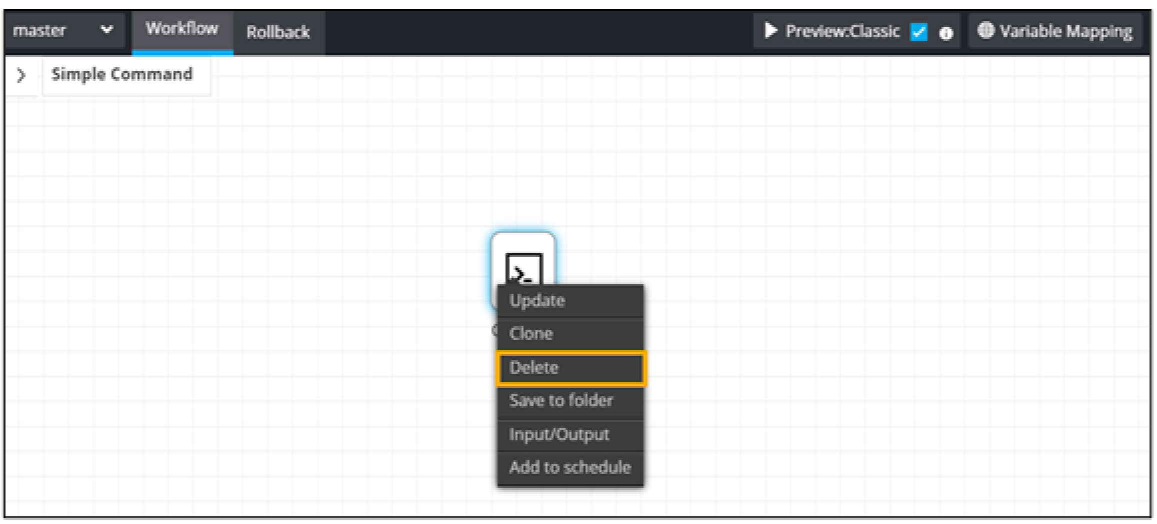
4. Click **Clone**.

A copy of the task is created in the workspace.



Deleting a Task

1. Right-click on the task to view options.
2. Click **Delete**.



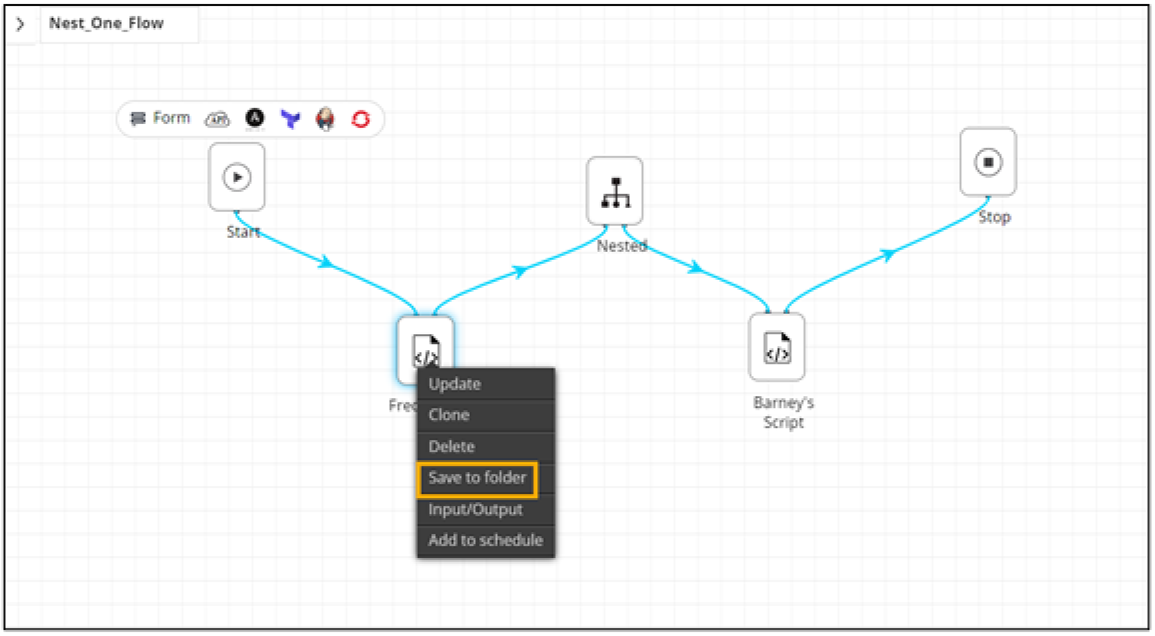
3. Click **Yes** in the **Confirmation** pop-up window.

   The task is deleted.

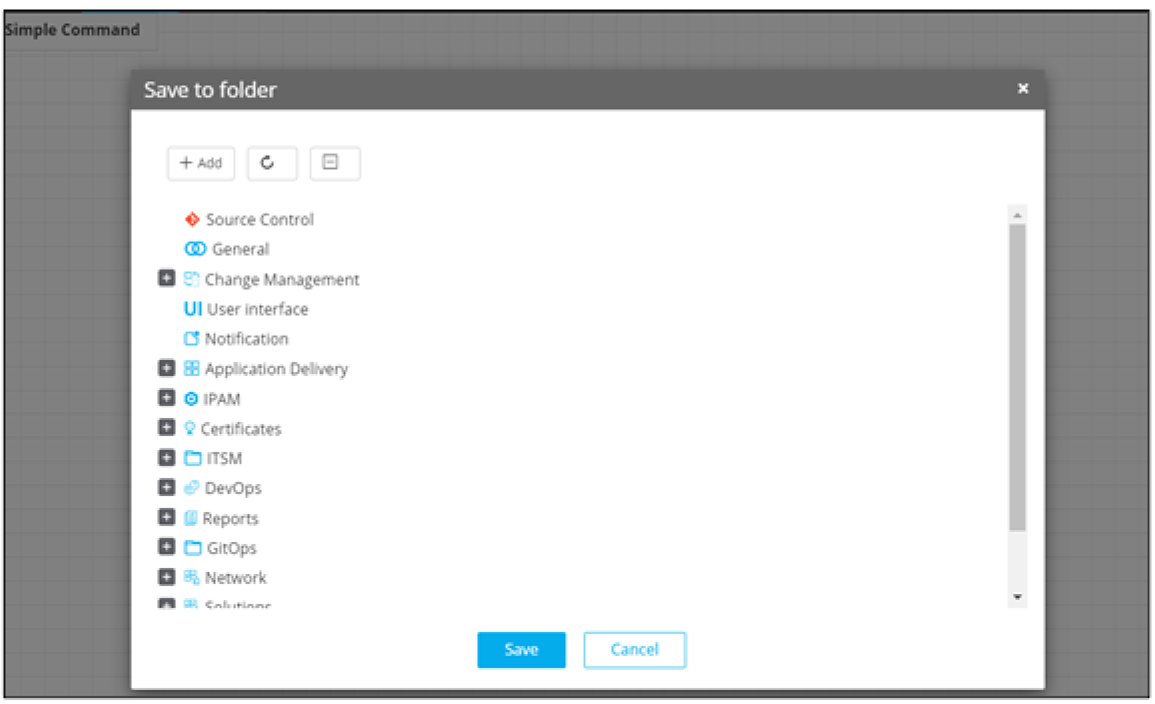
Saving and Reusing a Task

You can save specific tasks to a folder(s) in order to import/reuse them in other workflows.

1. Open an existing workflow in the Workflow Studio.
2. Right-click on the task to be saved.
3. From the options displayed, select **Save to folder**.

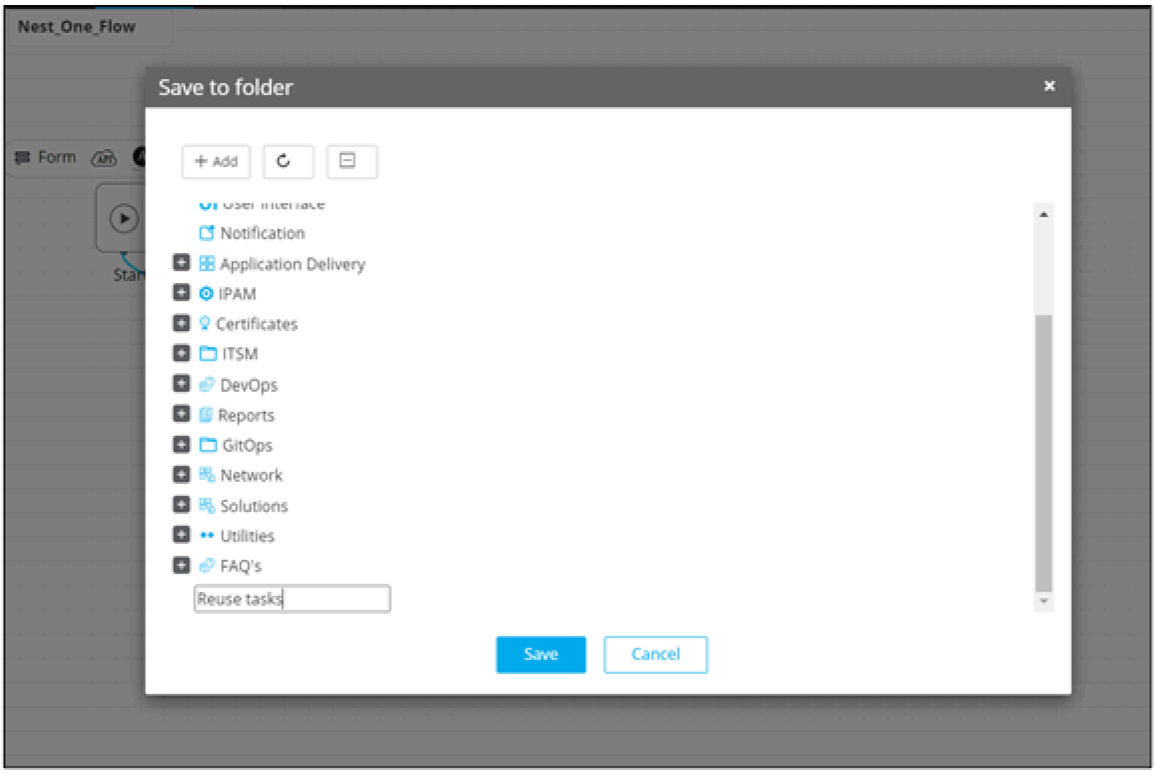


4. To save the task to an existing folder, in the **Save to folder** window, select a folder from the list.



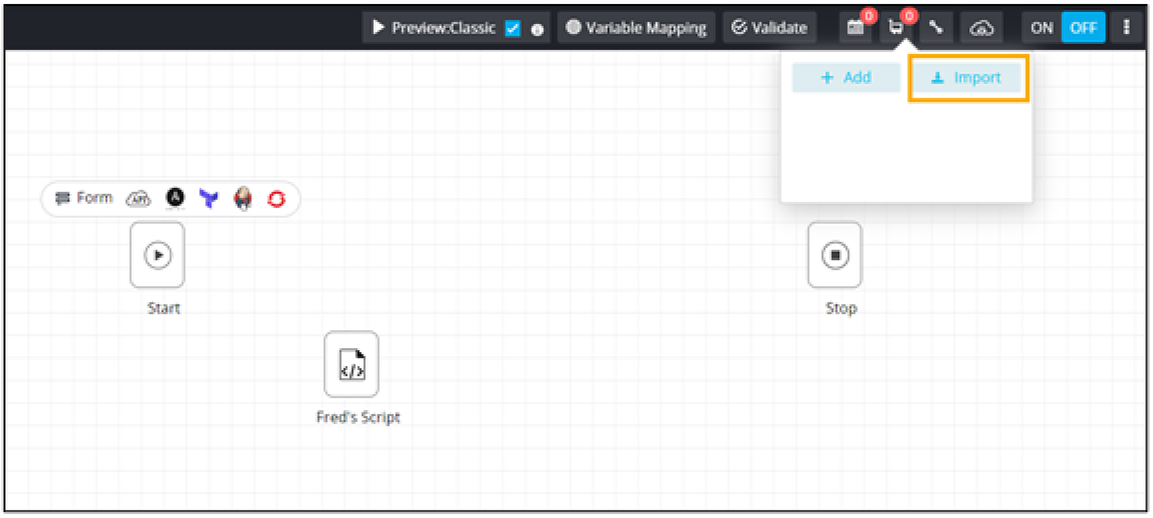
Note: When naming a folder, the same naming conventions must be followed as those when naming a workflow. If a folder is created without providing a name, it will show up as **untitled**.

5. Click **Save**.
6. To save the task to a new folder, click **Add**.
7. Enter the folder name and click **Save**.

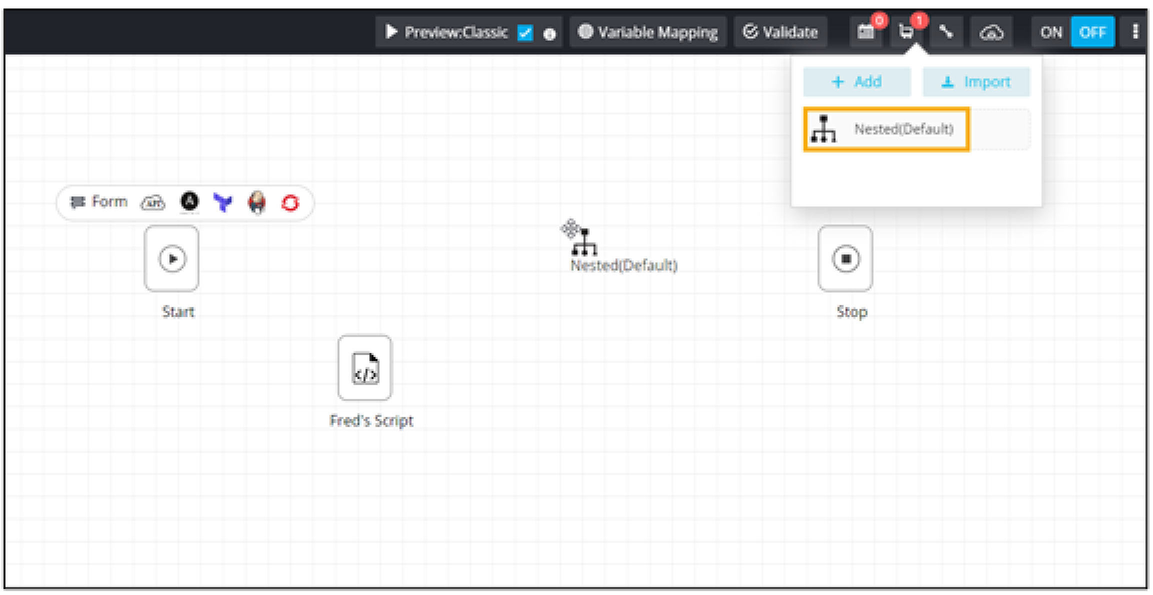


The new folder is added to the left menu.

8. Exit the workflow.
9. Design a new workflow.
10. Drag and drop the task saved to the folder in the previous workflow and reuse in the current workflow.
11. To reuse a workflow, click  **(Cart)** icon.
12. From the options displayed, click **Import**.

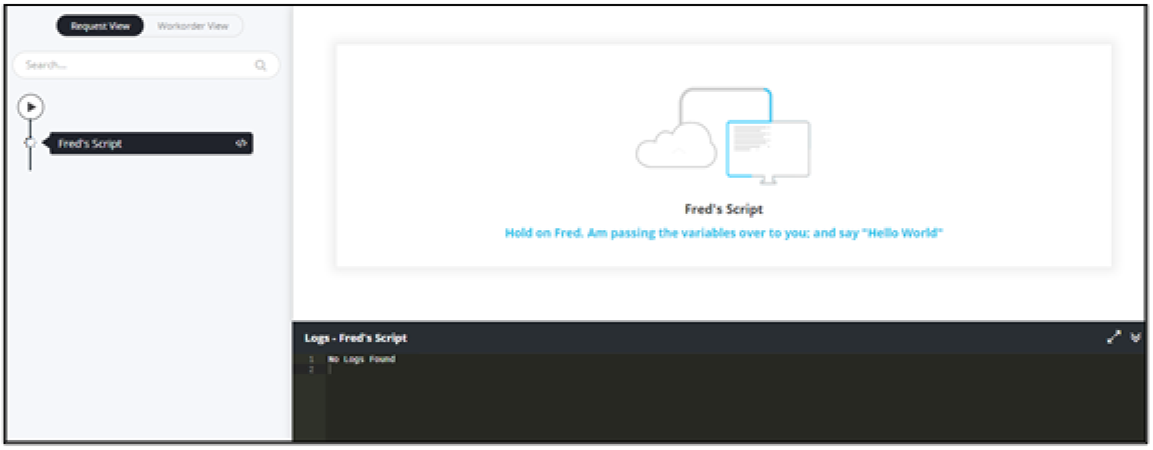


13. In the **Import flows**, window, select the workflow(s) and click **Import**.
14. Drag and drop the imported workflow into the workspace.



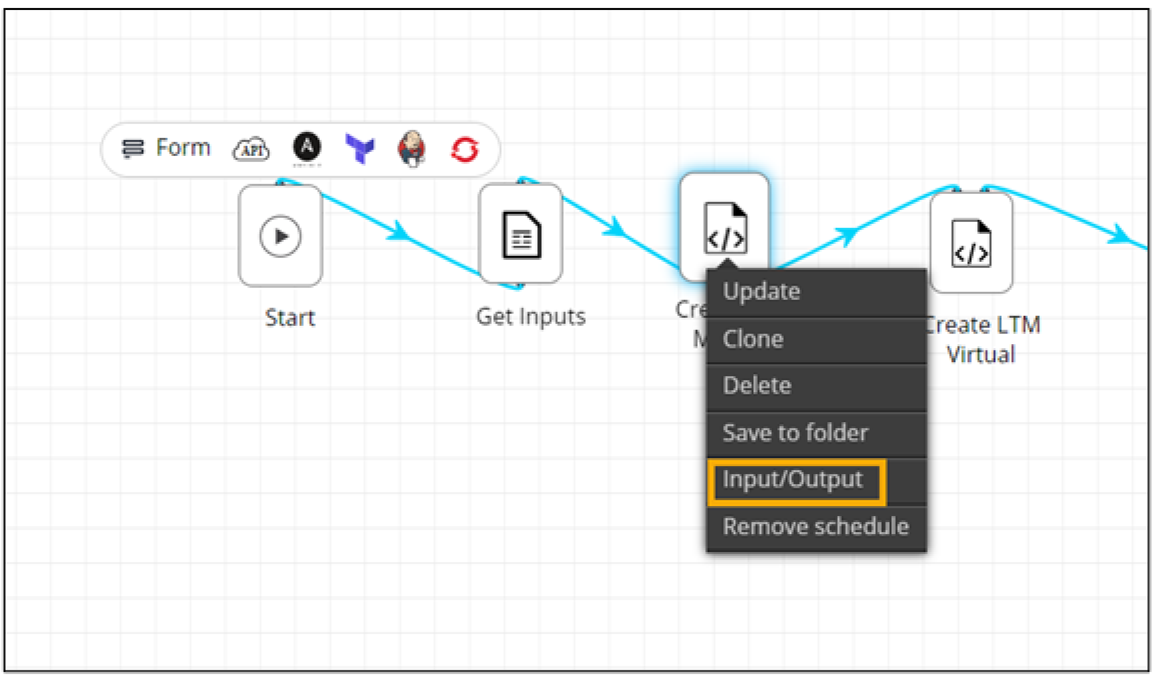
15. Refer the Global variables if needed.
16. Connect the workflow tasks and enable the workflow.
17. Trigger the workflow.

Workflow executed with the reused task.

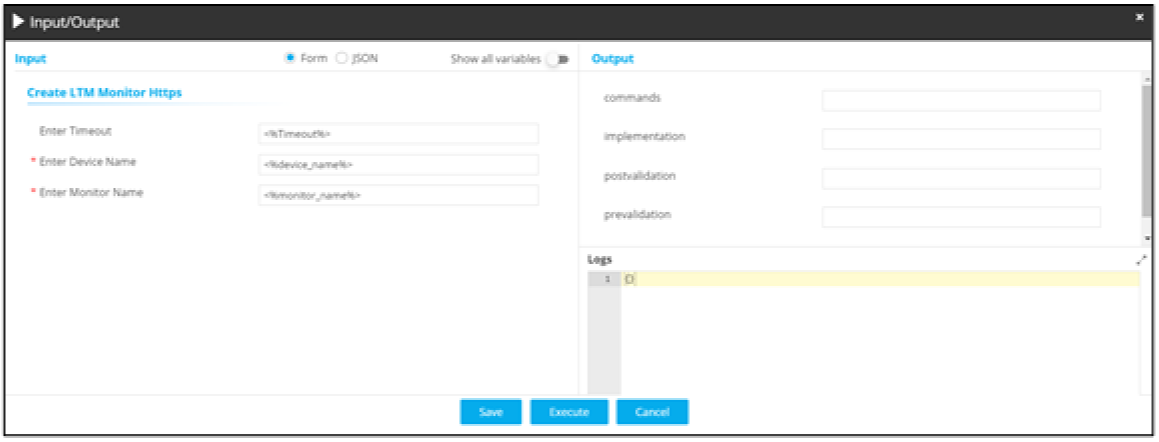


Input/Output Variable Mapping

1. To see the mapping of input/output variables in a task, right-click on the task to see the options.
2. Select **Input/Output** from the list of options.



The input/output variables defined in the task are displayed.



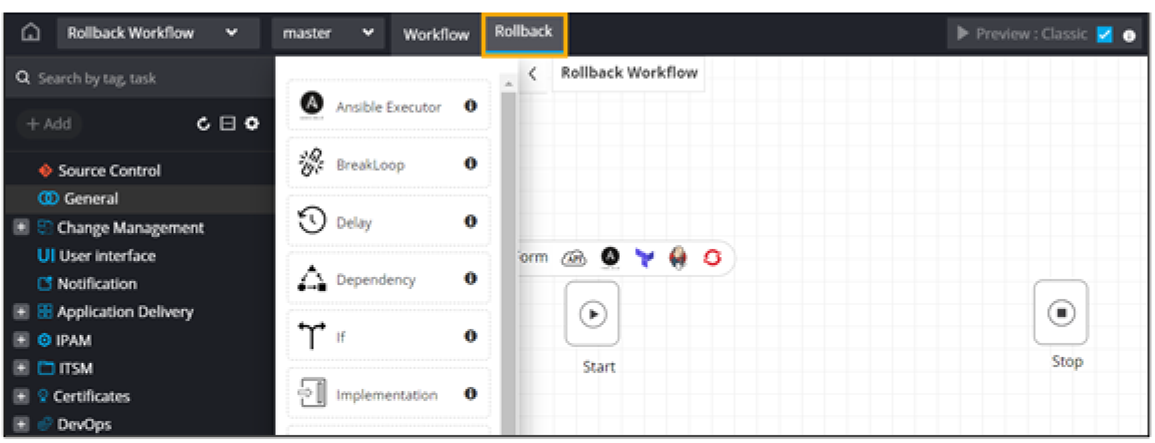
Rollback Workflow

You can either design new rollback workflow(s) or import existing rollback workflow(s) under a parent workflow. A manual rollback can be triggered by accessing Workflow Request and performing a right-click against a workflow request ID.

- Manual Rollback
- Auto Rollback

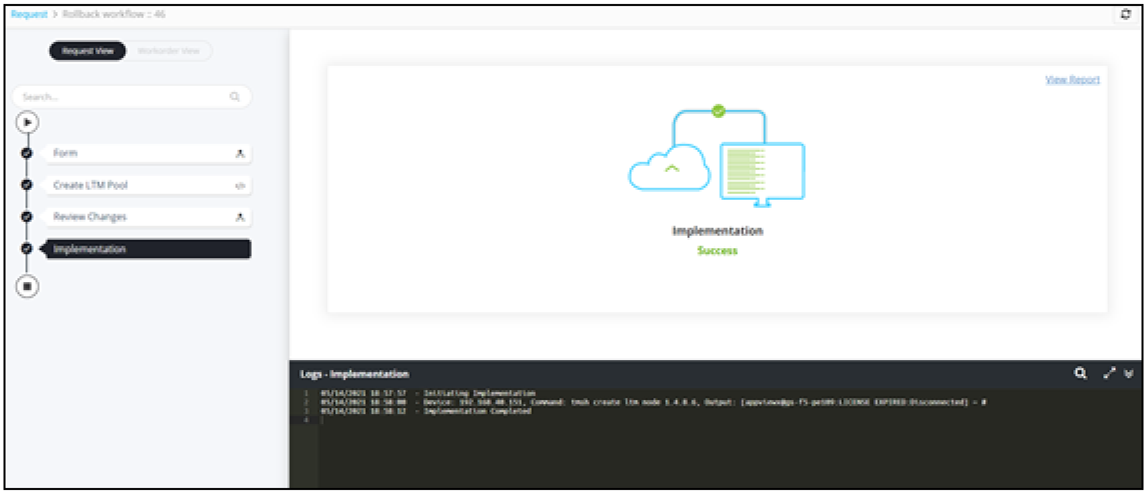
Manual Rollback

1. Design a workflow using workflow tasks.
2. To add a rollback workflow, from the menu on the top, click **Rollback**.

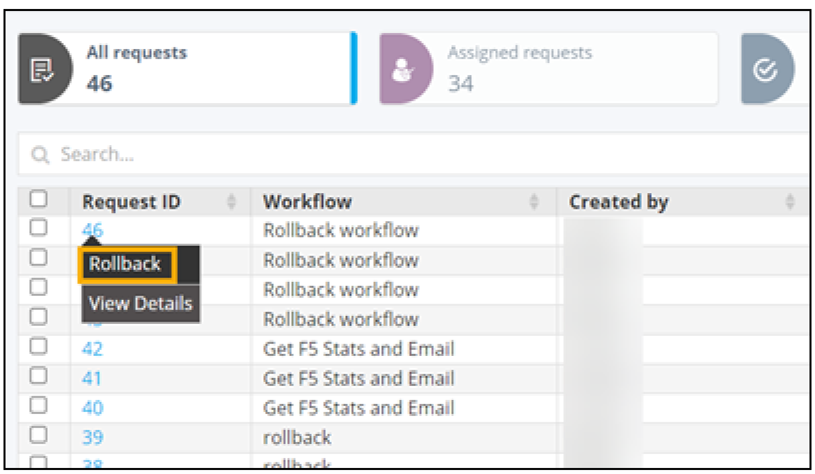


3. From the General section, drag and drop the **Rollback** task.
4. Connect the workflow tasks and enable the workflow.
5. Trigger the workflow.

   Workflow is executed successfully.

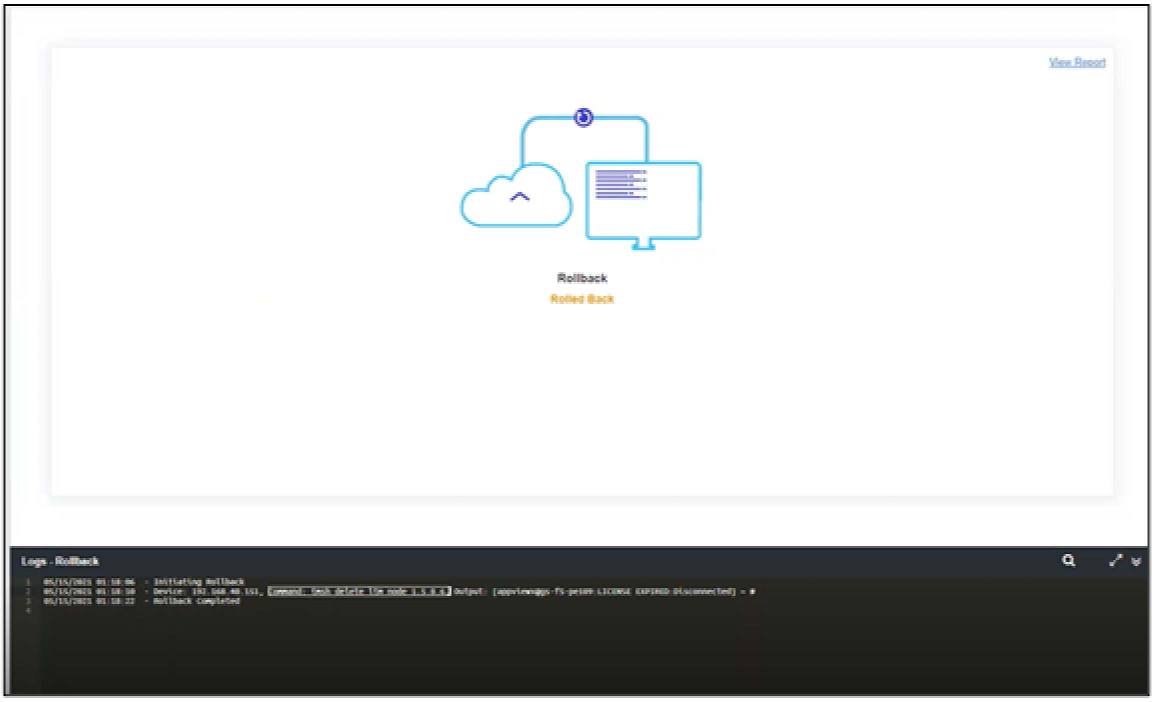


6. On the Request : All page. right-click on the workflow Request ID and select **Rollback**.



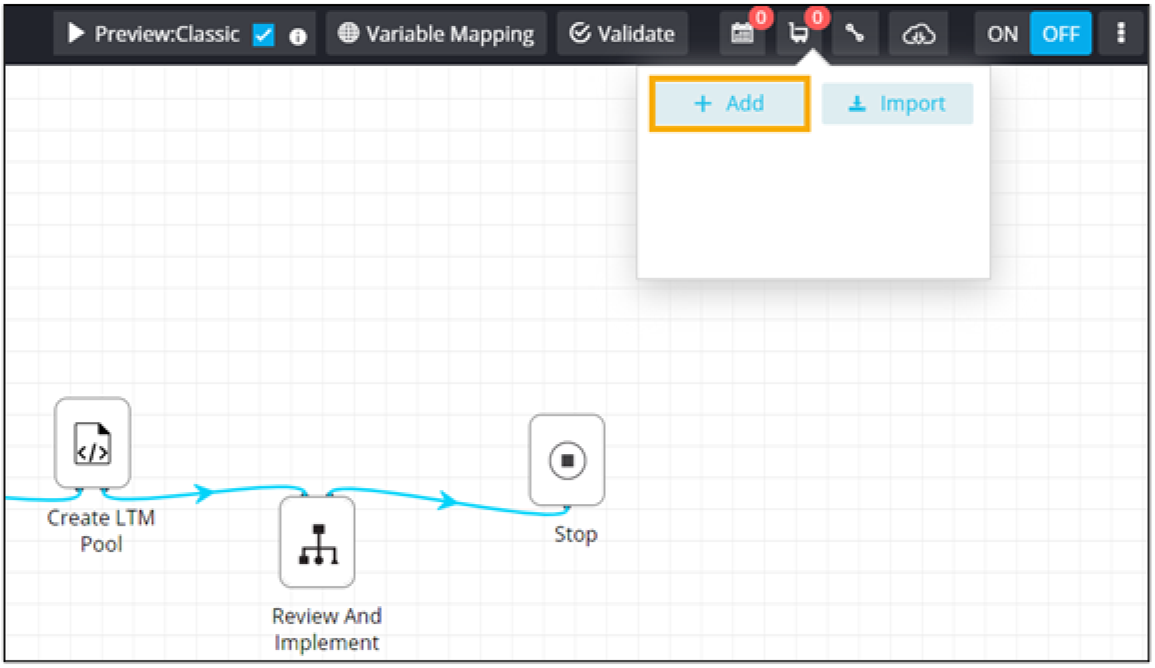
7. In the Rollback window, select **Rollback type** as **Request**.

Workflow is **Rolled Back**.

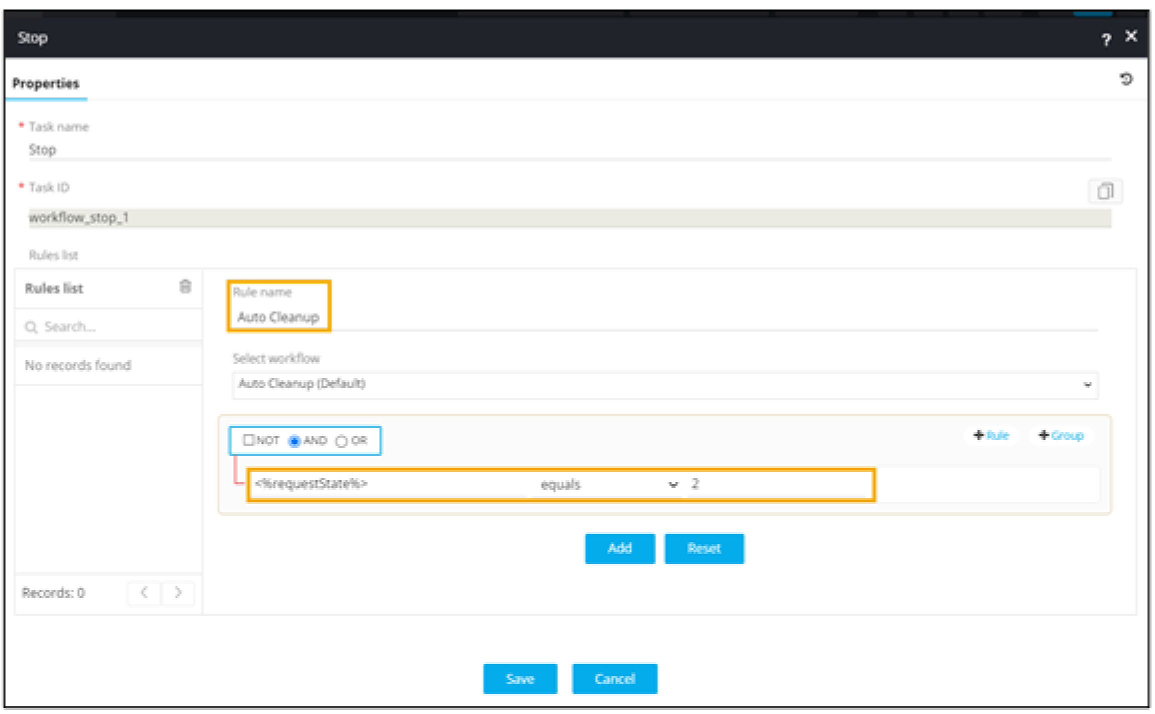


Auto Rollback

1. Design a workflow using workflow tasks.
2. To create a subflow, from the command bar on the top, click **(Cart)** icon.
3. From the options displayed, click **Add**.



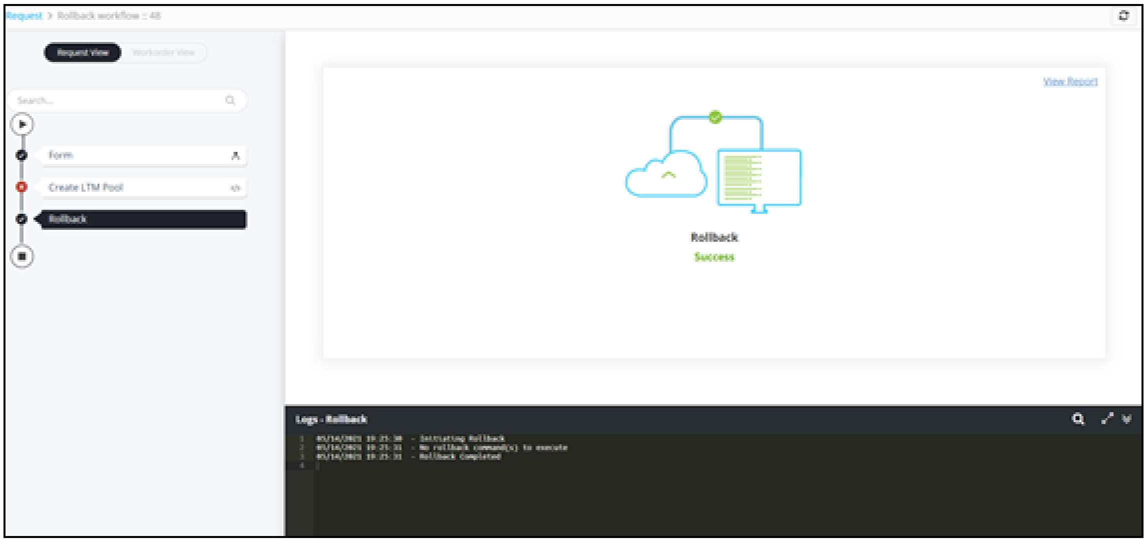
4. In the New subflow window, enter a workflow **Name**.
5. From the **General** section, drag and drop the **Rollback** task.
6. Double-click on the **Stop** task of your main workflow.
7. In the **Stop** task window, define a rule to trigger the auto rollback.



8. Click **Add**.

9. Connect the workflow tasks and enable the workflow.
10. Trigger the workflow.

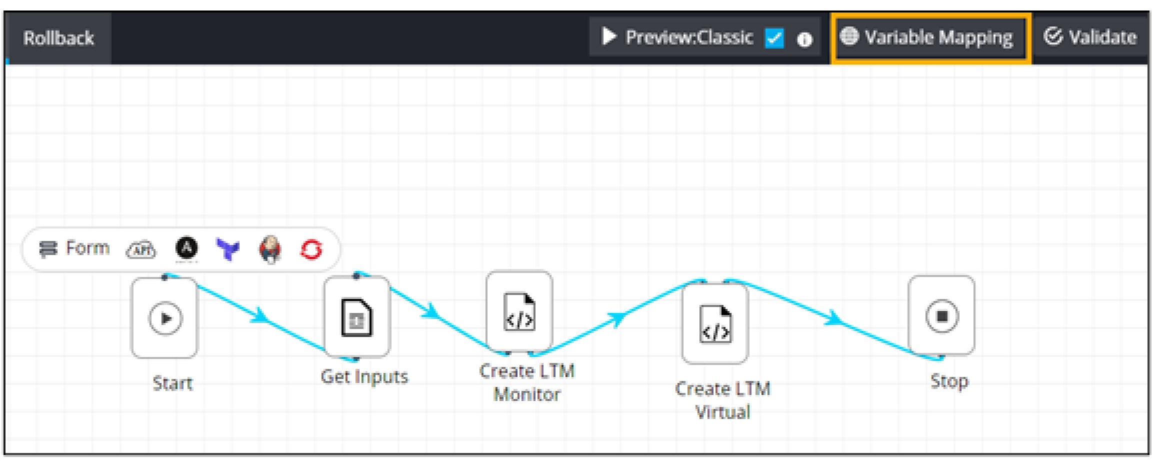
    Auto Rollback is initiated based on workflow state.



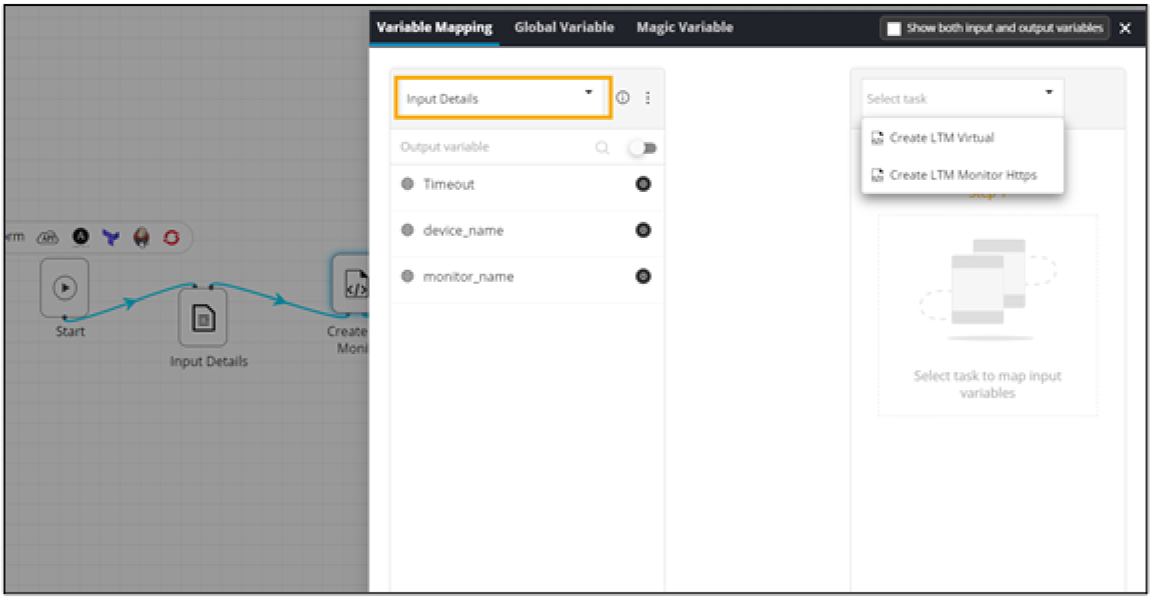
Variable Mapping

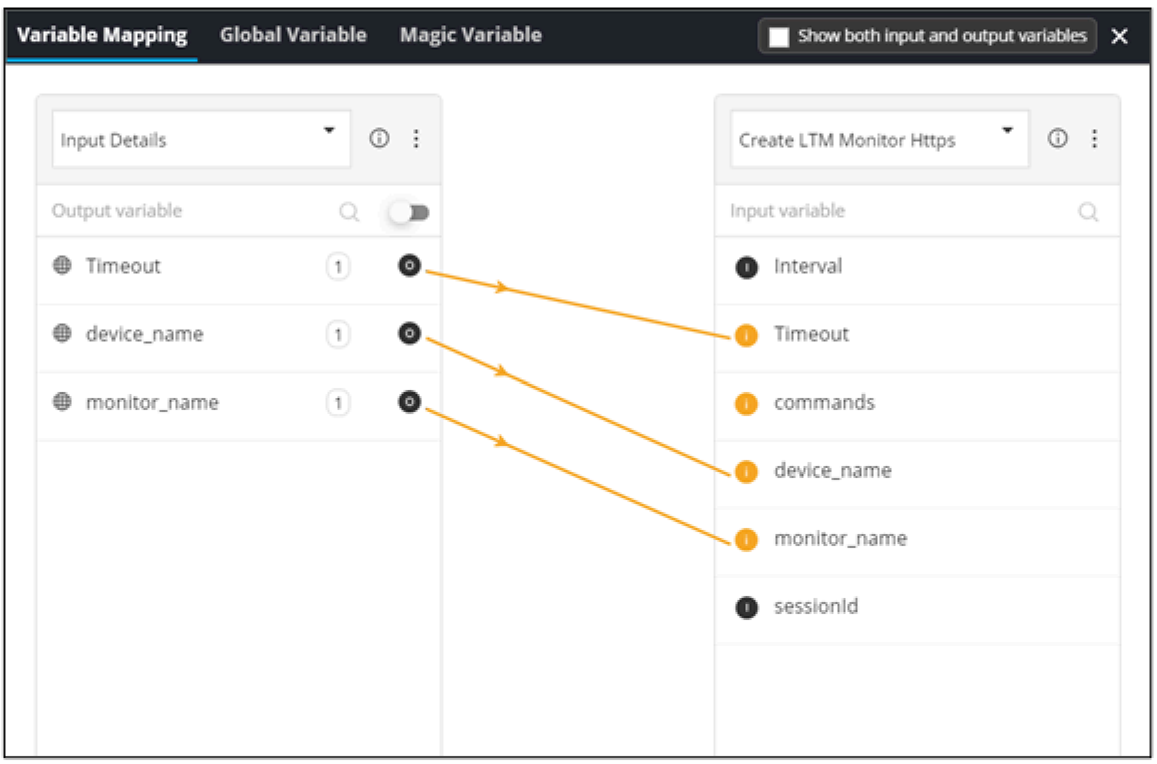
Variable mapping allows you to map the variables between tasks. You can select one or more tasks in order to map variables and get a quick view of the number of variable associations across tasks.

1. Design a workflow or open an existing workflow,
2. From the command bar on the top, click **Variable Mapping**.

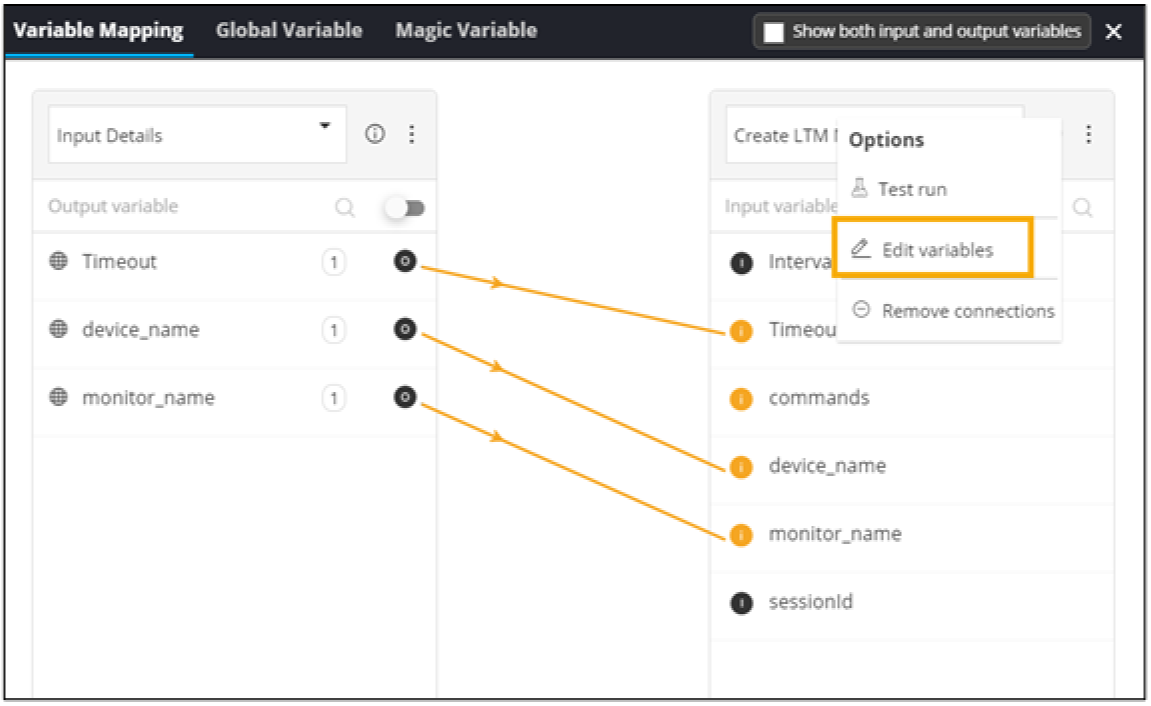


3. Under **Variable Mapping**, select the tasks for variable mapping.

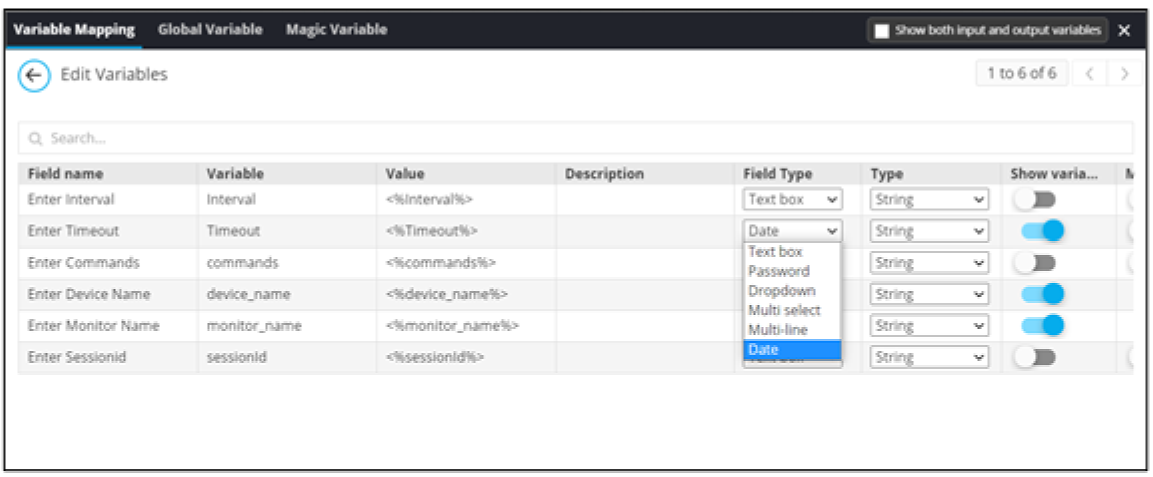




4. To see options, from the top right corner of the window, click ⋮.
5. To make changes in the field parameters, select **Edit variables**.



6. Edit the parameters of the variables as required.

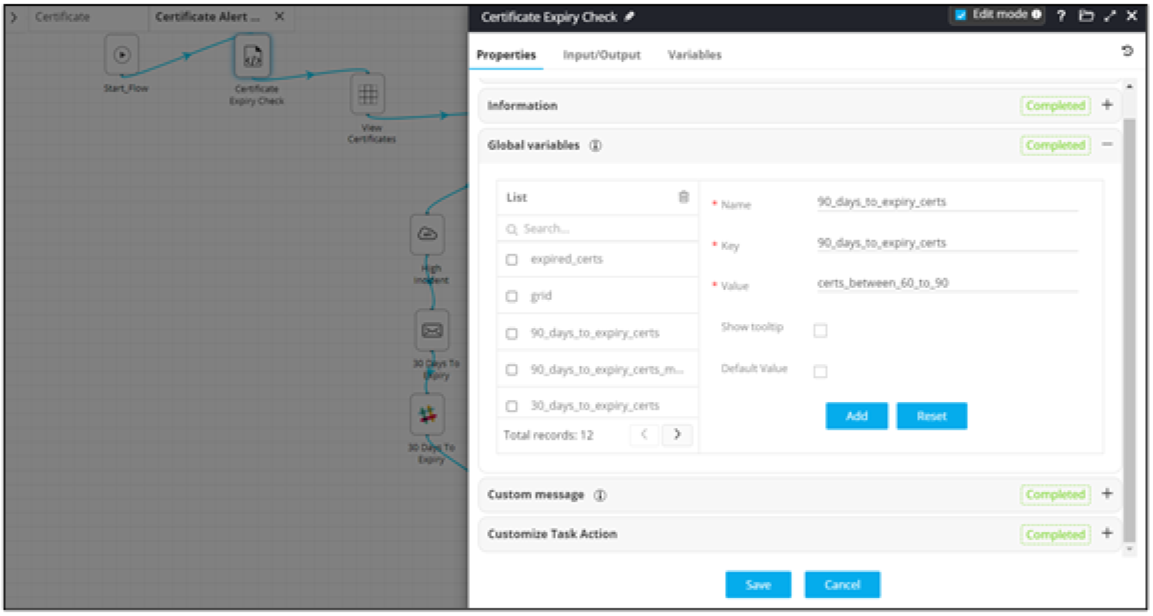


- Global Variables
- Magic Variable

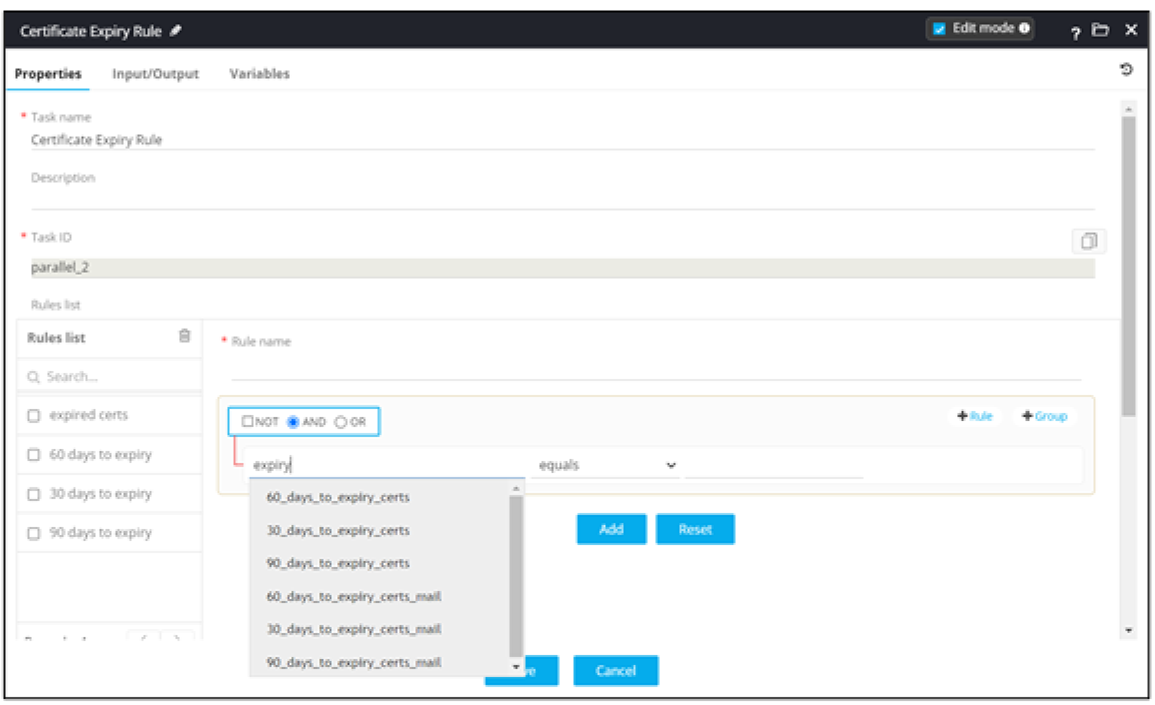
Global Variables

Variables can be declared globally and referenced across any stage of the workflow thus allowing data to flow seamlessly across the entire workflow process.

1. Design a workflow.
2. Click on any workflow task.
3. In the task window, under **Properties**, in the **Global Variables** tab, define the **Key** and **Value**.



4. Click **Add**.
5. Auto-populate and refer the global variables using keyword match.

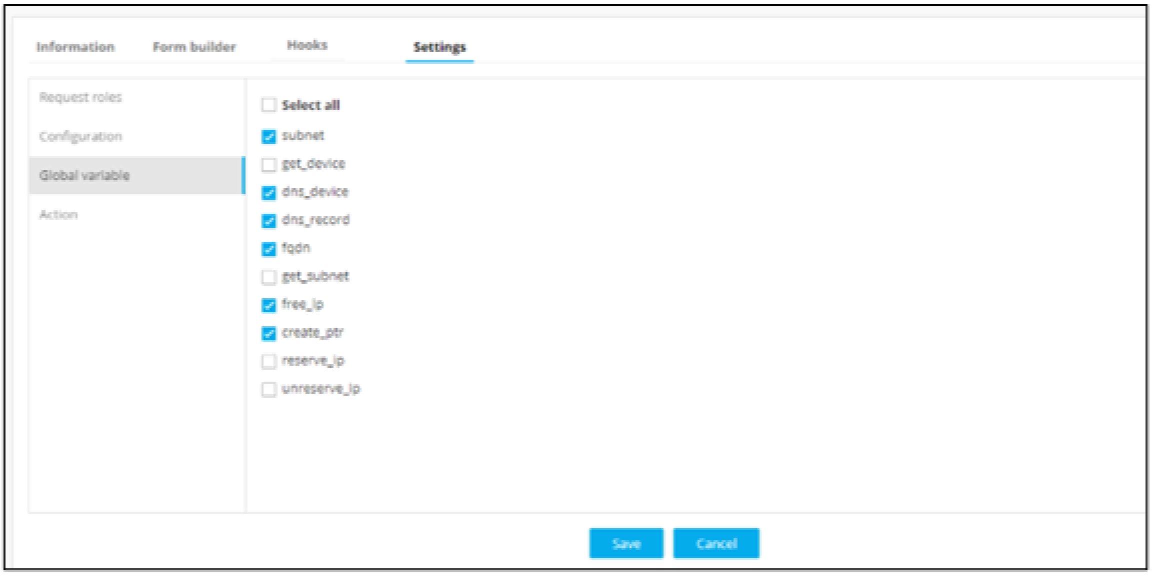


- Declaring Variables from Tasks
- Referring Variables

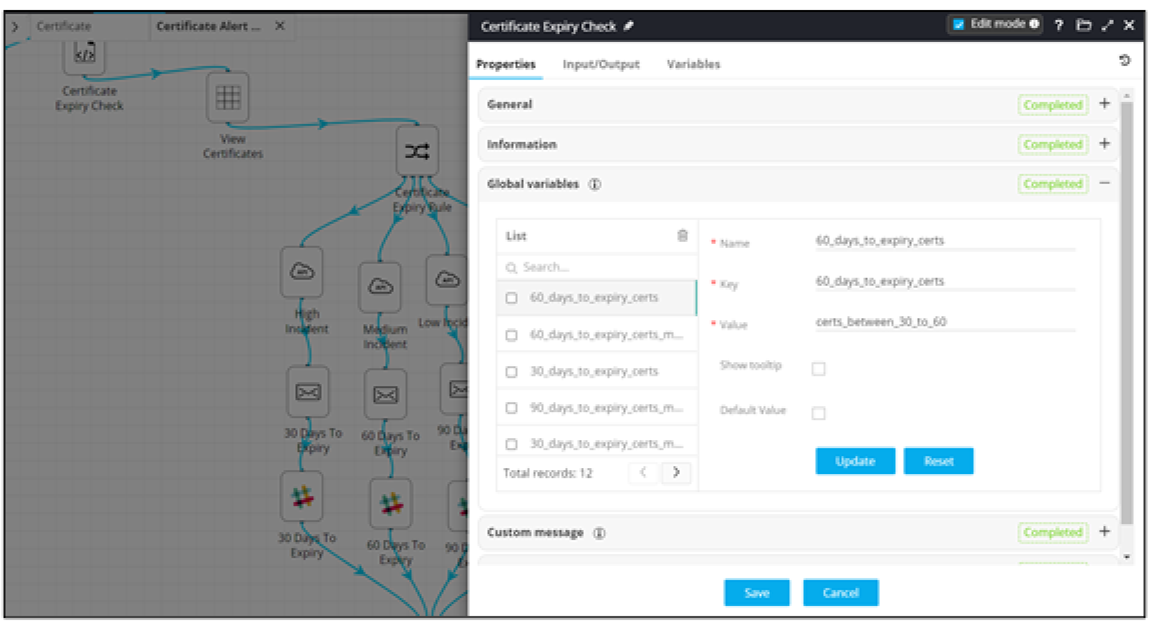
- [Global Variable Inventory]
- [Configuring Tooltips]

Declaring Variables from Tasks

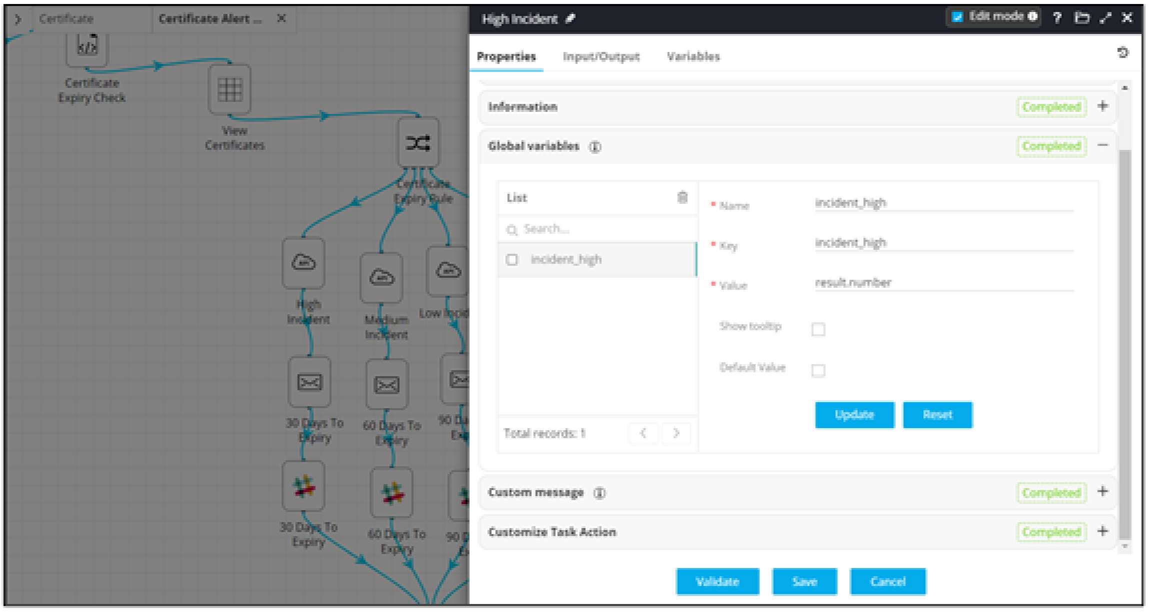
- Declaring variables from a Form task



- Declaring variables from a Script task



• Declaring variables from a REST API task.



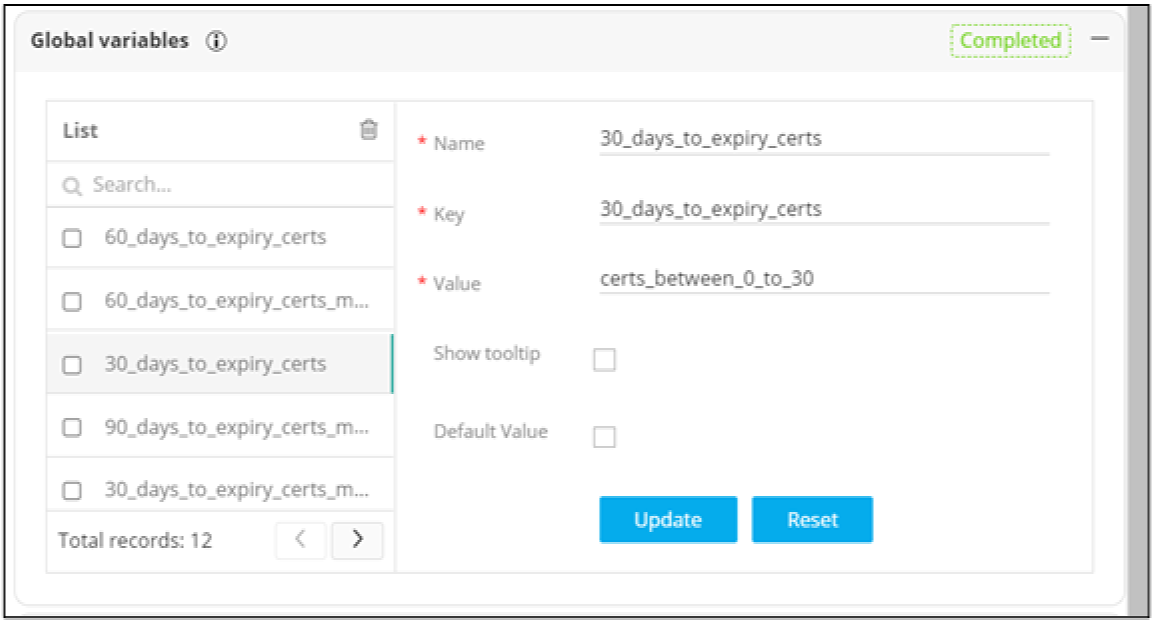
Referring Variables

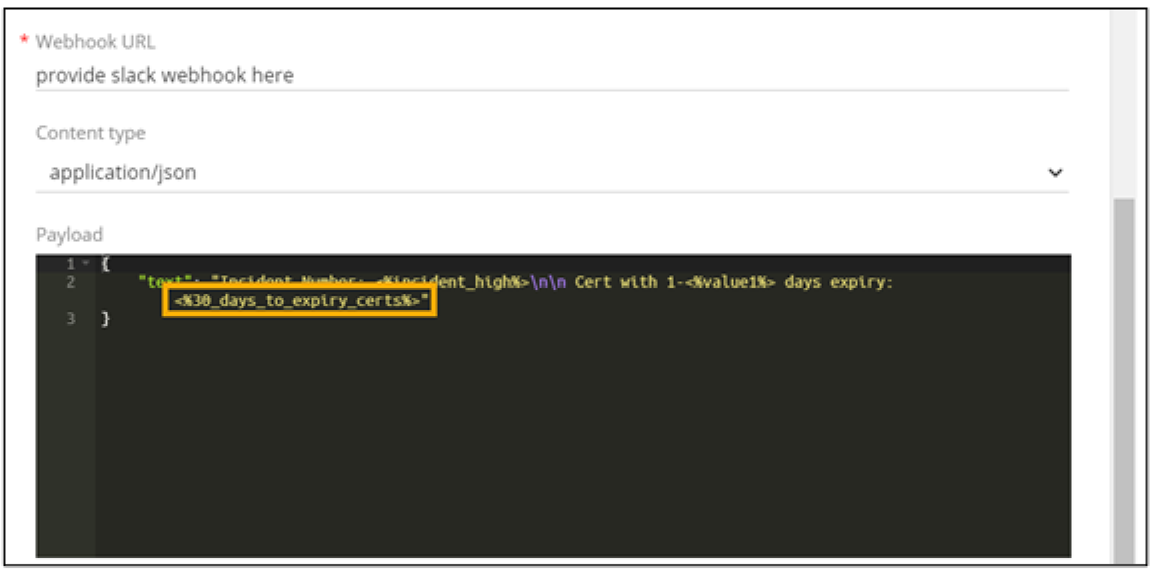
Once declared, global variables can be simply referred across any other stage using the following syntax:

<%variable%>

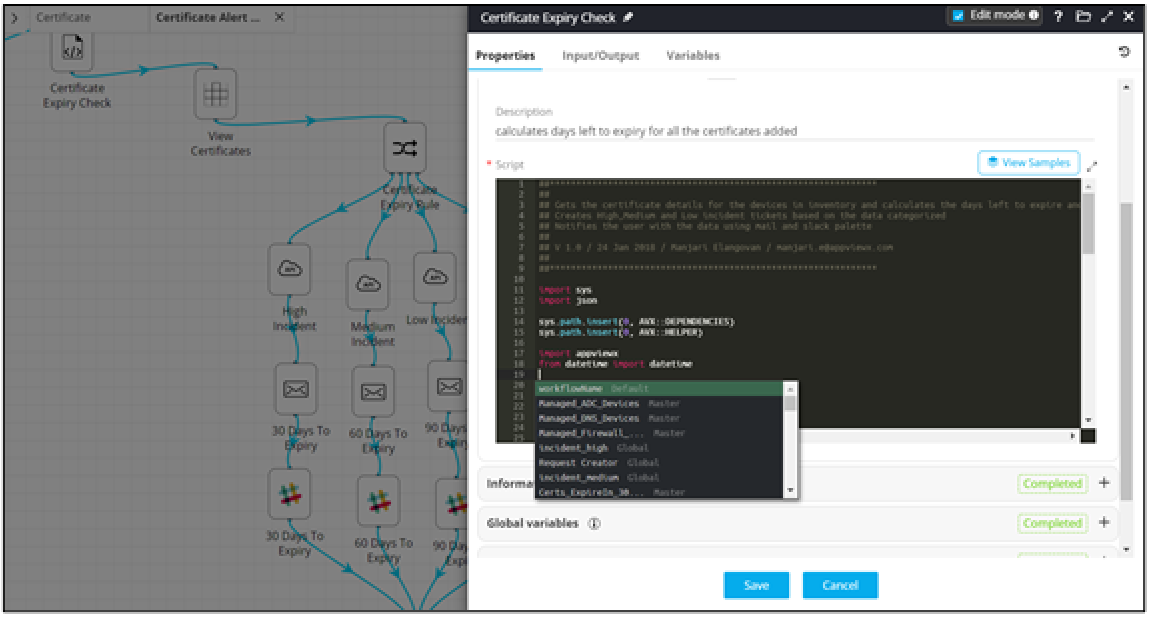
For example, once the global variable “30_days_to_expiry_certs” is declared in the Script task, it can be referenced anywhere in the workflow as

<%30_days_to_expiry_certs%>





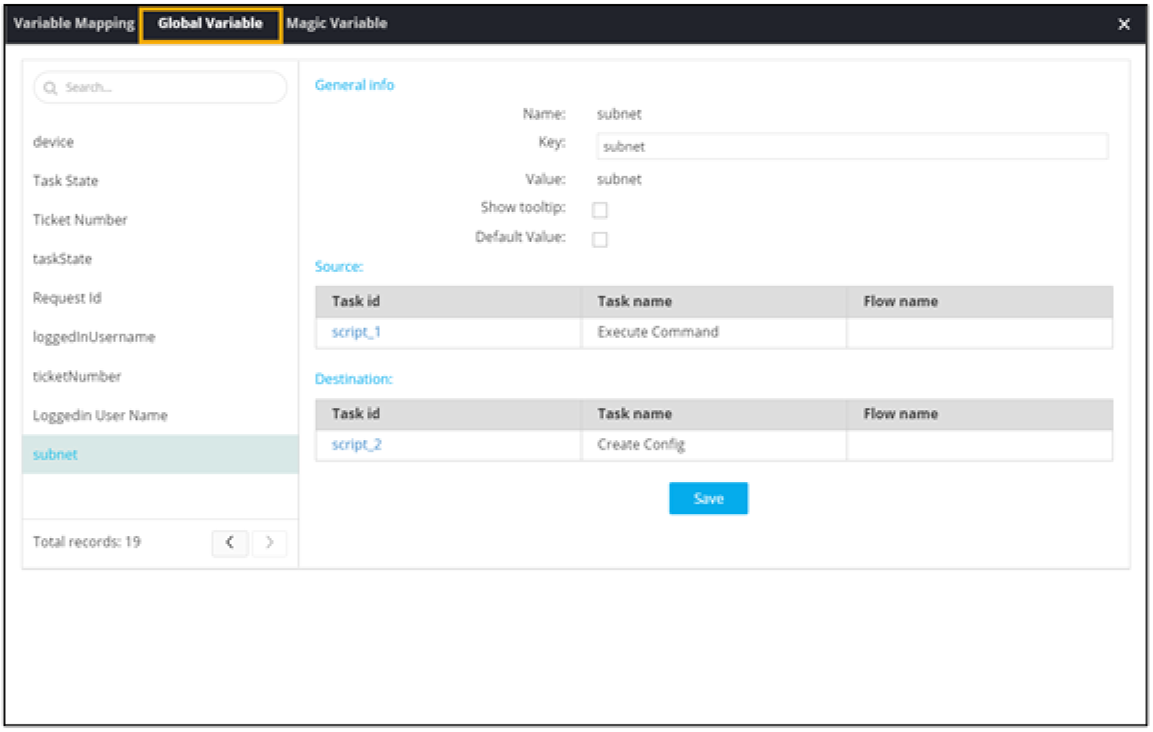
Note: Use keyboard shortcut **Ctrl+g** to view and refer global variables within a Script task.



Global Variable Inventory

The Global variable inventory shows a single view of the source and destination tasks and stages across which a global variable has been referenced within a workflow.

- Provision to view and search the list of global variables declared and referenced across various stages of the workflow.
- Provision to view the source and destination tasks, nested workflow name across which a global variable has been declared and referred.
- Provision to click and update the specific workflow task (via Task id).
- Provision to define one or more global variables as ‘tooltip’ information for a workflow request.



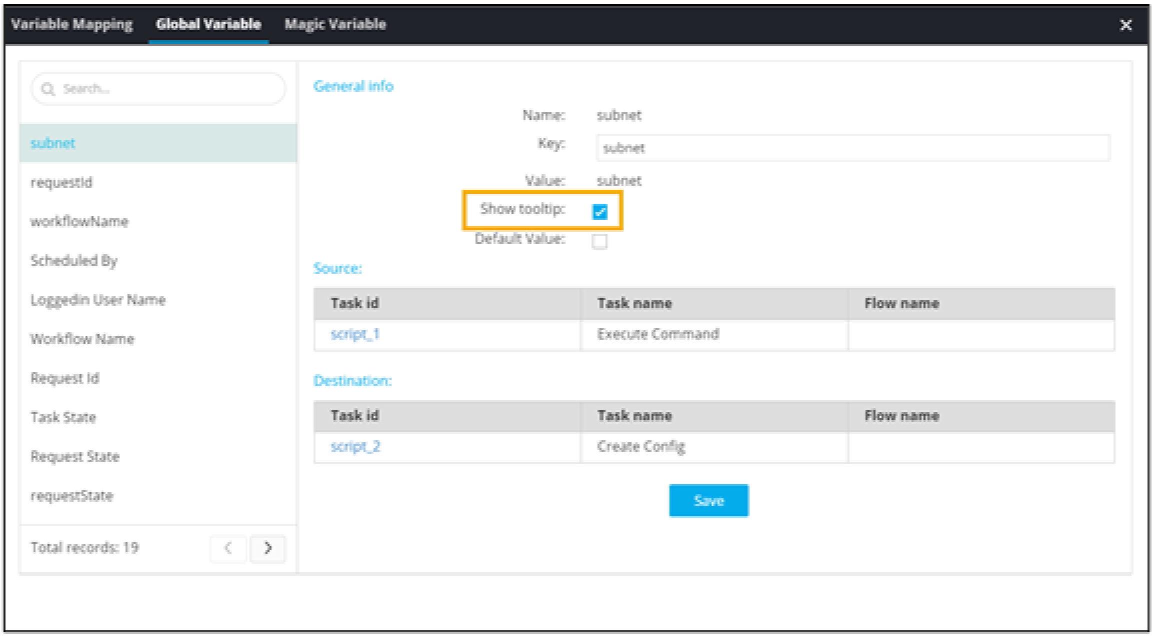
Note: Any updates to the Key field will be reflected globally across the workflow.

Configuring Tooltips

This allows you to define one or more global variables as 'tooltip' information. Once the tooltip is defined, you can right-click a workflow request and view details. Since typically workflows carry a significant amount of data, configuring a tooltip provides the flexibility to select one or more global variable values to be displayed as tooltip information.

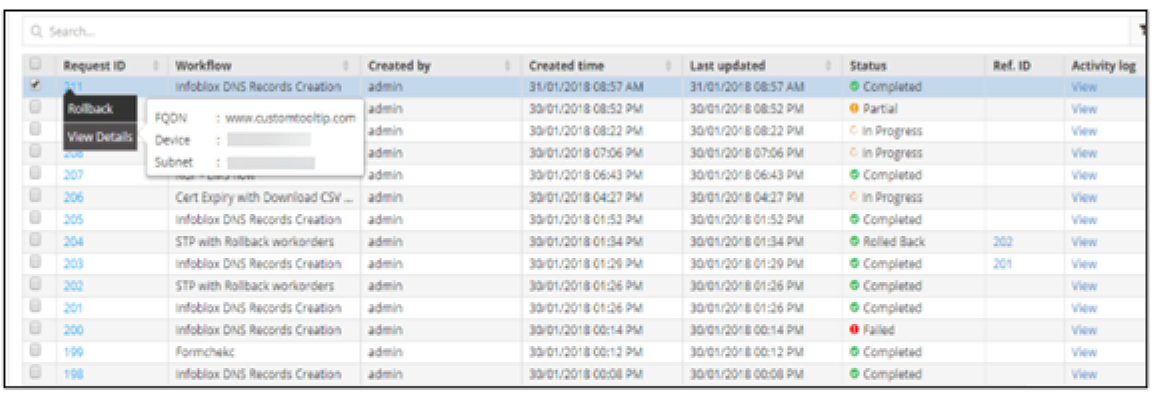
To configure a tooltip:

1. Design a workflow.
2. Drag and drop the required workflow tasks.
3. Assign the required values as global variables.
4. To view all global variables, from the command bar on the top, click **Variable Mapping**.
5. Under **Global Variable**, select the **Show tooltip** checkbox.



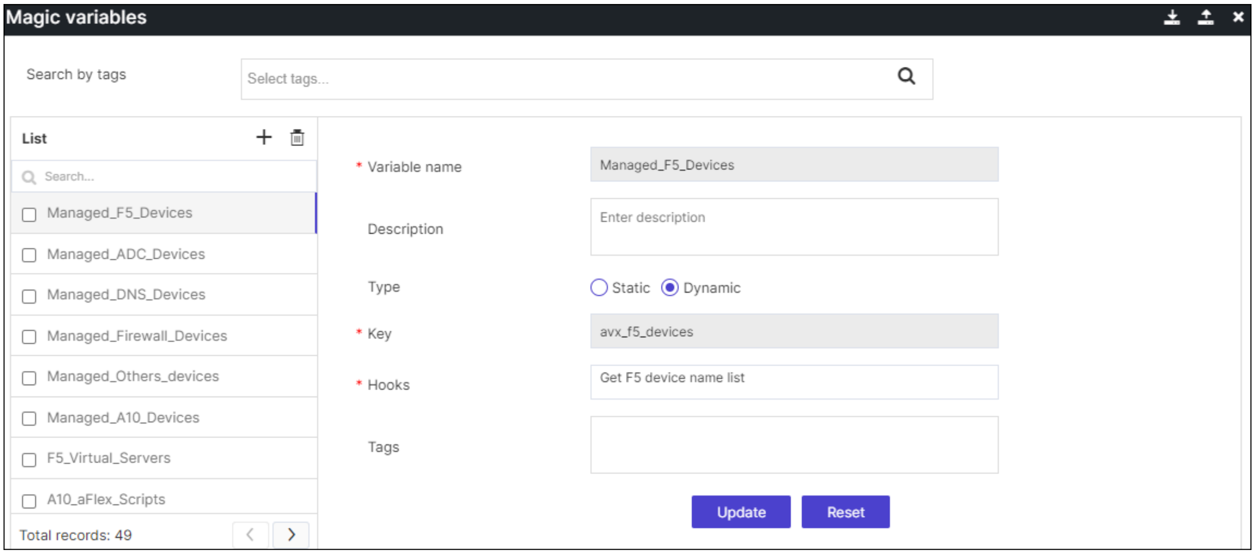
6. Connect the workflow tasks and enable the workflow.
7. Trigger the workflow.

   Workflow is executed successfully.
8. On the Request :: All page, right-click on the Request ID of the workflow.
9. To see the tooltip information, click **View Details**.



Magic Variable

This shows a list of all the Magic variables defined in various tasks across workflows.

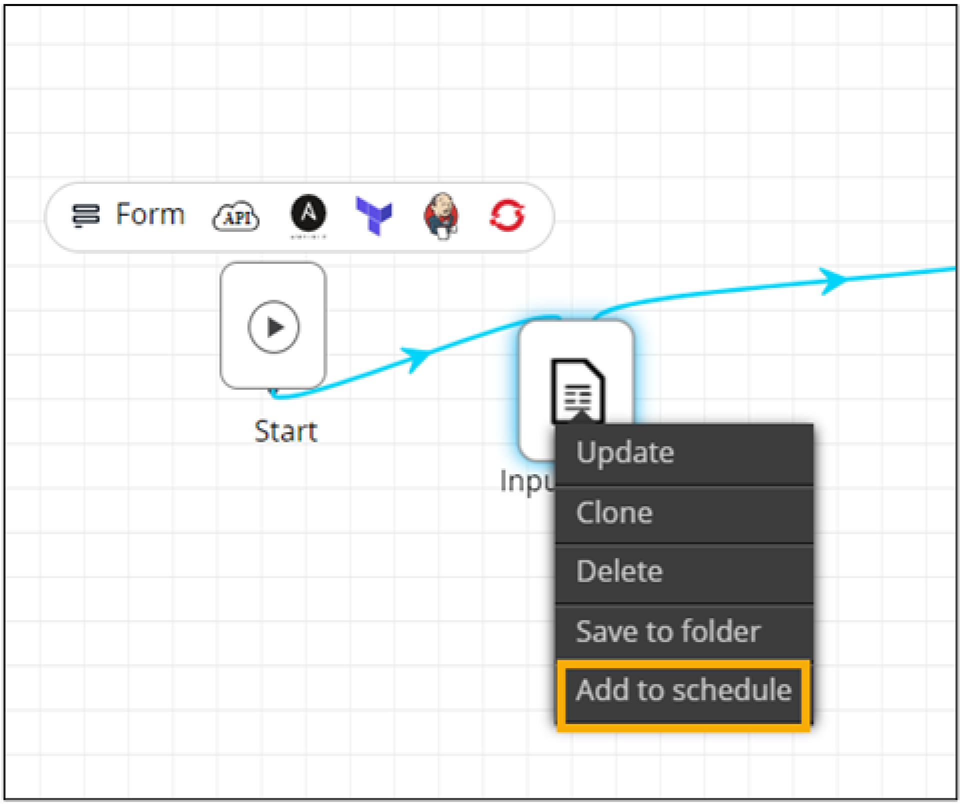


- Accessing Magic Variables
- Defining a Hook and declaring it as a Magic Variable
- Using a Magic Variable within a Workflow

Scheduling Tasks

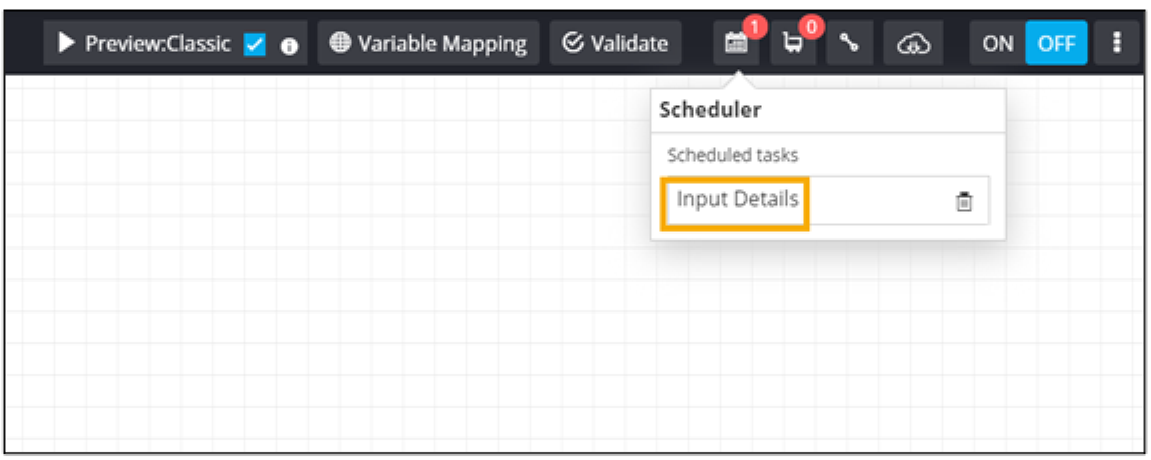
You can schedule workflows and also schedule specific tasks within the workflows that require user inputs.

1. Design a new workflow or open an existing workflow.
2. Right-click the task to be scheduled to view options.
3. Select **Add to schedule**.



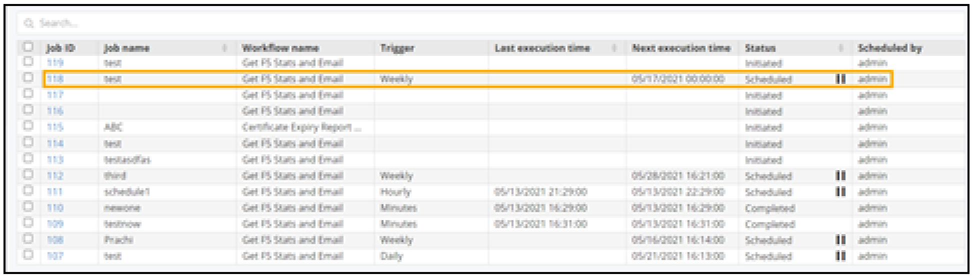
4. Click **Yes** in the **Confirmation** pop-up window.

   The task is added to the **Scheduler**.



5. Connect the workflow tasks and enable the workflow.
6. On the Request :: View/Run page, search for the workflow and schedule the workflow.

The request is added to the **Scheduled Jobs**.



Search...

Job ID	Job name	Workflow name	Trigger	Last execution time	Next execution time	Status	Scheduled by
119	test	Get F5 Stats and Email				Initiated	admin
118	test	Get F5 Stats and Email	Weekly		05/17/2021 00:00:00	Scheduled	admin
117		Get F5 Stats and Email				Initiated	admin
116		Get F5 Stats and Email				Initiated	admin
115	ABC	Certificate Expiry Report ...				Initiated	admin
114	test	Get F5 Stats and Email				Initiated	admin
113	testasdfas	Get F5 Stats and Email				Initiated	admin
112	third	Get F5 Stats and Email	Weekly		05/28/2021 16:21:00	Scheduled	admin
111	schedule1	Get F5 Stats and Email	Hourly	05/13/2021 21:29:00	05/13/2021 22:29:00	Scheduled	admin
110	newone	Get F5 Stats and Email	Minutes	05/13/2021 16:29:00	05/13/2021 16:29:00	Completed	admin
109	testnow	Get F5 Stats and Email	Minutes	05/13/2021 16:31:00	05/13/2021 16:31:00	Completed	admin
108	Prachi	Get F5 Stats and Email	Weekly		05/16/2021 16:14:00	Scheduled	admin
107	test	Get F5 Stats and Email	Daily		05/21/2021 16:13:00	Scheduled	admin

Workflow Cart

Workflow cart allows you to associate or nest one or more flow(s) within a parent workflow

- **Nested flow**: Provision to design new nested workflows or sub workflows
- **Reuse workflow**: Provision to import, reuse one or more workflow(s); and nest them under the parent flow.
- **Rollback**: Provision to design a new Rollback workflow for a parent workflow.

- Nested Workflows

Nested Workflows

“Design once, reuse forever…”

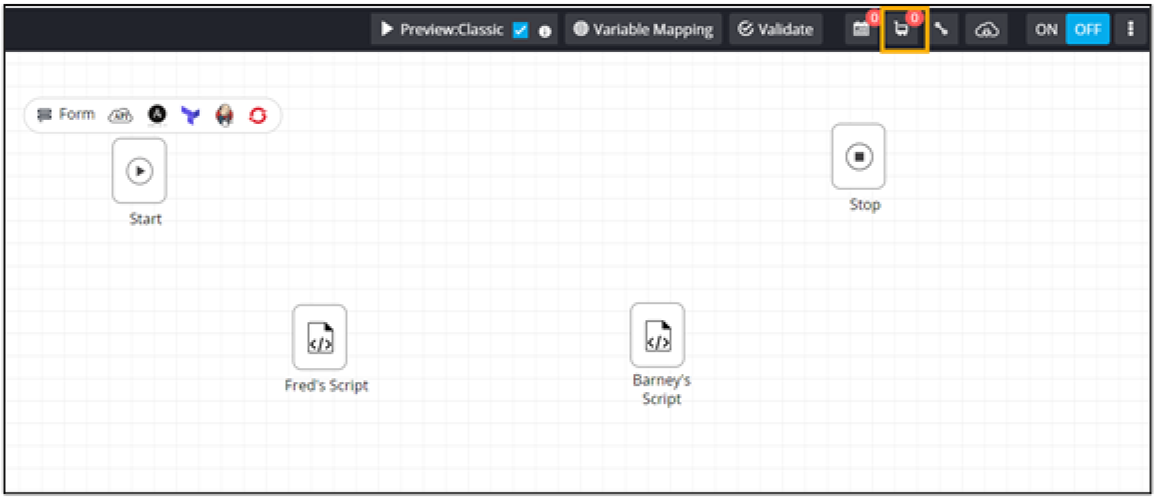
Nested workflows allow you to reuse existing workflows and automate quickly. You can design new workflow(s) or import workflow(s) under a parent workflow. Once nested, the workflows are depicted in green. You can double click on the nested workflow(s) and modify them.

> **Note:** Relevant global variables must be declared and referenced in order for nested workflows to work seamlessly.

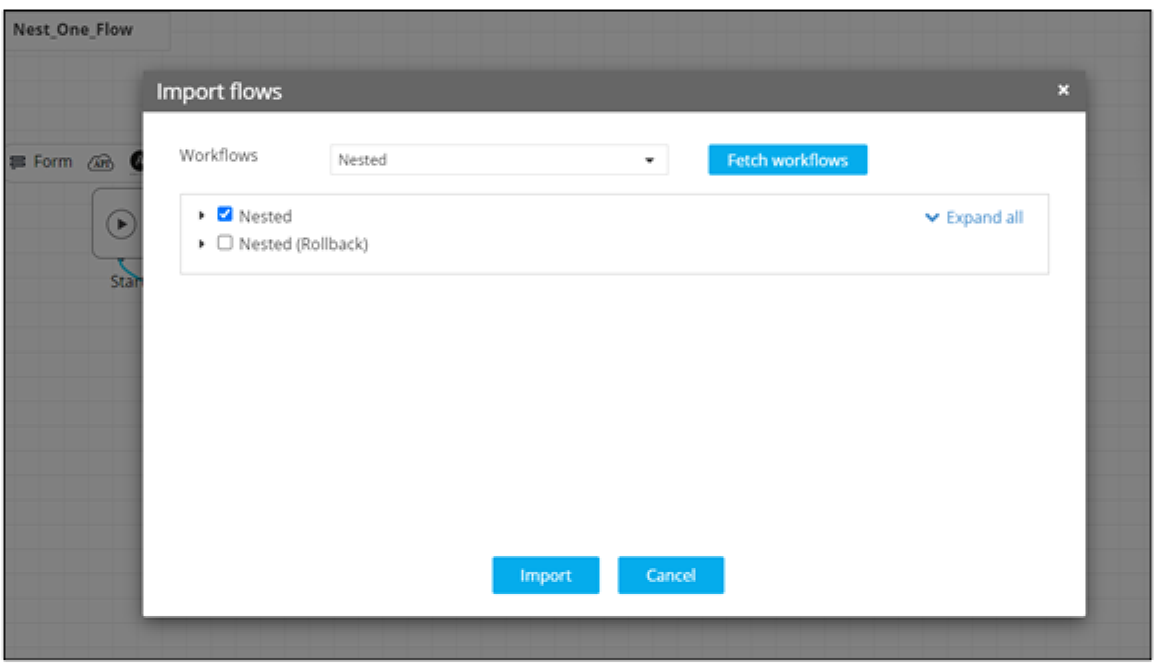
- How to Add/Import a Nested Workflow?

How to Add/Import a Nested Workflow?

1. Design a new workflow.
2. To add or import a nested workflow, click **(Cart)** icon.

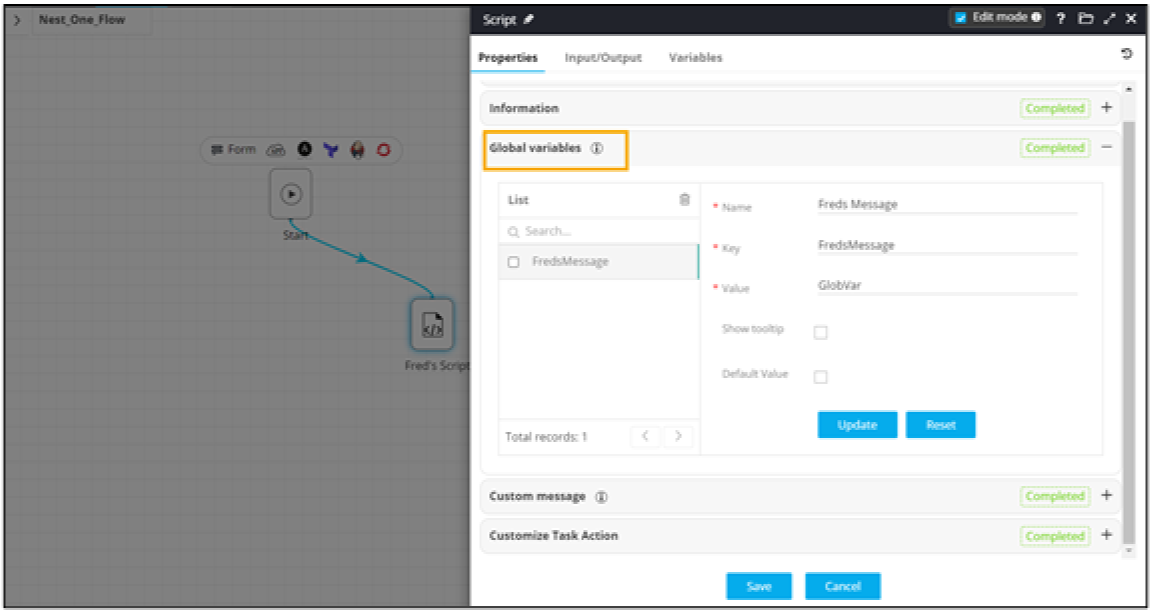


3. To define a new subflow, from the options displayed in the dropdown, click **Add** or to reuse an existing workflow, click **Import**.
4. In the **Import flows** window, select the workflow to be reused or nested.

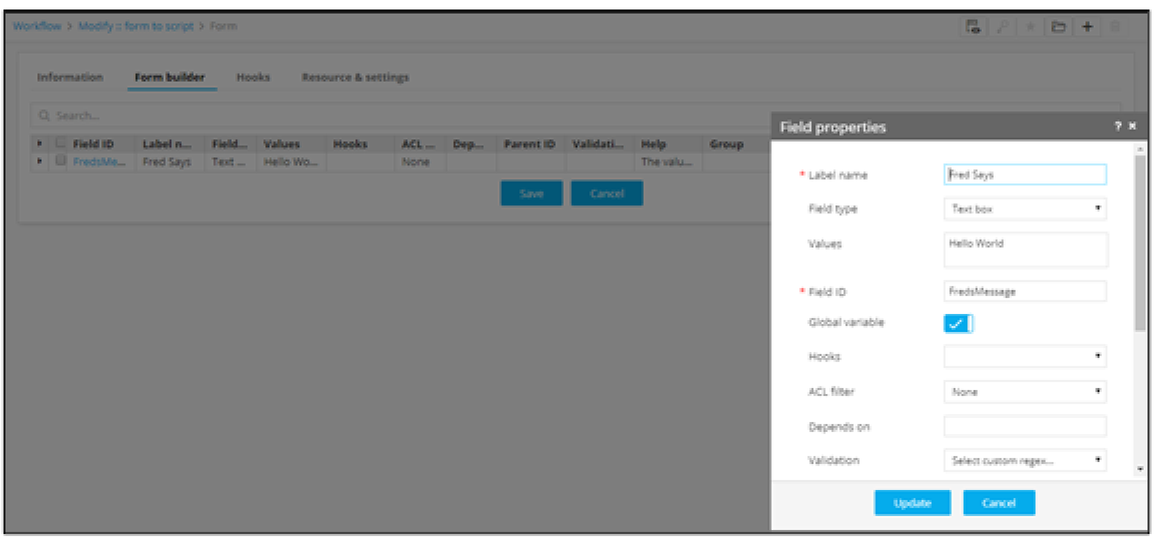


5. Drag and drop the workflow to be nested into the workspace.

6. In the **Script** task window, under **Properties**, in the **Global variables** section, define the global variables in order to pass the values between task(s) in the existing flow and the nested flow.
   For example, The value ‘Fred says: Hello world’ from the script is declared as a global variable and used as an input into the form task of the nested flow.

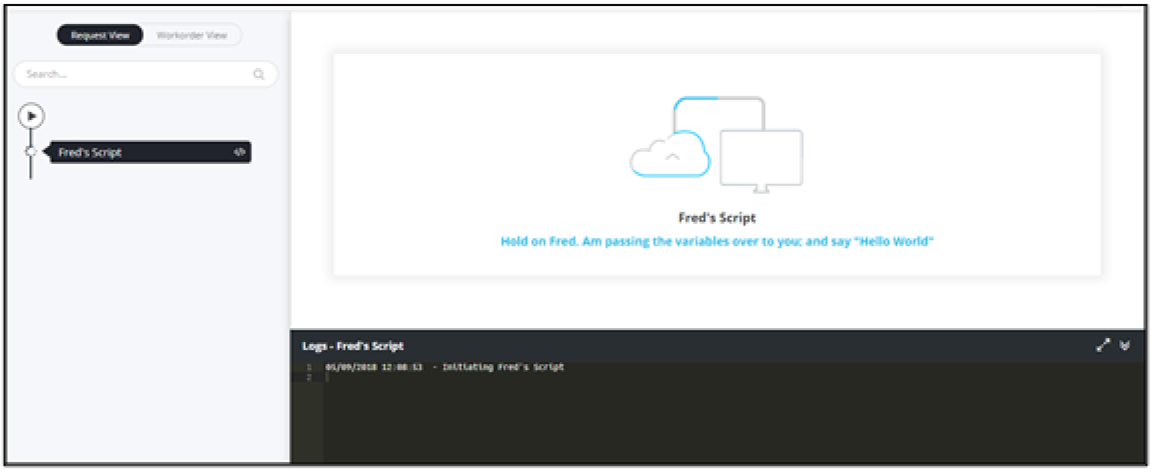


7. Refer the Global Variable key into one or more task(s) of the nested flow.



8. Connect the workflow tasks and enable the workflow.
9. Trigger the workflow.

- Variable values being passed from one task to another task within a nested flow.



- The value being passed into the destination task where it is referred.

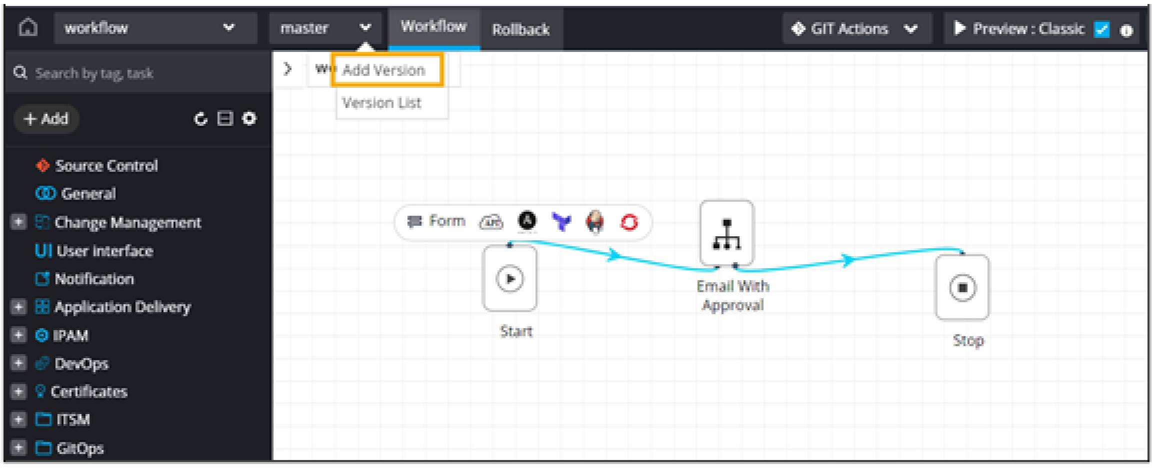
Version Control

You can design and have multiple versions of a workflow and enable one master version.

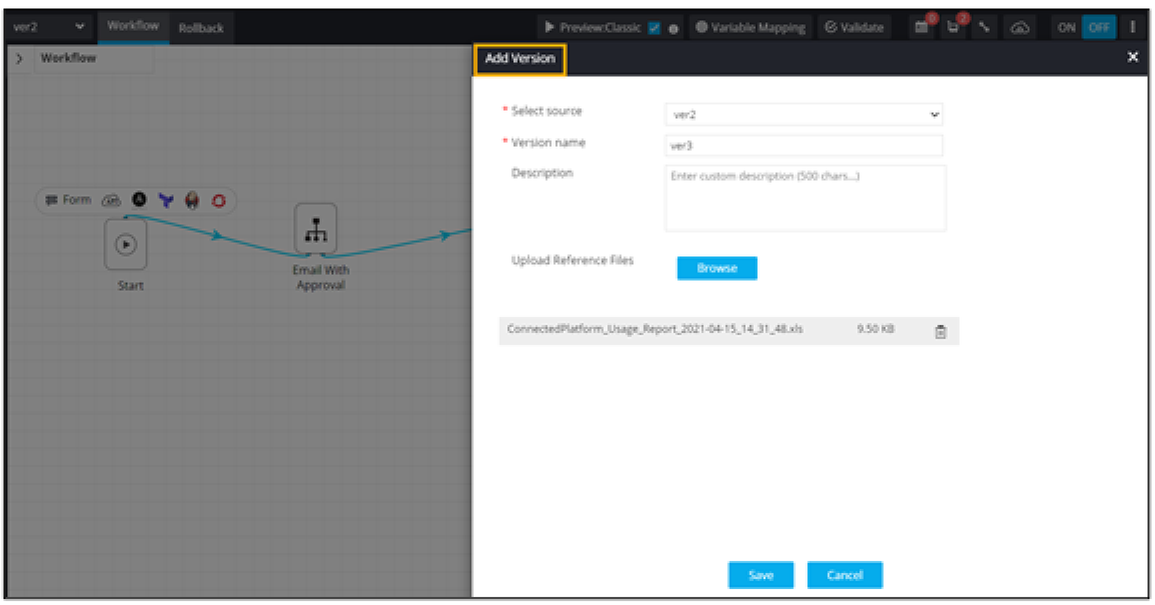
- Provision to version control the workflows.
- Provision to upload and audit requirements/artifacts against each version.
- Version control allows for tracking up to ten versions of the workflow.
- Provision to update, modify version.
- Provision to audit changes made to workflow tasks via ‘Task history’.
- Adding a Version
- Viewing the Workflow Version List

Adding a Version

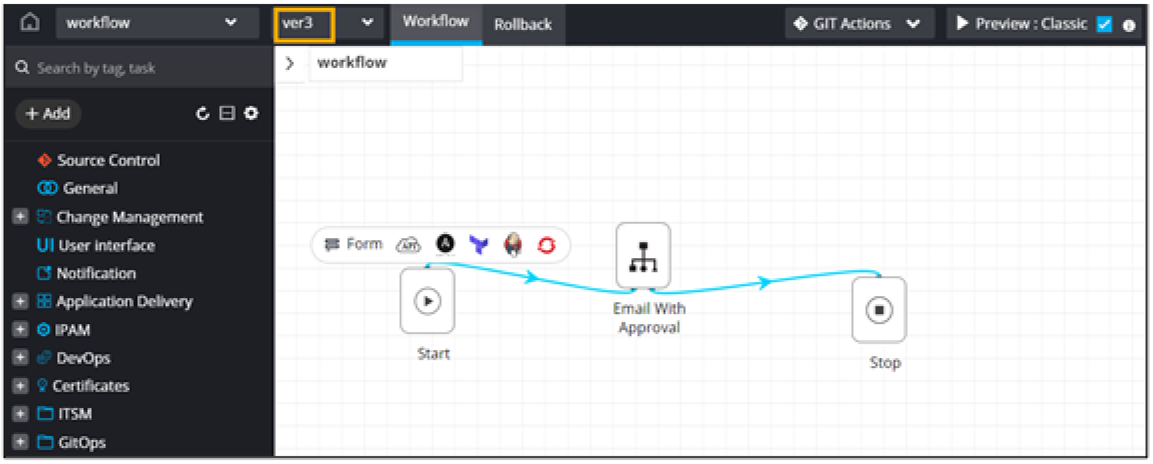
1. Design a workflow.
2. From the master dropdown menu, select **Add Version**.



3. In the **Add Version** window, add the version details and click **Save**.



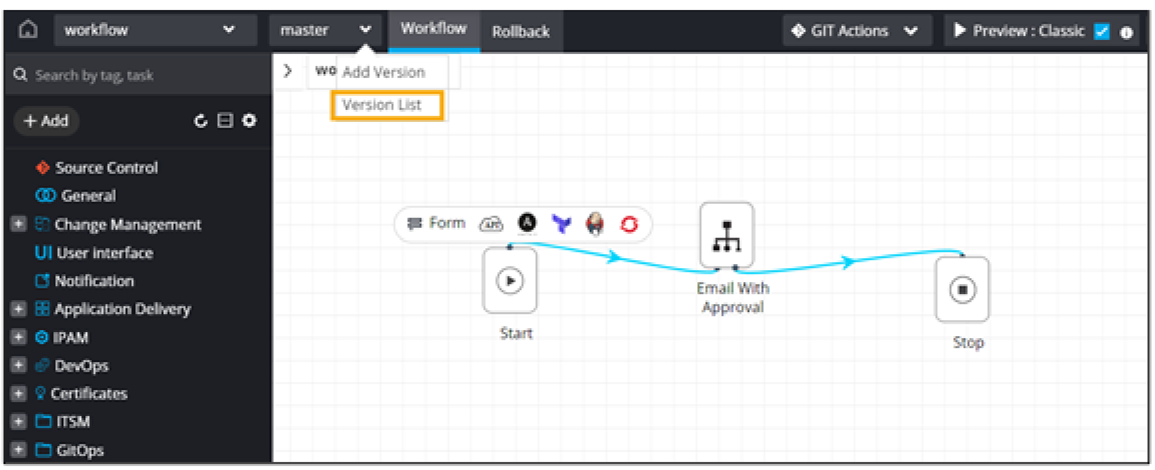
- The newly added workflow version is activated and the workflow version is also displayed on the Workflow page.



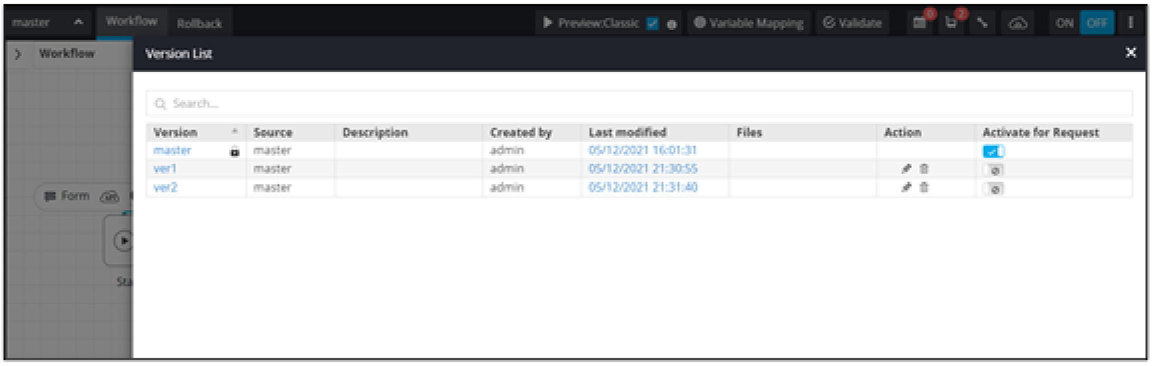
Viewing the Workflow Version List

To check the number of versions created for a workflow:

1. Go to the Workflow page.
2. Open the workflow for which you want to check versions.
3. From the master dropdown menu, select **Version List**.



The **Version List** window displays the list of all versions existing for the workflow.



4. To enable a particular workflow version, turn on the toggle.

> **Note:** Only one version can be enabled at a time.

The enabled version is added to the **Workflow** page.

Workflow Options

This section talks about the different features available in the Workflow Studio.

- Provision to add/remove versions of a workflow
- Provision to customize workflow task connectors
- Provision to modify the workflow alignment
- Provision to view the task history within a workflow
- Provision to validate a workflow
- Provision to switch between edit mode and citizen mode
- Provision to preview a workflow in classic or wizard mode
- Provision to configure access control for user(s).

- Preview
- Workflow Alignment
- History
- Workflow Settings
- Validating a Workflow
- Switching between Edit mode/Citizen mode

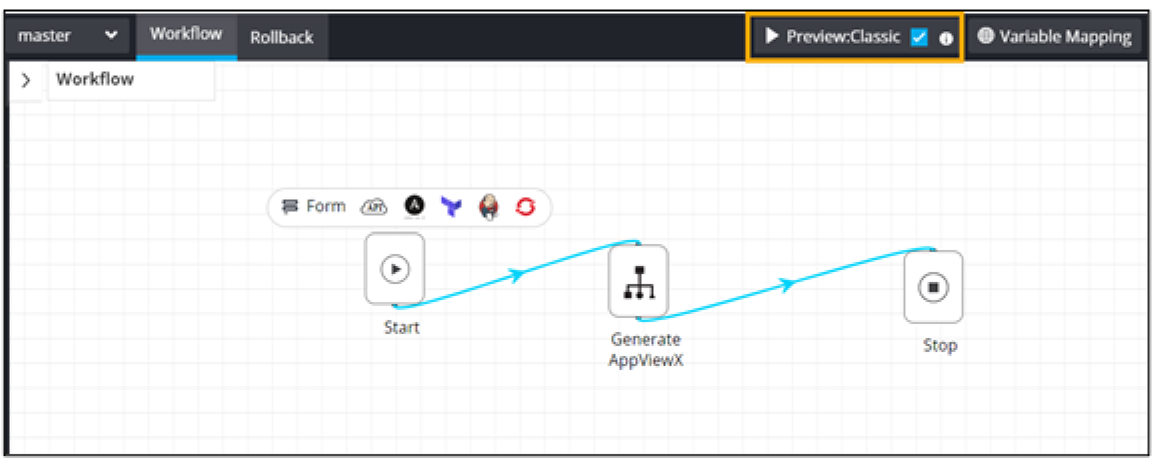
Preview

You can perform a quick preview of the workflow to validate the outcome before triggering the workflow request. Preview action cannot be performed on enabled workflows. Two preview modes are available:

- **Classic**: This is the default Preview option
- **Wizard**: This option can be used when you want to validate individual workflow tasks without enabling the workflow. A pseudo form is generated for stage-wise execution of tasks without triggering the workflow. A pseudo form is only supported for the following tasks:
  - Script
  - REST
  - REST (I)
  - Schedule
  - Email
  - Slack.

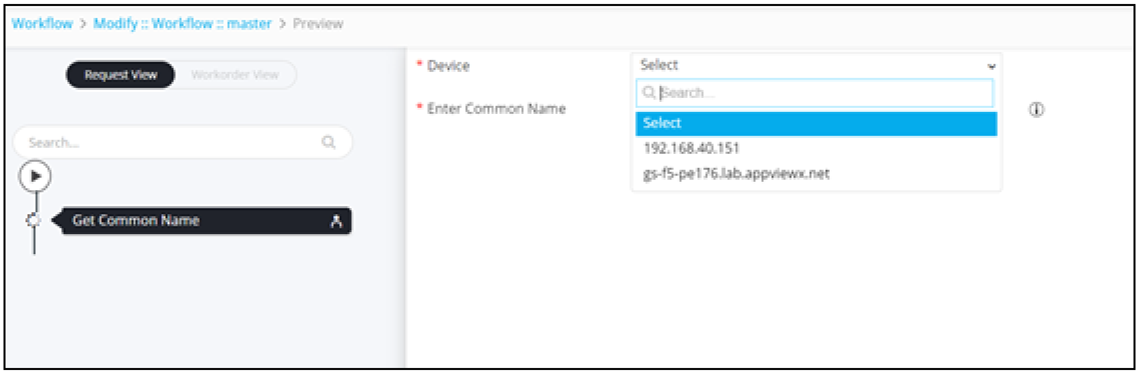
To preview a workflow in Classic mode:

1. Go to the Workflow page.
2. Open a workflow.
3. From the command bar on the top,, select the **Preview:Classic** checkbox.

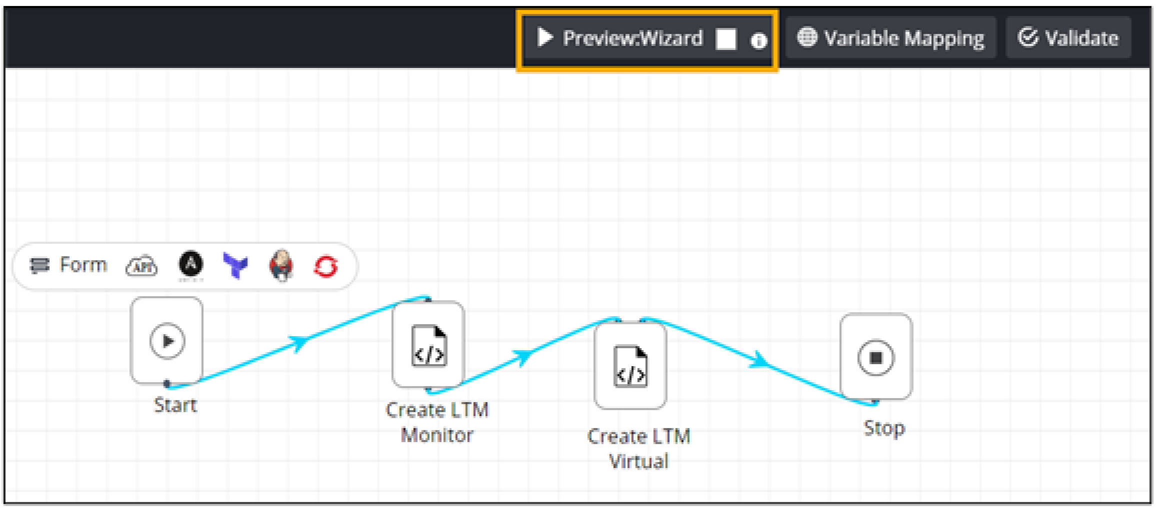


4. Click **Preview**.

   Workflow preview is displayed.

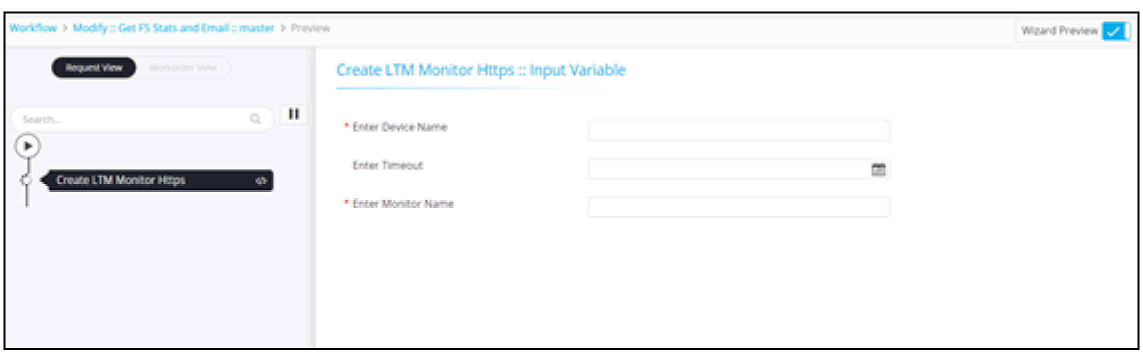


5. To perform Preview in wizard mode, clear the **Preview:Wizard** checkbox.

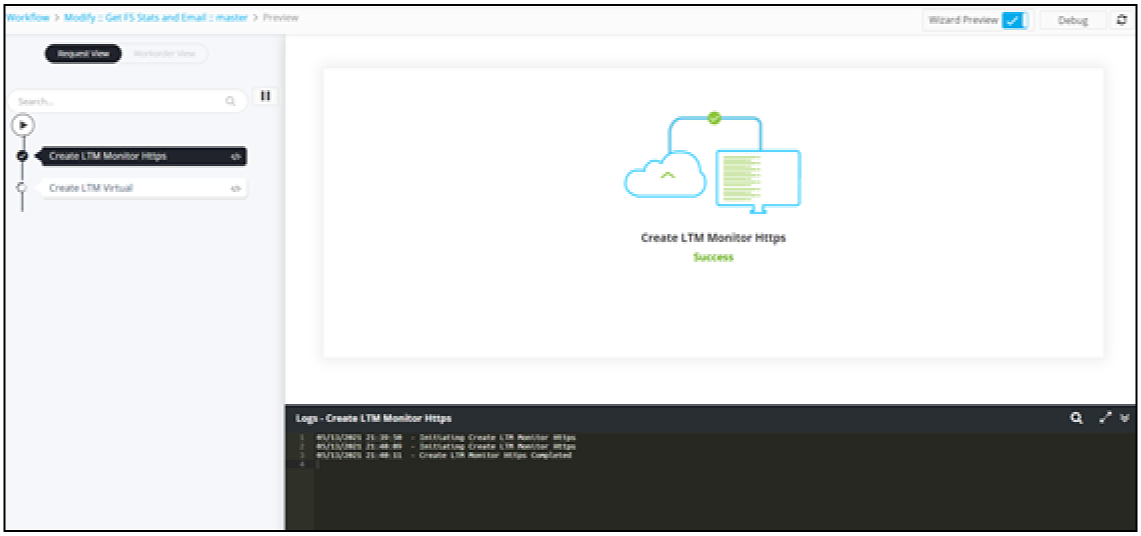


6. Click **Preview**.

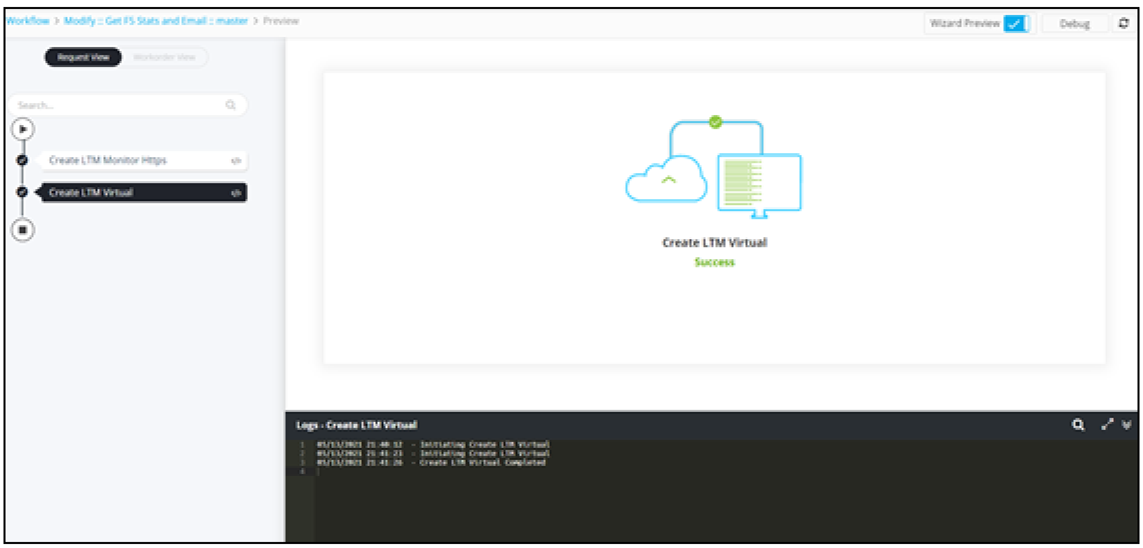
   A pseudo form is generated with the input variables in the Script task.



7. Enter the field information to execute the task.



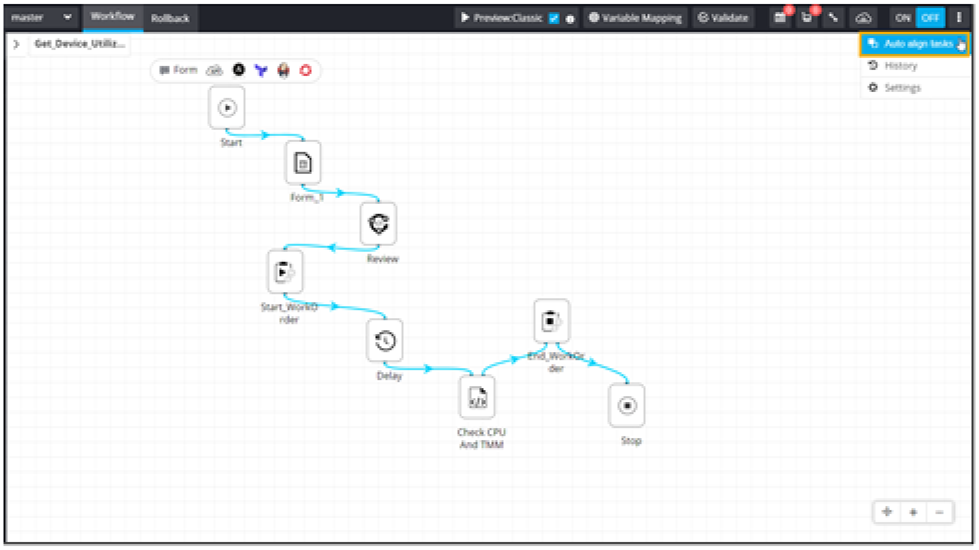
The second task executes automatically once the first task is successful.



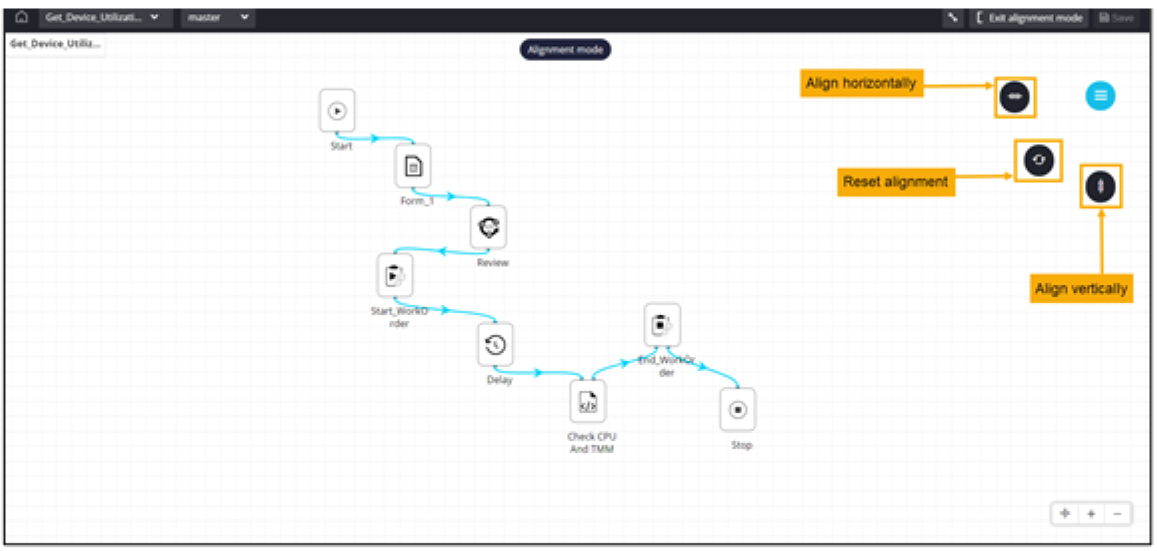
Workflow Alignment

You can align a workflow either vertically or horizontally.

1. Go to the Workflow page.
2. Open a workflow.
3. To see alignment options, from the right upper corner of the screen, click .
4. From the options displayed, select **Auto align tasks**.



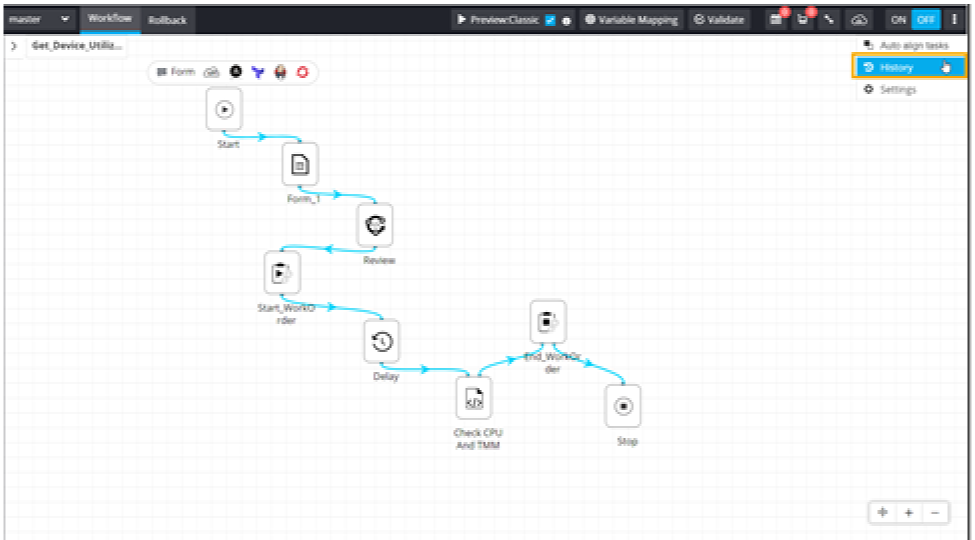
5. Click **Yes** in the **Confirmation** pop-up window.
6. Select the alignment for the workflow from the options available.



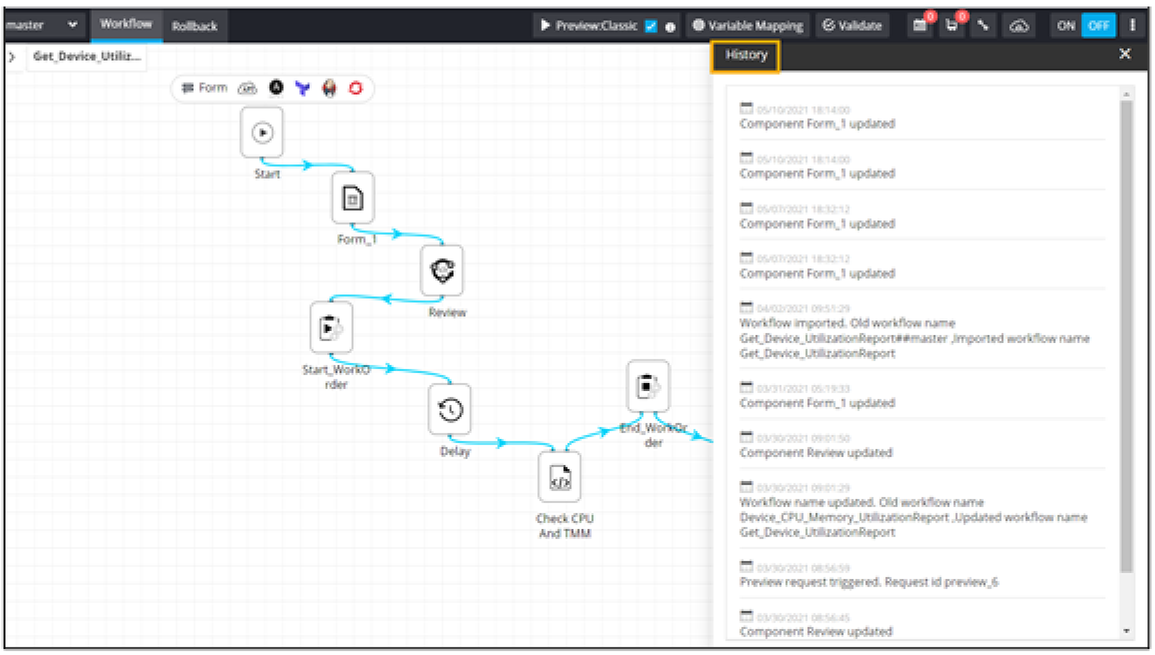
History

You can track the complete audit of changes made against a workflow.

1. Go to the Workflow page.
2. Open a workflow.
3. From the top right corner of the screen, click .
4. From the options displayed, select **History**.



The **History** window opens displaying the complete history of actions for this workflow.



Workflow Settings

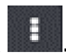
On creating a new workflow, the workflow settings option allows users to perform the following actions:

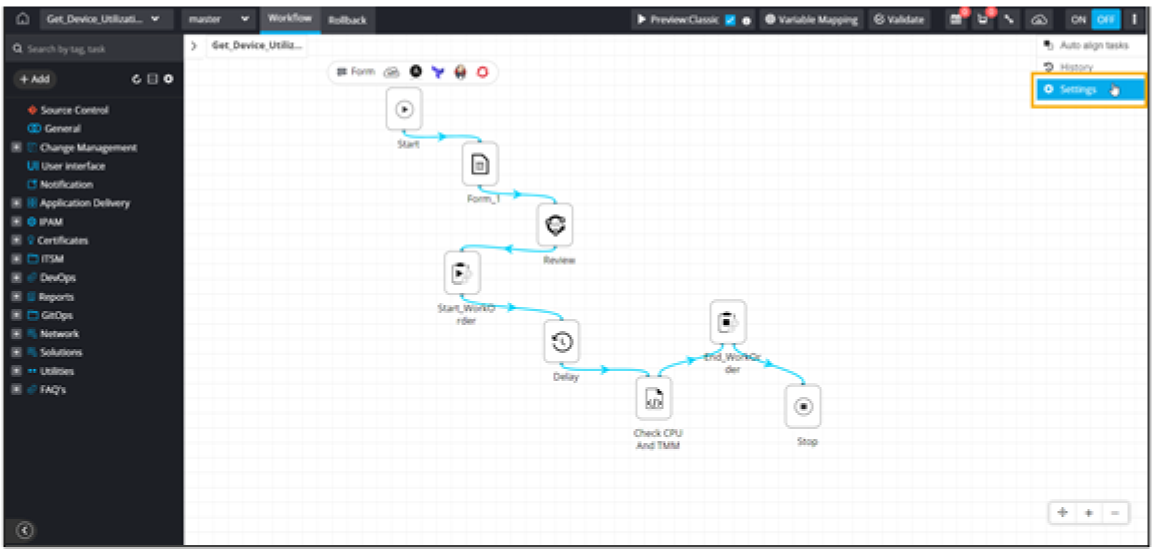
Provision to generate a unique workflow request ID for every workflow triggered.

- Role based access control (RBAC) to assign workflows.
- Role based access control (RBAC) to assign workflow requests.

- Provision to exclude specific workflow task(s) for calculating Request status.
- Provides an option to opt-out specific task(s) from being considered for calculating the overall workflow request status.

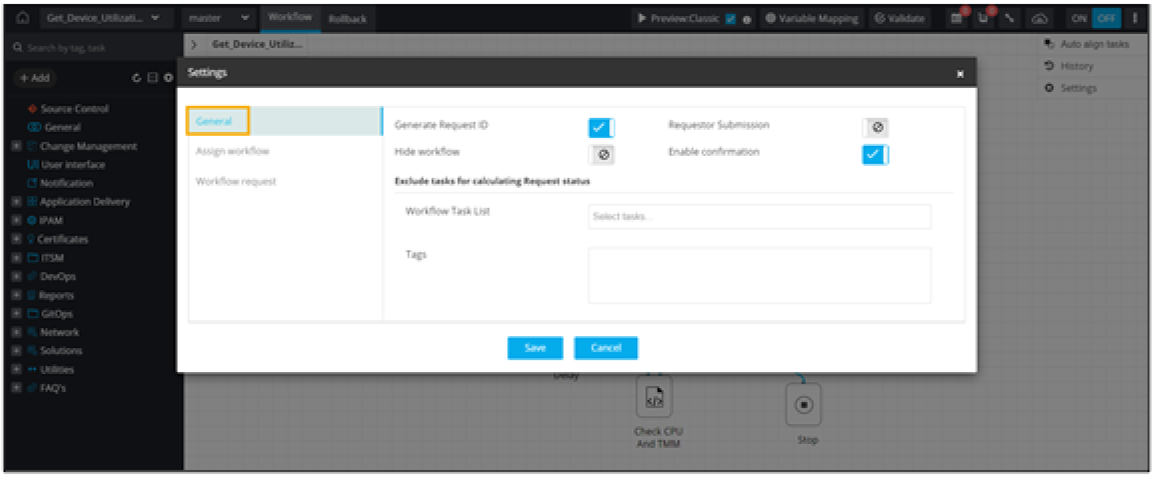
To access workflow settings:

1. Go to the Workflow page.
2. Open a workflow.
3. From the top right corner of the screen, click .
4. From the options displayed, select **Settings**.



- General
- Assign Workflow
- Workflow Request

General

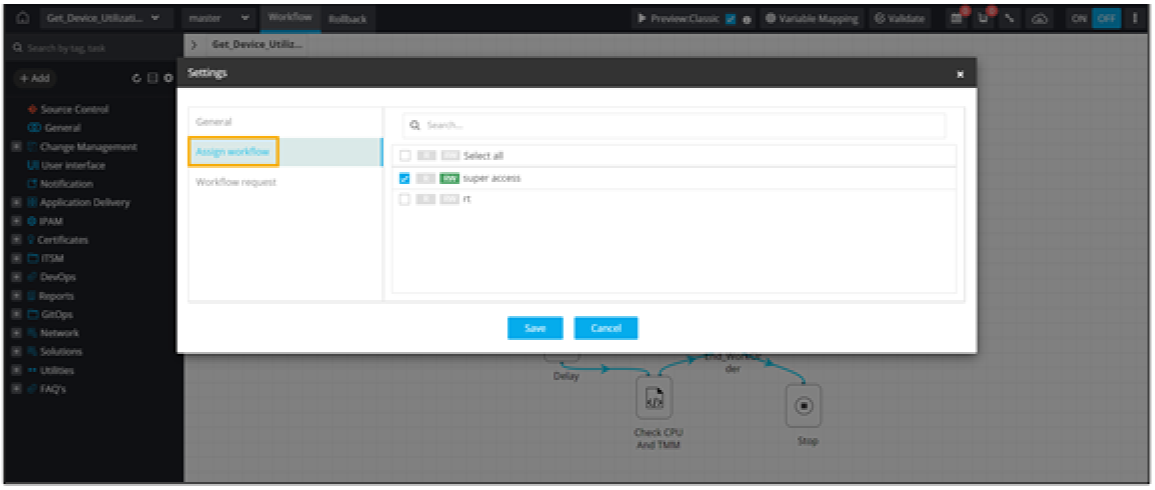


Under Workflow Settings, the following options are available under the General section:

- **Generate Request ID**: Allows you to choose generating or not generating a request ID for a workflow
- **Requestor Submission**: Allows you to restrict/allow actions/approvals for the user who requested the task
- **Hide workflow**: Allows you to keep the workflow hidden or visible
- **Enable confirmation**: Allows you to enable or disable the Confirmation popup window
- **Exclude tasks for calculating Request status**: Allows you to select workflow tasks that will be excluded when calculating request status.

Assign Workflow

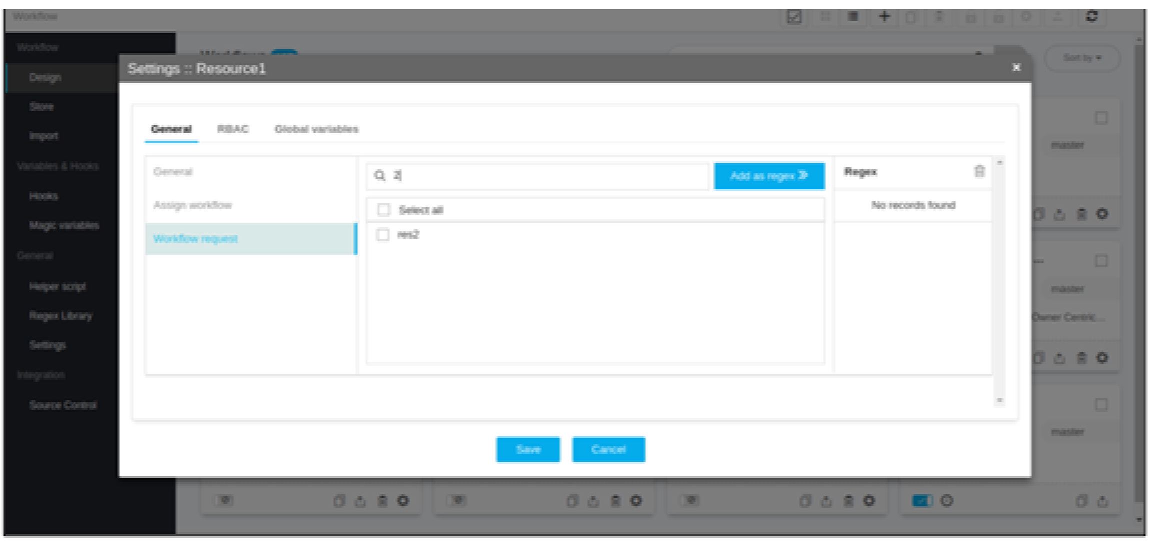
Under Workflow Settings, you can set up RBAC for workflows and assign workflows to specific users or roles.



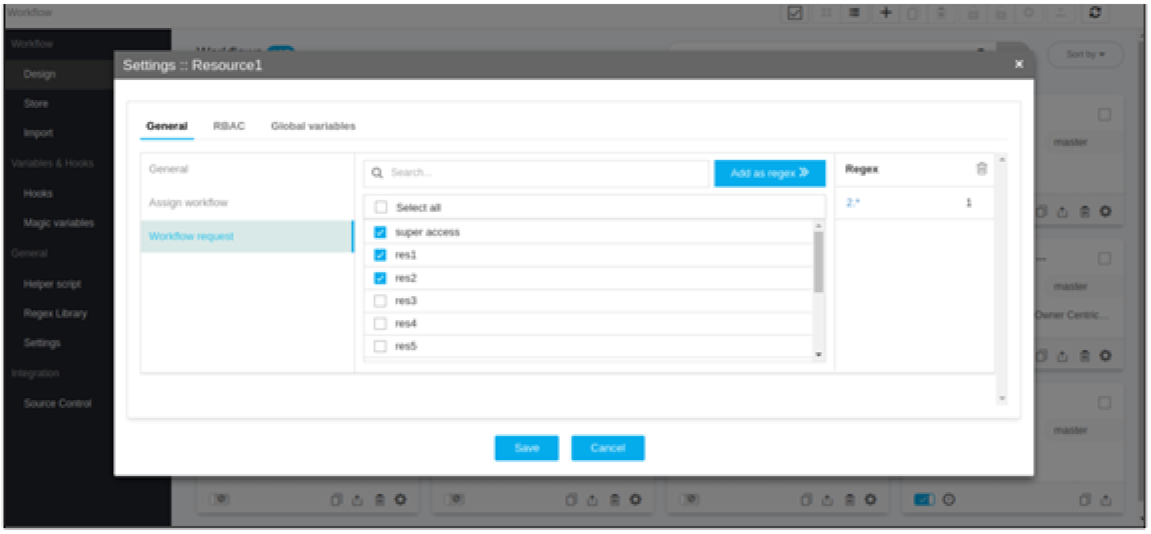
Workflow Request

Under Workflow Settings, you can set up RBAC for workflow requests.

- Provision to search for resources by entering values in the search field.



- Provision to Add a Regex and select the list of available resources using a regex pattern. For example, regex names can be used to select the list of all available resources.



- When trying to add the regex "*", a validation error will be triggered, resulting in the disabling of the "Add as regex" option.
- Provision to delete the regex.

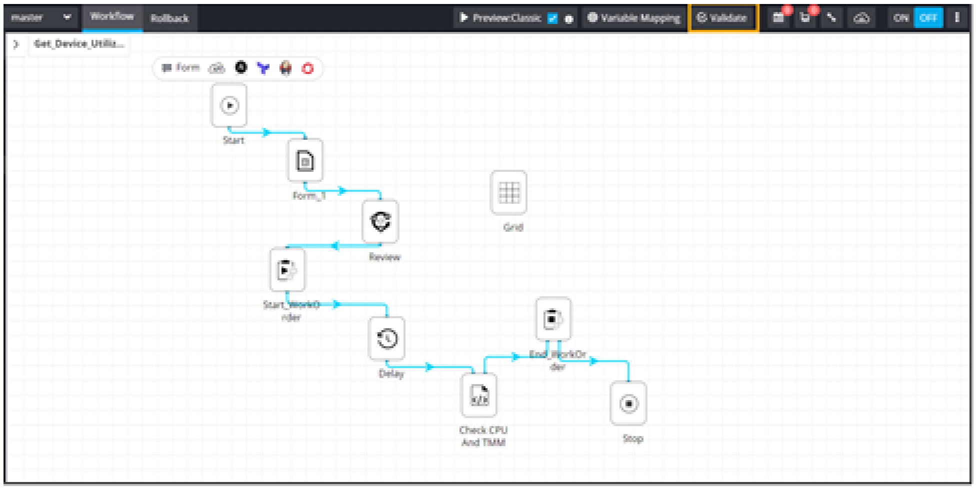
Validating a Workflow

Once a workflow is designed, users can validate the workflow and its elements. Basic workflow validation includes:

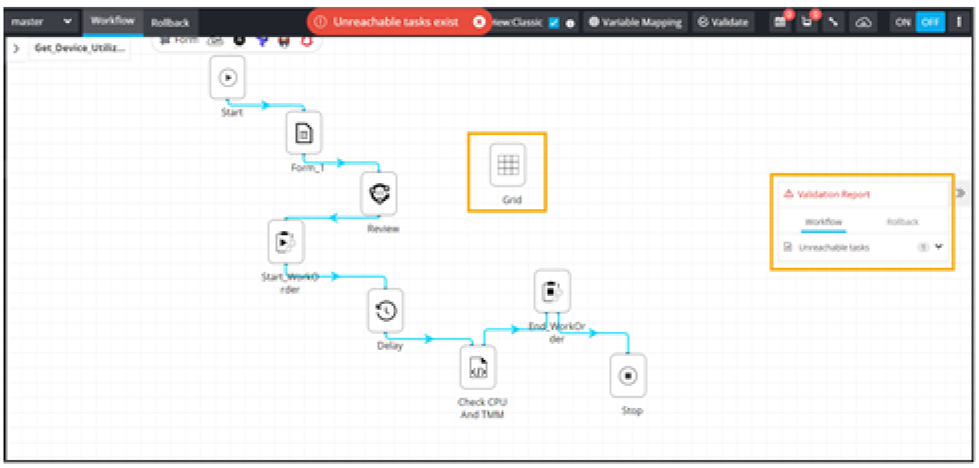
- Tasks with no relevant connectors associated.
- Missing tasks which are dependent on other workflow tasks.
- Invalid links or connections.

To validate a workflow:

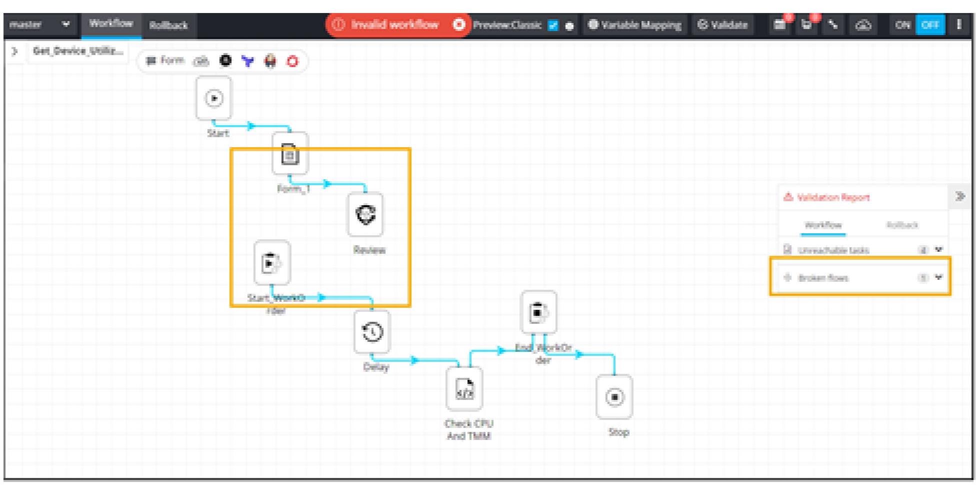
1. Design a workflow.
2. In the design space, from the command bar on the top, click **Validate**.



• Validation Report shows unreachable tasks.



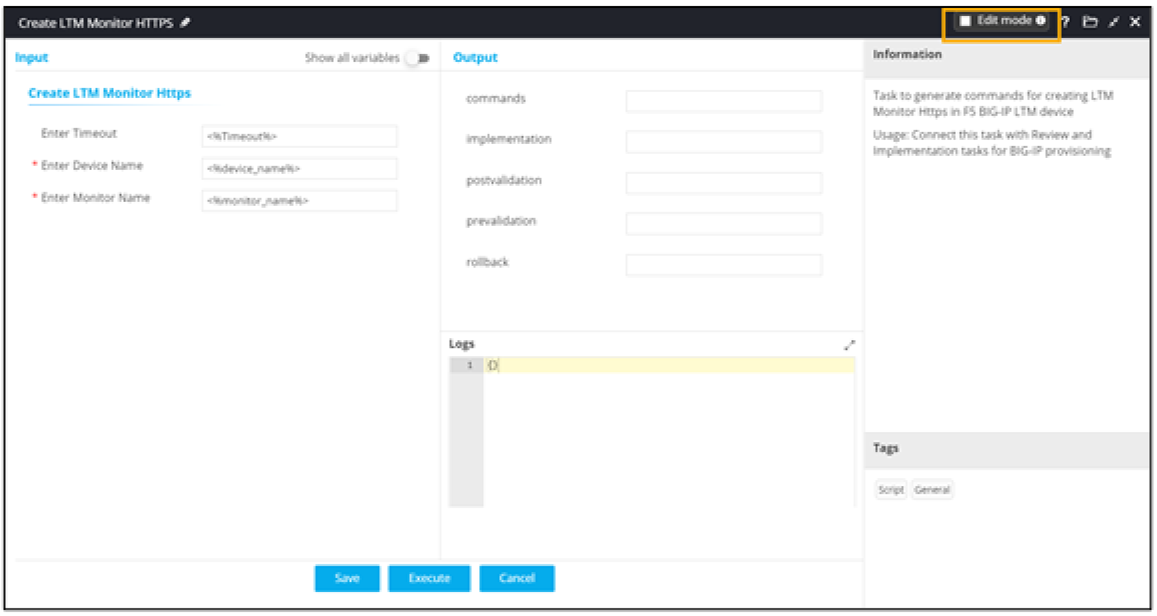
• Validation report shows broken flows.



Switching between Edit mode/Citizen mode

Users can switch between Edit mode and Citizen mode when creating/updating tasks. All tasks are displayed in Edit mode by default.

1. Design a workflow with workflow tasks.
2. To switch to citizen mode, in the task window, deselect the **Edit mode** checkbox.
   In the Citizen mode of the task window, variables referred in the script are listed under **Input**, while the values declared as Global Variables are listed under **Output**.

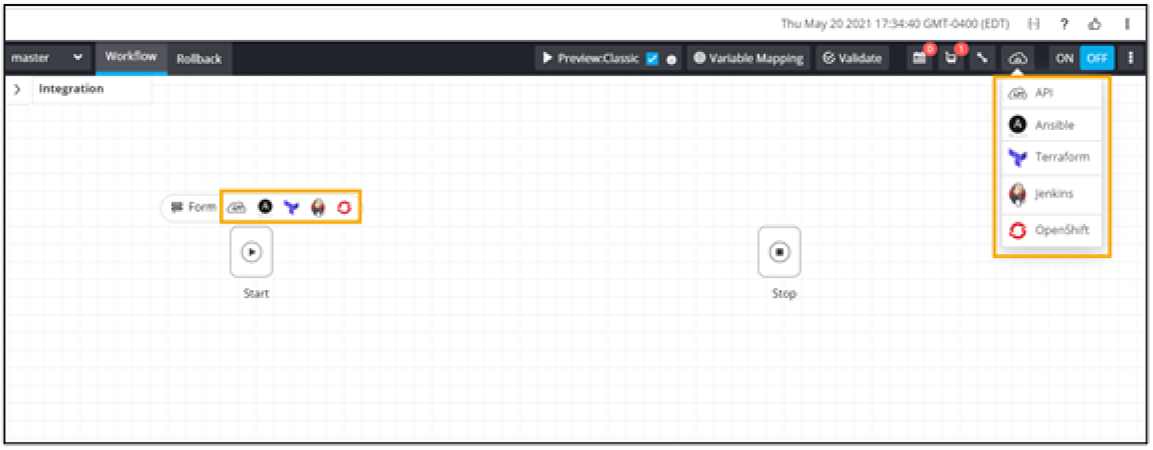


3. To switch back to Edit mode, select the **Edit mode** checkbox.

Northbound Integrations

Organizations have multi-vendor tools for various network needs. AppViewX allows you to integrate with automation tools such as Ansible, Jenkins, Terraform, and Openshift in order to automate and orchestrate the network.

- Provision for native Northbound integration with multiple platforms
- Provision to generate an API and trigger a workflow from an external source Python code, Javascript, Web browsers, and Postman.



- Postman - Visual Workflow Northbound Integration
- Jenkins - Visual Workflow Northbound Integration
- Terraform - Visual Workflow Northbound Integration
- Ansible - Visual Workflow Northbound Integration
- Openshift PaaS Orchestration

Postman - Visual Workflow Northbound Integration

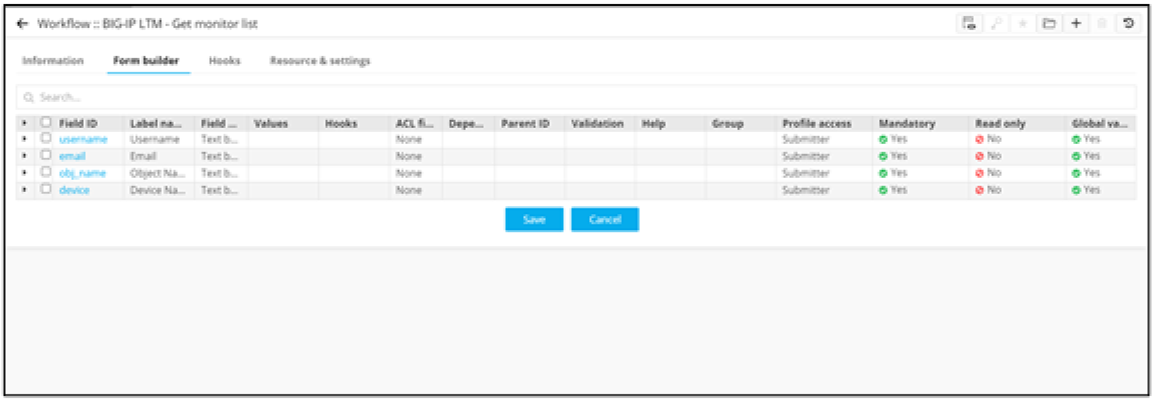
This integration enables dynamic API generation from a third-party payload generator such as Postman, which can be used to design custom automation workflows.

- Dynamic API Generation

Dynamic API Generation

You can design and enable a workflow and dynamically generate custom API to ease the automation and orchestration process. This allows a simple API payload builder/generator to aid you in integrating with external tools and systems.

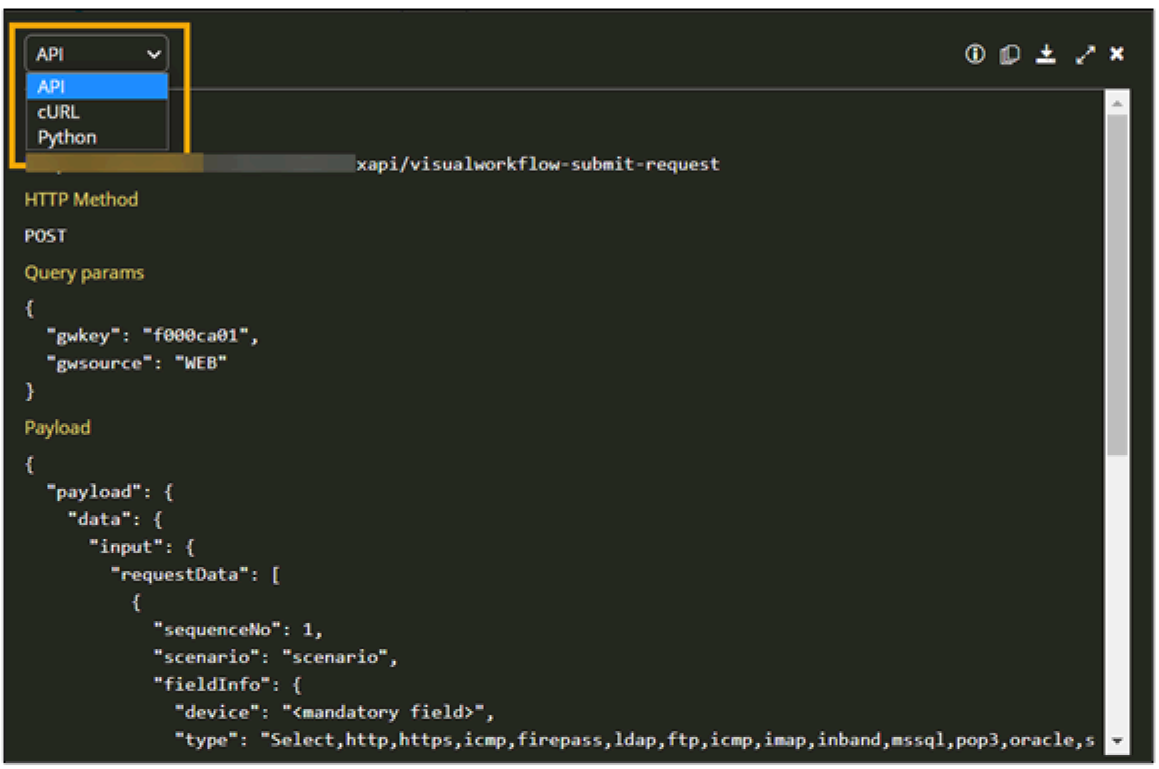
1. Design a new workflow.
2. From the **User Interface** section, drag and drop a **Form** task.
3. Define the required form fields.



4. From the **General** section, drag and drop a **Script** task.
5. From the **User Interface** section, drag and drop an **Email** task.
6. To generate an API for this workflow, click icon above the **Start** task.

   The URL, HTTP method, Query params, Payload, and the Header for the API are displayed.

> **Note:** The API can also be generated as cURL or Python.

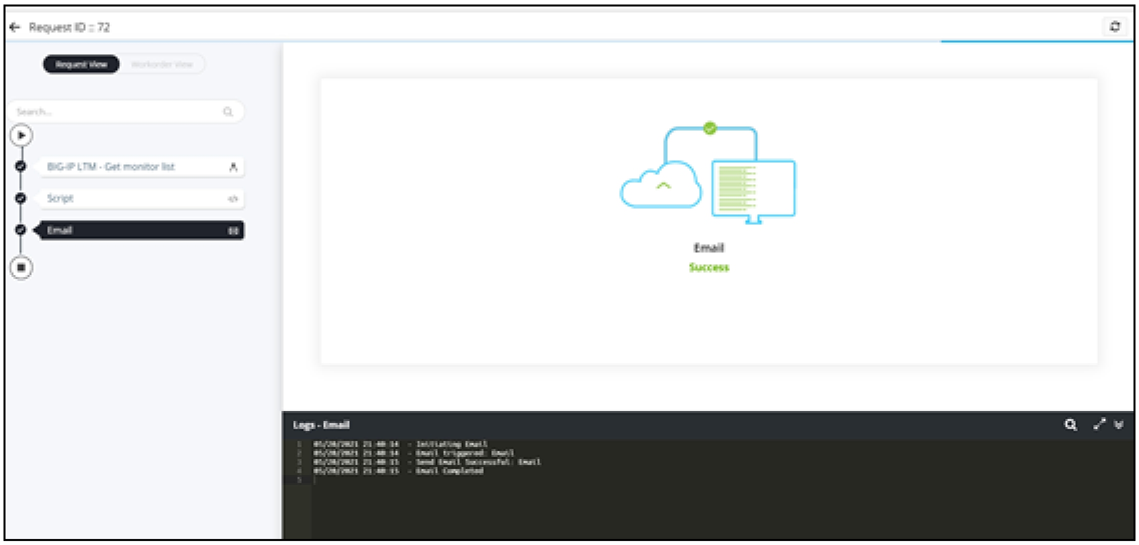


7. To download this API, form the top right corner of the API window, click **(Download)** icon.
8. Connect and enable the workflow.

9. To trigger the workflow from Postman, copy the URL and payload from the Workflow Studio.
10. On the Postman Utility tool, under **Body**, paste the URL and Payload details.
11. Under **Headers**, update the username and password details.
12. To trigger the workflow from Postman, click **Send**.

    Response with a Request ID is generated.
13. Navigate to the **Request :: All** page and click on the **Request ID** generated in Postman.

    Workflow is executed successfully.



Jenkins - Visual Workflow Northbound Integration

This integration enables rapid network service orchestration of CI/CD Pipelines, using a Jenkins plugin, and AppViewX's Visual workflow in order to automate and orchestrate application and network security services. This section will guide you to,

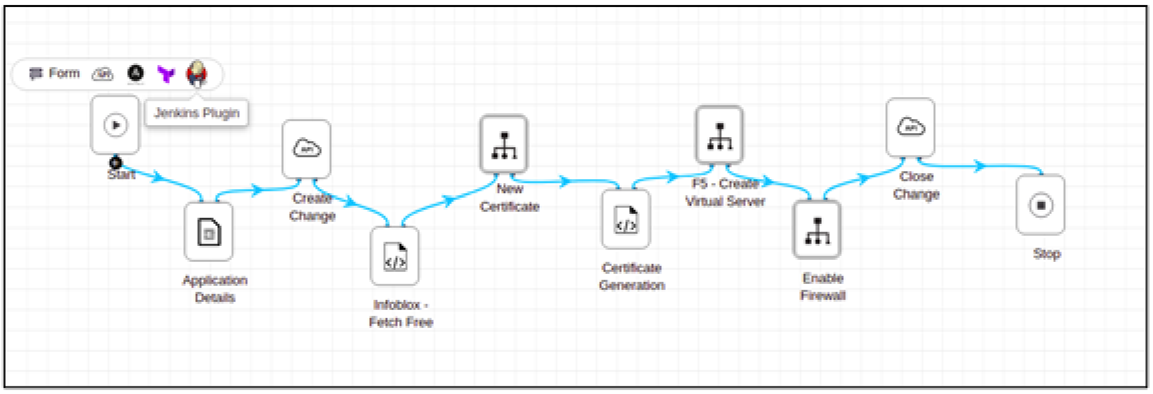
- Download a Jenkins plugin from Visual workflow
- Install the Jenkins plugin and set it up
- Execute Jenkins job to trigger workflows
- Create Pipelines
- Build the Application Service Framework via Jenkins.

- Prerequisites
- Global Configuration
- Creating a Freestyle Job

- Creating a Pipeline Job
- Jenkins pipeline for Application provisioning

Prerequisites

Download the AppViewX plugin for Jenkins, design a sample workflow, and click on the Jenkins logo (AppViewX_Jenkins_Plugin.zip):



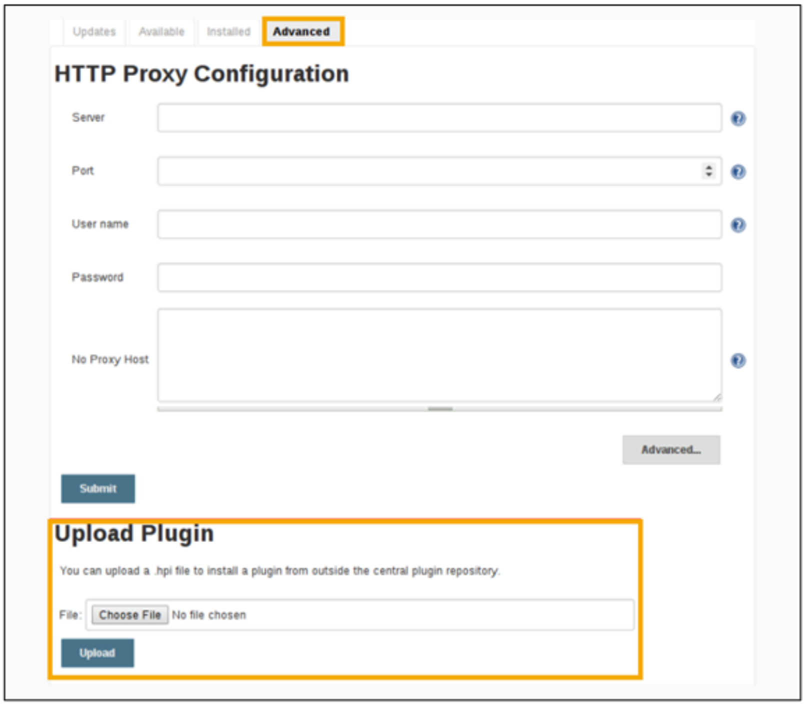
Note: Files present within the downloaded zip file are:

- AppViewX.hpi (the plugin to be installed in the Jenkins)
- Jenkins and AppViewX Integration Guide <.html>
- Jenkins and AppViewX Integration Guide <.pdf>.

To install the AppViewX plugin in Jenkins:

1. Go to the **Manage Jenkins** > **Manage Plugins** page in the web UI (available to administrators of a Jenkins environment).
2. Click on the **Advanced** tab.
3. Under the Upload Plugin section, click the <.hpi> file.

4. **Upload** the plugin file.



Global Configuration

In Jenkins, credentials can be added in global configuration settings. This information can be used in the job configuration page. Click on **Check connection** to validate connectivity of hostname, port, username, and password.

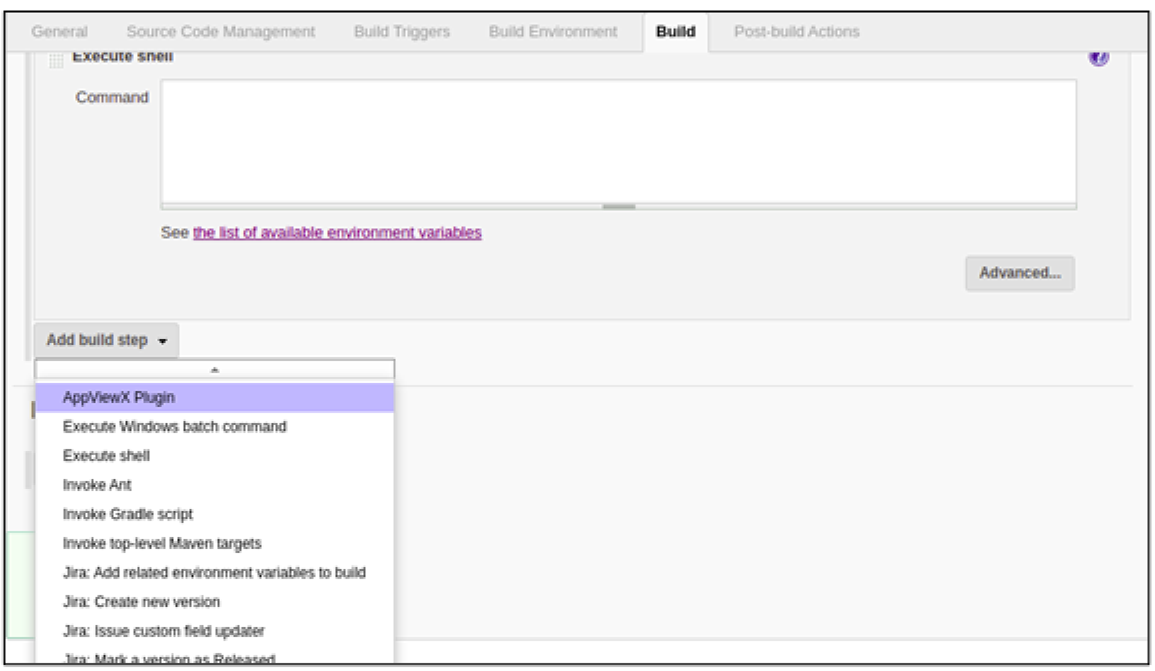
Creating a Freestyle Job

The freestyle build job is a highly flexible and easy-to-use option. It is a repeatable build job which contains steps and post-build actions.

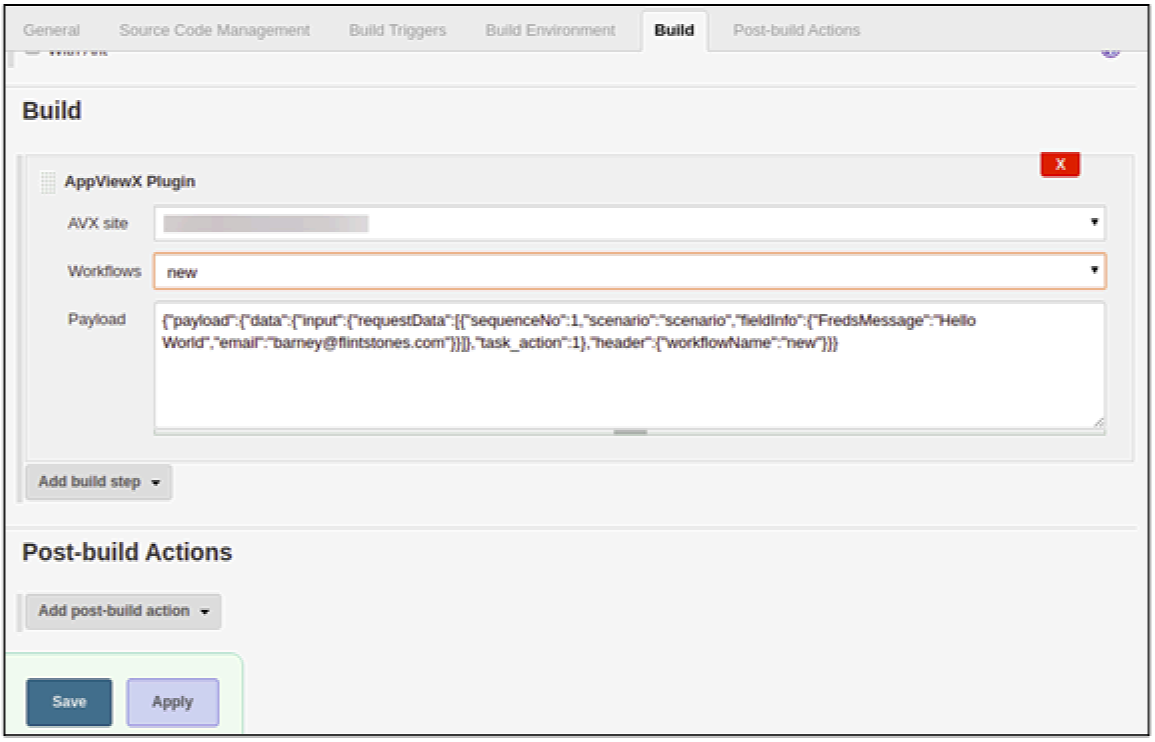
To create a freestyle job in Jenkins:

1. Click **New Item** in the upper left corner of the admin console.
2. Enter the name of the project.
3. Select **Freestyle Project**.

4. Create a build step using the AppViewX plugin in order to execute the automation workflow. The AppViewX plugin that has been installed will be available in the build step in the job configuration page.
5. Under **Add build step**, select **AppViewX Plugin** from the dropdown list.



6. Select the AppViewX site and the required workflow from the list.
   On saving the configuration, the payload will be auto-populated in the payload text area.



Creating a Pipeline Job

In Jenkins, a pipeline is a group of events or jobs which are interlinked with one another in a sequence.

Use the following exposed extension points in the pipeline script in the configuration page:

- To run the workflow in AppViewX and show each task with its name, logs and status in the console output:

```
appViewXAutomation(String siteName, String workflow, String payload)
```

- To submit a workflow request in AppViewX and print the request Id in the console output:

```
appViewXRaiseRequest(String siteName, String payload, String workflow)
```

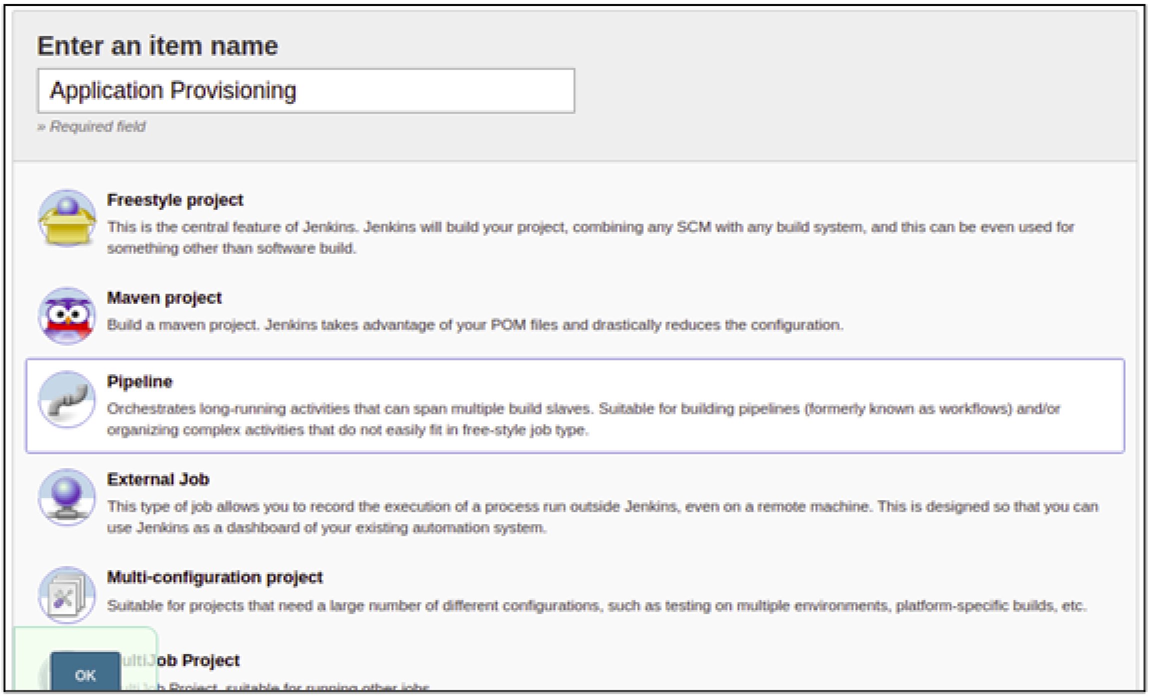
- To show the details of a set of tasks in Jenkins pipeline job (It shows the name, logs and the status of the tasks from the start task and end task):

```
appViewXCheckStageStatus(String siteName, String requestId, String startTask, String endTask)
```

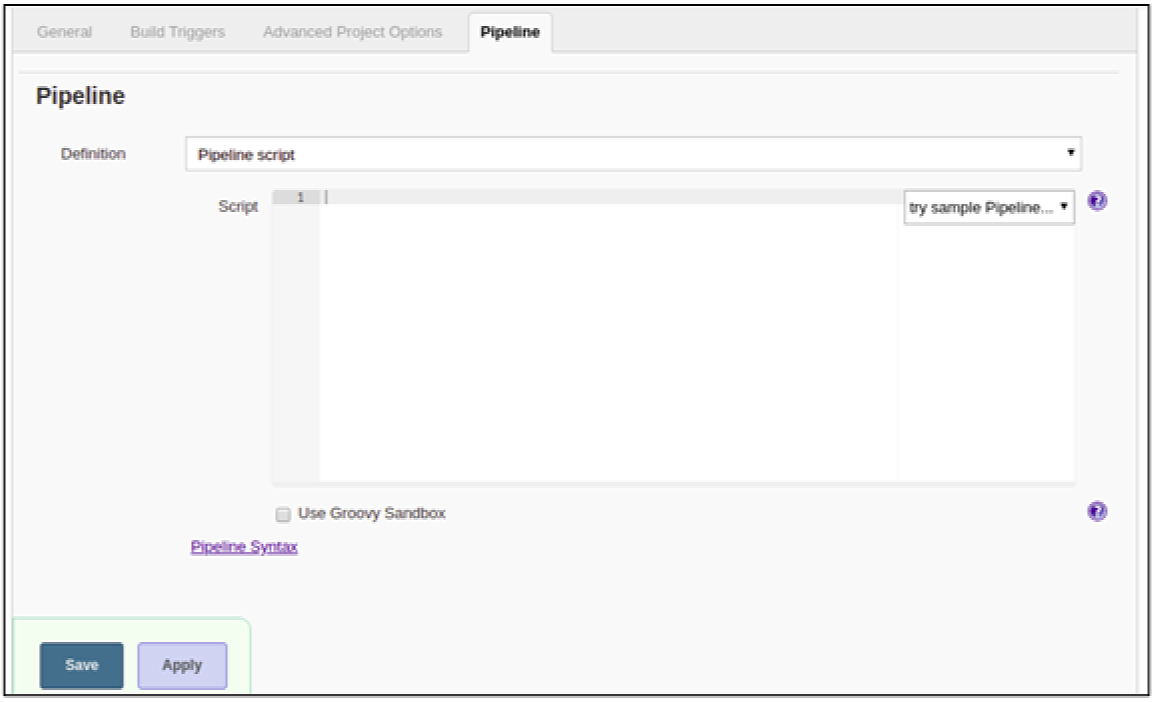
Jenkins pipeline for Application provisioning

To create a pipeline job in Jenkins:

1. Click **New Item** in the upper left corner of the admin console.
2. Enter the name of the project.
3. Select **Pipeline**.



4. Under **Pipeline**, enter the pipeline script, refer to the example pipeline script given here.

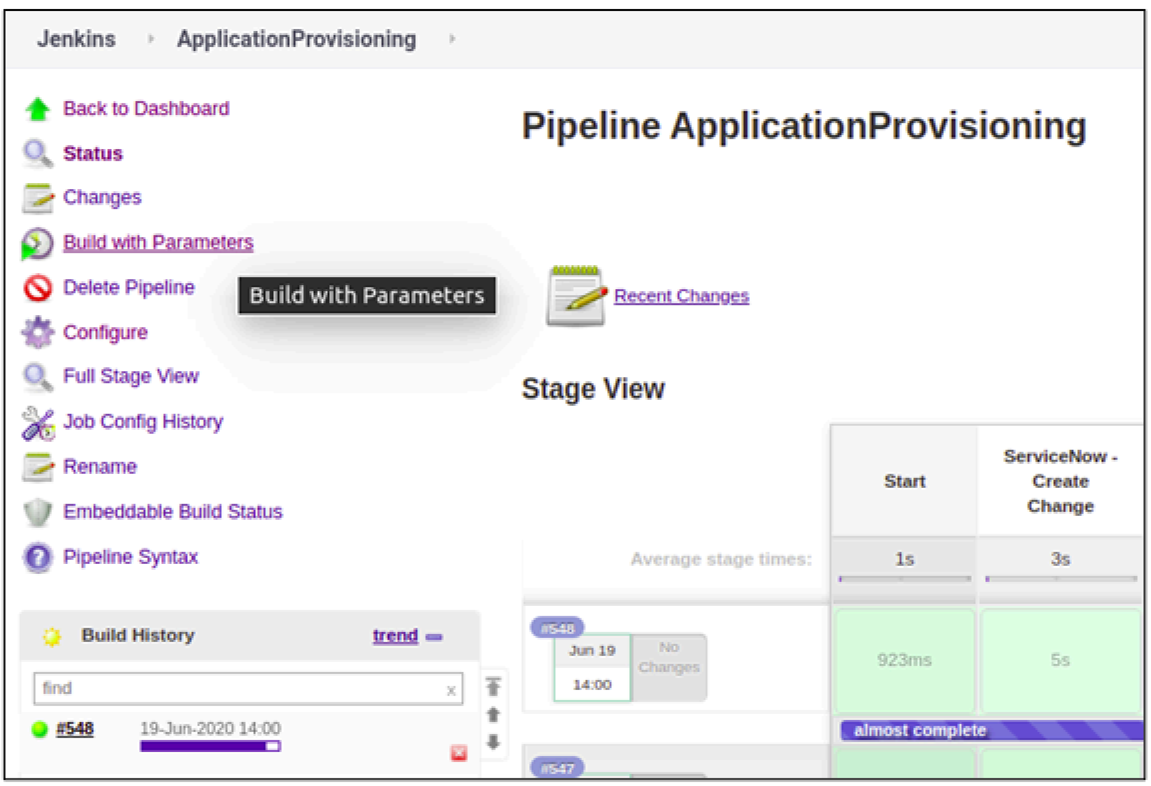


Example Pipeline Script:

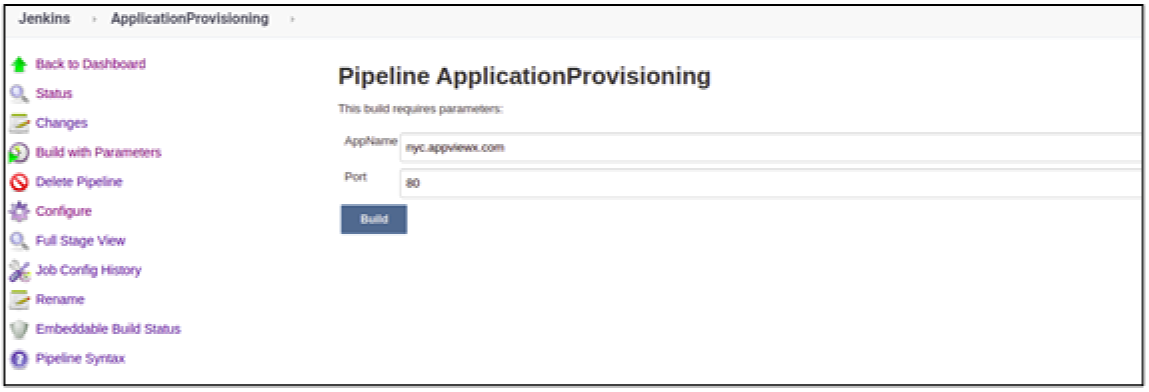
```
node("master") { stage ('Start')
{ appViewXRaiseRequest(siteName: "admin@https://192.168.142.172:5300",
payload:"{\"payload\":{\"data\":{\"input\":{\"requestData\":[{\"sequenceNo\":1,
\"scenario\":\"scenario\",\"fieldInfo\":{\"fqdn\":\"${AppName}\",\"port\":\"${Port}
\",\"name\":\"admin\"}}]},\"task_action\":1},\"header\":{\"workflowName\":
\"Application Provisioning\"}}}",workflow:"Application Provisioning")
script { env.logContent = Jenkins.getInstance().getItemByFullName(env.JOB_NAME)
.getBuildByNumber(Integer.parseInt(env.BUILD_NUMBER)).logFile.text scr = '''
LOG="$logContent" arr=() while read -r line; do arr+=("$line") done <<< "$LOG" for
(( counter = 0; counter <= 15; counter++ ))
do if grep -q "requestId" <<< ${arr[counter]};
then echo ${arr[counter]} | sed -e 's/ //g' | cut --complement -d ":" -f 1 fi done
''' env.RequestId = sh (script: scr,returnStdout: true).trim()
echo "Request Id - ${RequestId}" } } stage ('ServiceNow - Create Change')
 { appViewXCheckStageStatus(siteName:"admin@https://192.168.142.172:5300",
requestId: "${RequestId}", startTask: "",
endTask: "servicenow_create_change_2_1_1_2_2_1_3_1_1_1:Create Change Ticket") }
stage('DNS Provisioning') { appViewXCheckStageStatus
(siteName:"admin@https://192.168.142.172:5300",
requestId: "${RequestId}", startTask: "", endTask:
```

```
"script_1:Infoblox - Fetch Free IP's") }

stage('Certificate Generation') { appViewXCheckStageStatus

(siteName:"admin@https://192.168.142.172:5300", requestId: "${RequestId}",

startTask: "", endTask: "script_2:Certificate Generation") }

stage('F5 BIG IP Provisioning')

{ appViewXCheckStageStatus(siteName:"admin@https://192.168.142.172:5300",

requestId: "${RequestId}", startTask: "", endTask: "endWorkOrder_1:WorkOrder") }

stage('Firewall Provisioning') { appViewXCheckStageStatus

(siteName:"admin@https://192.168.142.172:5300",

requestId: "${RequestId}", startTask: "",

endTask: "bigip_firewall_policy_1:bigip_firewall_policy") }

stage ('ServiceNow - Close Change')

{ appViewXCheckStageStatus(siteName:"admin@https://192.168.142.172:5300",

requestId: "${RequestId}", startTask: "",

endTask: "servicenow_close_change_1_1_1_2_1_1:Close Change Ticket") } }
```

5. Add the contents and click **Save**.
6. On the stage view page, where the Pipeline build will be executed, choose **Build with Parameters** option.

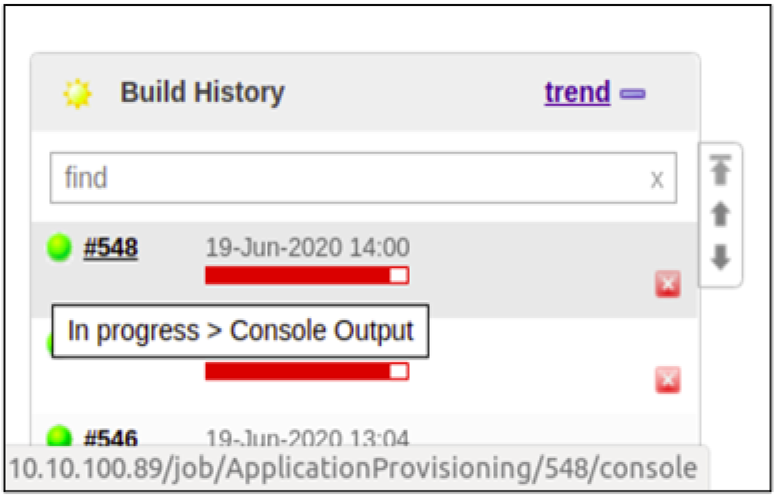


7. Enter the required inputs to run the workflow and click **Build**.

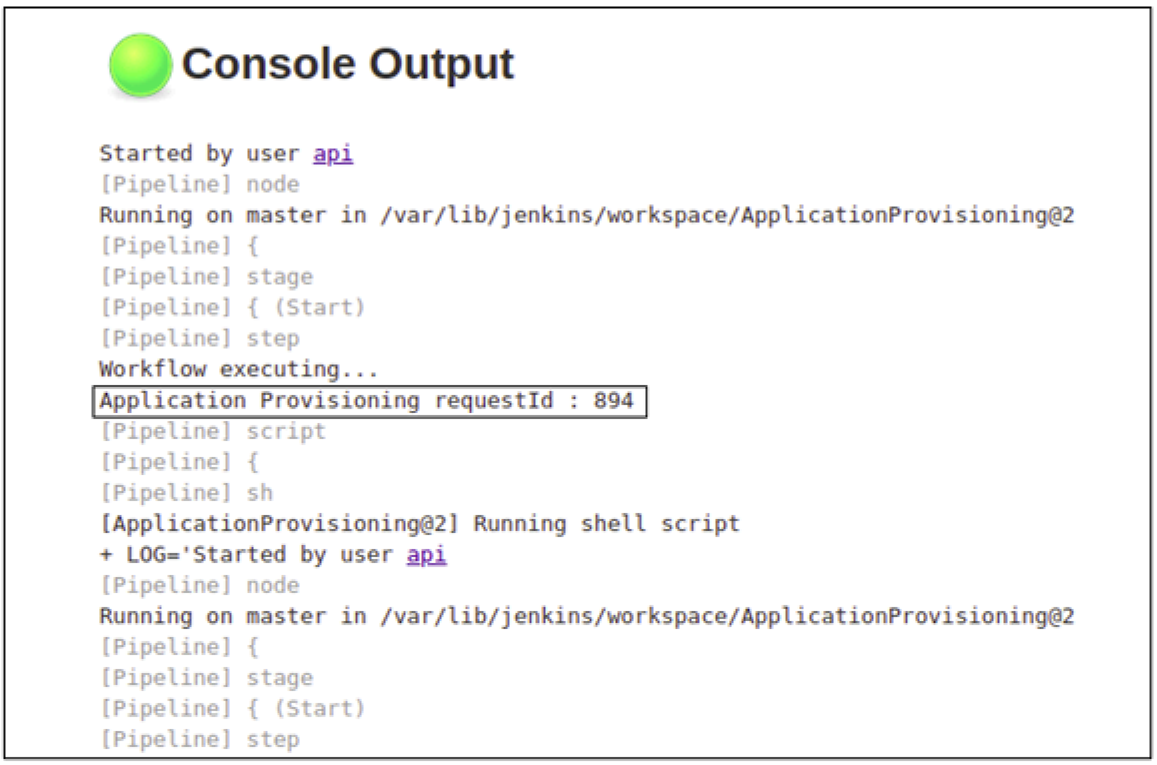


8. The stage view and console output can be seen when the pipeline is executed.
   - Pipeline stages

   - Console Output: Select the globe near the build number under Build History to see the outputs.



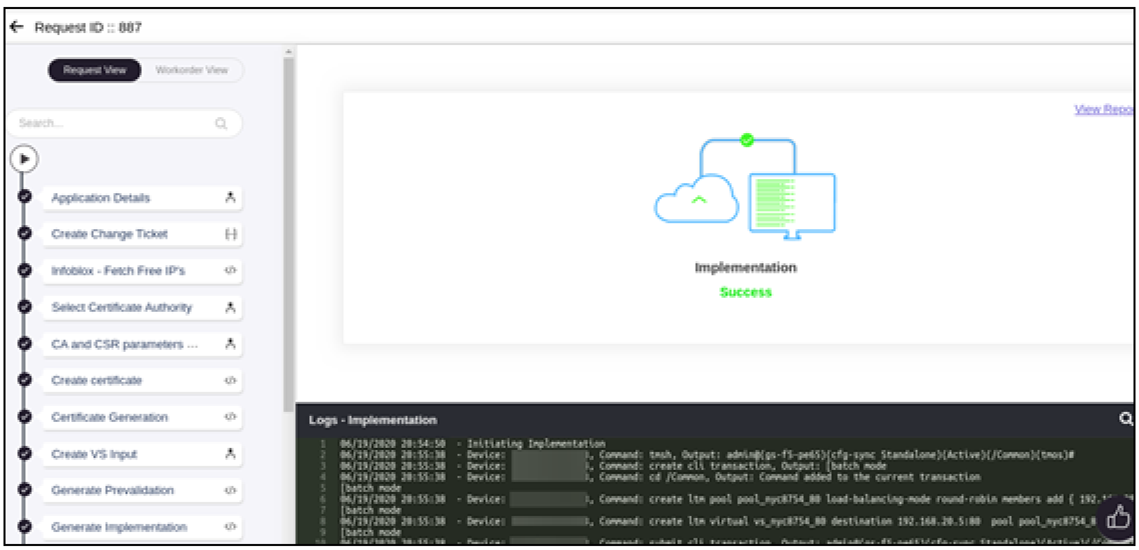
- The request Id of the workflow is shown in the console logs.



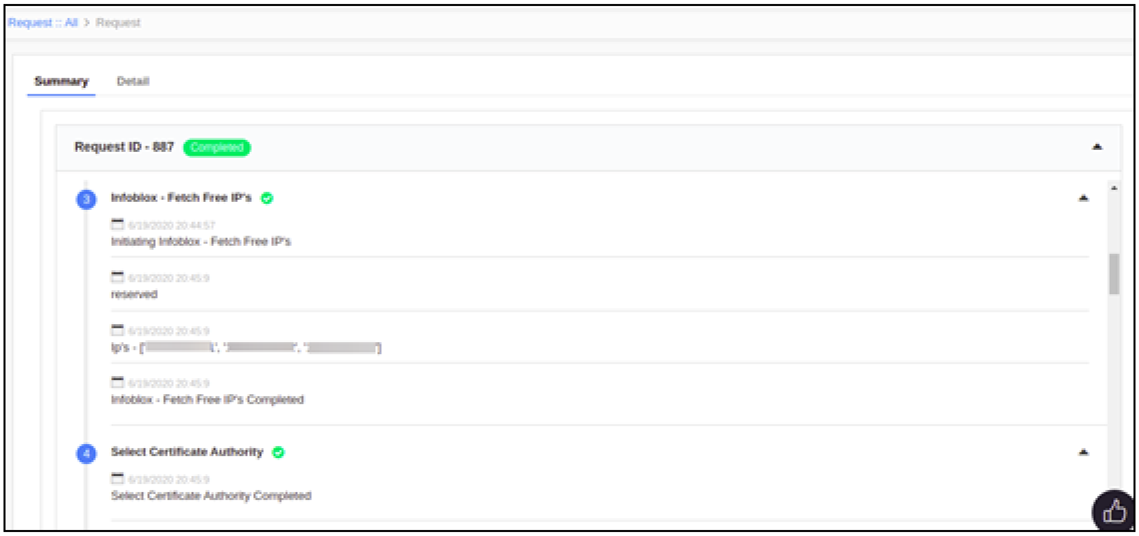
9. The task name, task logs and task status for each task in the workflow also shown in the console logs.


```
task name : Infoblox - Fetch Free IP's
Initiating Infoblox - Fetch Free IP's
reserved
Ip's - [[redacted]l', '[redacted]', '[redacted]']
Infoblox - Fetch Free IP's Completed
task status : Success


task name : Select Certificate Authority
Select Certificate Authority Completed
task status : Success


task name : CA and CSR parameters - AppViewX
Form has been submitted by user:admin
CA and CSR parameters - AppViewX Completed
task status : Success
```

10. On the Request :: All page, click on the **Request ID** of any workflow to see the execution of each task in the workflow.



11. To see the summary of each task in the workflow, under **Activity**, click **View**.

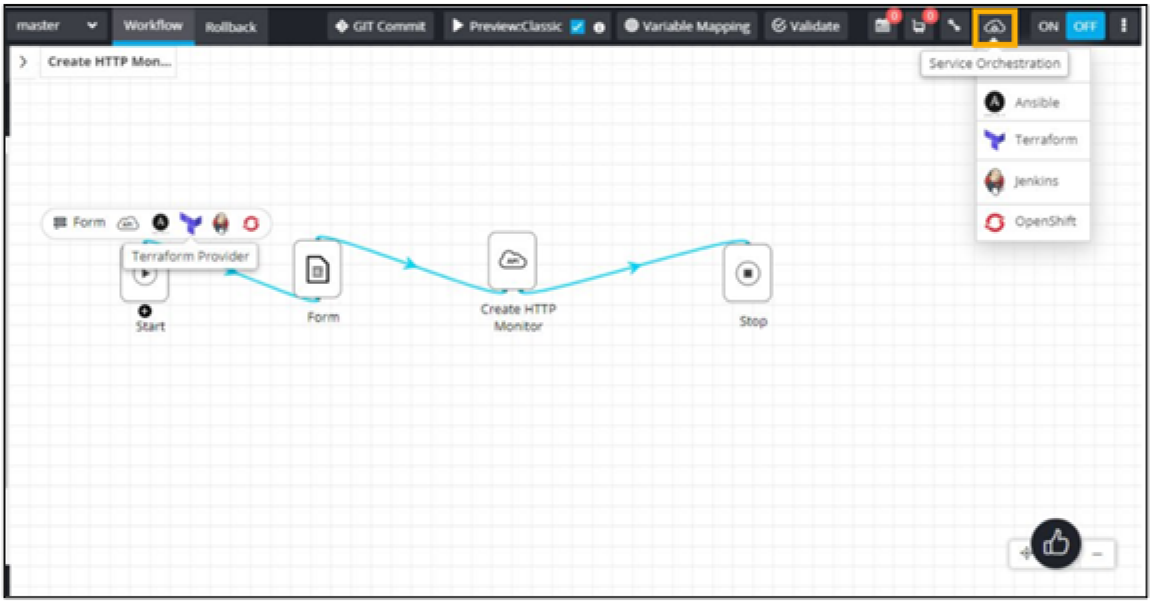


Terraform - Visual Workflow Northbound Integration

AppViewX Visual Workflow provides you with an interface to download Terraform (TF) files for all automation workflows. A TF file can be used with AppViewX Provider to submit automation requests on AppViewX Visual Workflow from Terraform.

1. Design a new workflow.
2. In the design space, to download the .zip file, click above the **Start** task.

> **Tip:** You can also click on the **Service Orchestration** icon in the command bar on the top, to download the <.zip> file.



3. Extract the <.zip> file.
4. From the extracted <.zip> file, copy the <workflowname>.tf file to the installation path of AppViewX Terraform provider.
5. In <workflowname>.tf file, update AppViewX instance details - AppViewX ID/FQDN, Username, Password, Protocol.

Sample:

```
provider "appviewx" {
  appviewx_username =
  appviewx_password = "
  appviewx_environment_is_https = true
  appviewx_environment_ip = "
  appviewx_environment_port =
}
```

6. Update Terraform meta arguments.

Sample:

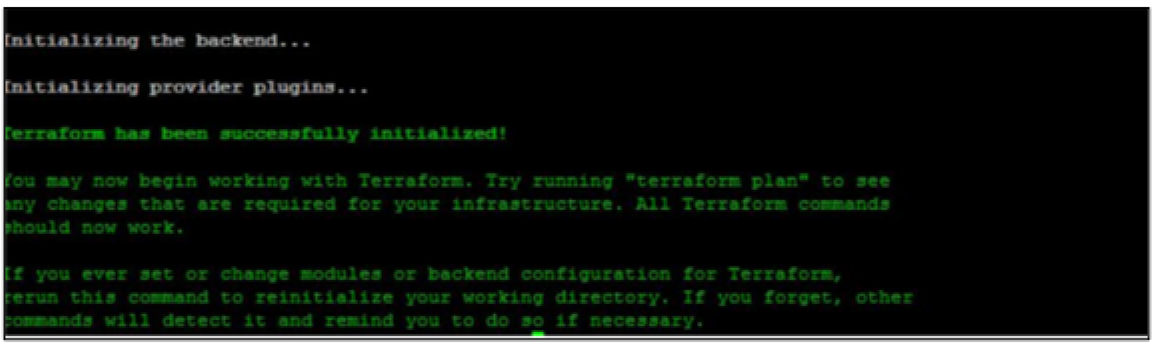
```
resource "appviewx_automation" "vip_lab.appviewx.com"
```

7. Enter workflow request details under the payload section.

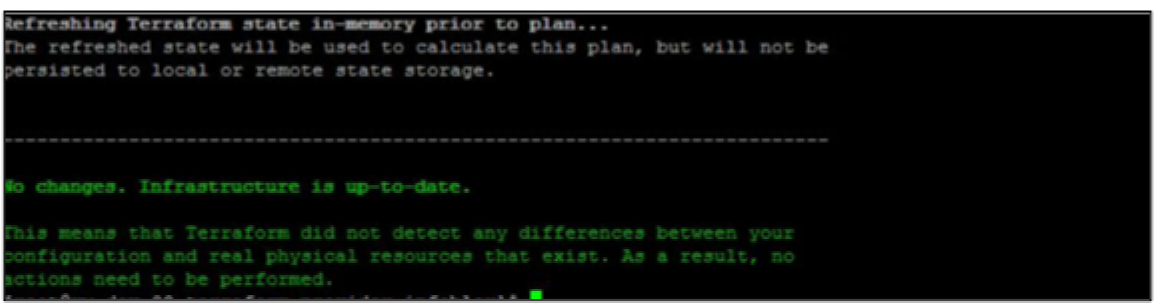
Sample:

```
payload= <<EOF
 {
{
 "payload" : {
  "header" : {
   "workflowName" : "command task"
  }
 }
}
EOF
action_id= "visualworkflow-submit-request"
 }
```

8. Save the changes and execute the file via Terraform commands.
   - terraform init



   - terraform plan



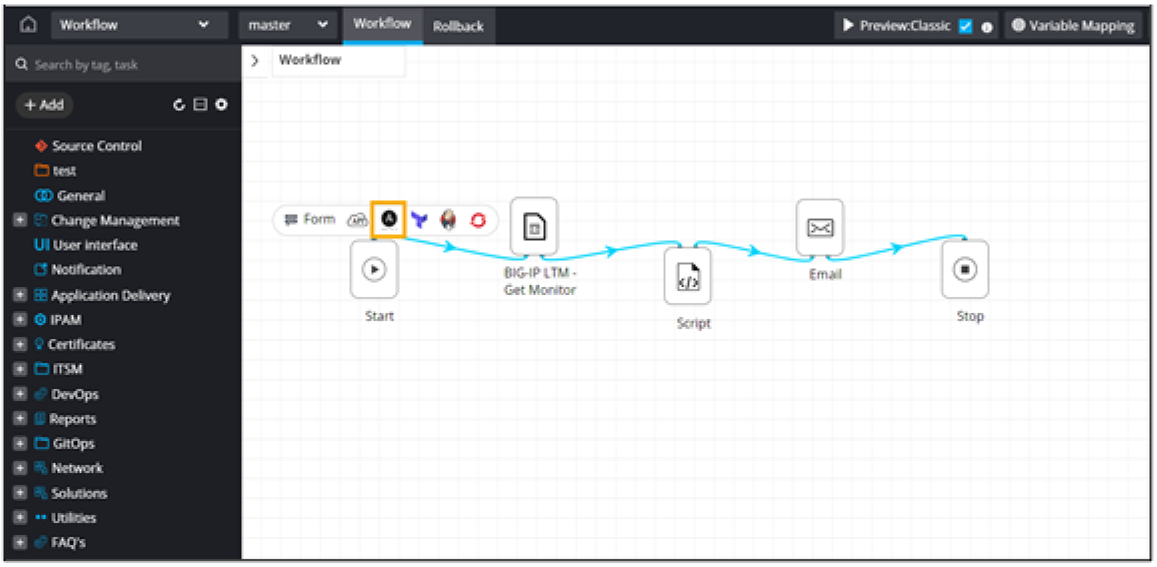
   - terraform apply.

Ansible - Visual Workflow Northbound Integration

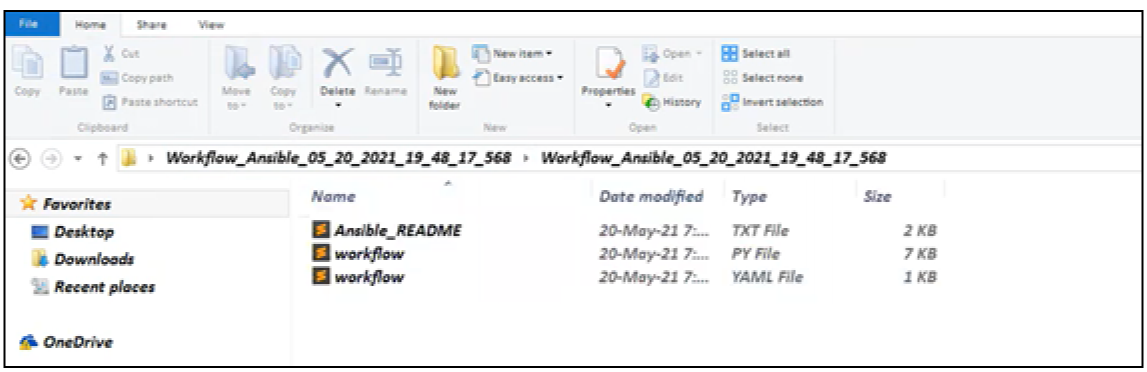
You can automate network configurations through Ansible as the Northbound.

To trigger a workflow by creating an Ansible playbook:

1. Design a workflow.
2. Connect all workflow tasks and enable the workflow.
3. To download the Ansible <.zip> file, click icon above the **Start** task.

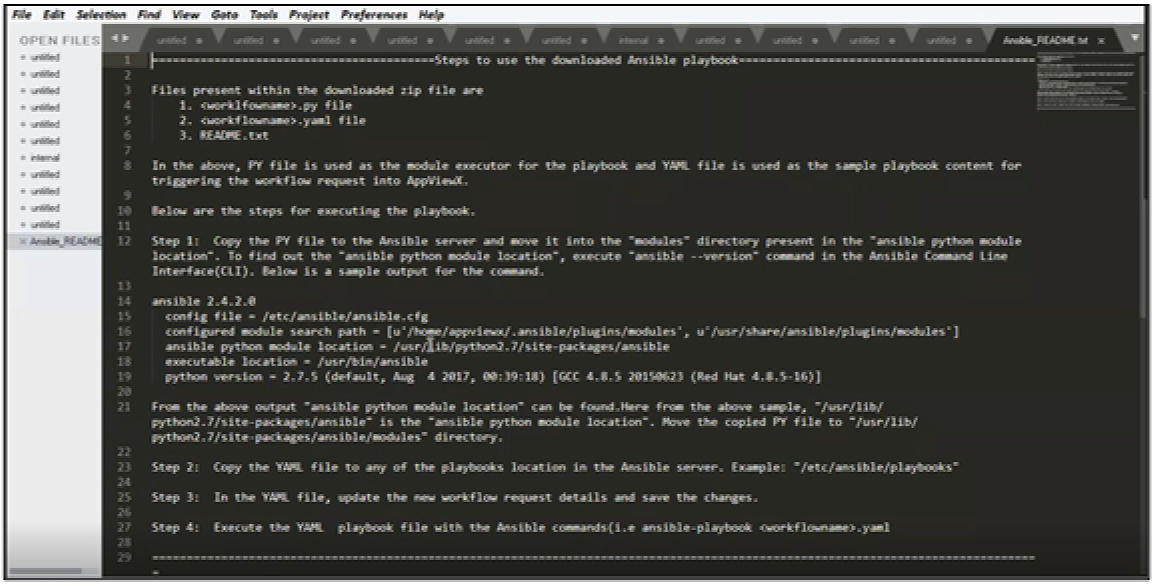


4. Extract the files.

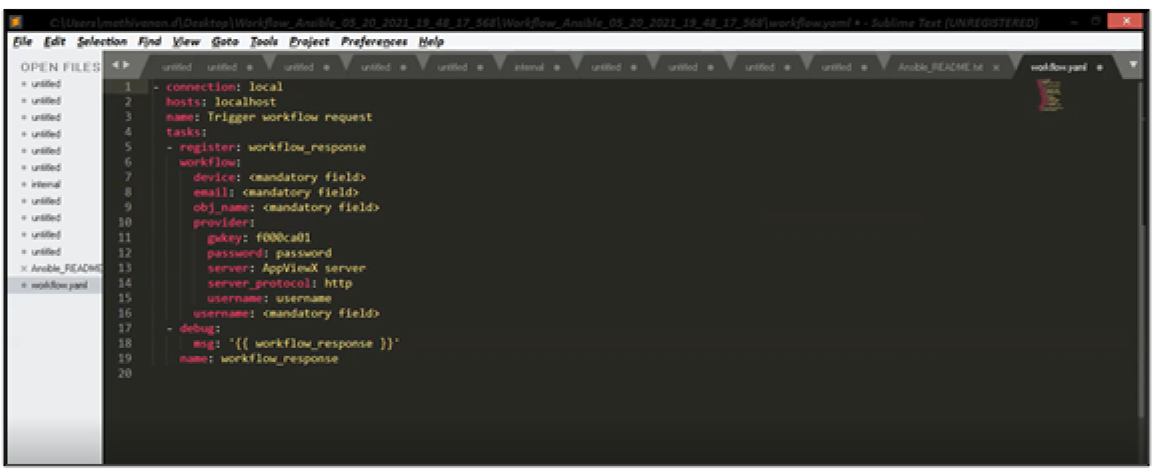


The downloaded <.zip> folder contains the following files:

- Ansible_README txt file - This contains the steps to use the downloaded Ansible playbook.



- <workflowname>.py - Workflow Python file (Ansible executor file)
- <workflowname>.yaml - Workflow YAML file



In the above, the Python (.py) file is used as the module executor for the playbook and the YAML file (.yaml) is used as the sample playbook content for triggering the workflow request into AppViewX.

5. To execute the playbook, copy the Python <.py> file to the Ansible server and move it into the modules directory present in the ansible python module location.
6. To find out the ansible python module location, execute the ansible --version command in the Ansible Command Line Interface (CLI).

   Here is a sample output for the command:

```
ansible 2.4.2.0
  config file = /etc/ansible/ansible.cfg
```



```
configured module search path = u'/home/appviewx/.ansible/plugins/modules', u'/usr/share/ansible/plugins/modules']
ansible python module location = /usr/lib/python2.7/site-packages/ansible
executable location = /usr/bin/ansible
python version = 2.7.5 (default, Aug  4 2017, 00:39:18) [GCC 4.8.5 20150623 (Red Hat 4.8.5-16)]
```

In this sample command, the ansible python module location is `/usr/lib/python2.7/site-packages/ansible`

7. Copy the YAML file to any of the playbooks location in the Ansible server. For example: `/etc/ansible/playbooks`
8. In the YAML file, update the new workflow request details and save the changes.
9. Execute the YAML playbook file with the Ansible commands - `ansible-playbook <workflowname>.yaml`

Openshift PaaS Orchestration

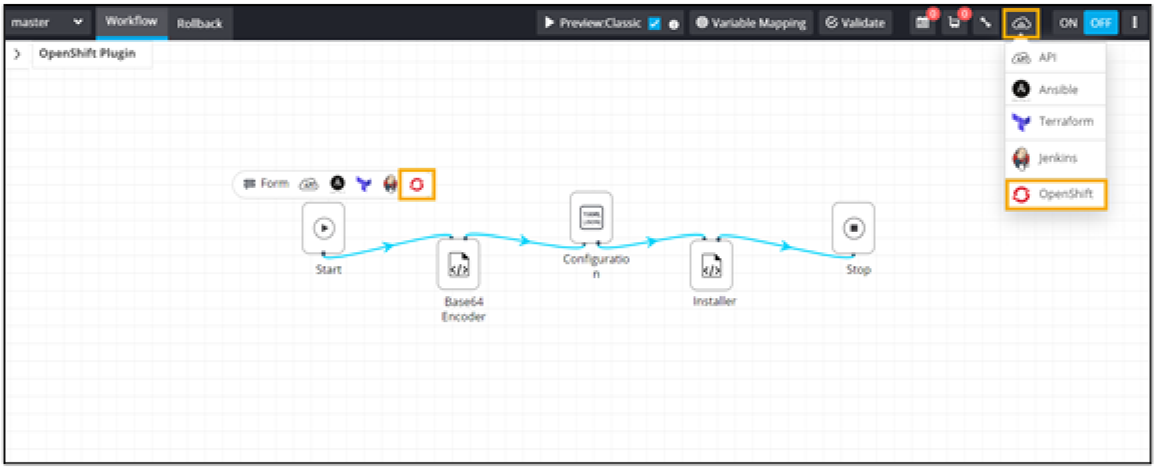
AppViewX's Visual Workflow provides you with an interface to install Open service Broker (OSB) that manages applications defined by AppViewX catalogs and accelerates application deployment.

- Offers a Kubernetes container platform to manage hybrid cloud deployments.
- Allows you to connect application deployed in OSE Container Platform to a variety of service brokers.
- OSB API manages applications defined by AppView catalogs.
- Enables PaaS Orchestration of application and PKI services.
- Accelerated application deployment.

To facilitate Openshift PaaS Orchestration:

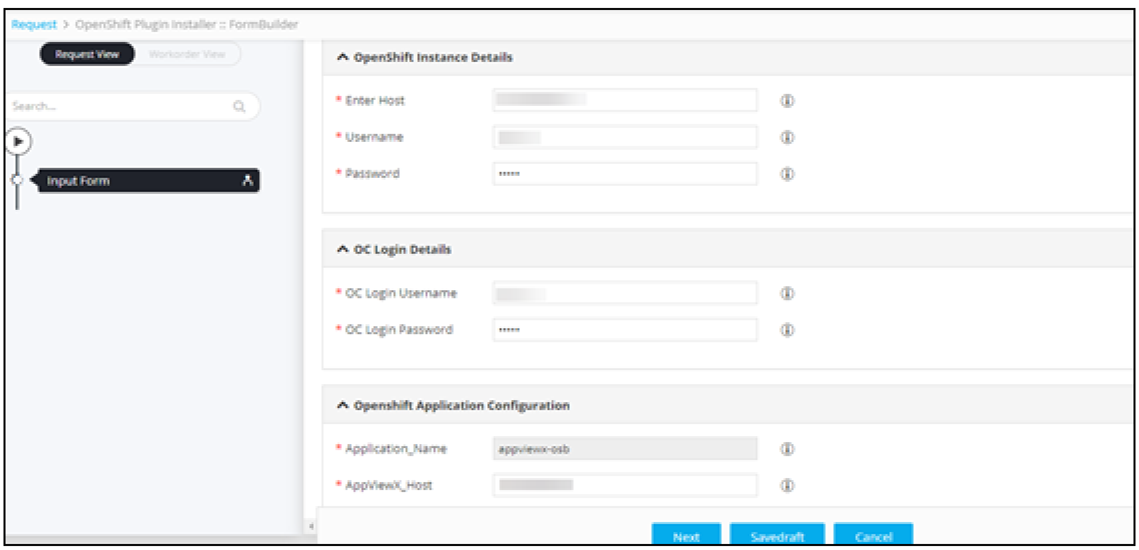
1. Design a new workflow.
2. To download the OSB Plugin manually, click icon above the **Start** task.

> **Tip:** You can also click on the **Service Orchestration** icon to install the plugin.

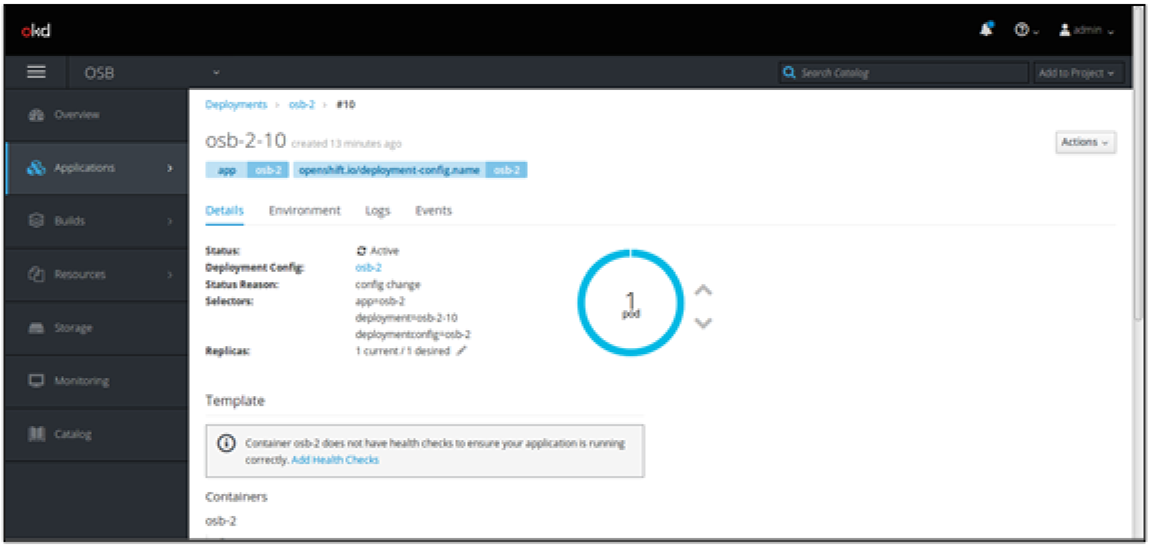


3. To install the AppViewX OSB plugin automatically, trigger the OpenShift Plugin Installer workflow from the **Request :: View/Run** page.

   The workflow execution page is displayed with the workflow inputs requested at the first stage.
4. Enter the OSB details in the fields.



5. To proceed with plugin installation, click **Next**.
6. Open the AppViewX Open Service Broker.



7. View all automation catalogs in OpenShift once the AppViewX broker is installed.
8. To provision application and security services through AppViewX, click on any Automation catalog.

> **Note:** The following REST API is used to orchestrate and provision services between openshift and appviewx.

- **API**: visualworkflow-get-all-workflows

  Description: To fetch the list of workflows in the AppViewX environment

  Method: POST.

Type	Name	Mandatory	Description
Header	username	yes	AppViewX username
Header	password	yes	AppViewX password
Header	Content-Type	yes	application/json
Header	Accept	yes	application/json
Query	gwkey	yes	Tenant key
Query	gwsource	yes	Source from which the request is triggered (Example, external)

Format: https://<application ip>:<gatewayport>/avxapi/visualworkflow-get-all-workflows?gwkey=<tenant key>&gwsource=<gateway source>

Sample Payload: {"payload":{"sSearch":""}}

Sample Response:{"response":{"workflowTemplateList":[{"workflowName":"Create LTM HTTP Monitor","accessType":"RW","properties":{"version":"19.3.0","category":38,"categoryName":"F5 BIG-IP","subCategory":2,"subCategoryName":"Ansible","internalWorkflow":false,"readOnly":false,"hidden":false,"locked":false},"settings":{"disableRequestorSubmission":false,"tags":[],"hideWorkflow":false,"enableRequest":true,"confirmationAlert":true},"globalData":{},"_id":"5f1011f4b958f463955d2b6e","description":"","created_by":"admin","status":"Enabled","workflowCategory":"Default","workflowVersion":"mas

- **API**: visualworkflow-generate-api

  Description: To generate the payload for specific workflow

  Method: POST

Type	Name	Mandatory	Description
Header	username	yes	AppViewX username
Header	password	yes	AppViewX password
Header	Content-Type	yes	application/json
Header	Accept	yes	application/json
Query	gwkey	yes	Tenant key
Query	gwsource	yes	Source from which the request is triggered (Eg: external)

Format:https://<application ip>:<gateway port>/avxapi/visualworkflow-generate-api?gwkey=<tenant key>&gwsource=<gateway source>

Sample Payload: {"payload":{"workflowName":"Create LTM HTTP Monitor","workflowType":"Default","workflowVersion":"master"}}

Sample Response:{"response":{"url":"<protocol>://<IP>:<port>/avxapi/visualworkflow-submit-request?gwkey=f000ca01&gwsource=WEB","actionId":"visualworkflow-submit-request","httpMethod":"POST","payload":{"payload":{"data":{"input":{"requestData":[{"sequenceNo":1,"scenario":"scenario","fieldInfo":{"device_ip":"192.168.40.214","device_name":"192.168.40.214","interval":"5","timeout":"2","monitor_name":"demmoo_monitor"}}]},"task_action":1},"
{"workflowName":"Create LTM HTTP Monitor"}}}},"message":null,"appStatusCode":null,"tags":null,"headers":null}

- **API**: visualworkflow-submit-request

  Description: To submit a workflow request in AppViewX

  Method: POST

Type	Name	Mandatory	Description
Header	username	yes	AppViewX username
Header	password	yes	AppViewX password
Header	Content-Type	yes	application/json
Header	Accept	yes	application/json
Query	gwkey	yes	Tenant key
Query	gwsource	yes	Source from which the request is triggered (Eg: external)

Format:https://<application ip>:<gateway port>/avxapi/visualworkflow-submit-request?gwkey=<tenant key>&gwsource=<gateway source>

Sample Payload:"payload": { "data": { "input": { "requestData": [{ "sequenceNo": 1, "scenario": "scenario", "fieldInfo": { "device_ip": "192.168.40.214", "device_name": "192.168.40.214", "interval": "5", "timeout": "2", "monitor_name": "ltm_monitor_test" } }] }, "task_action": 1 }, "header": { "workflowName": "Create LTM HTTP Monitor" } } }

Sample Response:{"response": {"workorderId":"0","requestType":"default","requestId":"38","workflowVersion":"master","message":"Workflow Request is created with Id 33 . Request submitted to workflow engine for processing workorder.","status":"In Progress","statusCode":0}}

Service Requests

The Service Requests console allows you to execute workflow(s). Workflows can be executed manually or can be scheduled. The **Request :: Overview** page provides a one-stop view of all workflow requests by status - opened, closed, failed and total workflow requests. It allows you to generate dynamic reports on the status of the workflow requests.

- Configuring RBAC for Workflow Request Stage View
- Accessing Service Requests
- Request :: Overview
- Custom Reports
- Request :: View/Run

- Scheduled Jobs
- Request :: All
- Request :: Open
- Request :: Closed
- Request :: Failed
- Request :: Assigned
- Request :: Audit Logs
- Configuring RBAC for Archive and Restore Requests
- Archive Requests
- Restore Archives
- Preferences
- Service Request Actions

Configuring RBAC for Workflow Request Stage View

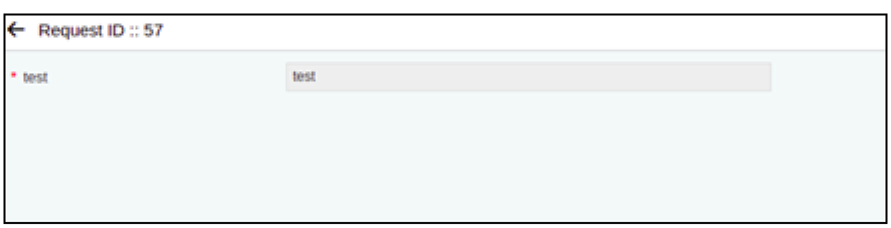
You can configure permissions to allow or restrict users to access the stage view when the workflow request is executed.

To hide the stage view of workflow stages in workflow execution:

1. Navigate to the **Role** page in the **Platform** module.
2. Select the role for which you want to enable access. For example, **admin**.
3. Under **Authorized functions**, expand **Automation** > **Service Requests** > **User Preference**.

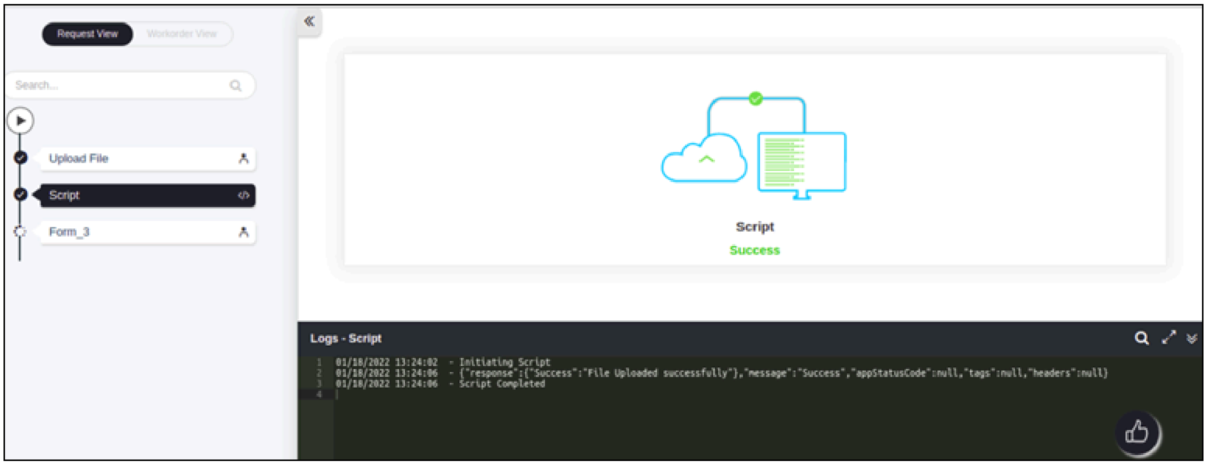
> **Note:** If you are using the old menu, under the **Authorized Functions** tab, expand **Request** > **Workflow**.

4. Under **User Preference**, ensure that the **Show request stage** checkbox is **not** selected.
5. Click **Save**.
   The stage view is not displayed for the user and only the details of the current task are displayed.



6. To display the stage-wise view in workflow execution, under **User Preference**, select the **Show request stage** checkbox.

   List of tasks is displayed on the left side of the screen.

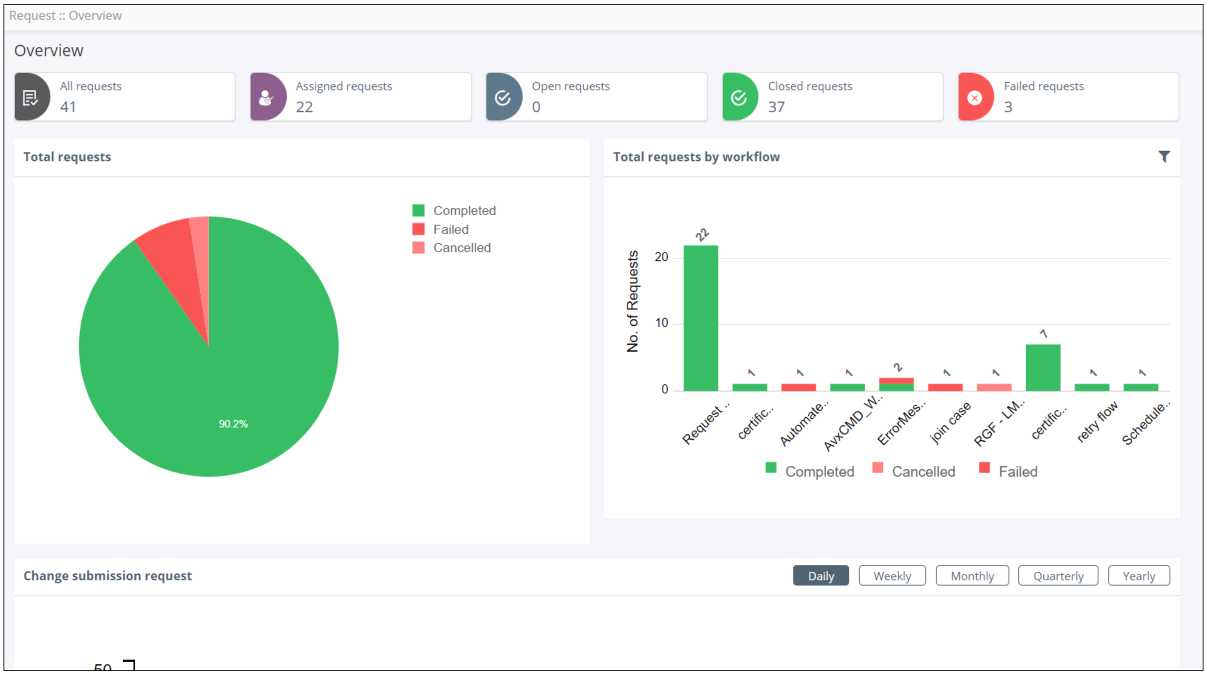


Accessing Service Requests

New Menu	Old Menu
From the main navigation menu, select **Automation**. The Request :: Overview page is displayed.	From the main navigation menu, select **Request**. The Request :: Overview page is displayed.
Note: For more information on how to switch between menus, click here.	

Request :: Overview

The **Request :: Overview** page provides a one-stop view of all workflow requests by status - opened, closed, failed, assigned, and total workflow requests. It allows you to generate dynamic reports on the status of the workflow requests.



Click on any section of the chart or metric and navigate directly to the inventory. Following are the possible statuses for a workflow request:

Status Descriptions

Status	Description
In-progress	The request is awaiting a response to proceed.
Completed	The request is complete and successful.
Failed	The request failed during execution.
Partial	The request is partially executed.
Rolled Back	The request is complete and has rolled back to its previous state.
Paused	The request is suspended/paused.
Aborted	The request is aborted.

Total requests

The **Total Requests** pie chart displays all the service requests as per their status.

Total requests by workflow

The **Total Requests by workflow** bar chart displays the number of requests triggered for each workflow along with their statuses

Change Submission Request

The **Change submission requests** line chart helps you visualize the request trend over a period of time.

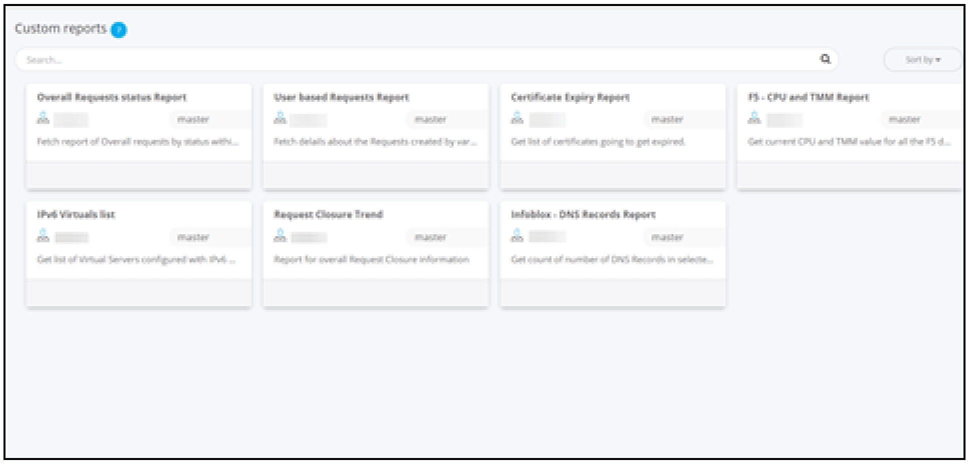
• Provision to perform trend analysis on the requests triggered over a period of time
• Provision to create daily, weekly, monthly and yearly views for the requests triggered.

Custom Reports

This section allows you to view and trigger custom reports.

• Any workflow created under the category Reports will be available under Custom report section.
• Provision to search for a custom report.
• Provision to apply custom sort based on ascending, descending, recently created.

- Provision to execute/run a custom report workflow.



- Triggering Custom Reports

Triggering Custom Reports

To execute a custom report workflow:

1. On the Request :: Overview page, from the menu on the left, click **Custom Reports**.
2. To execute any report workflow on the **Custom Reports** page, hover your mouse over the workflow and click **(Trigger)** icon.
   The workflow execution page is displayed with the workflow inputs requested at the first stage.

> **Note:** The inputs requested and workflow execution steps will vary as per the workflow that is triggered.

Request :: View/Run

The **Request :: View/Run** page displays the workflow catalogs.

- Provision to search for a workflow by keyword.
- Provision to trigger a workflow on demand.
- Provision to schedule a workflow.

- Triggering a Workflow
- Scheduling a Workflow

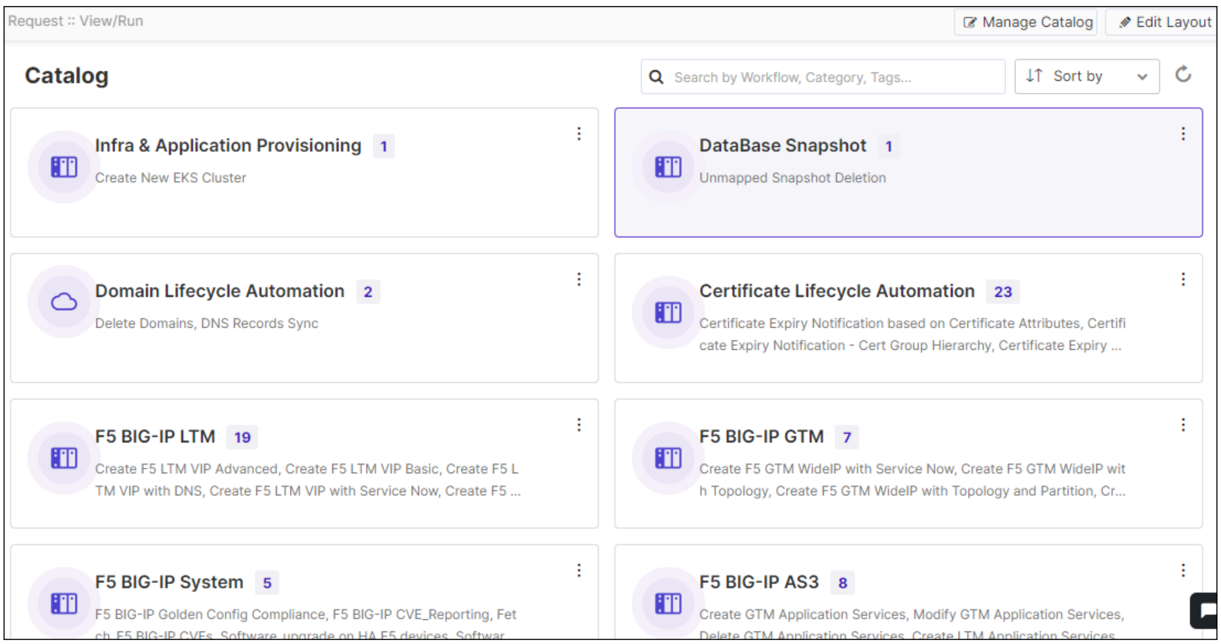
- Editing the Workflow Catalog Layout
- Editing Subcategories and Workflows within a Catalog

Triggering a Workflow

To trigger a workflow from the Workflow **Request :: View/Run** page:

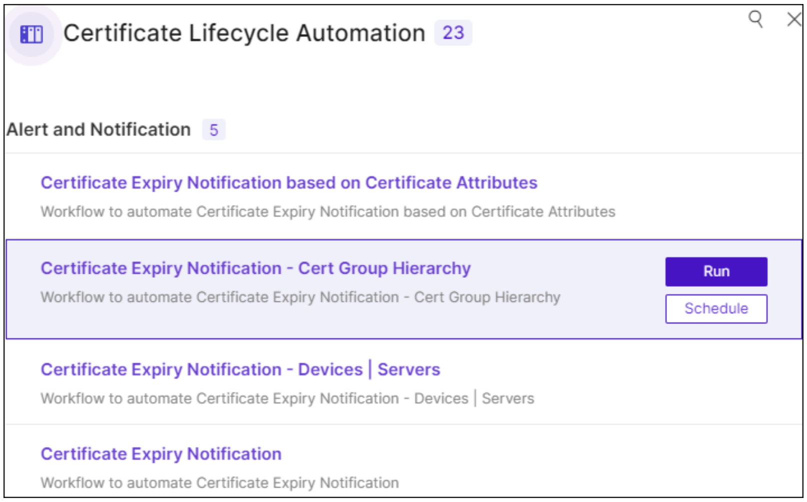
1. On the Request :: Overview page, from the menu on the left, click **View/Run**.

   The **Request :: View/Run** page is displayed.

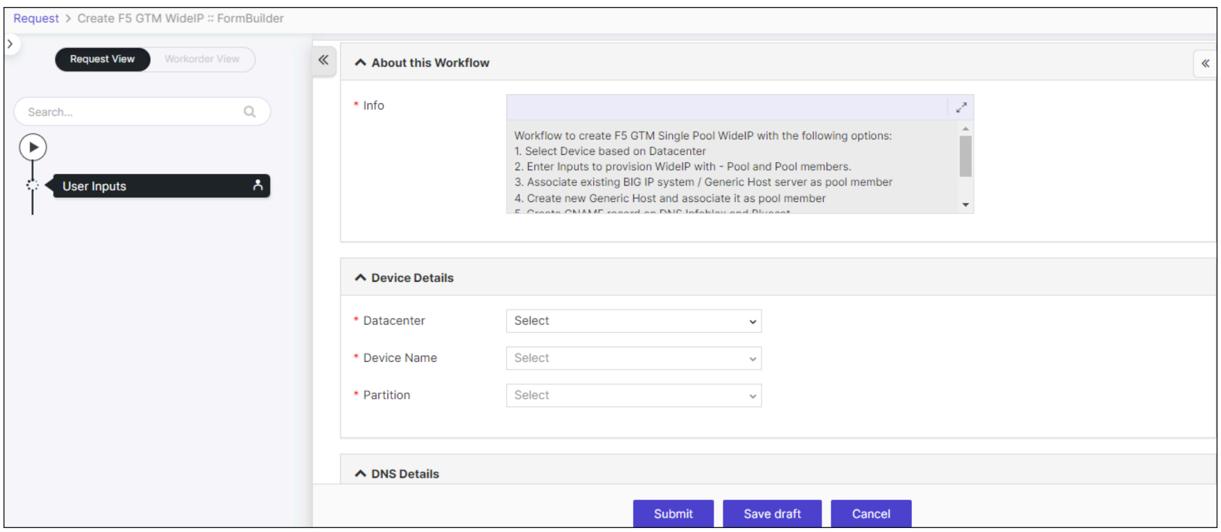


Note: Only enabled workflows are displayed on the **Request :: View/Run** page.

2. To trigger a workflow, on the **Request :: View/Run** page, search for the workflow.
3. Locate the workflow in the catalog and click the catalog name.
4. Hover your mouse over the workflow name, and then click Run **(Run)** button to initiate the workflow.



The workflow is executed with the user inputs requested at the first stage.



Scheduling a Workflow

This task allows you to schedule a workflow to trigger once or daily, weekly, monthly and yearly as per requirement.

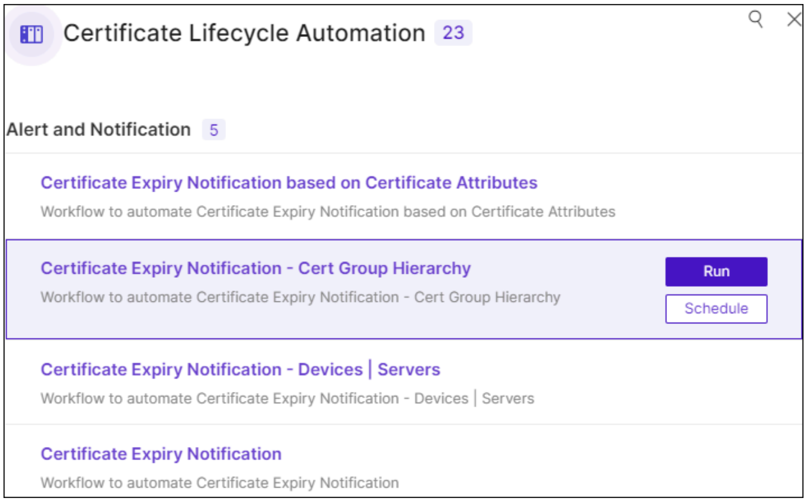
1. On the Request :: Overview page, from the menu on the left, click **View/Run**.

   The **Request :: View/Run** page is displayed.

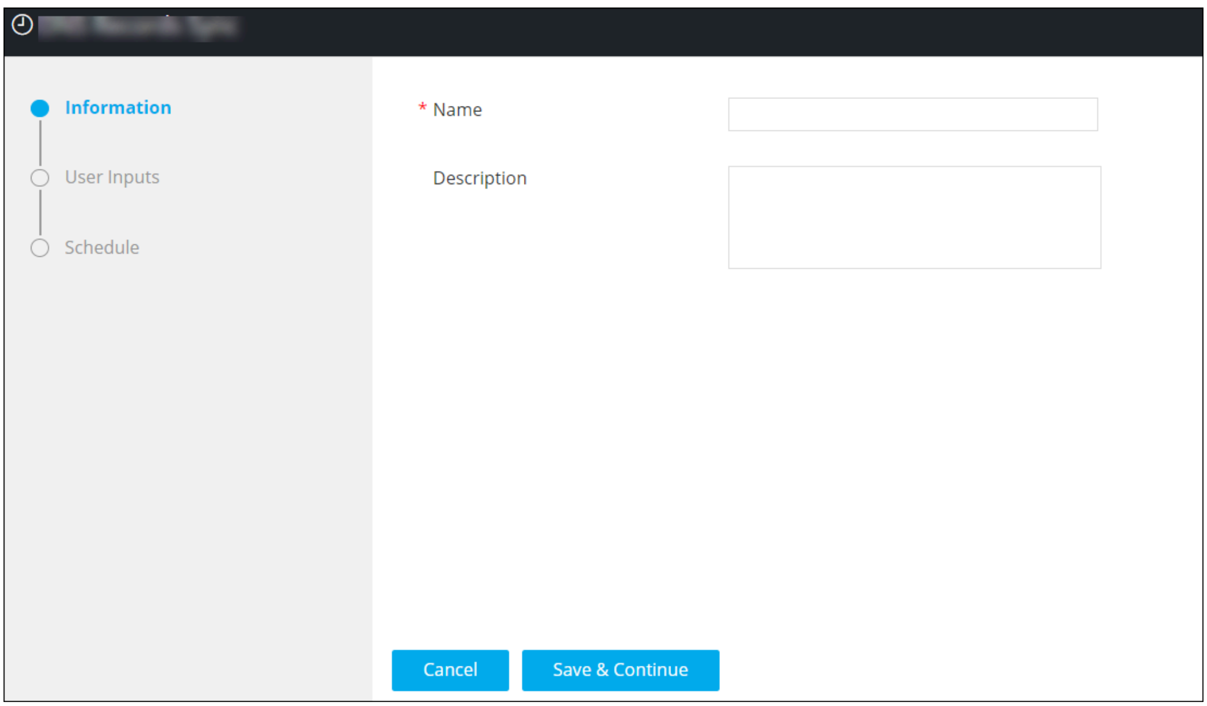
   Note: Only enabled workflows are displayed on the **Request :: View/Run** page.

2. To schedule a workflow, on the **Request :: View/Run** page, search for the workflow.

3. Locate the workflow in the catalog and click the catalog name.
4. Hover your mouse over the workflow name, and then click Schedule **(Schedule)** button.



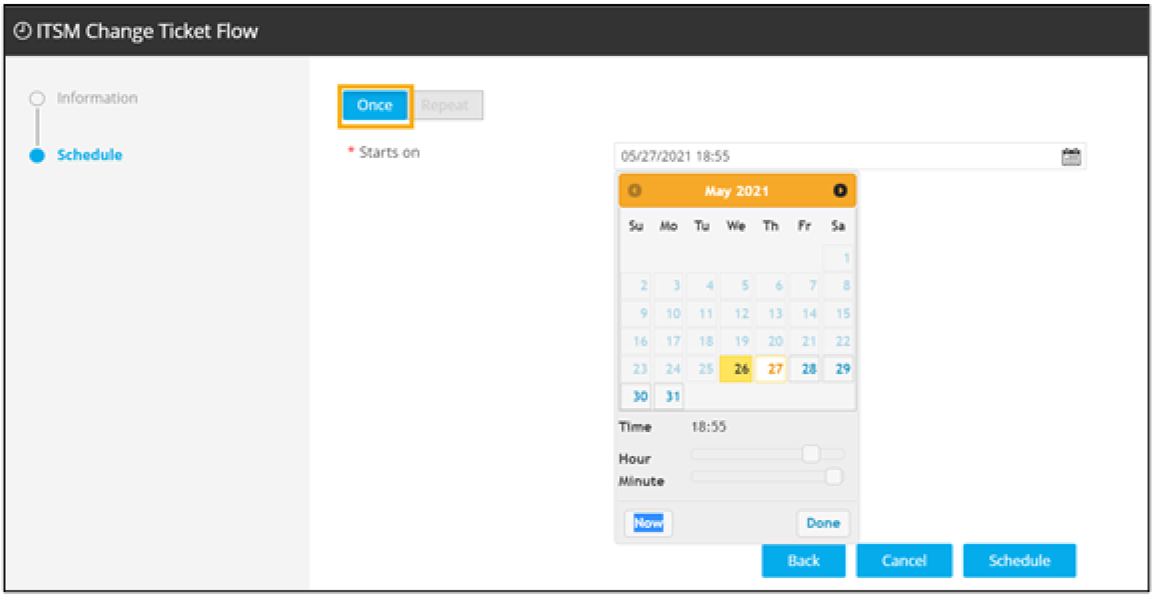
The schedule workflow page is displayed.



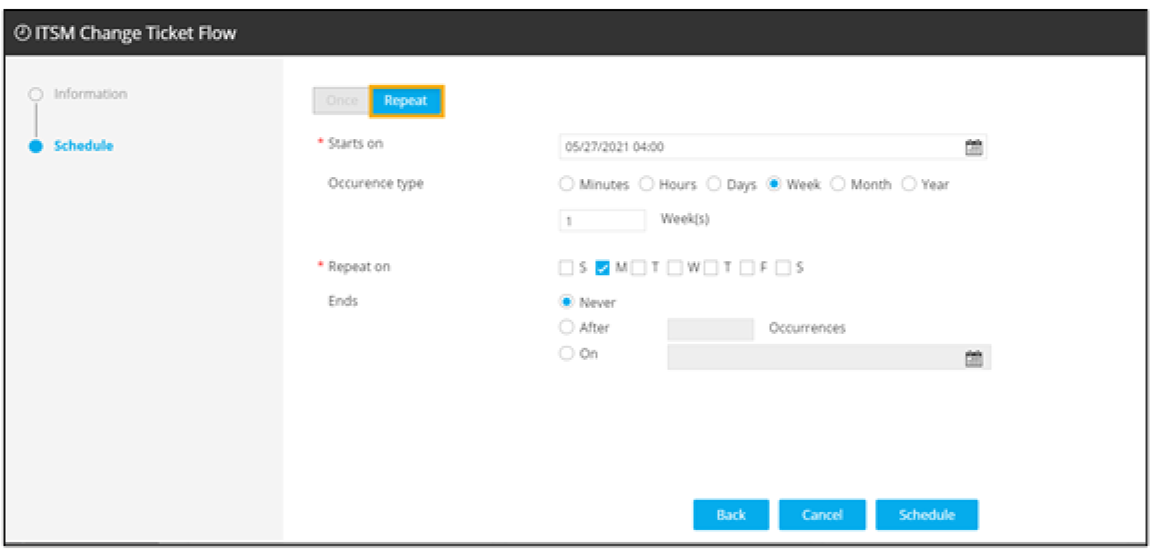
5. In the **Information** section, enter the **Name** of the scheduled job.
6. In the **User Inputs** section, enter the required field information.

> **Note:** This field is displayed only if the workflow requires user inputs. The user inputs required will vary according to the workflow.

7. In the **Schedule** section, assign the time for which the workflow is to be scheduled.
   - **Once**



   - **Repeat -** Provision to schedule the workflow to trigger daily, weekly, monthly and yearly.



8. Click **Schedule**.

   The scheduled workflow is added to the **Scheduled jobs** section.

Editing the Workflow Catalog Layout

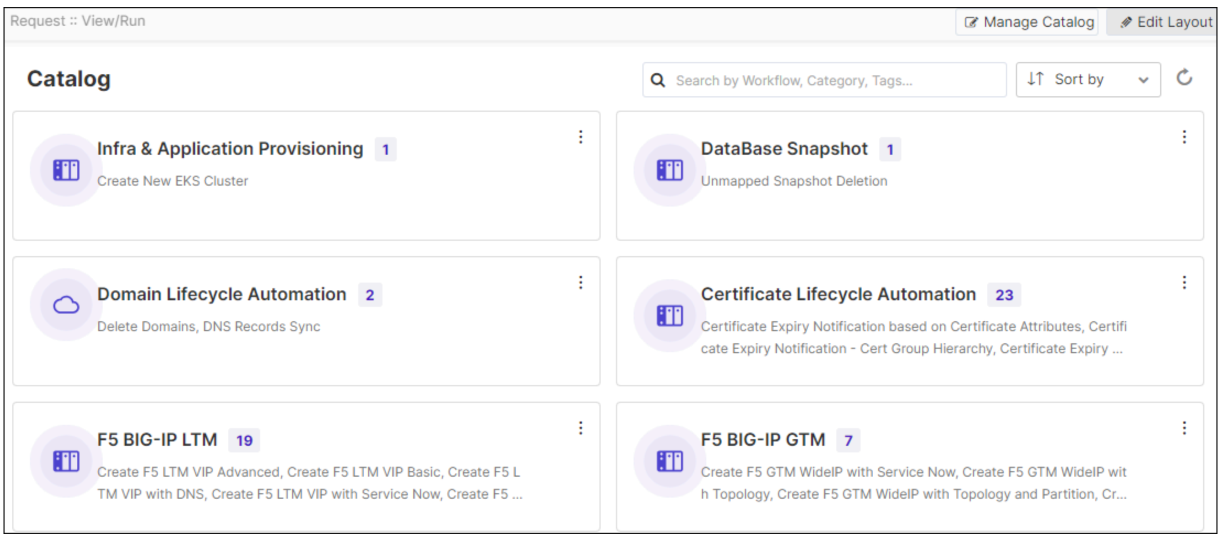
You can move the catalogs around and place them according to your preference on the View/Run page. You can also expand or collapse all workflow catalogs. Expanding the catalogs displays the list of workflows with a catalog.

To edit the layout:

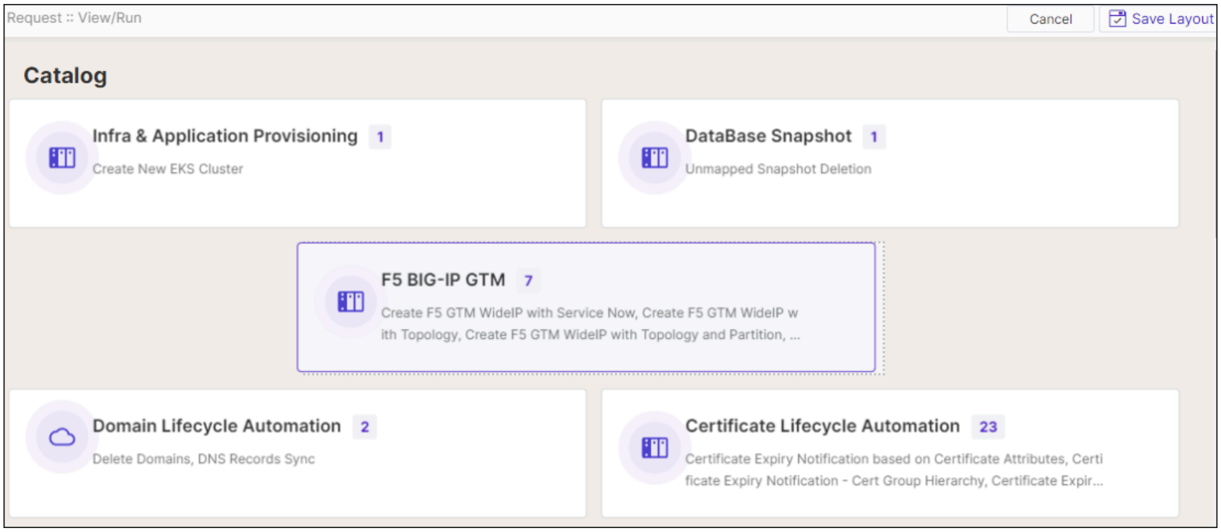
1. On the Request :: Overview page, from the menu on the left, click **View/Run**.

   Note: Only enabled workflows are displayed on the **Request :: View/Run** page.

2. On the **Request :: View/Run** page, from the top right corner of the screen, click **Edit Layout**.



3. Hold and drag the workflow catalog to another area on the page.



4. To save your layout preferences, from the top right corner of the page, click **Save Layout**.

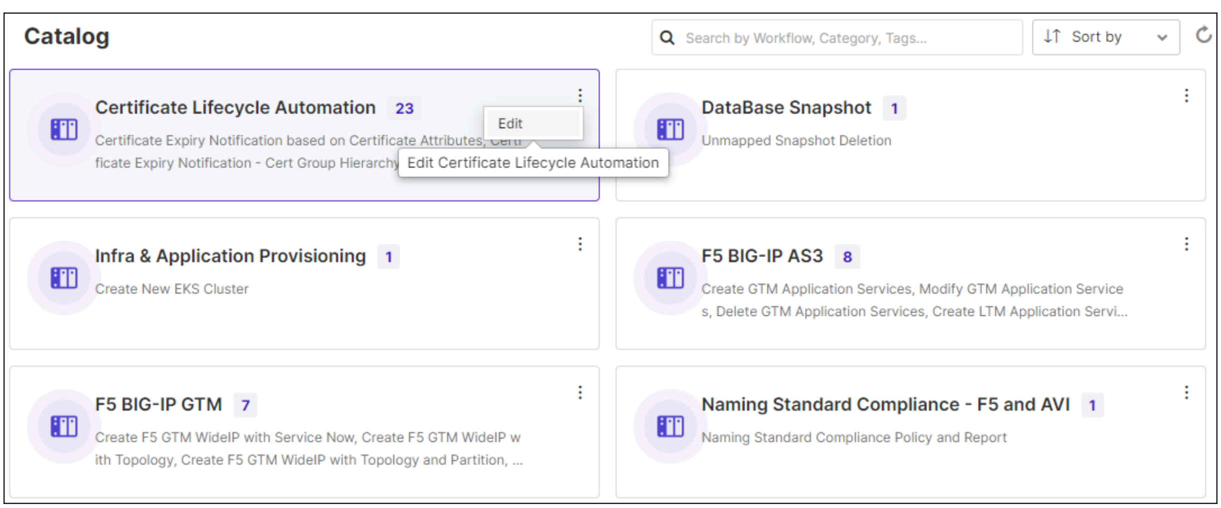
Editing Subcategories and Workflows within a Catalog

You can select the workflows to be placed under a workflow catalog and modify the order of sub-categories within the workflow catalog.

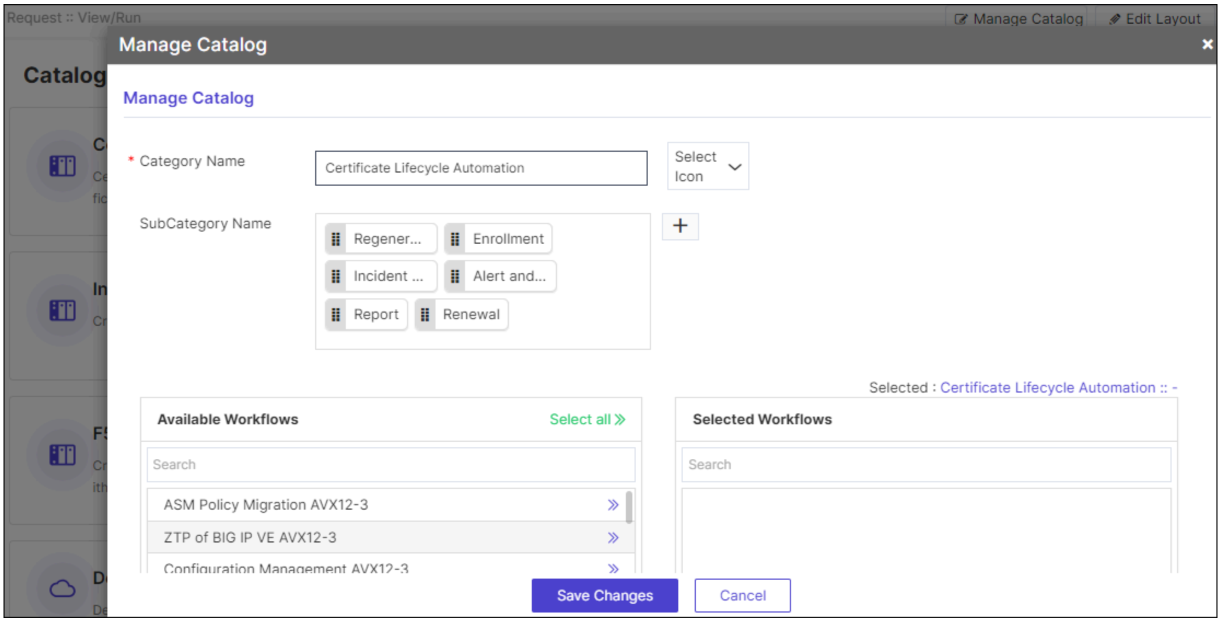
1. On the Request :: Overview page, from the menu on the left, click **View/Run**.

> **Note:** Only enabled workflows are displayed on the **Request :: View/Run** page.

2. On the **Request :: View/Run** page, from the top right corner of the screen, click ⋮.
3. From the options displayed, select **Edit**.



4. In the **Manage Catalog** window that is displayed, under **SubCategory Name**, hold and drag folders to modify the order in which they are displayed on the workflow card.

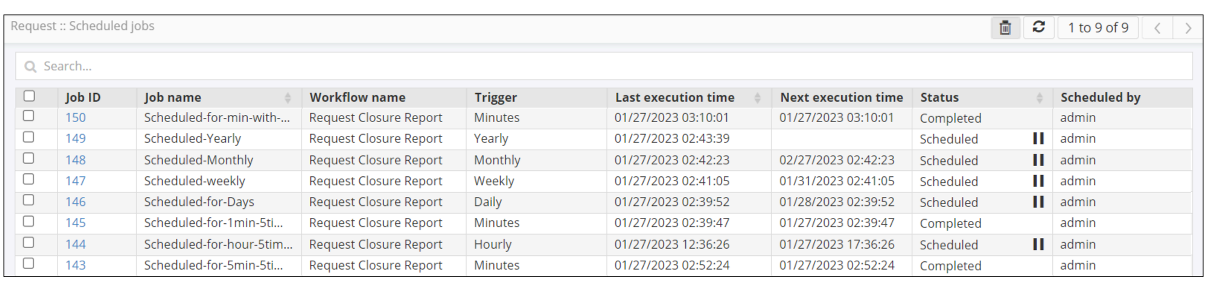


5. To add more workflows under a subcategory, select the subcategory and click » next to the workflow name under **Available Workflows**.
6. To move a workflow out of the selected workflows, click « next to the workflow under **Selected Workflows**.
7. Click **Save Changes**.

Scheduled Jobs

This section shows the inventory of scheduled workflows.

- Provision to ‘pause’ and ‘play’ a workflow.
- Displays details of execution time, trigger time and logs.
- Provision to ‘delete’ scheduled workflows.



Request :: Scheduled jobs

1 to 9 of 9

Search...

Job ID	Job name	Workflow name	Trigger	Last execution time	Next execution time	Status	Scheduled by
150	Scheduled-for-min-with-...	Request Closure Report	Minutes	01/27/2023 03:10:01	01/27/2023 03:10:01	Completed	admin
149	Scheduled-Yearly	Request Closure Report	Yearly	01/27/2023 02:43:39		Scheduled	admin
148	Scheduled-Monthly	Request Closure Report	Monthly	01/27/2023 02:42:23	02/27/2023 02:42:23	Scheduled	admin
147	Scheduled-weekly	Request Closure Report	Weekly	01/27/2023 02:41:05	01/31/2023 02:41:05	Scheduled	admin
146	Scheduled-for-Days	Request Closure Report	Daily	01/27/2023 02:39:52	01/28/2023 02:39:52	Scheduled	admin
145	Scheduled-for-1min-5ti...	Request Closure Report	Minutes	01/27/2023 02:39:47	01/27/2023 02:39:47	Completed	admin
144	Scheduled-for-hour-5tim...	Request Closure Report	Hourly	01/27/2023 12:36:26	01/27/2023 17:36:26	Scheduled	admin
143	Scheduled-for-5min-5ti...	Request Closure Report	Minutes	01/27/2023 02:52:24	01/27/2023 02:52:24	Completed	admin

1. On the Request :: Overview page, from the menu on the left, click **Scheduled Jobs**.
2. To pause a scheduled job, on the **Request :: Scheduled jobs** page, click **(Pause)** icon.
3. To resume a paused scheduled job, click **(Resume)** icon.
4. To delete scheduled jobs, select the **Job ID**.
5. Click **(Delete)** icon.
6. In the **Confirmation** pop-up window, click **Yes**.

   The selected scheduled jobs are deleted.

Request :: All

You can see a list of all (open, closed, failed) requests on the **Request :: All** page.

1. On the Request :: Overview page, from the navigation pane on the left, click **All**.

   The **Request :: All** page is displayed.
2. To see the stage-wise execution of a workflow, click on the **Request ID** for that workflow.
3. To clone a workflow request, select the workflow and click **(Clone)** icon from the command bar in the top right corner of the screen.
4. To refresh the **Request :: All** page, from the command bar, click **(Refresh)** icon.

Request :: Open

You can see a list of all the open requests on the **Request :: Open** page.

1. On the Request :: Overview page, from the navigation pane on the left, click **Open**.

   The **Request :: Open** page is displayed.
2. To see the stage-wise execution of a workflow, click on the **Request ID** for that workflow.
3. To clone a workflow request, select the workflow and click **(Clone)** icon from the command bar in the top right corner of the screen.
4. To refresh the **Request :: Open** page, from the command bar, click **(Refresh)** icon.

Request :: Closed

You can see a list of all the closed or completed requests on the **Request :: Closed** page.

1. On the Request :: Overview page, from the navigation pane on the left, click **Closed**.
   The **Request :: Closed** page is displayed.
2. To see the stage-wise execution of a workflow, click on the **Request ID** for that workflow.
3. To clone a workflow request, select the workflow and click **(Clone)** icon from the command bar in the top right corner of the screen.
4. To refresh the **Request :: Closed** page, from the command bar, click **(Refresh)** icon.

Request :: Failed

You can see a list of all the failed requests on the **Request :: Failed** page.

1. On the Request :: Overview page, from the navigation pane on the left, click **Failed**.
   The **Request :: Failed** page is displayed.
2. To see the stage-wise execution of a workflow, click on the **Request ID** for that workflow.
3. To clone a workflow request, select the workflow and click **(Clone)** icon from the command bar in the top right corner of the screen.
4. To refresh the **Request :: Failed** page, from the command bar, click **(Refresh)** icon.

Request :: Assigned

To see a list of all requests assigned to the logged in user:

1. On the Request :: Overview page, from the navigation pane on the left, click **Assigned**.
   The **Request :: Assigned** page is displayed.
2. To see the stage-wise execution of a workflow, click on the **Request ID** for that workflow.
3. To clone a workflow request, select the workflow and click **(Clone)** icon from the command bar in the top right corner of the screen.
4. To refresh the **Request :: Assigned** page, from the command bar, click **(Refresh)** icon.

Request :: Audit Logs

You can view the audit logs of all scheduled workflows and report-based workflows that were triggered on the **Request :: Audit Logs** page. Navigate to Request :: Overview > **Audit logs**.

Request :: Audit logs | 1 to 25 of 28

Search...

	Time	Message	User	Reference ID	Request type	Workflow Name
▸	01/27/2023 07:37:00	Request triggered for scheduled j...	system	41	Scheduled Job	Request Closure Report
▸	01/27/2023 05:30:11	Temporary request completed. R...	system	temp_3	Request(Temporary)	License_check
▸	01/27/2023 05:30:04	Temporary request triggered. Re...	system	temp_3	Request(Temporary)	License_check
▸	01/27/2023 03:17:06	Temporary request completed. R...	system	temp_2	Request(Temporary)	License_check
▸	01/27/2023 03:17:01	Request triggered for scheduled j...	system	temp_2	Scheduled Job	License_check
▸	01/27/2023 03:11:00	Request triggered for scheduled j...	system	40	Scheduled Job	Request Closure Report
▸	01/27/2023 03:06:01	Request triggered for scheduled j...	system	37	Scheduled Job	Request Closure Report
▸	01/27/2023 03:01:01	Request triggered for scheduled j...	system	36	Scheduled Job	Request Closure Report
▸	01/27/2023 02:56:01	Request triggered for scheduled j...	system	35	Scheduled Job	Request Closure Report
▸	01/27/2023 02:53:01	Request triggered for scheduled j...	system	34	Scheduled Job	Request Closure Report
▸	01/27/2023 02:51:01	Request triggered for scheduled j...	system	33	Scheduled Job	Request Closure Report
▸	01/27/2023 02:48:01	Request triggered for scheduled j...	system	29	Scheduled Job	Request Closure Report
▸	01/27/2023 02:46:01	Request triggered for scheduled j...	system	25	Scheduled Job	Request Closure Report
▸	01/27/2023 02:44:01	Request triggered for scheduled j...	system	23	Scheduled Job	Request Closure Report
▸	01/27/2023 02:43:00	Request triggered for scheduled j...	system	22	Scheduled Job	Request Closure Report
▸	01/27/2023 02:43:00	Request triggered for scheduled j...	system	21	Scheduled Job	Request Closure Report
▸	01/27/2023 02:42:01	Request triggered for scheduled j...	system	19	Scheduled Job	Request Closure Report
▸	01/27/2023 02:40:01	Request triggered for scheduled j...	system	16	Scheduled Job	Request Closure Report
▸	01/27/2023 02:40:01	Request triggered for scheduled j...	system	17	Scheduled Job	Request Closure Report
▸	01/27/2023 02:39:01	Request triggered for scheduled j...	system	15	Scheduled Job	Request Closure Report
▸	01/27/2023 02:38:01	Request triggered for scheduled j...	system	14	Scheduled Job	Request Closure Report
▸	01/27/2023 02:38:01	Request triggered for scheduled j...	system	13	Scheduled Job	Request Closure Report
▸	01/27/2023 02:37:01	Request triggered for scheduled j...	system	12	Scheduled Job	Request Closure Report
▸	01/27/2023 02:37:01	Request triggered for scheduled j...	system	11	Scheduled Job	Request Closure Report
▸	01/27/2023 02:36:01	Request triggered for scheduled j...	system	9	Scheduled Job	Request Closure Report

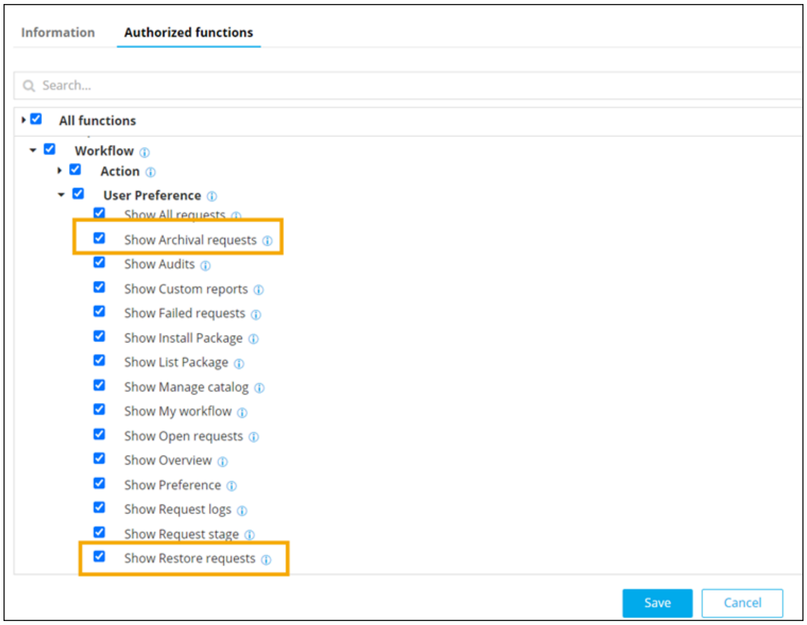
Configuring RBAC for Archive and Restore Requests

You can allow users to access the **Archive & Restore** section or restrict them from viewing this section on the Workflow Request page.

1. Go to the **Role** page in the Platform module.
2. Select the role for which you want to enable access. For example, **admin**.
3. Under **Authorized functions**, expand **Automation** > **Service Requests** > **User Preference**.

> **Note:** If you are using the old menu, under the **Authorized Functions** tab, expand **Request** > **Workflow**.

4. Under **User Preference**, select from the following checkboxes for the option to be displayed in the Service Requests console.
   - **Show Archival requests**
   - **Show Restore requests**.



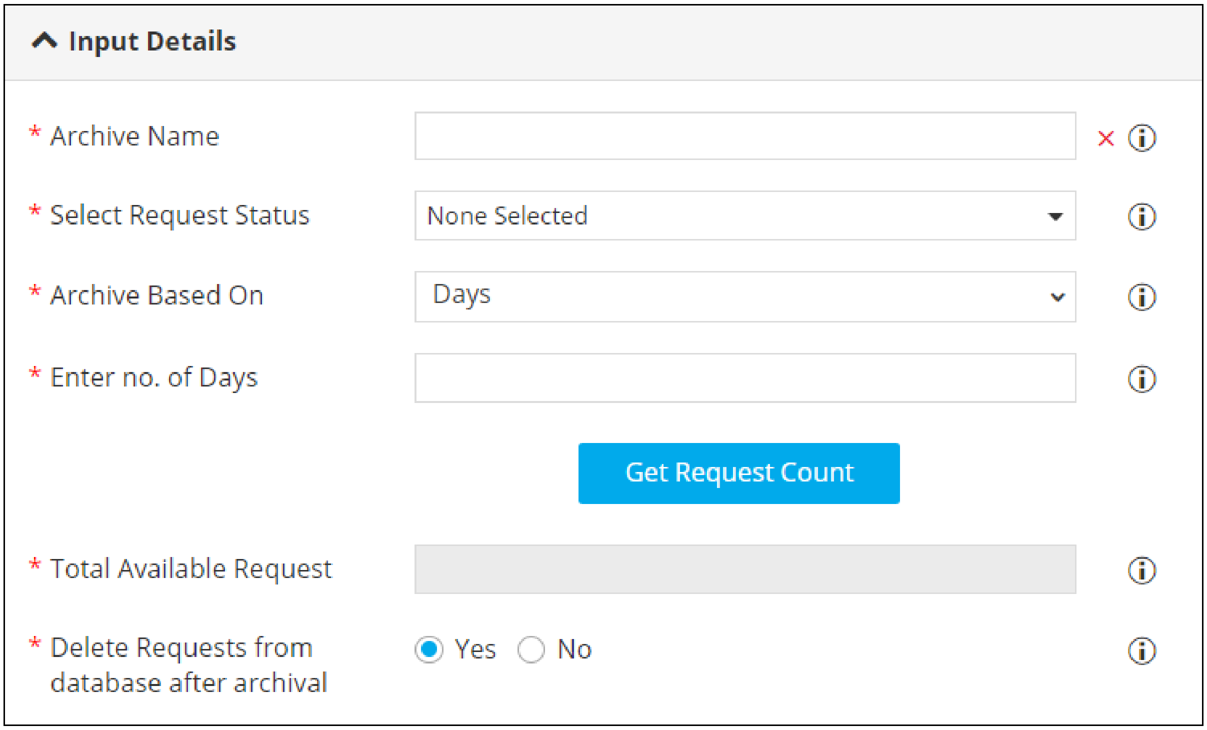
Note: Only the option that is selected under **User Preference** will be displayed under **Service Requests**.

Archive Requests

You can archive workflow service requests by status, time period and so on. You can also schedule archivals and also purge service requests.

To trigger an archive workflow request:

1. On the Request :: Overview page, from the navigation pane on the left, click **Archive Requests**. The **Archive Workflow Requests** workflow is triggered with the workflow inputs requested at the first stage.
2. In the **Input Details** section, enter or select the required field information.



Field descriptions for Input Details

Field	Description
***Archive Name**	Enter the name of the archive.
***Select Requests Status**	Select one or more type(s) of requests to be archived from the options available in the dropdown.
***Archive Based On**	Select the duration for which the requests will be archived as: • **Days**: This option allows you to enter the number of days for archiving service requests. • **Between Time**: This option allows you to select the Start and End date for archiving service requests.
***Enter no. of Days**	Enter the number of days for which you want to archive the service requests. **Note:** This field is displayed when you select **Days** in the **Archive Based On** field.
***Select Start Date**	Select the date and time from when the service requests are to be archived.

Field	Description
	Note: This field is displayed when you select **Between Time** in the **Archive Based On** field.
***Select End Date**	Select the date and time till which the service requests are to be archived. **Note:** This field is displayed when you select **Between Time** in the **Archive Based On** field.
Note: The asterisk (*) symbol indicates mandatory fields.	

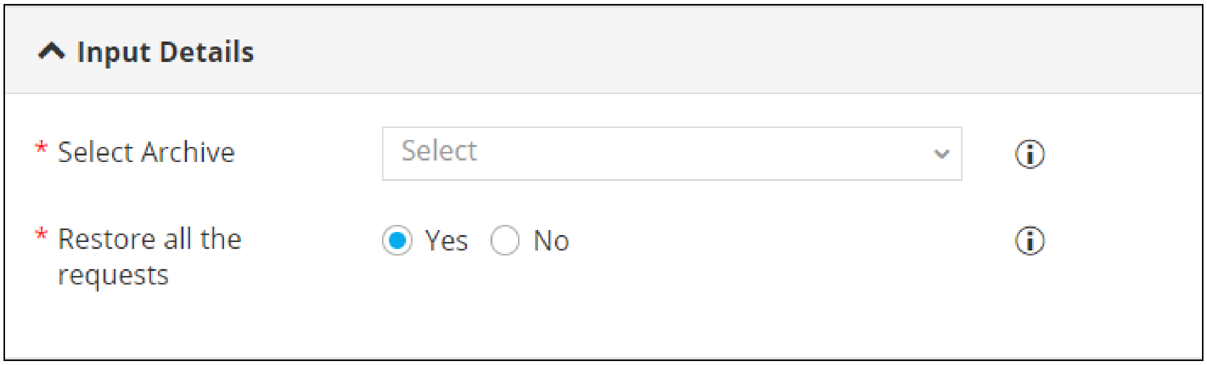
3. Click **Get Request Count**.
   The total number of requests that match the selection are displayed in the **Total Available Request** field.
4. To delete the requests from the database after archival, select **Yes**.
5. Click **Submit**.
   The service requests will be archived based on the selection.

Restore Archives

You can restore workflow service requests from the Workflow Request page as per your requirement. The requests can be restored either in bulk or by selecting individual workflow requests.

To trigger a restore workflow request:

1. On the Request :: Overview page, from the navigation pane on the left, click **Restore Archives**.
   The **Restore Archived Requests** workflow is triggered with the workflow inputs requested at the first stage.

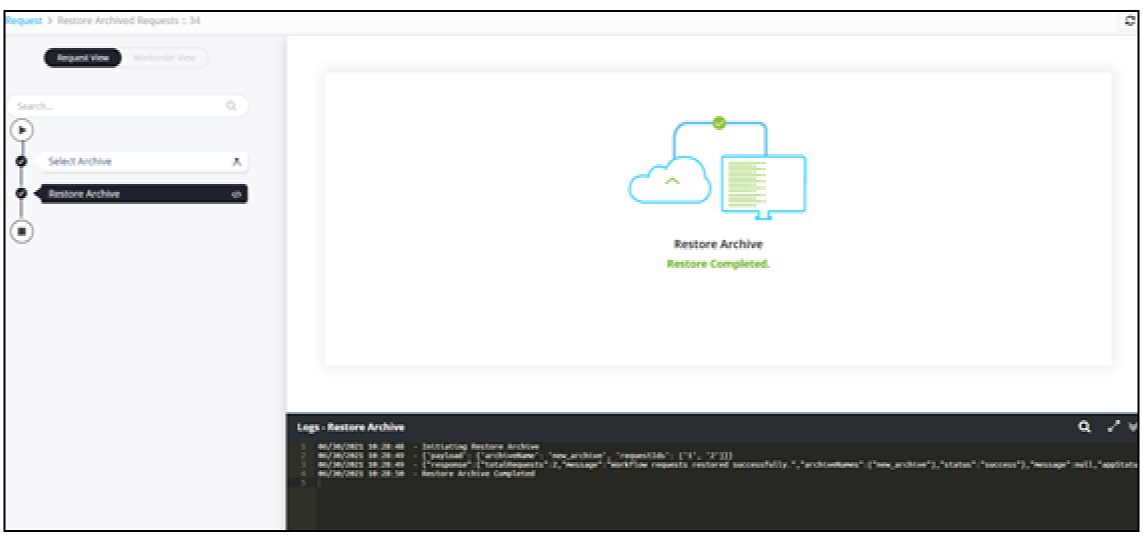


2. Under **Input Details**, select the archive that is to be restored.
3. To restore all the requests from the archive, select **Yes**.
4. To select specific request(s) from the archive, select **No**.
5. To auto populate the list of workflow requests, click **Get Request Details**.

> **Note:** This button is displayed only when you select **No** in the **Restore all the requests** field.

6. Select the workflows that you want to restore.
7. To restore the selected workflow request(s) from the archive, click **Submit** .
8. In the **Confirmation** window, click **Ok**.

   The workflow to Restore Archive is completed.



Preferences

This section describes the following settings that can be configured as per your requirement in the Service Requests console:

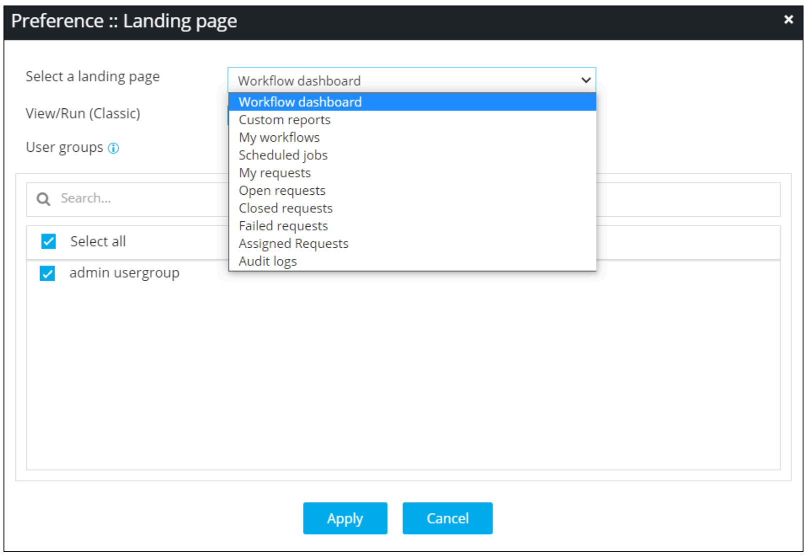
- Selecting the Landing Page
- Viewing the View/Run page in Classic mode
- Managing the Catalogs

Selecting the Landing Page

You can select the page that is displayed when you access the Service Requests console.

1. On the Request :: Overview page, from the navigation pane on the left, click **Preference**.

   The **Preference :: Landing page** window is displayed.
2. In the **Preference :: Landing page** window, **Select a Landing page** from the list of options.



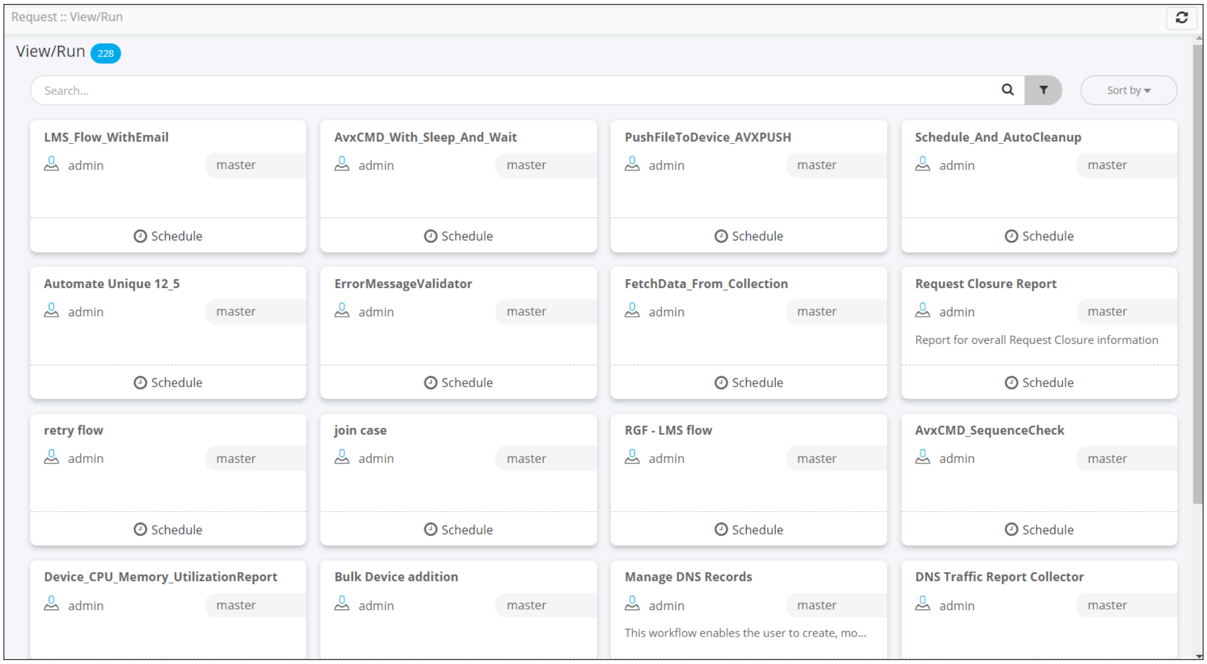
3. Under **User groups**, select the user groups for which the selected page will be set as the landing page.
4. Click **Apply**.

Viewing the View/Run page in Classic mode

1. On the Request :: Overview page, from the navigation pane on the left, click **Preference**.

   The **Preference :: Landing page** window is displayed.
2. To display the **View/Run** page in **Classic** mode, in the **Preference :: Landing page** window, turn on the **View/Run (Classic)** toggle.
3. Click **Apply**.

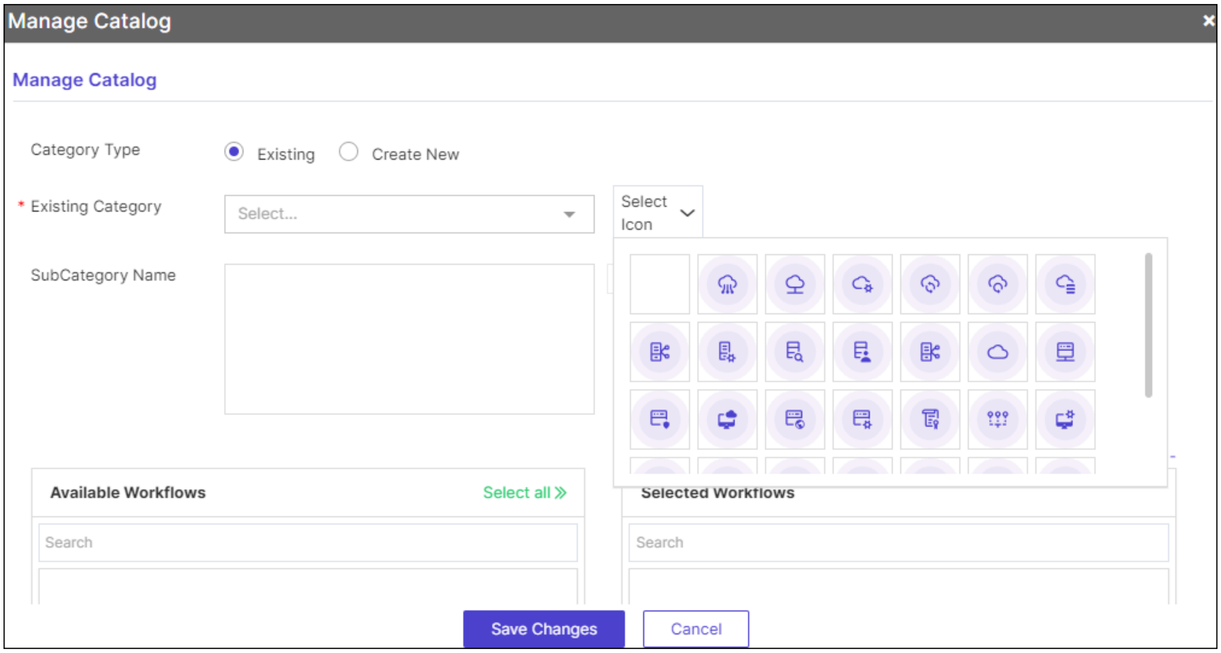
The **View/Run** page is displayed in Classic mode.



Managing the Catalogs

You can edit the catalog properties only when the View/Run page is not in Classic mode. This feature allows you to create/modify catalog categories, select icons for catalogs etc.

1. On the Request :: Overview page, from the top right corner of the page, click **Manage Catalog**.
   The **Manage Catalog** window is displayed.



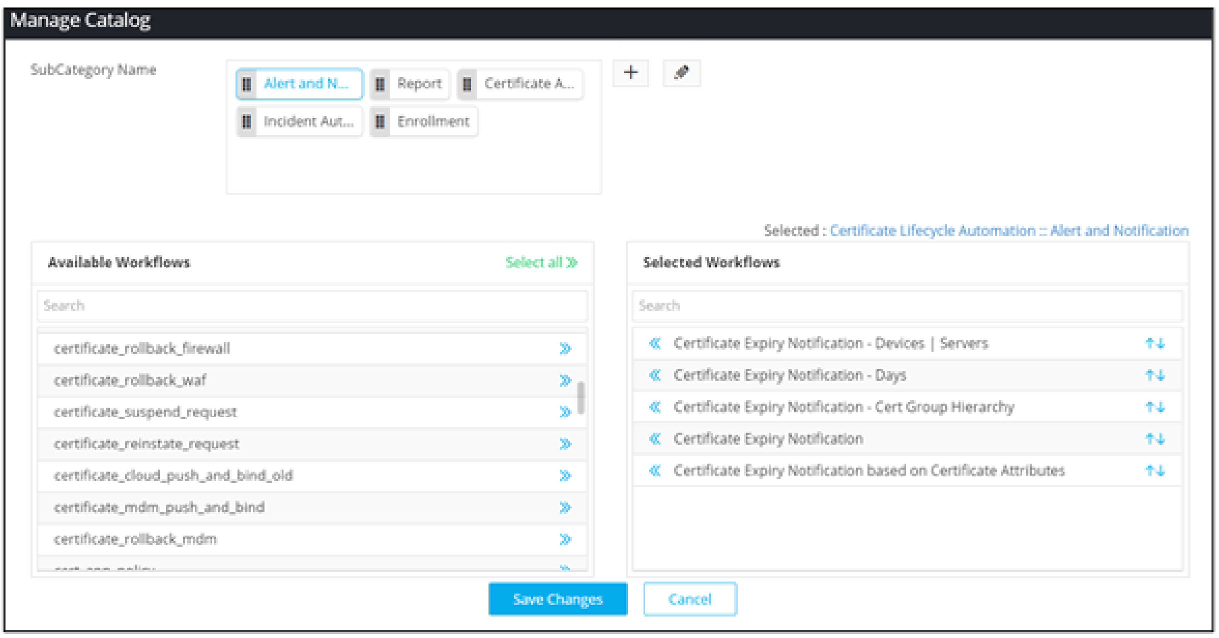
2. To modify the existing workflow categories, under **Category Type**, select the **Existing** option.
3. To create a new category, under **Category Type**, select the **Create New** option.

Options and Descriptions for Manage Catalogs

Option	Description
Category Type	Select the **Category Type** as: • **Existing**: Selecting this option allows you to select the category of workflows from the options available in the Existing Category dropdown. • **Create New**: Selecting this option allows you to create a new category of workflows.
***Existing Category**	Select the category of workflows from the options displayed in the dropdown. **Note:** If the **Create New** option is selected in the **Category Type** field, then this field becomes a text box and allows you to type a new category name.
Select Icon	Select an icon for the workflow card from the options available in the dropdown.
SubCategory Name	This field is auto-populated when you select the Existing Category option in the Category Type field.

Option	Description
+	Allows you to add more subcategories of workflows.
Note: The asterisk (*) symbol indicates mandatory fields.	

4. To add more workflows to a subcategory, select the subcategory in the **SubCategory Name** field and click » next to the workflows to be added.
5. To move a workflow out of the selected workflows, click « next to the workflow under **Selected Workflows**.



6. To modify the order in which the workflows are displayed within the subcategory, click ↑↓.
7. Click **Save Changes**.

Service Request Actions

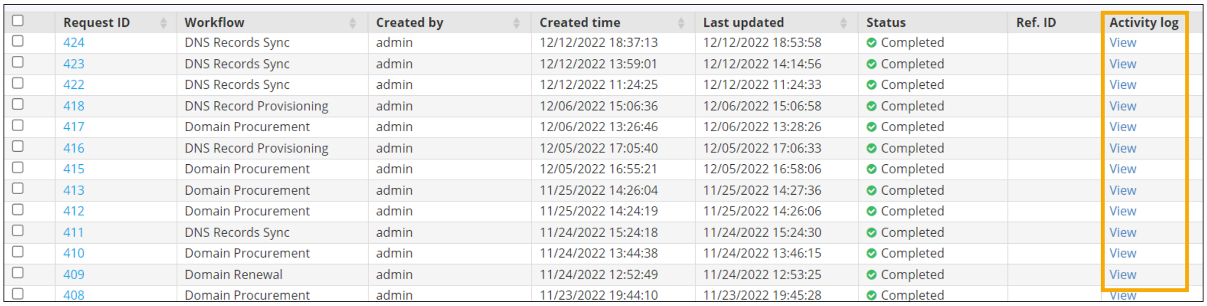
- Viewing the Workflow Log Summary
- Pausing/Resuming a Workflow

- Aborting a Workflow
- Rollback a Workflow

Viewing the Workflow Log Summary

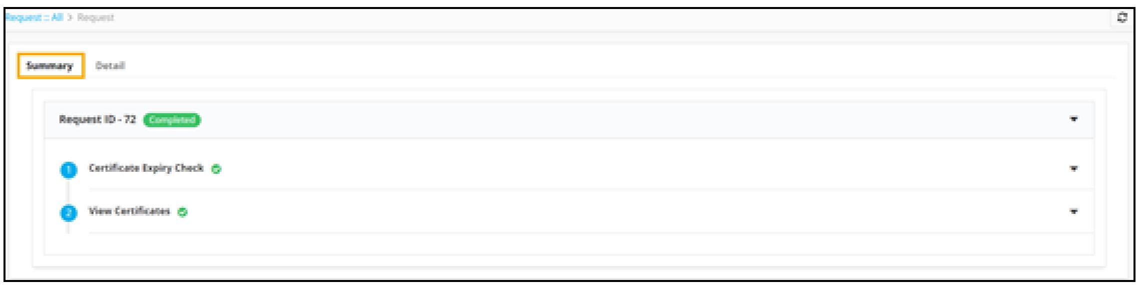
To view the activity log of a specific workflow:

1. On the Request :: All page, under the **Activity log** column, click **View** next to the workflow for which you want the summary.



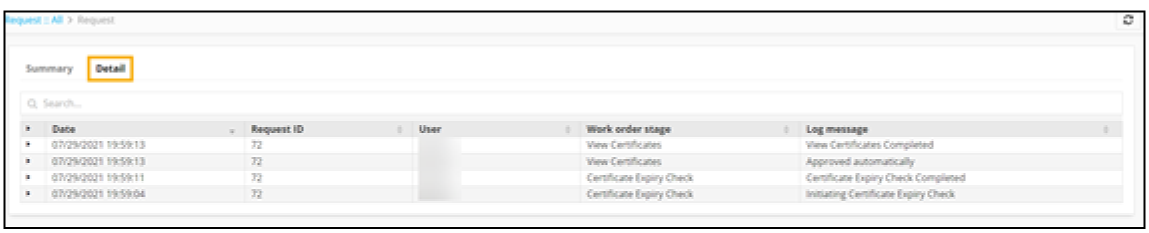
	Request ID	Workflow	Created by	Created time	Last updated	Status	Ref. ID	Activity log
☐	424	DNS Records Sync	admin	12/12/2022 18:37:13	12/12/2022 18:53:58	Completed		View
☐	423	DNS Records Sync	admin	12/12/2022 13:59:01	12/12/2022 14:14:56	Completed		View
☐	422	DNS Records Sync	admin	12/12/2022 11:24:25	12/12/2022 11:24:33	Completed		View
☐	418	DNS Record Provisioning	admin	12/06/2022 15:06:36	12/06/2022 15:06:58	Completed		View
☐	417	Domain Procurement	admin	12/06/2022 13:26:46	12/06/2022 13:28:26	Completed		View
☐	416	DNS Record Provisioning	admin	12/05/2022 17:05:40	12/05/2022 17:06:33	Completed		View
☐	415	Domain Procurement	admin	12/05/2022 16:55:21	12/05/2022 16:58:06	Completed		View
☐	413	Domain Procurement	admin	11/25/2022 14:26:04	11/25/2022 14:27:36	Completed		View
☐	412	Domain Procurement	admin	11/25/2022 14:24:19	11/25/2022 14:26:06	Completed		View
☐	411	DNS Records Sync	admin	11/24/2022 15:24:18	11/24/2022 15:24:30	Completed		View
☐	410	Domain Procurement	admin	11/24/2022 13:44:38	11/24/2022 13:46:15	Completed		View
☐	409	Domain Renewal	admin	11/24/2022 12:52:49	11/24/2022 12:53:25	Completed		View
☐	408	Domain Procurement	admin	11/23/2022 19:44:10	11/23/2022 19:45:28	Completed		View

A **Summary** of the workflow stages is displayed.



2. To view the workflow log requests details, click **Detail**.

A detailed log of the workflow stages is displayed.

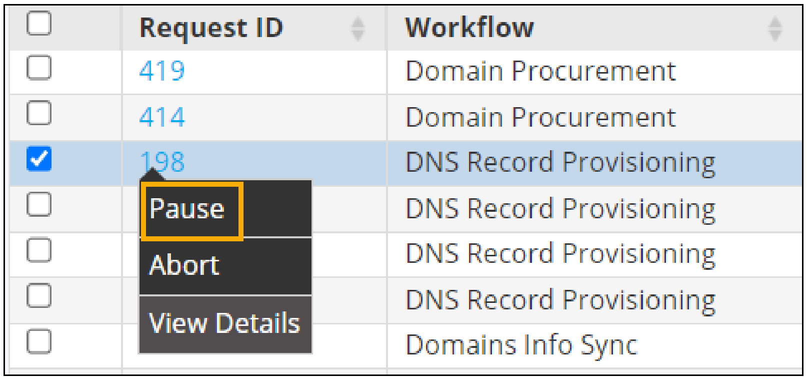


Date	Request ID	User	Work order stage	Log message
07/29/2021 19:59:13	72		View Certificates	View Certificates Completed
07/29/2021 19:59:13	72		View Certificates	Approved automatically
07/29/2021 19:59:11	72		Certificate Expiry Check	Certificate Expiry Check Completed
07/29/2021 19:59:04	72		Certificate Expiry Check	Initiating Certificate Expiry Check

Pausing/Resuming a Workflow

You can pause and resume a workflow request which is in the **In Progress** status.

1. On the Request :: All page, right-click on the **Request ID** of the workflow that you want to pause.
2. From the options displayed, select **Pause**.



3. Click **Yes** in the pop-up window.

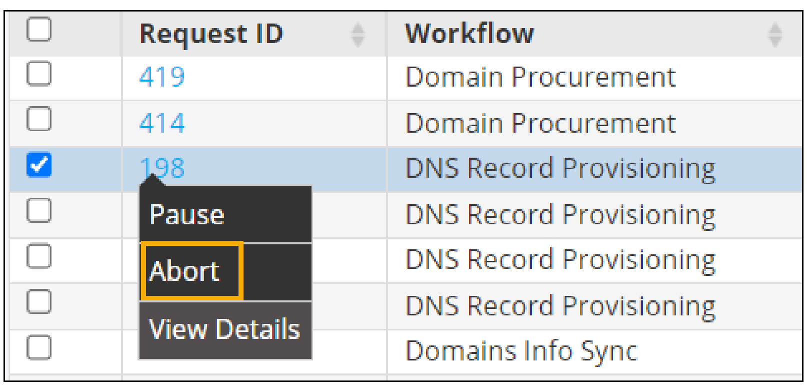
   The workflow is paused.
4. To resume the workflow, right-click and select **Resume**.
5. Click **Yes** in the confirmation pop-up window.

   The workflow resumes.

Aborting a Workflow

You can only abort workflows that are still open or in-progress.

1. On the Request :: All page, right-click on the **Request ID** of the workflow that you want to abort.
2. Select **Abort** from the available options.



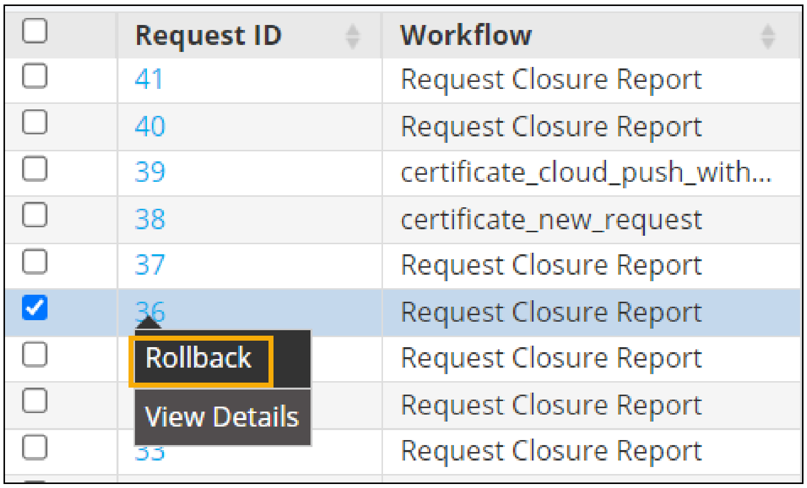
3. Click **Yes** in the confirmation pop-up window.

> **Note:** You can abort a workflow only if it is in the **In progress** state.

Rollback a Workflow

You can manually rollback a workflow request that has been completed (either successfully or failed).

1. On the Request :: All page, right-click on the **Request ID** of the workflow that you want to rollback.
2. From the options displayed, select **Rollback**.



3. Click **Yes** in the **Rollback** pop-up window.

> **Tip:** You can also rollback a workflow, by selecting the workflow and clicking in the upper right corner of the screen.

Automation and Service Catalog

Once you have designed your workflows, you can view them all in one place through AppViewX's self-service catalog. To access the AppViewX Service Catalog, navigate to **Automation** > **SERVICE CATALOG**.

- Provision to enable self-service automation workflows by persona
- Provision to logically select and group workflows such as CLM Automation, Application Delivery and so on
- Provision to drag and drop workflow(s) to design custom self-service catalogs
- Provision to view workflows through a single pane of glass
- Provision to add automation, self service request alerts to a customized page
- Provision to view workflow request status (open, closed, failed) from the self-service catalog
- Provision to run/trigger and schedule workflows from self-service catalog pages.

Note: For more information, refer to the Service Catalog User Guide.

Chapter 2: Automation API Guide

AppViewX's Automation module is a hybrid cloud automation and orchestration platform that allows for simple and complex network automation within your infrastructure. The module supports a wide range of device types, vendors, actions and integrations, in order to help you manage and automate your entire network with a unified automation and orchestration platform. Its primary interface is through both the Web GUI interface and a HTTP Restful API. A developer would use this API for programmatic access.

- Login to retrieve active SessionId
- Submit Request
- Update Request
- Get Current Task Detail of the Request
- Get Request Summary/Log
- Get Associated Workorder
- Get Tasks which are in InProgress State
- Upload File
- Download File
- Delete File

Login to retrieve active SessionId

This API retrieves the sessionId. This sessionId retrieved can be used for another API that ensures a valid session.

URL: /acctmgmt-perform-login

Type: POST

Parameters

Parameter Details

Param Type	Name	Description	Field Type	Constraints
Header	username	AppViewX login username	String	NA
Header	password	AppViewX login password	String	NA

Parameter Details (continued)

Param Type	Name	Description	Field Type	Constraints
Header	Content-Type	Specifies the nature of the data in the payload	String	The value of the param should be application/json
Query	gwkey	Tenant Key. Needed only in the case of multi-tenant installations.	String	NA
Query	gwsource	Source from which the request is triggered (For example: web, external)	String	NA
Body	payload	Contains an empty payload.	String	NA

Response Structure

200 OK returns string of type application/json with the following body params:

Name	Description	Field Type
response	Contains the response params for the search object request.	response
message	Success message or failure description in case of error.	String
appStatusCode	Application specific status code for the response. Will be non-null for failure response.	String
tags	More info in case of failure response.	NA

Response

Name	Description	Field Type
status	Status of the request	String
appStatusCode	appStatusCode	String
statusDescription	Description of the status	String
sessionId	SessionId to be retrieved	String

Name	Description	Field Type
availableLoginAttemptCount	Count of login attempts	String

Status Codes

HTTP Status Code	appStatusCode	Message	Possible remediation
200 OK	NA	Login successful	NA
400 Bad Request	NA	Username or password cannot be null or empty	Check and ensure if a non-null/non-empty value is given in the header field for username or password
401 Unauthorized	NA	Login failed. Invalid credentials.	Check and ensure if the username and password provided are valid and correct

Sample Request/Response

This API retrieves the sessionId that can be used for another API that ensures a valid session.

Use Case

Submit a request for the sessionId retrieval.

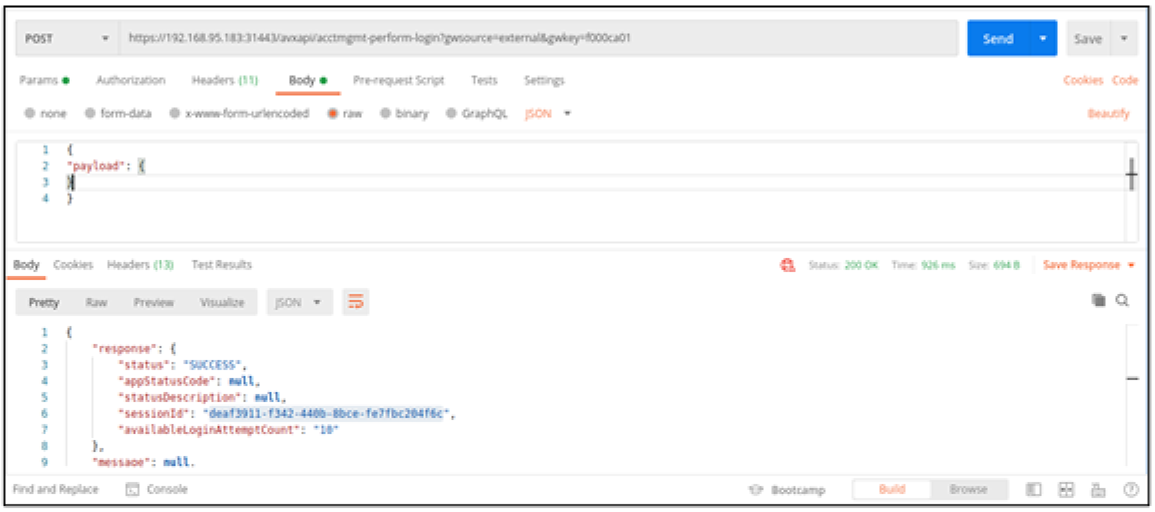
Sample Request


```
{
 "payload": {
   }
}
```


Sample Response


```
{
   "response": {
      "status": "SUCCESS",
      "appStatusCode": null,
```

```
        "statusDescription": null,
        "sessionId": "dea911-f342-440b-8bce-fe7fbc204g6c",
        "availableLoginAttemptCount": "10"
    },
    "message": null,
    "appStatusCode": null,
    "tags": null,
    "headers": null
}
```

Submit Request

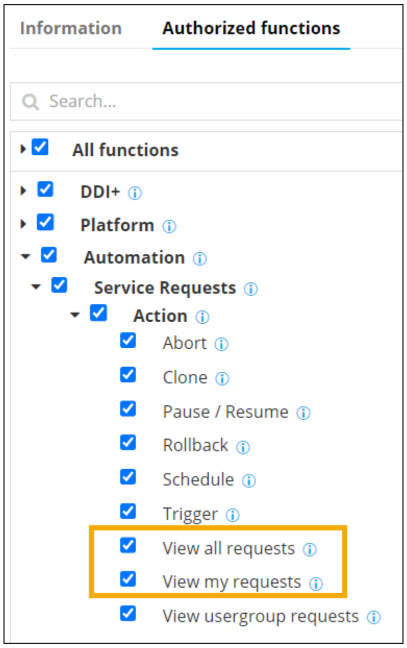
Creates a new workflow request for the given workflow name. A request id is generated and associated with each submitted workflow request which can be used later for retrieving any information about the request.

Before you begin

Before attempting to submit a request with this API, you must ensure that the following permissions are available:

- **Service Requests** > **Action** > **View all requests**
- **Service Requests** > **Action** > **View my requests**

You can check this in the **Platform** module under **Authorized functions**.



Request Structure

URL: /visualworkflow-submit-request

Type: POST

Parameters

Parameter Details

Param Type	Name	Description	Field Type	Constraints
Header	`sessionId`	sessionId received after login.	`String`	Required if username and password are not provided.
Header	`username`	AppViewX login username	`String`	Required if sessionId is not provided.
Header	`password`	AppViewX login password	`String`	Required if sessionId is not provided.
Header	`Content-Type`	Specifies the nature of the data in the payload.	`String`	Value of the param should be `application/json`
Query	`gwkey`	Tenant Key.	`String`	NA

Parameter Details (continued)

Param Type	Name	Description	Field Type	Constraints
		Needed only in the case of multi-tenant installations.		
Query	gwsource	Source from which the request is triggered (For example: web, external)	String	NA
Body	payload	Contains all the params to be sent in the request body for the post request.	Any one of the following: Service Task Payload User Task Payload	NA

Task Specific Information

Service Task Payload

Service tasks will not have any user input for submitting the request.

Name	Mandatory	Description	Field Type	Constraints
Header	Yes	Contains header param details of the request	Header Payload	NA

Header Payload

Name	Mandatory	Description	Field Type	Constraints
workflowName	Yes	The workflow for which the request is submitted.	String	NA

Sample Request/Response

Use Case

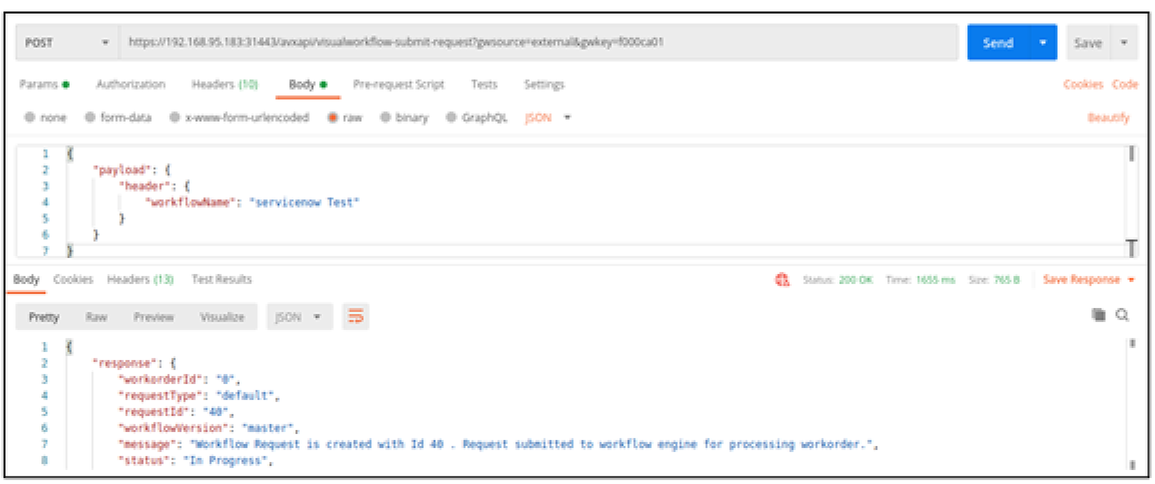
Submit a request for the workflow **servicenow Test**.

Sample Request

```
{
  "payload": {
    "header": {
      "workflowName": "servicenow Test"
    }
  }
}
```

Sample Response

```
{
  "response": {
    "workorderId": "0",
    "message": "Workflow Request is created with Id 401 . Request submitted to workflow engine for processing workorder.",
    "requestType": "default",
    "requestId": "401",
    "workflowVersion": "master",
    "status": "In Progress",
    "statusCode": 0
  },
  "message": null,
  "appStatusCode": null,
  "tags": null
}
```



User Task Payload

Name	Mandatory	Description	Field Type	Constraints
header	Yes	Contains header param details of the request	Header Payload	NA
data	Yes	Contains params for the user task	Data Payload	NA

Data Payload

Name	Mandatory	Description	Field Type	Constraints
input	Yes	The input data for the user task.	Input Payload	NA
task_action	Yes	Action to be performed on the task.	Integer	1 for success case and 2 for reject case

Input Payload

Name	Mandatory	Description	Field Type	Constraints
requestData	Yes	The request data for which the request is submitted.	List of Request Data Payload	NA

Header Payload

Name	Mandatory	Description	Field Type	Constraints
workflowName	Yes	The workflow for which the request is submitted.	String	NA

Request Data Payload

Name	Mandatory	Description	Field Type	Constraints
scenario	Yes	The scenario name for the form,	String	NA
sequenceNo	Yes	The sequence number of the scenario,	Integer	NA
fieldInfo	Yes	The field info configured in the form. The value should be key value, field id and its value.	Json	NA

Sample Request/Response - Form Task

Submit a request for the workflow with the first task as **Form**.

Sample Request

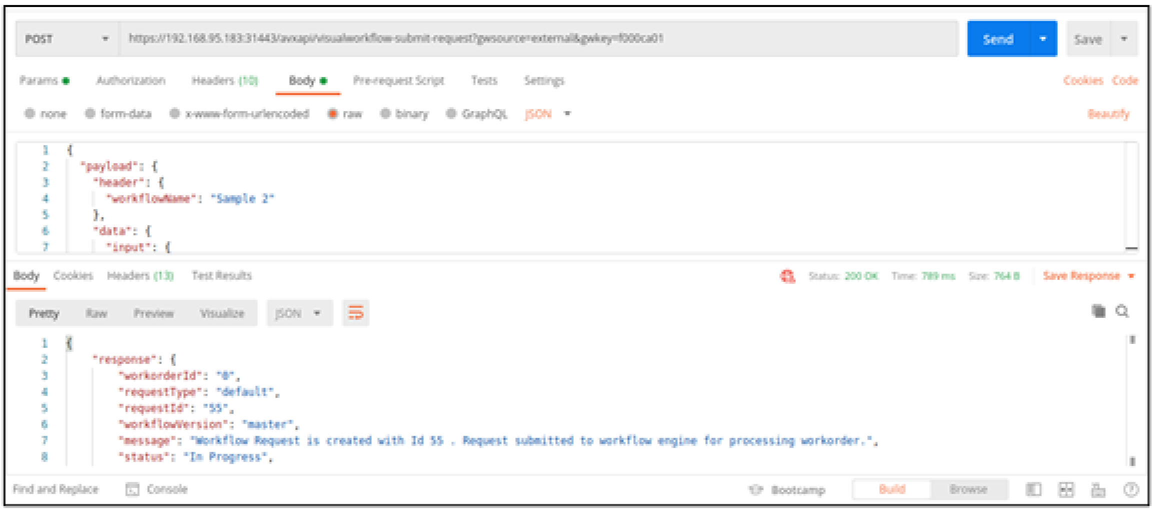

```
{
  "payload": {
    "header": {
      "workflowName": "Sample 2"
    },
    "data": {
      "input": {
        "requestData": [
          {
            "scenario": "scenario",
            "sequenceNo": 1,
            "fieldInfo": {
              "deviceName": "ADC"
            }
          },
          {
            "scenario": "scenario2",
            "sequenceNo": 2,
            "fieldInfo": {
              "deviceName": "A10"
            }
          }
        ]
      },
      "task_action": 1
    }
  }
}
```


Sample Response


```
{
  "response": {
    "workorderId": "0",
    "message": "Workflow Request is created with Id 55 . Request submitted to workflow engine for processing workorder.",
    "requestType": "default",
    "requestId": "401",
```



```
    "workflowVersion": "master",
   "status": "In Progress",
   "statusCode": 0
  },
  "message": null,
  "appStatusCode": null,
  "tags": null
}
```



Sample Request/Response - Grid Task

Submit a request for the workflow with the first task as **Grid**.

Sample Request

```
{
"payload": {
   "data": {
     "input": {
       "columnLabels": {
         "IP": "IP",
         "Device Name": "Device Name"
       },
       "gridData": [
         {
           "IP": "","Device Name": ""
         }
```

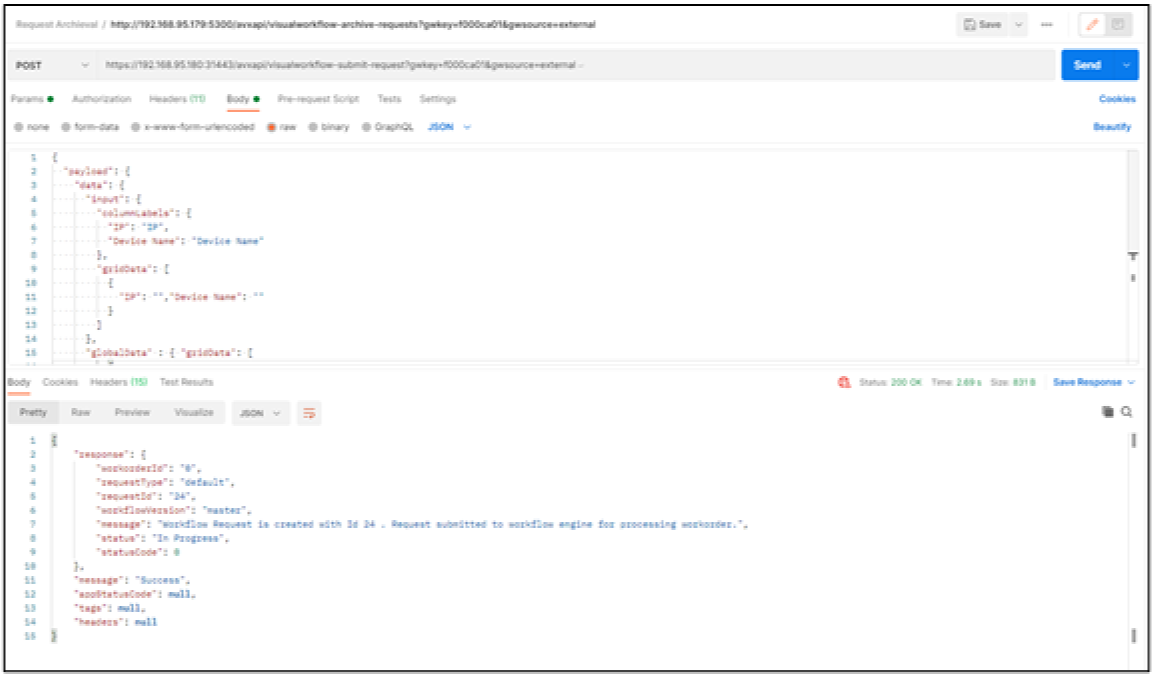


```
      ]
    },
    "globalData" : { "gridData": [
      {
        "IP": "192.168.15.35",
        "Device Name": "Device F5"
      },
      {
        "IP": "192.168.15.40",
        "Device Name": "Device AVI"
      }
     ]},
    "task_action": 1
   },
   "header": {
    "workflowName": "Grid"
   }
  }
}
```


Note: Global Variable `<%gridData%>` should be present in the “values” section in Grid Task.

Sample Response


```
{
  "response": {
    "workorderId": "0",
    "message": "Workflow Request is created with Id 24 . Request submitted to workflow engine for processing workorder.",
    "requestType": "default",
    "requestId": "401",
     "workflowVersion": "master",
    "status": "In Progress",
    "statusCode": 0
  },
  "message": null,
  "appStatusCode": null,
  "tags": null
}
```

Sample Request/Response - Script Task

Submit a request for the workflow with the first task as **Script**.

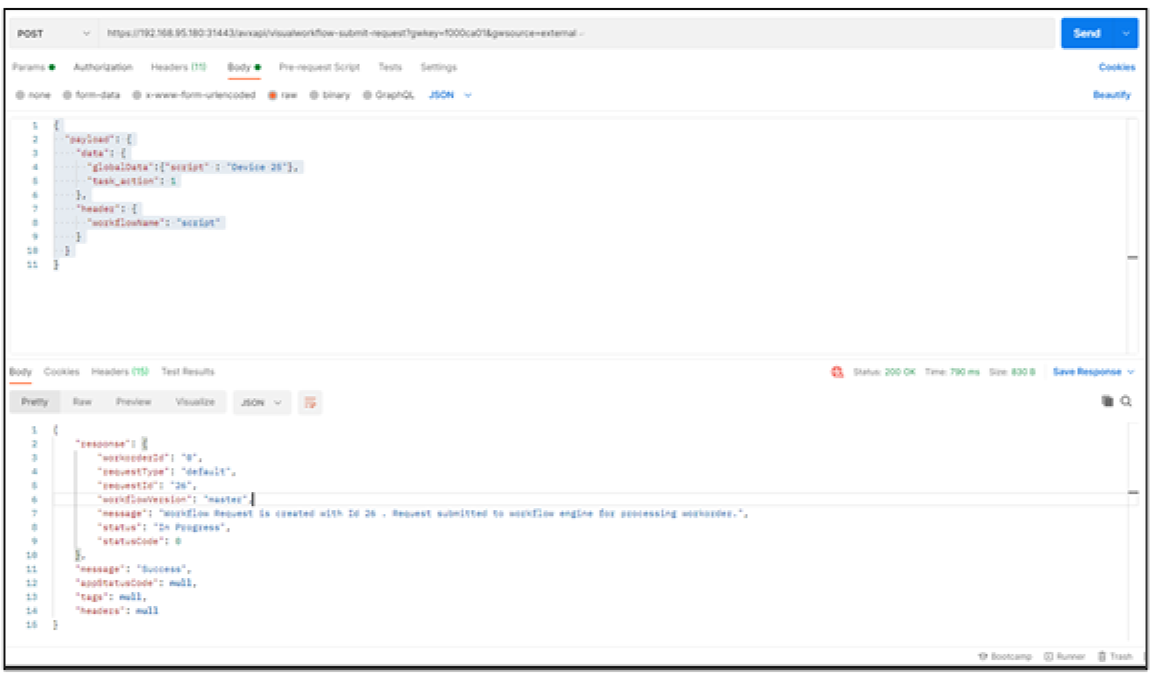
Sample Request

```
{
  "payload": {
    "data": {
      "globalData":{"script" : "Device 25"},
      "task_action": 1
    },
    "header": {
      "workflowName": "script"
    }
  }
}
```

Note: Values can be passed to available Global variables present inside the Script task.

Sample Response


```
{
  "response": {
    "workorderId": "0",
    "message": "Workflow Request is created with Id 26. Request submitted to workflow engine for processing workorder.",
    "requestType": "default",
    "requestId": "401",
     "workflowVersion": "master",
    "status": "In Progress",
    "statusCode": 0
  },
  "message": null,
  "appStatusCode": null,
  "tags": null
}
```



Sample Request/Response - YAML Task

Submit a request for the workflow with the first task as **YAML**.

Sample Request

```
{
  "payload": {
    "data": {
```

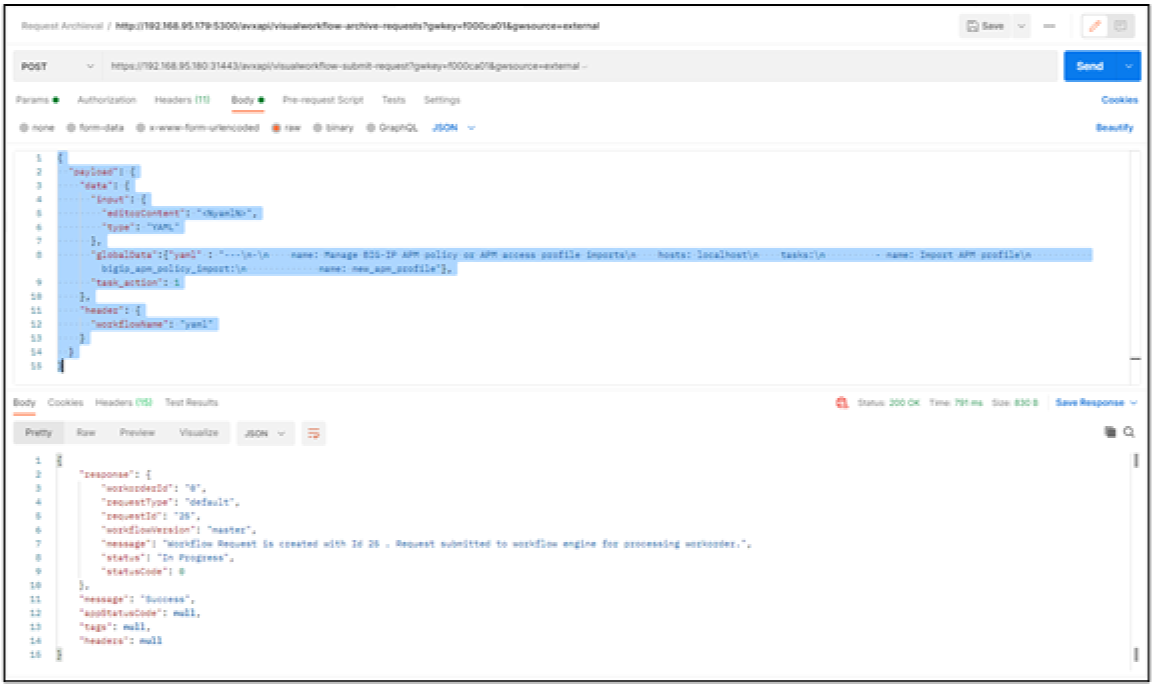


```
    "input": {
      "editorContent": "<%yaml%>",
      "type": "YAML"
    },
    "globalData":{"yaml" : "---\n-\n    name: Manage BIG-IP APM policy or APM access profile imports\n    hosts: localhost\n    tasks:\n        - name: Import APM profile\n          bigip_apm_policy_import:\n            name: new_apm_profile"},
    "task_action": 1
  },
  "header": {
    "workflowName": "yaml"
  }
 }
}
```


Note: Global Variable `<%yaml%>` should be present in the YAML task.

Sample Response


```
{
 "response": {
   "workorderId": "0",
   "message": "Workflow Request is created with Id 26 . Request submitted to workflow engine for processing workorder.",
   "requestType": "default",
   "requestId": "401",
    "workflowVersion": "master",
   "status": "In Progress",
   "statusCode": 0
 },
 "message": null,
 "appStatusCode": null,
 "tags": null
}
```

Response Structure

200 OK, returns string of type application/json with the following body params:

Name	Description	Field Type
response	Contains the response params for the search object request.	response
message	Success message or failure description in case of error.	String
appStatusCode	Application specific status code for the response, will be non-null for failure response.	String
tags	More info in case of failure response.	NA

Response

Name	Description	Field Type
requestID	Unique identifier for the request	String
workflowVersion	Version of the workflow	String
message	The message with the status and request number for the request	String
status	Status of the request	String

Name	Description	Field Type
statusCode	Status code of the request	Integer
requestType	The type of the request	String
workorderId	The workoderId for the request If there is no workorder the value will be 0.	String

Status Codes

HTTP Status Code	appStatusCode	Message	Possible remediation
200 OK	NA	Success	NA
400 Bad request	avx-common-028	Invalid/Incorrect payload	Check and ensure if a valid value is given in request payload field - input.
401 Unauthorized	WORKFLOW_1679	User is not authorized.	LoggedIn user is not authorized to access the workflow.
404 Not Found	engine-db-015	Workflow not found	Invalid workflow data
409 Conflict	WRKFLOW_1475	Given workflow is not in enabled state.	Given workflow is not in enabled state.
500 Internal Server Error	WORKFLOW_1617	Invalid form data. Please provide all mandatory data.	Check and ensure if a valid value is given in request payload field - input.
500 Internal Server Error	NA	Error while processing	• Task is already completed. • Task is being processed by another user.

Update Request

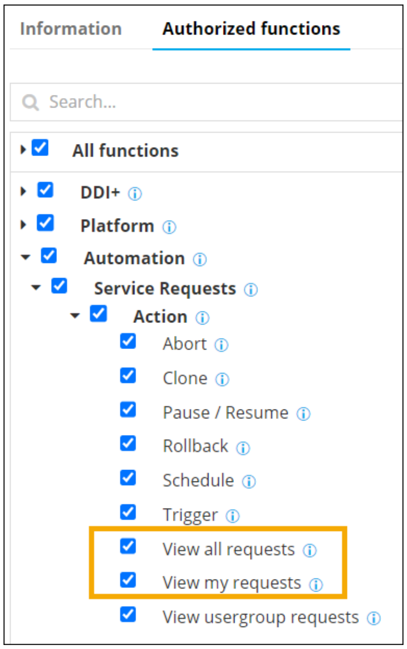
Updates an existing request waiting for a user input to progress and proceed to the next task of the workflow. This is applicable only for tasks that need manual intervention. For example, Form task.

Before you begin

Before attempting to submit a request with this API, you must ensure that the following permissions are available:

- **Service Requests** > **Action** > **View all requests**
- **Service Requests** > **Action** > **View my requests**

You can check this in the **Platform** module under **Authorized functions**.



Request Structure

URL: /visualworkflow-update-request

Type: POST

Parameters

Parameter Details

Param Type	Name	Description	Field Type	Constraints
Header	sessionId	SessionId received after login.	String	Required if username and password are not provided.

Parameter Details (continued)

Param Type	Name	Description	Field Type	Constraints
Header	username	AppViewX login username	String	Required if sessionId is not provided.
Header	password	AppViewX login password	String	Required if sessionId is not provided.
Header	Content-Type	Specifies the nature of the data in the payload.	String	Value of the param should be application/json
Query	gwkey	Tenant Key. Needed only in case of multi-tenant installations.	String	NA
Query	gwsource	Source from which the request is triggered (For example: web, external)	String	NA
Body	payload	Contains all the params to be sent in the request body for the post request.	Any one of the following: Review Task Payload User Task Payload	NA

Task Specific Information

Review Task Payload

Service tasks will not have any user input for submitting the request.

Name	Mandatory	Description	Field Type	Constraints
header	Yes	Contains header param details of the request.	Header Payload	NA

Name	Mandatory	Description	Field Type	Constraints
data	Yes	Contains params for the user task.	Data Payload	NA

Header Payload

Name	Mandatory	Description	Field Type	Constraints
request_id	Yes	The request ID to update the request details.	String	NA
workorder_id	Yes	The workorder ID to update the request details.	String	If there is no workorder in the workflow then the default value will be zero (0).
task_id	Yes	The task ID which will be updated.	String	NA

Data Payload

Name	Mandatory	Description	Field Type	Constraints
input	Yes	The input data for the user task.	Input Payload	NA
task_action	Yes	Action to be performed on the task.	Integer	1 for success case and 2 for reject case

Input Payload

Nam	Mandatory	Description	Field Type	Constraints
config	Yes	The prevalidation, implementation, post validation, and rollback configuration for the devices.	Object	NA

Sample Request/Response - Review Task

Update a request with a **Review** task.

Sample Request


```
{
  "payload": {
    "header": {
   "request_id": "936",
         "workorder_id": "0",
         "task_id" : "reviewComponent_1"
    },
    "data":{
     "input" : {"config": {
    "implement": {
      "deviceList": [
        {
          "deviceName": "192.168.41.101",
          "commands": [
            {
              "command": "<push></push>",
              "deviceName": "",
              "uniqueNumber": 2,
              "sequenceNumber": 0,
              "properties": {
                "sleep": 0
              },
              "type": "push"
            }
          ]
        }
      ]
    },
    "prevalidation": {
      "deviceList": [
        {
          "deviceName": "192.168.41.101",
          "commands": [
            {
              "command": "<push></push>",
              "deviceName": "",
              "uniqueNumber": 2,
```

```
            "sequenceNumber": 0,
            "properties": {
              "sleep": 0
            },
            "type": "push"
          }
        ]
      }
    ]
  },
  "postvalidation": {
    "deviceList": [
      {
        "deviceName": "192.168.41.101",
        "commands": [
          {
            "command": "<push></push>",
            "deviceName": "",
            "uniqueNumber": 2,
            "sequenceNumber": 0,
            "properties": {
              "sleep": 0
            },
            "type": "push"
          }
        ]
      }
    ]
  }
 }
 },

   "task_action":1
  }
 }
}
```


Sample Response


```
{
  "response": {
    "message": "Request submitted to workflow engine for processing workflow request 936",
    "requestId": "936"
  },
  "message": null,
  "appStatusCode": null,
  "tags": null
}
```


Sample Request/Response - Review Task within a Subflow

Update a request with a review palette, where the **Review** task is within a subflow.

Sample Request


```
{
  "payload": {
    "header": {
   "request_id": "936",
        "workorder_id": "0",
        "task_id" : "reviewComponent_1#subflowid"
    },
    "data":{
     "input" : {"config": {
    "implement": {
      "deviceList": [
        {
          "deviceName": "192.168.41.101",
          "commands": [
            {
              "command": "<push></push>",
              "deviceName": "",
              "uniqueNumber": 2,
              "sequenceNumber": 0,
              "properties": {
                "sleep": 0
              },
              "type": "push"
```

```
        }
      ]
    }
  ]
},
"prevalidation": {
  "deviceList": [
    {
      "deviceName": "192.168.41.101",
      "commands": [
        {
          "command": "<push></push>",
          "deviceName": "",
          "uniqueNumber": 2,
          "sequenceNumber": 0,
          "properties": {
            "sleep": 0
          },
          "type": "push"
        }
      ]
    }
  ]
},
"postvalidation": {
  "deviceList": [
    {
      "deviceName": "192.168.41.101",
      "commands": [
        {
          "command": "<push></push>",
          "deviceName": "",
          "uniqueNumber": 2,
          "sequenceNumber": 0,
          "properties": {
            "sleep": 0
          },
```

```
            "type": "push"
          }
        ]
      }
    ]
  }
 }
 },

    "task_action":1
  }
 }
}
```


Sample Response


```
{
  "response": {
    "message": "Request submitted to workflow engine for processing workflow request 936",
    "requestId": "936"
  },
  "message": null,
  "appStatusCode": null,
  "tags": null
```


Grid Task Payload

Service tasks will not have any user input for submitting the request.

Name	Mandatory	Description	Field Type	Constraintss
header	Yes	Contains header param details of the request	Header Payload	NA
data	Yes	Contains params for the user task	Data Payload	NA

Header Payload

Name	Mandatory	Description	Field Type	Constraints
request_id	Yes	The request ID to update the request details.	String	NA
workorder_id	Yes	The workorder ID to update the request details.	String	If there is no workorder in the workflow then the default value will be zero (0).
task_id	Yes	The task ID which will be updated.	String	NA

Data Payload

Name	Mandatory	Description	Field Type	Constraints
input	Yes	The input data for the user task.	Input Payload	NA
task_action	Yes	Action to be performed on the task.	Integer	1 for success case and 2 for reject case

Input Payload

Name	Mandatory	Description	Field Type	Constraints
gridData	Yes	The grid configuration to be updated in the task.	Object	NA

Sample Request/Response - Grid Task

Update a request with a **Grid** task.

Sample Request

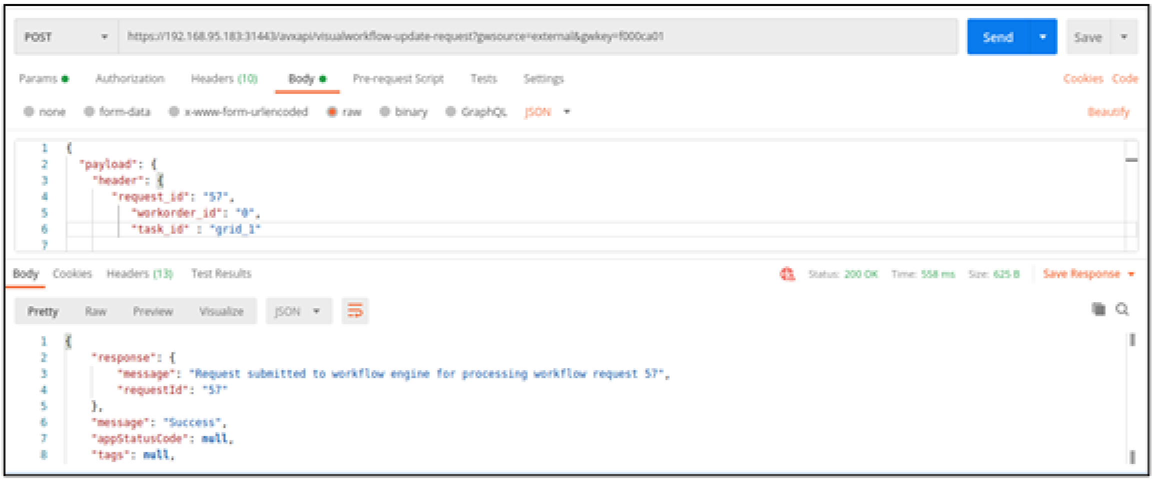

```
{
  "payload": {
 "header": {
   "request_id": "937",
   "workorder_id": "0",
```



```
   "task_id": "gridComponent_1"
  },
  "data": {
   "input": {
    "gridData": [
     {
      "name": "Appviewx",
      "designation": "Network"
     },
     {
      "name": "ADC",
      "designation": "Device"
     }
    ]
   },
   "task_action": 1
  }
 }
```


Sample Response


```
{
  "response": {
    "message": "Request submitted to workflow engine for processing workflow request 936",
    "requestId": "936"
  },
  "message": null,
  "appStatusCode": null,
  "tags": null
}
```



Response Structure

Name	Description	Field Type
response	Contains the response params for the search object request.	response
message	Success message or failure description in case of error.	String
appStatusCode	Application specific status code for the response. Will be non-null for failure response	String
tags	More info in case of failure response.	NA

Response

Name	Description	Field Type
requestId	Unique identifier for the request.	String
message	The message with the status and request number for the request.	String

Status Codes

HTTP Status Code	appStatusCode	Message	Possible remediation
200 OK	NA	Success	NA
400 Bad request	avx-common-028	Invalid/Incorrect payload	Check and ensure if a valid value is given in request payload field - input.

HTTP Status Code	appStatusCode	Message	Possible remediation
404 Not Found	engine-db-011	Request not found	Invalid workflow data
500 Internal Server Error	WORKFLOW_1056	Workflow request already closed.	Ensure workflow is not closed.
500 Internal Server Error	NA	Internal Server Error	• Task is already completed. • Task is being processed by another user. • Service Request is paused. • Service Request has been aborted. • Service request is closed. • Given task is not the right task to be executed with Id.

Get Current Task Detail of the Request

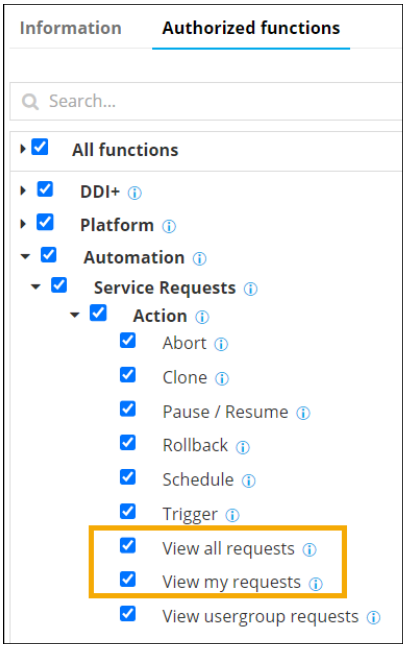
To get the details of the current active task of a specific workflow request. An action can be performed only on the current active task of a workflow.

Before you begin

Before attempting to submit a request with this API, you must ensure that the following permissions are available:

- **Service Requests** > **Action** > **View all requests**
- **Service Requests** > **Action** > **View my requests**

You can check this in the **Platform** module under **Authorized functions**.



Request Structure

URL: /visualworkflow-get-request-current-task-details

Type: POST

Parameters

Parameter Details

Param Type	Name	Description	Field Type	Constraints
Header	sessionId	Session Id received after login	String	Required if username and password are not provided.
Header	username	AppViewX login username	String	Required if sessionId is not provided.
Header	password	AppViewX login password	String	Required if sessionId is not provided.
Header	Content-Type	Specifies the nature of the data in the payload.	String	Value of the param should be application/json
Query	gwkey	Tenant Key. Needed only in case of multi tenant installations.	String	NA

Parameter Details (continued)

Param Type	Name	Description	Field Type	Constraints
Query	gwsource	Source from which the request is triggered (For example: web, external)	String	NA
Body	payload	Contains all the params to be sent in the request body for the post request.	Any one of the following: Payload User Task Payload	NA

Payload

Name	Mandatory	Description	Field Type	Constraints
request_id	Yes	The request ID to update the request details	String	NA
workorder_id	Yes	The workorder ID to update the request details.	String	If there is no workorder in the workflow then the default value will be zero (0).
task_id	Yes	The task ID which will be updated.	String	NA

Response Structure

200 OK returns string of type application/json with the following body params:

Name	Description	Field Type
response	Contains the response object along with the hierarchy of the object mapped in field children.	Key value pair
message	Success message or failure description in case of error.	String

Name	Description	Field Type
appStatusCode	Application specific status code for the response. Will be non-null for failure response.	String
tags	More info in case of failure response.	NA

Status Codes

HTTP Status Code	appStatusCode	Message	Possible remediation
200 OK	NA	Success	NA
400 Bad request	avx-common-028	Invalid/ Incorrect payload	Check and ensure if a valid value is given in request payload field - input.
404 Not Found	engine-db-011	Request not found	Check and ensure if a valid requestId/ workorderId is given.
500 Internal Server Error	engine-db-022	Unable to fetch task status	Check and ensure if a valid task ID is given.

Sample Request/Response

Use Case

To get the current task details for a request.

Sample Request

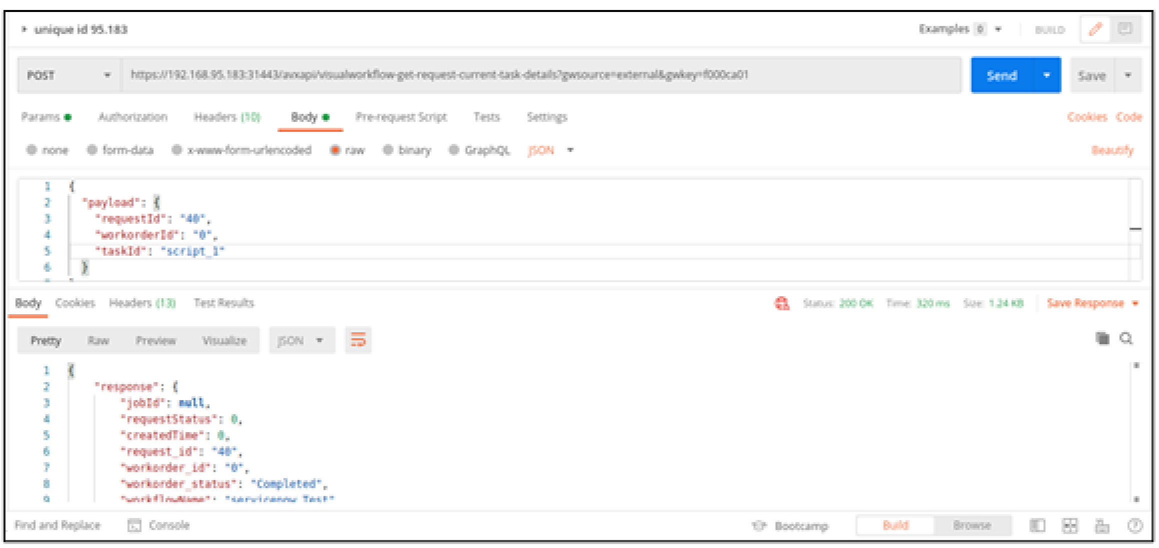

```
{
  "payload": {
    "requestId": "409",
    "workorderId": "1",
    "taskId": "reviewComponent_1"
  }
}
```


Sample Response


```
{
  "response": {
        "jobId": null,
        "requestStatus": 0,
        "createdTime": 0,
        "request_id": "409",
        "workorder_id": "1",
        "workorder_status": "Completed",
        "workflowName": "Sample",
    "data": {
      "comments": "Hi Tester of RC",
      "state": 1,
      "taskStatus": "Success",
      "task_id": "reviewComponent_1",
      "task_name": "Review Component",
      "input": {

      },
      "output": {
        "comments": "Hi Tester of RC",
        "configLabel": {
          "Prevalidation": "Pre Validation",
          "implementation": "Implementation",
          "PostValidation": "Post Validation"
        },
        "enablePrevalidation": true,
        "fetchOnLoad": true,
        "proceedProcess": "Submit",
        "ticketDetails": {
          "ticketNumber": "",
          "startTime": null,
          "endTime": null
        },
        "configureChangeManagement": true,
        "isAuthorized": true,
        "vendor": "CHANGE",
        "breakdownProcess": "Cancel",
```

```
"readWrite": true,
"prevalidationValue": "Check",
"config": {
  "Prevalidation": {
    "deviceList": [
      {
        "deviceName": "f5",
        "commands": [
          "tmsh",
          "list gtm pool pool41_161_demo"
        ]
      }
    ]
  },
  "implementation": {
    "deviceList": [
      {
        "deviceName": "f5",
        "commands": [
          "tmsh",
          "create gtm pool pool41_161_demo"
        ]
      }
    ]
  },
  "PostValidation": {
    "deviceList": [
      {
        "deviceName": "f5",
        "commands": [
          "tmsh",
          "show sys version"
        ]
      }
    ]
  }
},
```

```
        "logs": [
          "Request doesn't have valid ticket details."
        ],
        "implementationType": "Manual"
      },
      "is_picked": false,
      "roles": null,
      "tags": null,
      "label": null,
      "component_type": "review",
      "task_process_state": 1,
      "is_completed": true,
      "task_category": "review"
    }
  },
  "message": null,
  "appStatusCode": null,
  "tags": null
}
```



Get Request Summary/Log

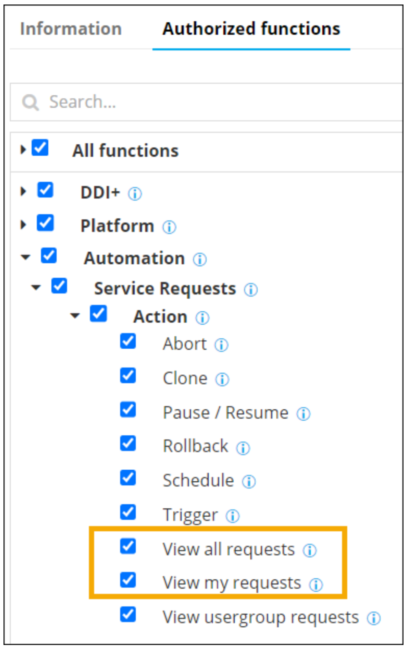
Get the request summary/logs details of a particular workflow request. The execution details of the workflow can be viewed from these logs.

Before you begin

Before attempting to submit a request with this API, you must ensure that the following permissions are available:

- **Service Requests** > **Action** > **View all requests**
- **Service Requests** > **Action** > **View my requests**

You can check this in the **Platform** module under **Authorized functions**.



Request Structure

URL: /visualworkflow-request-logs

Type: GET

Parameters

Parameter Details

Param Type	Name	Description	Field Type	Constraints
Header	sessionId	Session Id received after login	String	Required if username and password are not provided.

Parameter Details (continued)

Param Type	Name	Description	Field Type	Constraints
Header	username	AppViewX login username	String	Required if sessionId is not provided.
Header	password	AppViewX login password	String	Required if sessionId is not provided.
Header	Content-Type	Specifies the nature of the data in the payload.	String	Value of the param should be application/json
Query	gwkey	Tenant Key. Needed only in case of multi-tenant installations.	String	NA
Query	gwsource	Source from which the request is triggered (For example: web, external)	String	NA
Query	ids	Request IDs	String	NA

Response Structure

200 OK returns string of type application/json with the following body params:

Name	Description	Field Type
response	Contains the response object along with the hierarchy of the object mapped in field children.	Key value pair
message	Success message or failure description in case of error.	String
appStatusCode	Application specific status code for the response. Will be non-null for failure response.	String
tags	More info in case of failure response.	NA

Status Codes

HTTP Status Code	appStatusCode	Message	Possible remediation
200 OK	NA	Success	NA

HTTP Status Code	appStatusCode	Message	Possible remediation
400 Bad request	WORKFLOW_1300	Workflow Request ID is Mandatory.	Check and ensure if a valid request ID is given.
404 Not Found	engine-db-011	Request not found	Invalid workflow data

Sample Request/Response

To get the request summary/logs for a request.

Sample Request

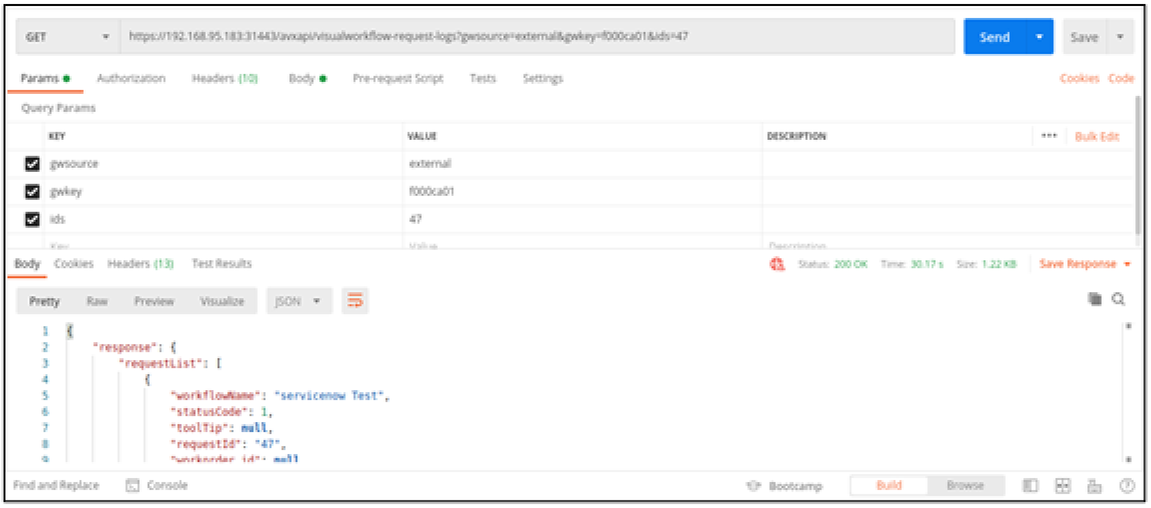
```
NA
```

Sample Response


```
{
    "response": {
        "workOrderList": [
            {
                "statusCode": 1,
                "requestId": "410",
                 "created_by": "admin",
                "status": "Completed",
                "workflowName": "Copy of reviewComponent Test",
                 "toolTip": null,
                "workorder_id": null,
                 "description": null,
                "ticket_details": null,
                "tasks": [
                    {
                        "state": 1,
                       “comments”:null,
                        "task_name": "WorkOrder",
                        "task_id": "startWorkOrder_1",
                        "user": "admin",
                        "task_status": "Success",
                        "task_start_time": 1502696313667,
                        "task_end_time": 1502696729781,
```



```
                        "logs": [
                            {
                                "time": 1502696313815,
                                "message": "Child WorkOrder ID: 1, has been
created for the Child workflow: WorkOrder"
                            },
                            {
                                "time": 1502696729781,
                                "message": "WorkOrder Completed"
                            }
                        ]
                    }
                ],
                "created_time": 1502696305239,
                "rollback_reference_id": "",
                "clone_reference_ids": [],
                "last_updated": 1502696305239,
                "runnning_time": 0
            }
        ]
    },
    "message": null,
    "appStatusCode": null,
    "tags": null
}
```

Get Associated Workorder

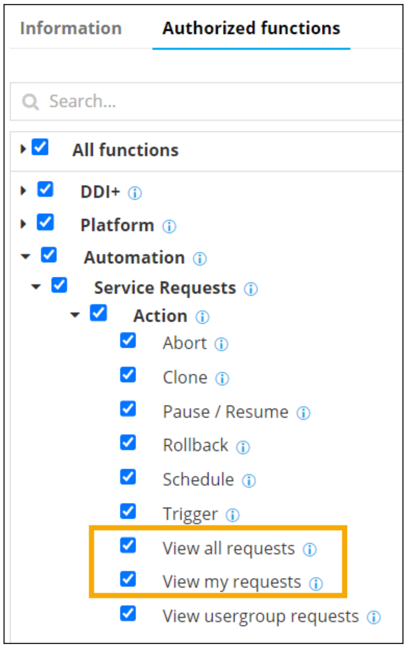
Retrieve the workorders associated with a specific workflow request. Workorders are used when there are multiple sub workflows involved with a workflow. Each workorder takes a separate parallel execution path.

Before you begin

Before attempting to submit a request with this API, you must ensure that the following permissions are available:

- **Service Requests** > **Action** > **View all requests**
- **Service Requests** > **Action** > **View my requests**

You can check this in the **Platform** module under **Authorized functions.**



Request Structure

URL: /visualworkflow-get-associated-workorder

Type: GET

Parameters

Parameter Details

Param Type	Name	Description	Field Type	Constraints
Header	sessionId	SessionId received after login.	String	Required if username and password are not provided.
Header	username	AppViewX login username	String	Required if sessionId is not provided.
Header	password	AppViewX login password	String	Required if sessionId is not provided.
Header	Content-Type	Specifies the nature of the data in the payload.	String	Value of the param should be application/json
Query	gwkey	Tenant Key. Needed only in case of multi-tenant installations.	String	NA

Parameter Details (continued)

Param Type	Name	Description	Field Type	Constraints
Query	gwsource	Source from which the request is triggered (For example: web, external)	String	NA
Query	requestId	Request id	String	NA

Response Structure

200 OK returns string of type application/json with the following body params:

Name	Description	Field Type
response	Contains the response object along with the hierarchy of the object mapped in field children.	Key value pair
message	Success message or failure description in case of error.	String
appStatusCode	.Application specific status code for the response. Will be non-null for failure response.	String
tags	More info in case of failure response.	NA

Status Codes

HTTP Status Code	appStatusCode	Message	Possible remediation
200 OK	NA	Success	NA
400 Bad request	NA	Workflow Request ID is Mandatory	Check and ensure if a valid request ID is given.
404 Not Found	NA	Request not found	Check and ensure if a valid request ID is given.

Sample Request/Response

Sample Request

```
NA
```


Sample Response


```
{
  "response": {
    "workorderList": [
      {
     "workorderId": "1",
     "createdTime": 1528957336385,
     "lastUpdatedTime": 1528957336385,
     "description": "startWorkOrder_1",
     "toolTip": {

     },
     "log": [
      {
        "time": 1528957341359,
        "message": "Initiating Script"
      }]
    }],
  "iTotalDisplayRecords": 1
  }
}
```


Get Tasks which are in InProgress State

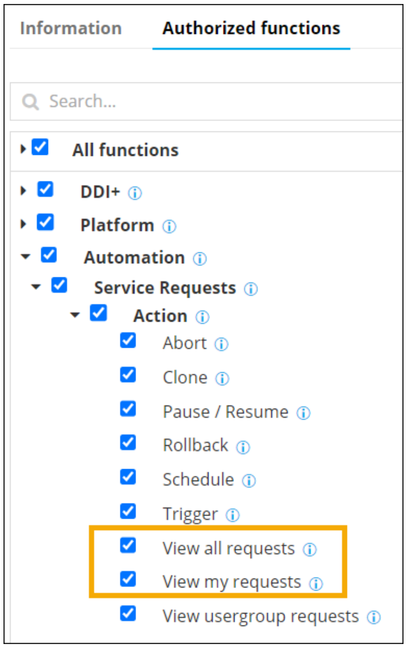
Retrieve all the tasks that are in progress for a specific request. When there are sub workflows involved there could be more than one active in-progress tasks.

Before you begin

Before attempting to submit a request with this API, you must ensure that the following permissions are available:

- **Service Requests** > **Action** > **View all requests**
- **Service Requests** > **Action** > **View my requests**

You can check this in the **Platform** module under **Authorized functions**.



Request Structure

URL: /visualworkflow-get-tasks-inprogress

Type: POST

Parameters

Parameter Details

Param Type	Name	Description	Field Type	Constraints
Header	sessionId	Session Id received after login	String	Required if username and password are not provided.
Header	username	AppViewX login username	String	Required if sessionId is not provided.
Header	password	AppViewX login password	String	Required if sessionId is not provided.
Header	Content-Type	Specifies the nature of the data in the payload.	String	Value of the param should be application/json
Query	gwkey	Tenant Key. Needed only in case of multi tenant installations.	String	NA

Parameter Details (continued)

Param Type	Name	Description	Field Type	Constraints
Query	gwsource	Source from which the request is triggered (For example: web, external)	String	NA
Body	payload	Contains all the params to be sent in the request body for the post request.	Payload	NA

Payload

Name	Mandatory	Description	Field Type	Constraints
request_id	Yes	The request ID to update the request details.	String	NA
workorder_id	Yes	The workorder ID to update the request details.	String	If there is no workorder in the workflow then the default value will be zero (0).

Response Structure

200 OK returns string of type application/json with the following body paras:

Name	Description	Field Type
response	Contains the response object along with the hierarchy of the object mapped in field children.	Key value pair
message	Success message or failure description in case of error.	String
appStatusCode	Application specific status code for the response. Will be non-null for failure response.	String
tags	More info in case of failure response.	NA

Status Codes

HTTP Status Code	appStatusCode	Message	Possible remediation
200 OK	NA	Success	NA
400 Bad request	avx-common-028	Invalid/ Incorrect payload	Check and ensure if a valid value is given in request payload field - input.
400 Bad request	common-service-007	Input data is not valid	Check and ensure if a valid value is given in request payload field - input requestId/workorderId
404 Not Found	engine-db-011	Request not found	Check and ensure if a valid value is given in request payload field - input requestId/workorderId

Sample Request/Response

To retrieve all the tasks of a request, which are in In progress state.

Sample Request

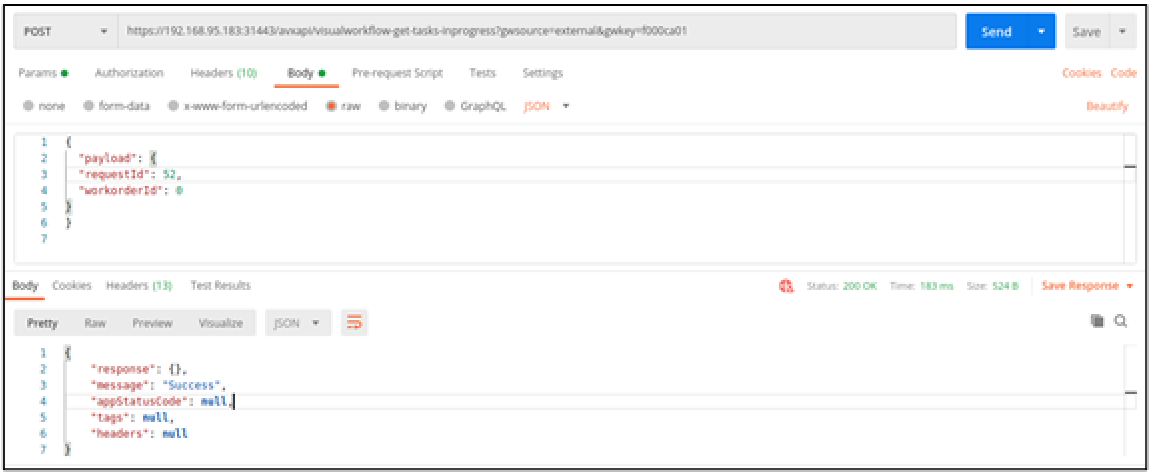

```
{
  "payload": {
  "requestId": "2",
  "workorderId": 0
}
}
```


Sample Response


```
{
  "response": {
 "tasksInProgress": [
   {
     "status": "In Progress",
     "taskName": "Implementation"
   }
 ]
  }
```



```
}
```



Upload File

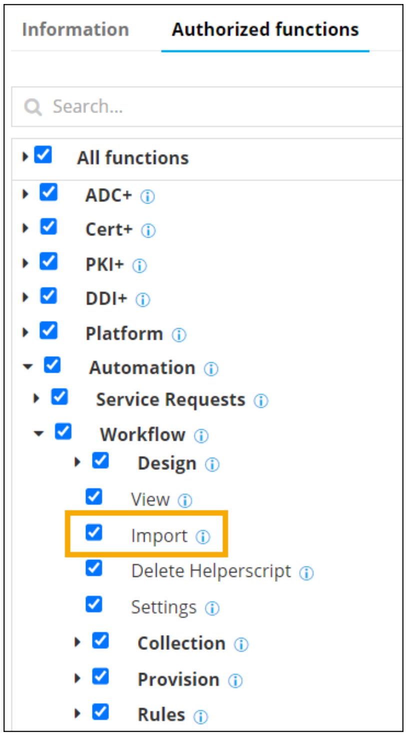
Upload file supports uploading a file and overwriting an uploaded file.

Before you begin

Before attempting to submit a request with this API, you must ensure that the following permissions are available:

- **Automation** > **Workflow** > **Import**

You can check this in the **Platform** module under **Authorized functions.**



Request Structure

URL: /visualworkflow-file-operation-upload

Type: POST

Parameters

Parameter Details

Param Type	Name	Description	Field Type	Constraints
Header	sessionId**	Session Id received after login	String	Required if username and password are not provided.
Header	username	AppViewX login username	String	Required if sessionId is not provided.
Header	password	AppViewX login password	String	Required if sessionId is not provided.
Header	Content-Type	Specifies the nature of the data in the payload.	String	Value of the param should be multipart/form-data

Parameter Details (continued)

Param Type	Name	Description	Field Type	Constraints
Query	gwkey	Tenant Key. Needed only in case of multi tenant installations.	String	NA
Query	gwsource**	Source from which the request is triggered (For example: web, external)	String	NA

Payload

Name	Mandatory	Description	Field Type	Constraints
fileName	Yes	File name, allowed file formats : img, jpg, jpeg, png, doc, pdf, txt, gif, zip, ppt, pptx, xls, xlsx, csv, p7b, p7c, pfx, p12, der, pem, csr, crt, cer, key	String	NA
uniqueId	Yes	Unique ID for the file being uploaded.	String	NA
fileOverWrite	Yes	Determines whether to overwrite the file being uploaded if a file with the same uniqueId exists.	Boolean	False (Upload file only if a file with same uniqueId not present.) True (Upload file and overwrite if file with same uniqueId is present. If not, then upload the given file.)
file	Yes	File to be uploaded	Binary	NA

Response Structure

200 OK, returns string of type application/json with the following body params:

Name	Description	Field Type
response	Contains the response object along with the hierarchy of the object mapped in field children.	Key value pair
message	Success message or failure description in case of error.	String
appStatusCode	Application specific status code for the response. Will be non-null for failure response.	String
tags	More info in case of failure response.	NA

Status Codes

HTTP code	appStatus code	Message	Possible Remediation
200 OK	NA	Success	NA
400 Bad Request	StudioService_078	uniqueId field not found	Check and ensure if valid uniqueId is given.
400 Bad Request	StudioService_084	Data cannot be null or empty	Check and ensure if valid data is provided.
400 Bad Request	StudioService_077	File content not found	Check and ensure for valid file.
400 Bad Request	StudioService_071	Invalid file name	Check and ensure if valid fileName is given.
400 Bad Request	StudioService_076	File Name not found	Check and ensure if valid fileName is given
400 Bad Request	StudioService_040	Invalid field format	Check and ensure if valid file and fileName is given.
400 Bad Request	StudioService_075	File with uniqueId already exists	File with uniqueId already exists.

Sample Request/Response

Use Case: Upload file to collection

Request URL

http://appviewxapi.com/avxapi/visualworkflow-file-operation-upload?gwkey=f000ca01&gwsource=external

Sample Request

```
{
  "payload": {
 “fileName”:”sample.zip”,
 “uniqueId”:”uniqueId”,
 “fileOverWrite”:false,
 “file”:<%File to be uploaded%>
  }
}
```

Sample Response

```
{
  "response": "File Uploaded successfully",
  "message": null,
  "appStatusCode": null,
  "tags": null,
  "headers": null
}
```

Use Case - Overwrite file: Upload a file and overwrite if a file with the same uniqueID is present in the database.

Request URL

http://appviewxapi.com/avxapi/visualworkflow-file-operation-upload?gwkey=f000ca01&gwsource=external

Sample Request

```
{
  "payload": {
 “fileName”:”sample.zip”,
 “uniqueId”:”uniqueId”,
 “fileOverWrite”:true,
 “file”:<%File to be uploaded%>
  }
}
```

Sample Response


```
{
  "response": "File Uploaded successfully",
  "message": "File Uploaded successfully",
  "appStatusCode": null,
  "tags": null,
  "headers": null
}
```


Download File

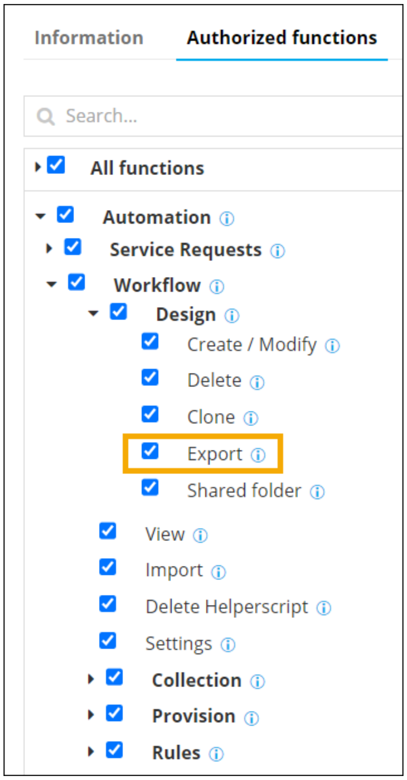
API to download the uploaded file from the database.

Before you begin

Before attempting to submit a request with this API, you must ensure that the following permissions are available:

- **Automation** > **Workflow** > **Import**

You can check this in the **Platform** module under **Authorized functions**.



Request Structure

URL: /visualworkflow-file-operation-download

Type: POST

Parameters

Parameter Details

Param Type	Name	Description	Field Type	Constraints
Header	sessionId**	Session Id received after login	String	Required if username and password are not provided.
Header	username	AppViewX login username	String	Required if sessionId is not provided.
Header	password	AppViewX login password	String	Required if sessionId is not provided.
Header	Content-Type	Specifies the nature of the data in the payload.	String	Value of the param should be application/json

Parameter Details (continued)

Param Type	Name	Description	Field Type	Constraints
Query	gwkey	Tenant Key. Needed only in case of multi tenant installations.	String	NA
Query	gwsource**	Source from which the request is triggered (For example: web, external)	String	NA

Payload

Name	Mandatory	Description	Field Type	Constraints
uniqueId	Yes	UniqueId of the file to be downloaded.	String	NA
decodeFile	No	Decide whether the file is to be decoded while downloading.	Boolean	NA

Response Structure

200 OK, returns string of type application/json with the following body params:

Name	Description	Field Type
response	Contains the response object along with the hierarchy of the object mapped in field children.	Key value pair
message	Success message or failure description in case of error.	String
appStatusCode	Application specific status code for the response. Will be non-null for failure response.	String
tags	More info in case of failure response.	NA

Status Codes

HTTP Code	appStatusCode	Description	Response message
200 OK	NA	Success	NA

HTTP Code	appStatusCode	Description	Response message
400 Bad Request	StudioService_078	uniqueID field not found	Check and ensure if valid uniqueId is given.
400 Bad Request	avx-common-028	Invalid/Incorrect payload	Check and ensure if a valid value is given in request payload field - input
404 Not Found	studio-db-043	File not found	Check and ensure if file with a uniqueId is not present.

Sample Request/Response

Request URL

http://appviewxapi.com/avxapi/?gwkey=f000ca01&gwsource=external

Sample Request

```
{
  "payload": {
 “uniqueId”:“uniqueId” }
}
```

Sample Response

```
File to be downloaded
```

Delete File

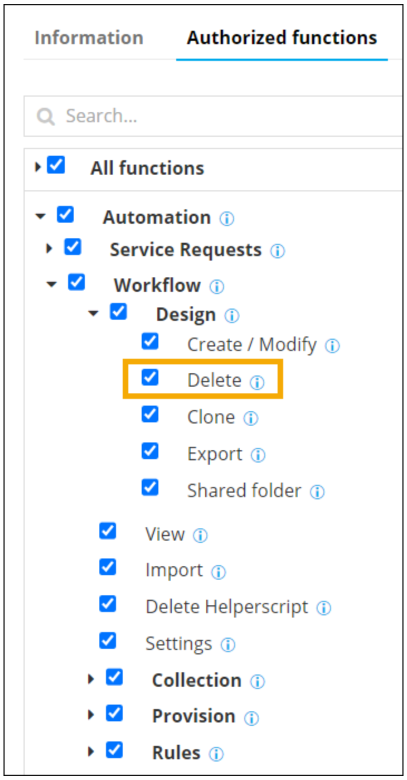
API to delete an uploaded file from the database.

Before you begin

Before attempting to submit a request with this API, you must ensure that the following permissions are available:

- **Automation** > **Workflow** > **Import**

You can check this in the **Platform** module under **Authorized functions**.



Request Structure

URL: /visualworkflow-file-operation-delete

Type: POST

Parameters

Parameter Details

Param Type	Name	Description	Field Type	Constraints
Header	sessionID**	Session Id received after login.	String	NA
Header	Content-Type	Specifies the nature of the data in the payload.	String	Value of the param should be application/json
Query	gwkey	Tenant Key. Needed only in case of multi tenant installations.	String	NA
Query	gwsource**	Source from which the request is triggered (For example: web, external)	String	NA

Payload

Name	Mandatory	Description	Field Type	Constraints
uniqueId	Yes	UniqueId of the file to be deleted.	String	NA

Response Structure

200 OK returns string of type application/json with the following body paras:

Name	Description	Field Type
response	Contains the response object along with the hierarchy of the object mapped in field children.	Key value pair
message	Success message or failure description in case of error.	String
appStatusCode	Application specific status code for the response. Will be non-null for failure response.	String
tags	More info in case of failure response.	NA

Status Codes

HTTP code	appStatus Code	Message	Possible Remediation
200 OK	NA	Success	NA
400 Bad Request	StudioService_078	uniqueId field not found	Check and ensure if valid uniqueId is given.
404 Not Found	studio-db-043	File not found	Check if file with uniqueId not present.
400 Bad Request	avx-common-028	Invalid/Incorrect payload	Check and ensure if a valid value is given in request payload field - input.

Sample Request/Response

Request URL

http://appviewxapi.com/avxapi/visualworkflow-file-operation-delete?gwkey=f000ca01&gwsource=external

Sample Request


```
{
  "payload": {
 “uniqueId”:“uniqueId” }
}
```


Sample Response


```
{
  "response": "File Deleted successfully",
  "message": null,
  "appStatusCode": null,
  "tags": null,
  "headers": null
}
```

Chapter 3: Automation Troubleshooting Guide

This document helps you troubleshoot the common problems that you might encounter when using the functionalities in the Automation module.

- Prerequisites
- Issues with Workflows
- Issues with Workflow Tasks
- Issues in Using Helper Scripts
- Issues in Using Hooks
- Issues in Creating a Workflow Request

Prerequisites

Web Browser Requirement

Browser	Version	Notes
Firefox	Till latest (Version 84.0.4147.135)	NA
Chrome	Till latest (Version 80.0)	NA
IE	Limited support in 9, Full support from 10+	No support for IE9 post AppViewX Version 11.0
Safari	Till latest (Windows - Version 5.1.7, macOS - Version 13.1.2)	From AppViewX Version 11.1
Opera	Till latest (Version 70)	From AppViewX Version 11.1

Issues with Workflows

- Issues in Creating a Workflow
- Issues in Importing a Workflow
- Issues in Exporting a Workflow
- Issues in Workflow Validation

Issues in Creating a Workflow

Following are the errors you may face when trying to create workflows, with their possible causes and solutions:

- Creating a new workflow or Renaming a workflow

Error Message	Possible Cause	Possible Solution
Workflow name already exists	Workflow name should be unique.	Enter some other name for the workflow.
Workflow name has invalid entry	Only a few special characters are allowed.	Allowed characters are '-' , '_' , '*' , ':' , '|' and space.

- Adding versions in a workflow

Error Message	Possible Cause	Possible Solution
Version limit exceeded	A workflow can have only 10 versions.	Create a workflow with 10 or less versions.

Issues in Importing a Workflow

Following are the errors you may face when trying to import workflows from other environment(s), with their possible causes and solutions:

- Workflow version already exists

Error Message	Possible Cause	Possible Solution
The workflow version already exists	Workflow is imported with the version which is already present.	Enter some other name for the version.

- Invalid workflow format

Error Message	Possible Cause	Possible Solution
Invalid workflow format	The file contains invalid entries and it cannot be parsed.	Choose an import file that has valid workflow details.

• Version limit exceeded

Error Message	Possible Cause	Possible Solution
Version limit exceeded	A workflow can have only 10 versions.	Check if the workflow has more than 10 versions and delete the versions that are not required.

Issues in Exporting a Workflow

Following are the errors you may face when trying to export workflows from AppViewX, with their possible causes and solutions:

Error Message	Possible Cause	Possible Solution
Cannot export more than 25 workflows at once	Only 25 workflows can be exported at a time.	Select only 25 or less workflows to export.

Issues in Workflow Validation

Following are the errors you may face when trying to validate workflows, with their possible causes and solutions:

Error Message	Possible Cause	Possible Solution
Invalid Workflow	There might be an issue in the workflow.	Avoid run time errors by validating the workflow.

• Illegal flow

Error Message	Possible Cause	Possible Solution
Invalid workflow - illegal flow	Multiple task outputs are given as an input to a single task.	Use Join task to provide multiple inputs to a task
Invalid workflow - illegal flow	A single task output is connected to multiple tasks.	Use Split task to connect single task output to multiple tasks.

• Broken flow

Error Message	Possible Cause	Possible Solution
Invalid workflow - broken flow	A task is not connected to any other tasks in the workflow.	Check if all the tasks are connected.

- Subflow Validation

Error Message	Possible Cause	Possible Solution
Invalid workflow	There could be a validation error in the subflows.	Check if the subflow(s) have any illegal or broken flows.

Issues with Workflow Tasks

- Issues in Adding Tasks
- Issues in Updating Tasks
- Issues in User Interface Tasks

Issues in Adding Tasks

Following are the errors you may face when trying add tasks to workflows, with their possible causes and solutions:

Error Message	Possible Cause	Possible Solution
Invalid Task component details	When an API is used to drag and drop a task, the payload might be wrong.	Drag and drop a task from another section such as User Interface or Notification and check for the payload.
Workflow not found	Workflow details might be missing in the database.	Check whether the workflow details exist in the database.
Workflow components detail not found	Workflow component details might be missing in the database.	Check whether the workflow details exist in the database.
Link Id is mandatory to perform the action	Task ID might be missing.	Check whether the task ID is available for the task.

Issues in Updating Tasks

Following are the errors you may face when trying update existing tasks in workflows, with their possible causes and solutions:

Error Message	Possible Cause	Possible Solution
Workflow Detail not found	Workflow details might be missing in the database	Check whether the workflow details exist in the database

Issues in User Interface Tasks

Following are the errors you may face when trying add User Interface tasks to your workflows, with their possible causes and solutions:

- Access Denied Issue in Form Task

Error Message	Possible Cause	Possible Solution
User is not authorized to perform the action	The logged in resource might not have access to the Form task.	1. Open the task and click **Resource & Settings.** 2. Under **Resource & Settings**, click **Request Resource**. 3. Give submitter access to the logged in resource.

- Access Denied Issue in Other User Interface Tasks

Error Message	Possible Cause	Possible Solution
User access denied	The logged in resource might not have access to the user interface task	1. Open the task and click **Resource & Settings.** 2. Under **Resource & Settings**, click **Request Resource**. 3. Give submitter access to the logged in resource.

- Global Variable Issue in Form Task

Error Message	Possible Cause	Possible Solution
The referred global variable will be null	The global variable toggle might not be enabled when adding the field in the Form task.	Enable the global variable toggle in the Form task to refer it across the workflow.

Issues in Using Helper Scripts

- Issues in Creating a Helper Script
- Issues in Updating a Helper Script
- Issues in Importing a Helper Script

Issues in Creating a Helper Script

Following are the errors you may face when trying to create Helper Scripts, with their possible causes and solutions:

Error Message	Possible Cause	Possible Solution
Helper script name already exists	Helper Script name should be unique.	Enter some other name for the Helper Script.
Name has invalid entry	Only few special characters are allowed.	Allowed characters: '-' , '_' , '(', ')'"

Issues in Updating a Helper Script

Following are the errors you may face when trying to update Helper Scripts, with their possible causes and solutions:

Error Message	Possible Cause	Possible Solution
Syntax error/Indentation Error	Validated python script may contain syntax or indentation errors.	Fix the syntax or indentation errors in the python script.

Error Message	Possible Cause	Possible Solution
Module not found	Validated python script may contain modules that are not available in python/AppViewX or imported module name is wrong.	Import the module with the right name or add the required modules to AppViewX.

Issues in Importing a Helper Script

Following are the errors you may face when trying to import Helper Scripts, with their possible causes and solutions:

Error Message	Possible Cause	Possible Solution
Invalid helper script format	The file contains invalid entries and it cannot be passed.	Choose an import file that has valid helper script details.

Issues in Using Hooks

- Issues in Creating a Query explorer Hook
- Issues in Creating a Script Hook
- Issues in Creating a REST API Hook
- Issues in Exporting Hooks
- Issues in Importing Hooks

Issues in Creating a Query explorer Hook

Following are the errors you may face when trying to create hooks based on Query Explorer, with their possible causes and solutions:

Error Message	Possible Cause	Possible Solution
Hook name already exists in inventory	The Hook name should be unique.	Enter some other name for the Hook.
Please enter mandatory fields	Hook name/Projection fields may be missing.	Enter the Hook name and select the projection fields.

Issues in Creating a Script Hook

Following are the errors you may face when trying to create hooks based on Script, with their possible causes and solutions:

Error Message	Possible Cause	Possible Solution
Hook name already exists	The Hook name should be unique.	Enter some other name for the Hook.
Name has invalid entry	Only a few special characters are allowed.	Allowed characters: '-' , '_' , '(', ')'"
Syntax error/ Indentation Error	Validated python script may contain syntax or indentation errors.	Fix the syntax or indentation errors in the python script.
Module not found	Validated python script may contain modules that are not available in python/AppViewX or the imported module name is wrong.	Import the module with the right name or add the required modules to AppViewX.

Issues in Creating a REST API Hook

Following are the errors you may face when trying to create hooks based on REST API, with their possible causes and solutions:

Error Message	Possible Cause	Possible Solution
Hook name already exists	The Hook name should be unique.	Enter some other name for the Hook.
Name has invalid entry	Only a few special characters are allowed.	Allowed characters: '-' , '_' , '(', ')'"

Issues in Exporting Hooks

Following are the errors you may face when trying to export hooks from AppViewX, with their possible causes and solutions:

Error Message	Possible Cause	Possible Solution
Cannot export more than 25 hooks at once	Only 25 hooks can be exported at a time.	Select only 25 or less hooks to export.

Issues in Importing Hooks

Following are the errors you may face when trying to import hooks from other environments, with their possible causes and solutions:

Error Message	Possible Cause	Possible Solution
Hook already exists	The hook being imported is already available in the Hooks Inventory.	Enter some other name for the Hook.
Invalid hook format	The file contains invalid entries and it cannot be parsed.	Select an import file that has valid hook details.

Issues in Creating a Workflow Request

The **Service Requests** section in the **Automation** module gives you a single point view of all the assigned workflows, service requests triggered, and scheduled workflows. You can face the following issues when trying to trigger a workflow service request.

- Issues in Submitting a User Task
- Issues in Submitting an Executed Service Request
- Issues in Reflection of Workflow Data Updates in a Service Request
- Issues in the Form task in a Service Request
- Issues in Data Manipulation in a Service Request
- Issues in Command Task in a Service Request

Issues in Submitting a User Task

Tasks that display the data in a specific format and require a user’s approval for further action are called user tasks. Following are the possible limitations that users might encounter during task execution:

- When an **unauthorised user** tries to submit a user task, unauthorized exception will be thrown.
- When a user task is the first task of a workflow and an unauthorised user tries to trigger that workflow, unauthorized exception will be thrown.
- When a user triggers a service request with following tasks as first task of the workflow, they will not be visible in the stage-wise view of the Service Request:
  - If
  - Switch and
  - Split task

  However, you can view them in the Activity log on the **Request :: All** page.

- Clean up flow cannot contain work order tasks in it

Issues in Submitting an Executed Service Request

- A rolled back request cannot be rolled back again or cloned
- A request which is in **In Progress** state cannot be cloned or rolled back. Only requests with status as Completed, Failed, or Partial can be cloned or rolled back.

Issues in Reflection of Workflow Data Updates in a Service Request

- Any changes made o the workflow after triggering a service request will not be reflected in that particular request.
- Cloning a request will only comprise the request's data from which it has been cloned and will not reflect any changes made to that workflow later.

Issues in the Form task in a Service Request

- **Auto fetch** of associate script output will not occur if the Form task is saved as draft and opened again. It will load the data only in case of a newly triggered Form
- When an **RGF flow** is enabled in a Form task, the request execution will take place in the prescribed order by the respective users who are allowed to create, review, or submit the task in order.
- **Pseudo form** is onl supported in following tasks:
  - Slack
  - REST
  - REST (I)
  - Schedule

- Script
- Email

Issues in Data Manipulation in a Service Request

- Any workflow containing **global key(s)** with dot (.) will not be allowed to create a request [Mongo behaviour].
- When the MongoDB document containing the request data exceeds **16mb in size** due to any uploaded file (or) any huge global variable data injection (or) heavy logs and so on, the request will hang.

Issues in Command Task in a Service Request

- Command Task **execution time** is based on the response time of the device in which the commands are being executed.
- If multiple commands are configured in Command Task, there is no possibility of selecting a particular command for execution; all commands are executed.
- Command Task **execution order** cannot be scheduled within the task for each command.

Chapter 4: Service Catalog User Guide

Welcome to the complete guide to getting started with the Service Catalog. This guide describes AppViewX's Page builder tool and gives you step by step instructions on creating custom pages, incorporating various page elements like HTML pages, reports, workflow catalogs, forms, tables etc to suit user-specific requirements.

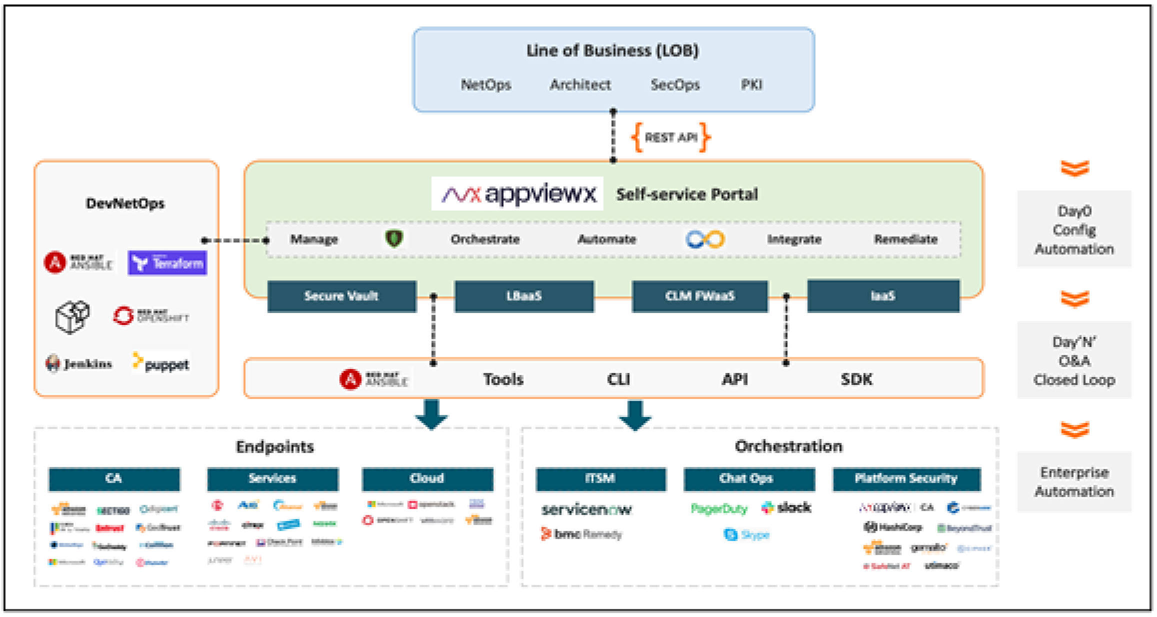
Module Overview

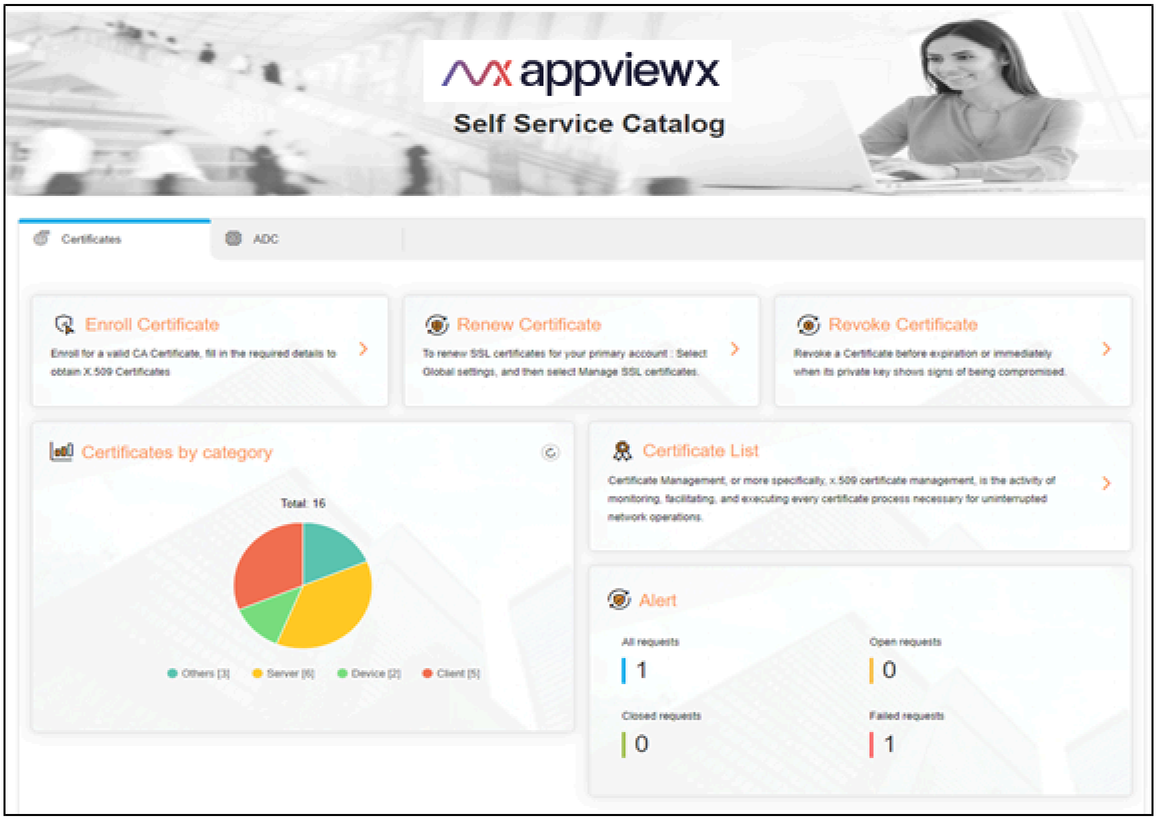
AppViewX's Service Catalog module allows you to design and customize your own catalog and access different modules within AppViewX from that catalog. You can design custom catalogs and add elements like HTML pages, quick links, buttons, search widgets and so on to suit specific requirements. Easy and quick access to multiple modules within one catalog saves time.

AppViewX's Service Catalog offers:

- Intuitive service catalogs to access service offerings for Line of Business (LOB), employees, and partners via customized pages.
- Service catalogs by persona for NetOps, PKI, Application Teams, SecOps.

- Low code page builder that allows customization of pages with personalized task views.
- Provision to design custom web pages using HTML.
- Configurable page elements and customised branding to provide a seamless intuitive experience for end users.
- Provision to share catalogs for collaboration between team members.
- Reusability of page components among multiple users.





Prerequisites

The avx_visual_page_builder plugin must be enabled during AppViewX deployment.

Web Browser Requirement

Browsers	Version
Internet Explorer	v11.0.9600.18817 or later
Firefox	V74.0.1 (64-bit) or later
Google Chrome	V85.0.4183.83 (64-bit) or later

Getting Started with the Service Catalog

AppViewX's Service Catalog is a self-service tool that allows you to:

- Use existing catalogs for immediate use
- Modify existing catalogs according to user-specific needs

- Design and customize catalogs by persona
- Assign permission for users to view, modify and share catalogs
- Publish self service catalogs.

Accessing the Service Catalog

New Menu	Old Menu
From the main navigation menu, select **Automation**. The **Automation** module is displayed with the **SERVICE CATALOG** accessible from the navigation pane on the left.	From the main navigation menu, select **Pages**. The **Pages** module is displayed with the **Published** section open by default.
Note: For more information on how to switch between menus, click here.	

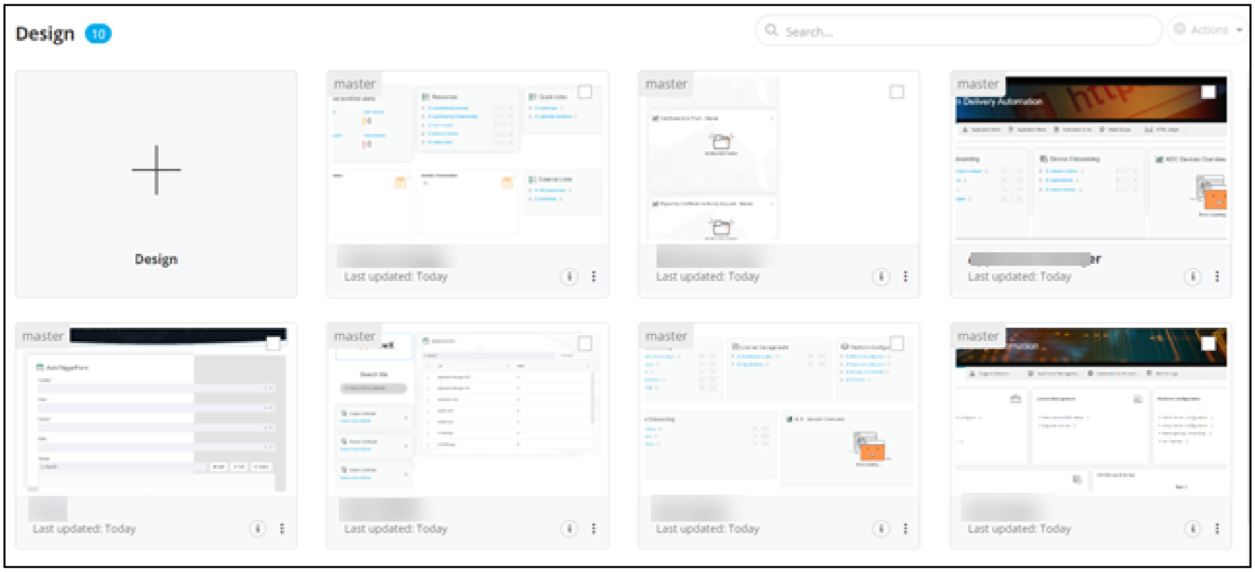
The Service Catalog module has the following sections:

- **Published**: This is an inventory of all published catalogs.
- **Design**: This is an inventory of all catalogs in design mode.
- **Store**: This is an inventory of out of the box (OOB) catalogs.
- **Import**: This section allows you to import catalogs from other environments.

- Design
- Published
- Store
- Import

Design

The **Design** page shows all catalogs created by a user that are in design mode and have not been published yet.



To access the inventory of catalogs in design mode:

New Menu	Old Menu
1. From the main navigation menu, select **Automation**. 2. In the **Automation** module, from the left menu, under **SERVICE CATALOG**, click **Design**.	1. From the main navigation menu, select **Pages**. 2. In the **Pages** module, from the left menu, click **Design**.

Note: For more information on how to switch between menus, click here.

Various options in design section & their descriptions

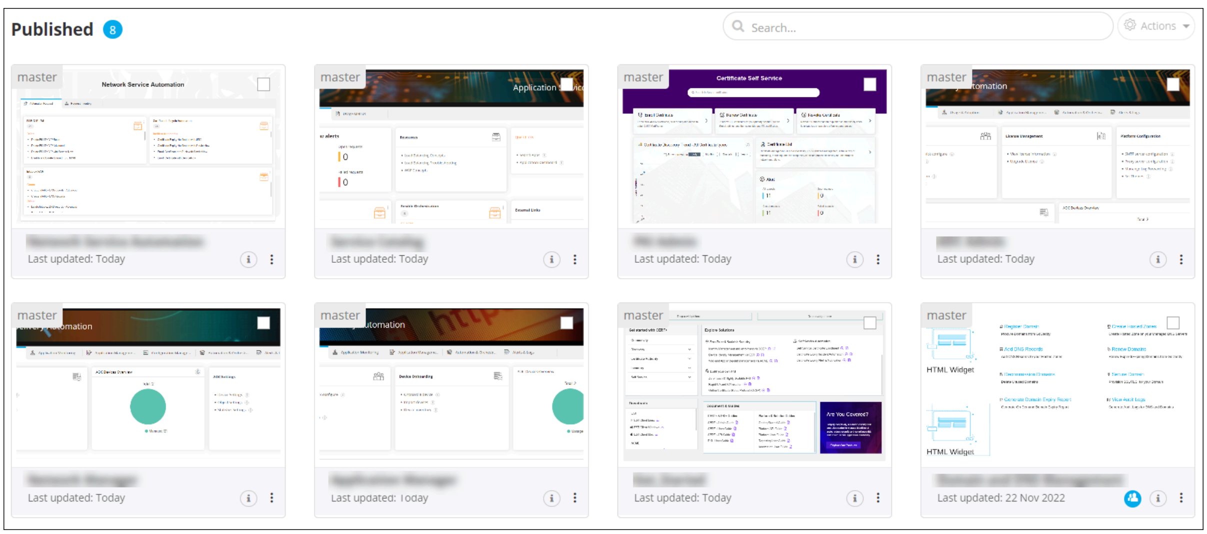
Options	Description
	Displays ‘Last Updated’ info and description of the page.
	Allows users to perform the following actions on the selected page: • **Clone** • **Delete** • **Export**.

Various options in design section & their descriptions (continued)

Options	Description
Search bar	Allows users to search for a specific page within the Design Inventory by using keyword(s).
Actions	Allows users to perform the following actions on selected page(s): • **Delete** • **Export**. **Note:** This option is enabled only when you select a page(s).

Published

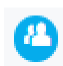
This **Published** page shows all the catalogs that are ready-to-use and published.



To access the inventory of published catalogs:

New Menu	Old Menu
1. From the main navigation menu, select **Automation**. 2. In the **Automation** module, from the left menu, under **SERVICE CATALOG**, click **Published**.	From the main navigation menu, select **Pages**. The **Pages** module is displayed with the **Published** section open by default.
Note: For more information on how to switch between menus, click here.	

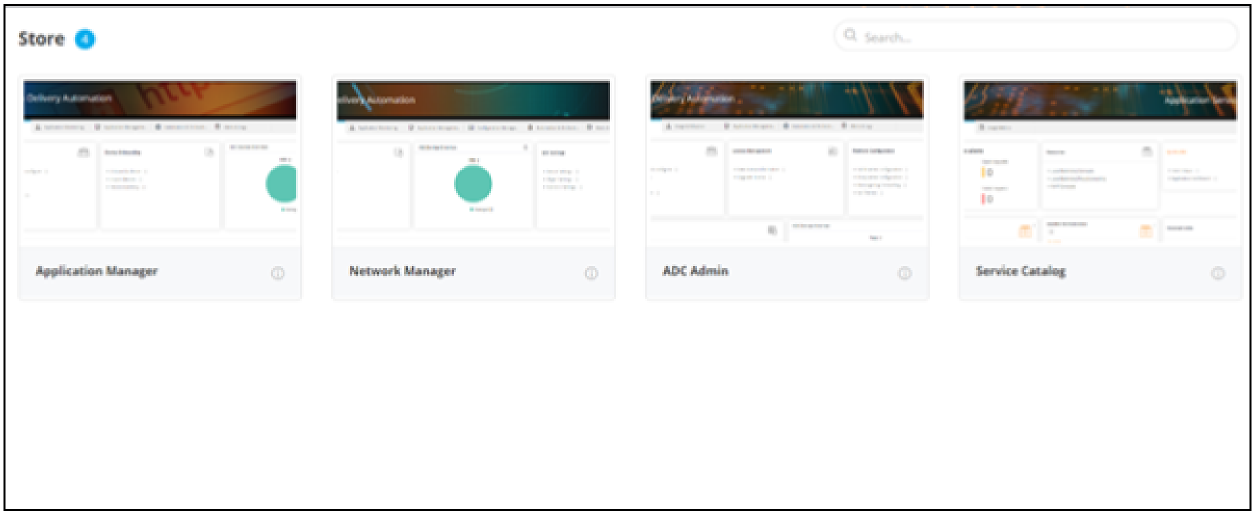
Various options in published section & their descriptions

Options	Description
	A highlighted icon on the bottom right corner of the card represents a shared page. Hovering over the icon displays the name of the person who created the page.
	Shows ‘Last Updated’ info and description of the page.
	Allows users to perform the following actions on the selected page: • **Clone**: Allows users to clone the selected page. • **Export**: Allows users to export the selected page.
Actions	Allows users to export the selected page(s). **Note:** This option is enabled only when you select a page.
Search bar	Allows users to search for a specific page within the Published inventory by using keyword(s).

Important: Published catalogs cannot be deleted.

Store

Store has a collection of prebuilt out of the box (OOB) catalogs that can be used as a reference for building your customized catalogs. Select the OOB Catalog you want to build based on the persona and customize it as per your preference.



To access the inventory of OOB catalogs:

New Menu	Old Menu
1. From the main navigation menu, select **Automation**. 2. In the **Automation** module, from the left menu, under **SERVICE CATALOG**, click **Store**.	1. From the main navigation menu, select **Pages**. 2. In the **Pages** module, from the left menu, click **Store**.
Note: For more information on how to switch between menus, click here.	

Note: For more information on Getting Started with OOB Catalogs, click here.

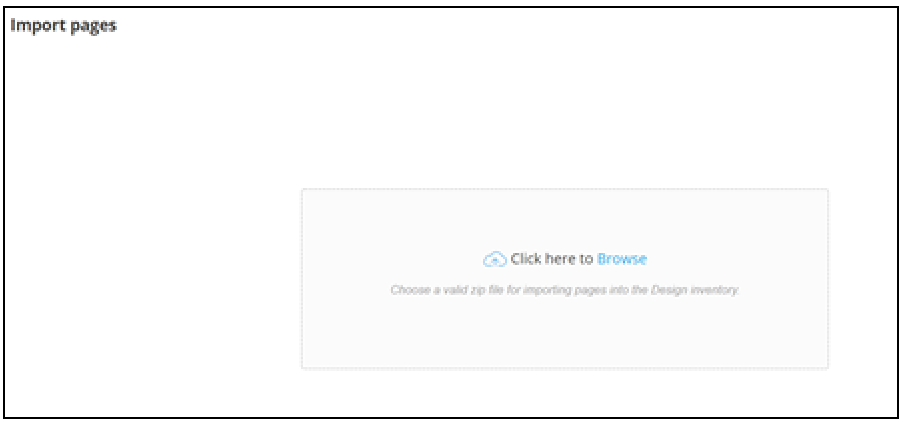
Import

You can import a page into the AppViewX Service Catalog. Importing catalogs allows you to reuse catalogs exported from one environment into another.

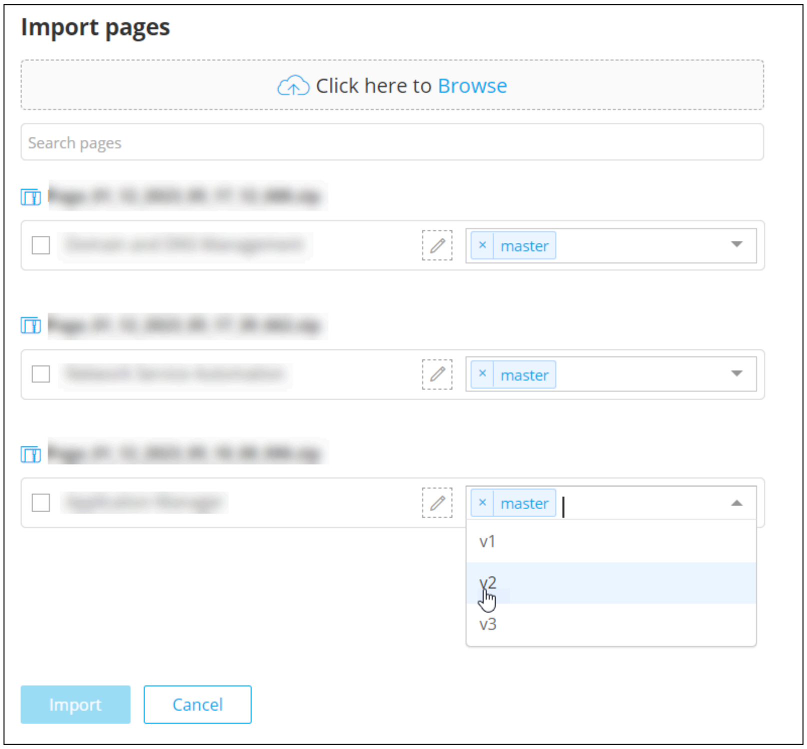
To import a catalog:

New Menu	Old Menu
1. From the main navigation menu, select **Automation**. 2. In the **Automation** module, from the left menu, under **SERVICE CATALOG**, click **Import**.	1. From the main navigation menu, select **Pages**. 2. In the **Pages** module, from the left menu, click **Import**.
Note: For more information on how to switch between menus, click here.	

1. To select a file to upload from your machine, click **Browse**.



2. Select the file(s) to be uploaded.
3. Once the files are uploaded, select the version for each catalog.



4. To change the name of the catalogs before importing it into the Design inventory, click **(Edit)** icon next to the name and type in the new name.

5. Once you have selected the required version, click **Import**.

   The imported catalog is added to the Design Inventory.

Getting Started with OOB Service Catalog

Before you begin designing your self-service catalogs, here's a brief overview of getting started with pre-built catalogs. These prebuilt catalogs give you an insight into the various actions that each persona (such as a Network Manager, ADC admin and so on) would be able to perform using their customized catalog.

These prebuilt catalogs enable you to perform the following key actions:

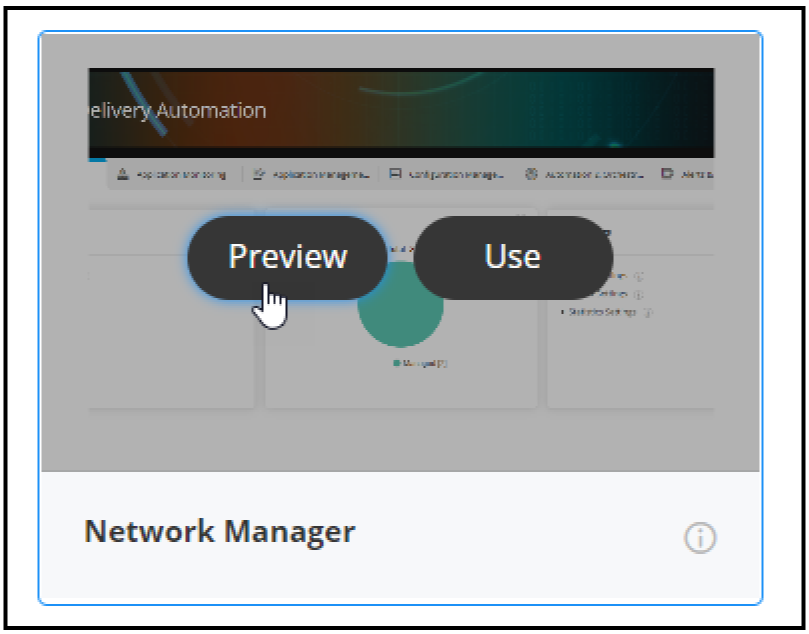
- Gain insights and visibility into application delivery services.
- Monitor and troubleshoot application services.
- Configuration management and upgrades.
- Self service and automation of application delivery services.

Based on the persona, be it a CA manager, ADC admin or an Application Manager, you can customize these catalogs and modify components to meet your requirements.

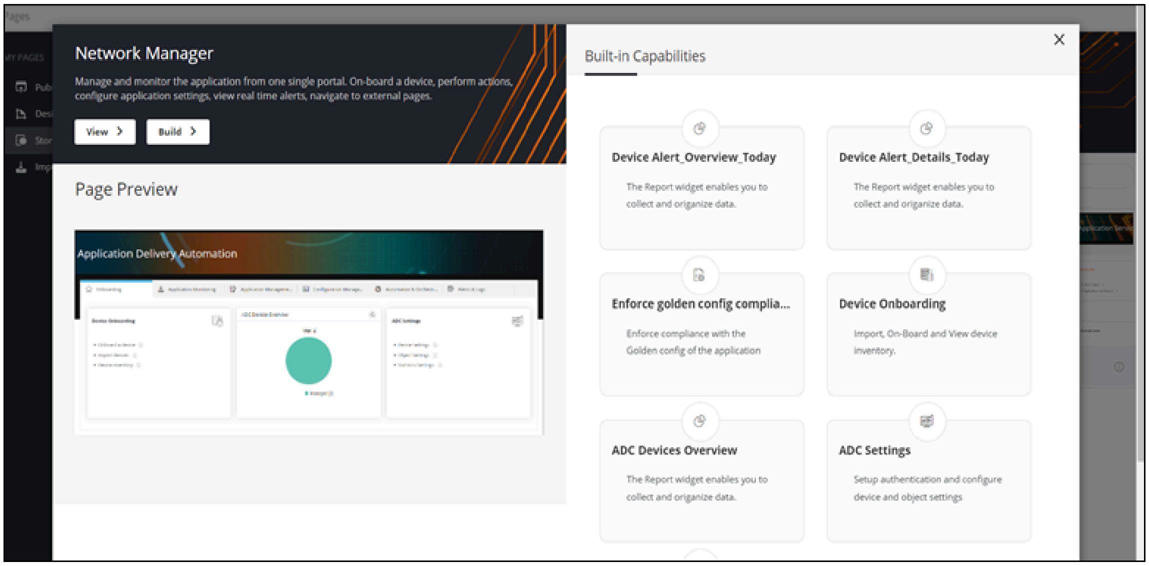
- Using OOB Catalogs
- Take a Tour

Using OOB Catalogs

1. Go to the Store.
2. To preview a prebuilt OOB catalog, hover your mouse over a catalog and click **Preview**. For example, **Network Manager**.



A preview of the Network Manager's page is displayed in a pop-up window.



3. To view the complete catalog, click **View**.

   The catalog opens in a new tab.
4. To modify this catalog, on the pop-up page, click **Build**.

   A copy of this catalog is added to the **Design** inventory and the page opens in the same tab.

   You now have the option to:
   - Publish and use this OOB catalog as is.
   - Customize this catalog to suit persona-specific needs.
   - Design a new custom page from scratch.

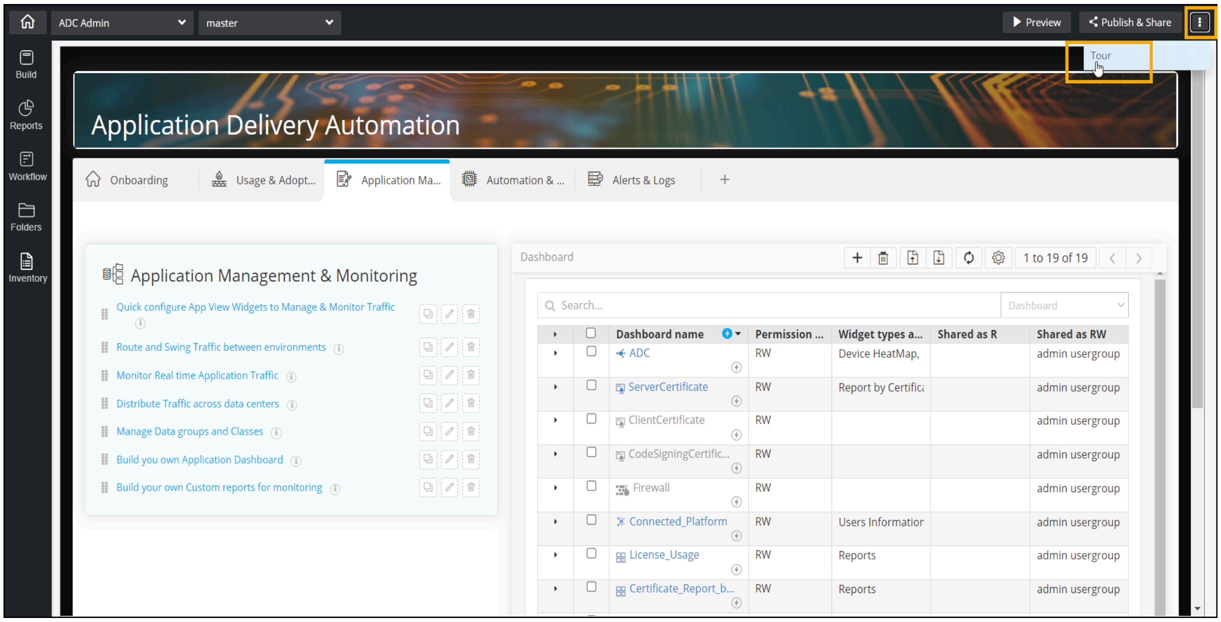
Take a Tour

Take a quick tour of the Service Catalog module and familiarize yourself with its various functionalities. Users will be prompted to take a tour when they access the Service Catalog and build a custom catalog for the first time. There is also a provision to explore the tool from any catalog in design mode.

To take a tour:

1. Open a page from the Design section.
2. From the top right corner of the screen, click .

3. To explore the Service Catalog module, click **Tour**.



Customizing Prebuilt Service Catalogs

The Service Catalog module allows you to preview out of the box (OOB) catalogs and customize them according to their specific requirements.

1. Navigate to the Design inventory and click **Design**.
2. In the **Design page** window, to customize an existing page, hover your mouse over the page and click **Use**.

   A copy of the selected page opens on your screen and is added to the Design inventory.

> **Note:** For steps on designing catalogs, click here.

> **Tip:** You can also access and customize these catalogs from the Store.

Designing a Custom Service Catalog

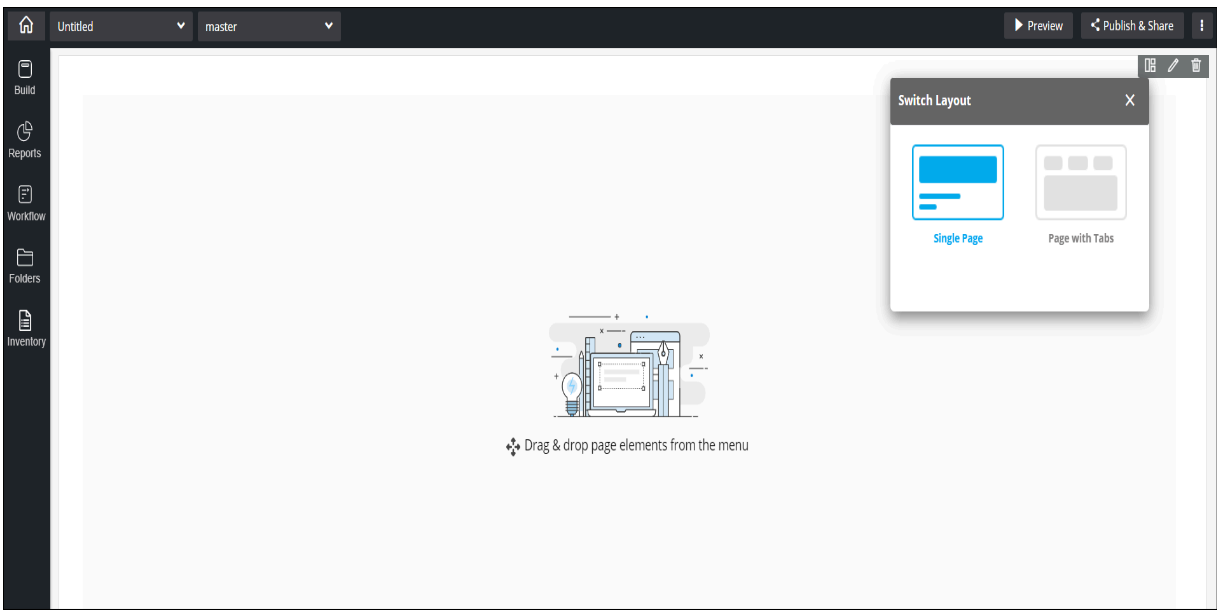
- Getting Started with Designing a Custom Catalog
- Page Layout
- Build
- Reports

- Workflow
- Folders
- Inventory
- How to Undo/Redo actions
- Moving widgets
- Cloning Widgets

Getting Started with Designing a Custom Catalog

1. Go to the Design inventory and click **Design**.
2. In the **Design page** window, click **New Page**.

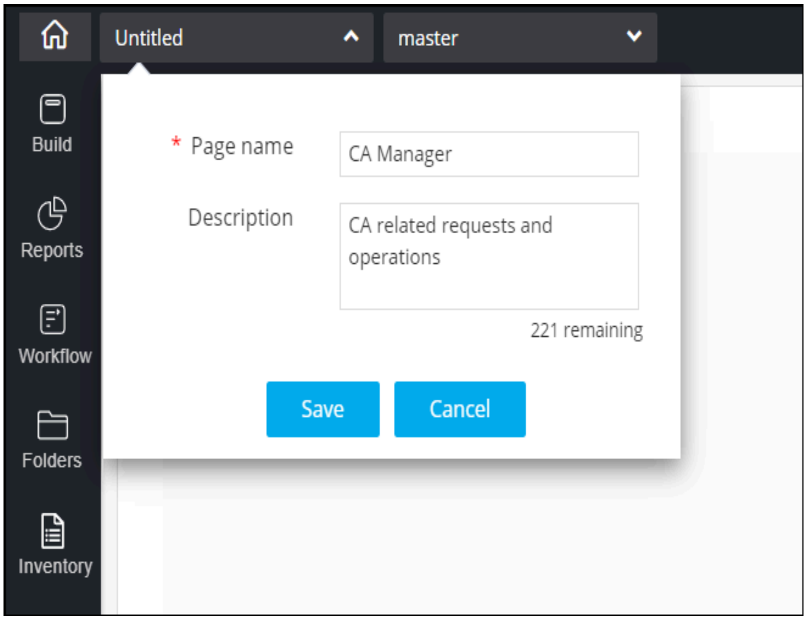
   A blank page is displayed with the **Switch Layout** window open by default.

   Note: For more information on page layouts, click here.

   The component panel on the left shows the following options for Page elements:

   - **Build**: Allows you to add widgets and form components to your catalog.
   - **Reports**: Allows you to add report widgets to your catalog.
   - **Workflow**: Allows you to add workflow catalogs to your catalog.
   - **Folders**: Allows you to save catalogs to personal and shared folders.
   - **Inventory**: Allows you to access OOB forms and customized forms.
3. To add a **Page name** and **Description,** click on the **Untitled** dropdown.



Note: Page name is a mandatory field.

4. Click **Save**.

Page Layout

You can choose between two layouts for your customized catalog:

- **Single page**: This is the default layout.
- **Page with tabs**: You can create catalogs with up to 6 tabs.

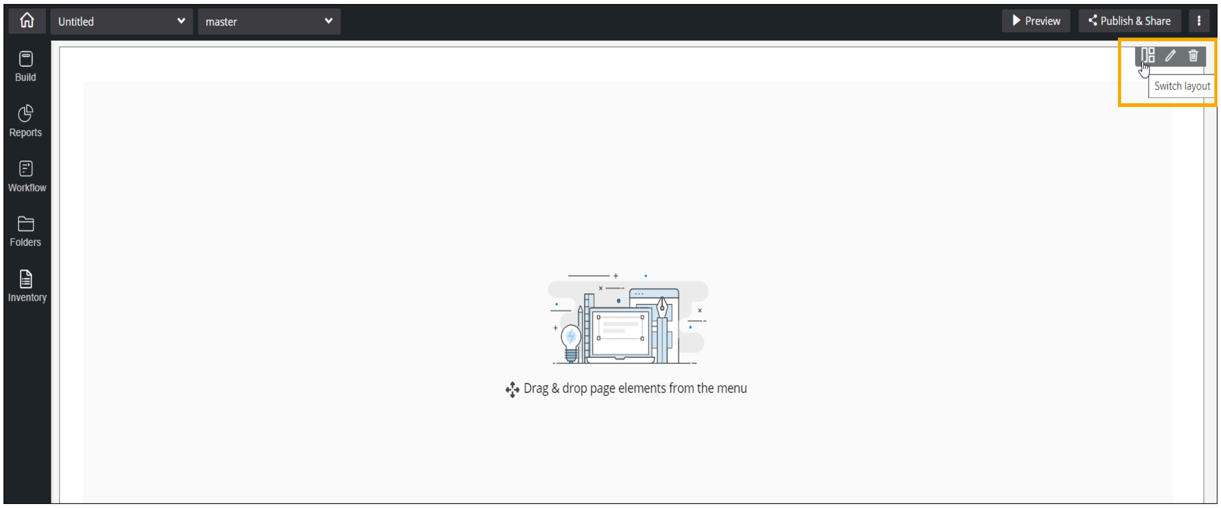
- Switching from Single Page Layout to Page with tabs Layout
- Switching from Page with tabs Layout to Single Page Layout
- Tab reordering
- Modifying page layout
- Deleting a Catalog

Switching from Single Page Layout to Page with tabs Layout

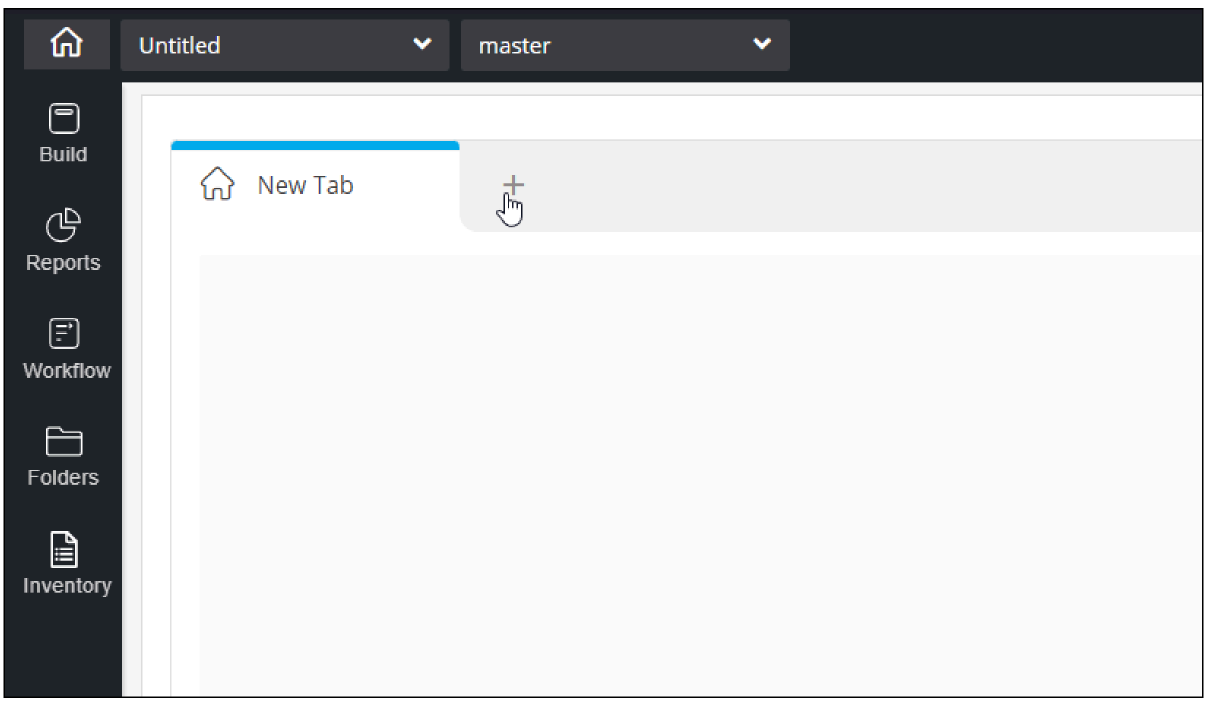
You can create a customized catalog with multiple tabs. There is a provision of adding up to 6 tabs in a catalog. The tabs can be arranged either horizontally or vertically.

To switch from single page layout to page with tabs layout:

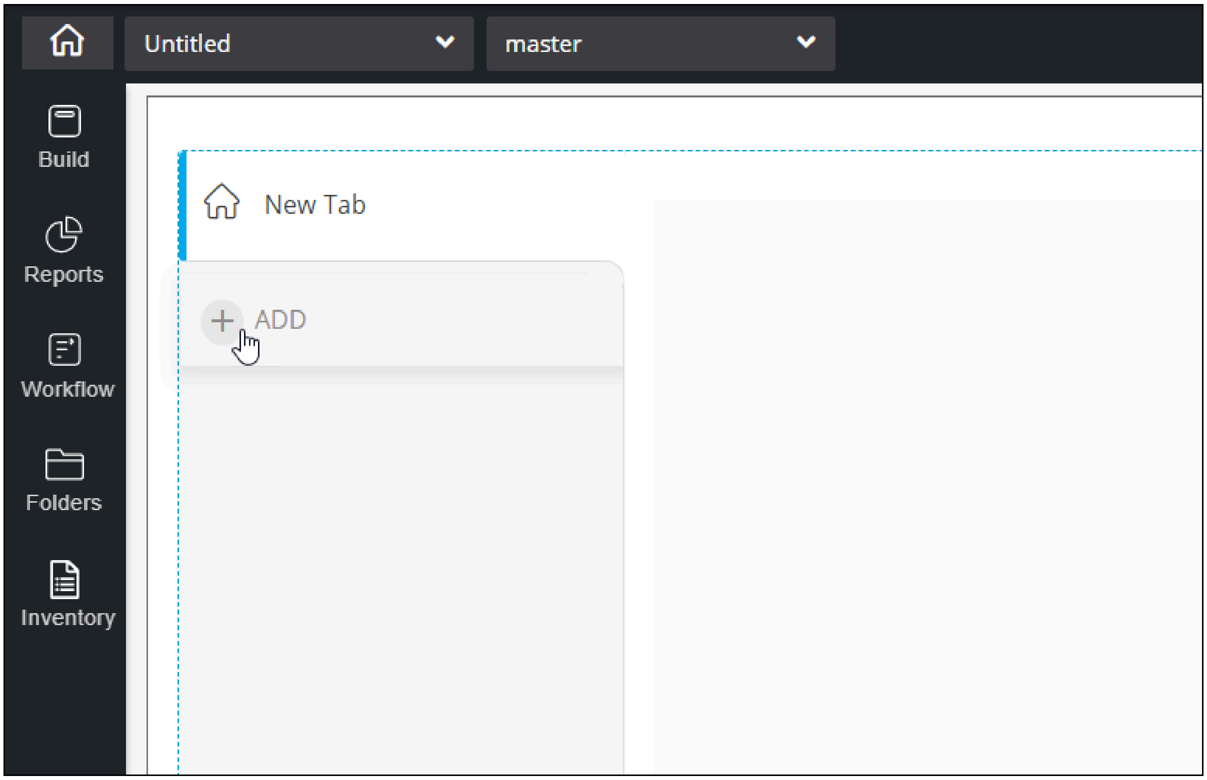
1. In the build area, from the top right corner of the screen, click **(Switch Layout)** icon.



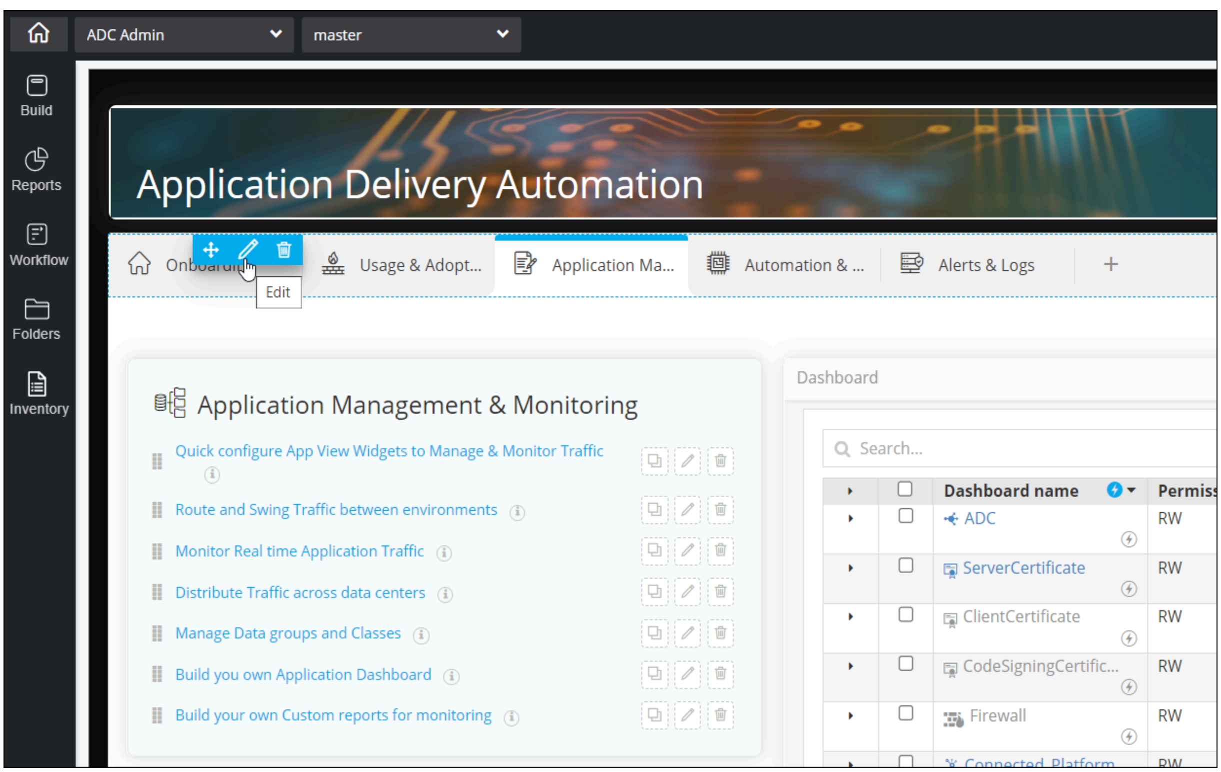
2. In the **Switch Layout** window that is displayed, click **Page with Tabs**.
3. Click **Yes** in the **Confirmation** pop-up window.
4. To flip between horizontal and vertical tabs, from the top right corner of the screen, click **(Flip Right)** icon.
5. To add a tab, when tabs are horizontally oriented, click **(Add)** icon.



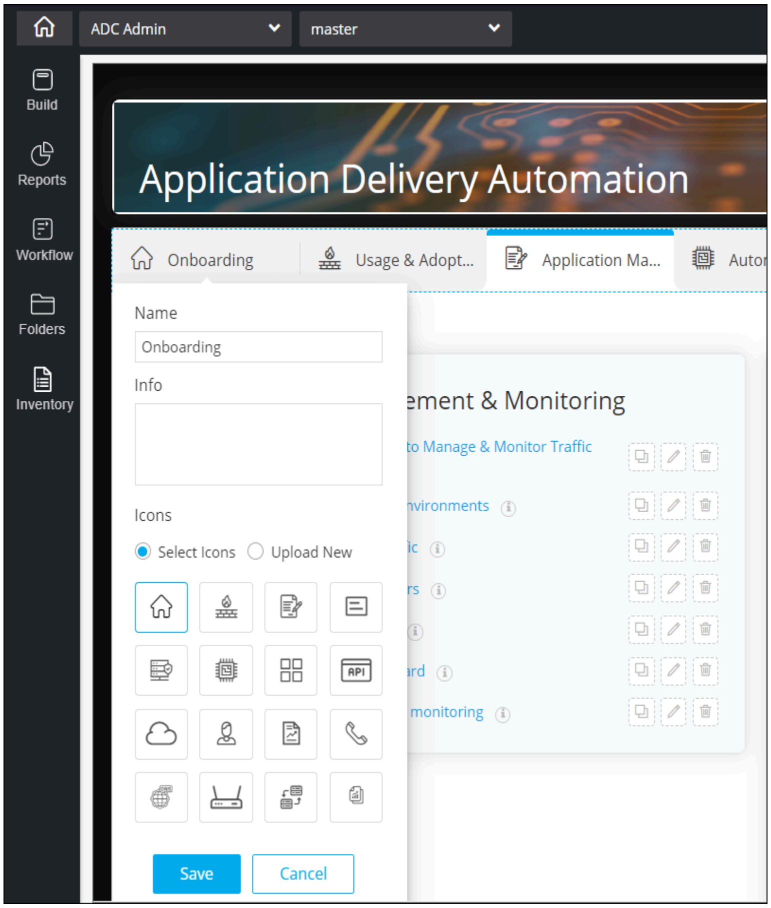
6. To add a tab, when tabs are vertically oriented, click **ADD**.



7. To modify the tab, hover your mouse over the tab and click  **(Edit)** icon.



A pop-up window is displayed right below the tab.



The following options are available here:

- **Name** - Allows you to change the name of the tab.
- **Info** - Allows you to add a tooltip for the tab.
- **Icons** - Allows you to select an icon for the tab from the options displayed or upload a new icon.

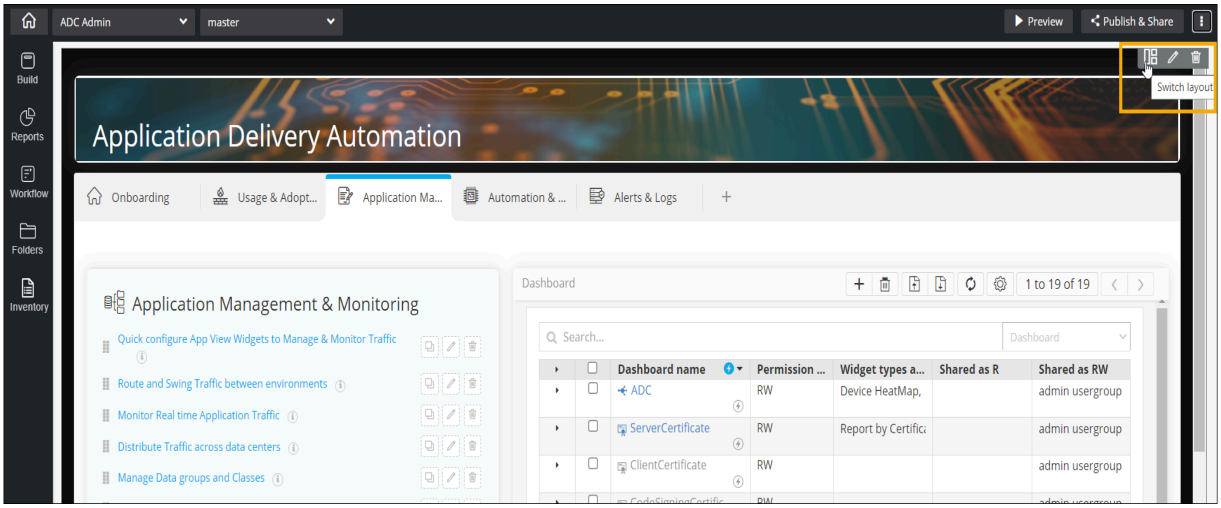
8. Click **Save**.

9. To delete a tab, hover your mouse over the tab and click  **(Delete)** icon.

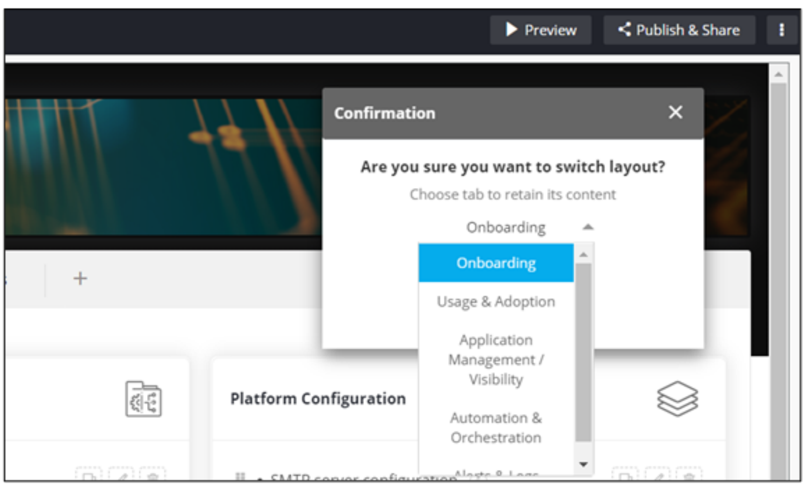
Switching from Page with tabs Layout to Single Page Layout

You can create a customized catalog with multiple tabs. There is a provision of adding up to 6 tabs in a catalog. The tabs can be arranged either horizontally or vertically.

1. In the build area, from the top right corner of the screen, click **(Switch Layout)** icon.



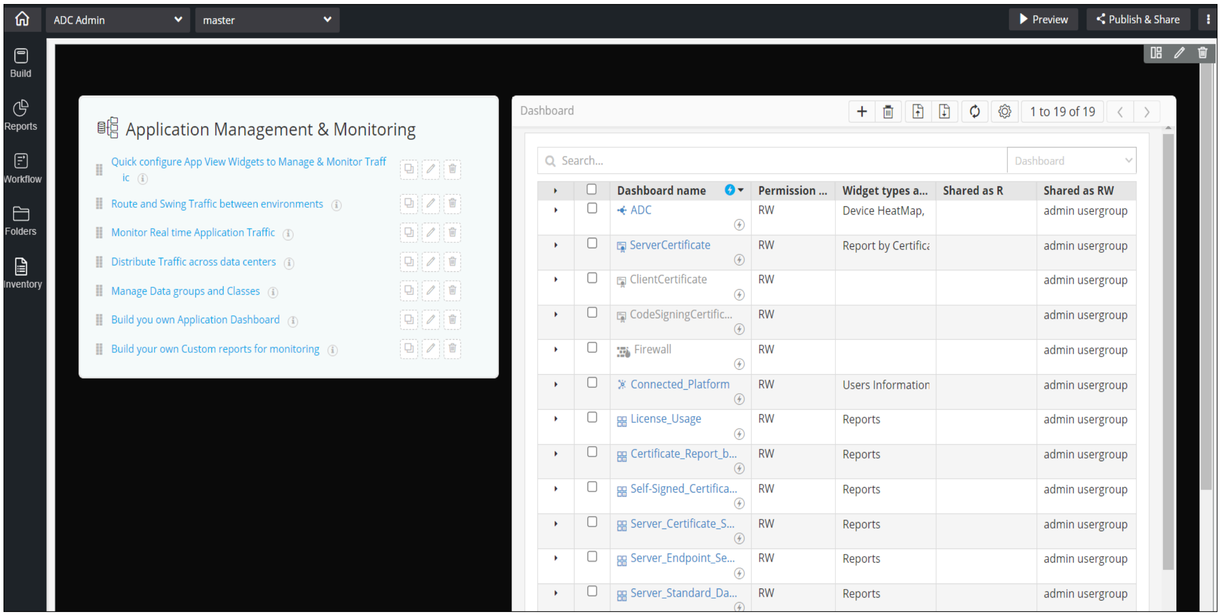
2. In the **Switch Layout** window that is displayed, click **Single Page**.
3. In the **Confirmation** window, from the dropdown, select the tab you want to retain on the Single page layout.



Warning: Remaining tabs will be deleted and cannot be recovered even after switching back to Page with Tabs layout.

4. Once you have selected the tab to be retained, click **Yes**.

The page layout is switched to Single Page displaying the retained tab.

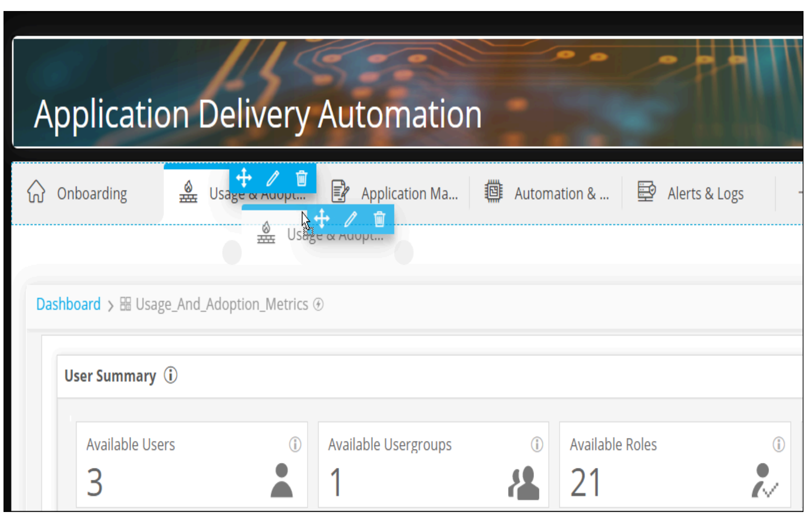


Tab reordering

When the catalog is in the Page with tabs mode, you can change the order in which the tabs are arranged.

To move the tabs:

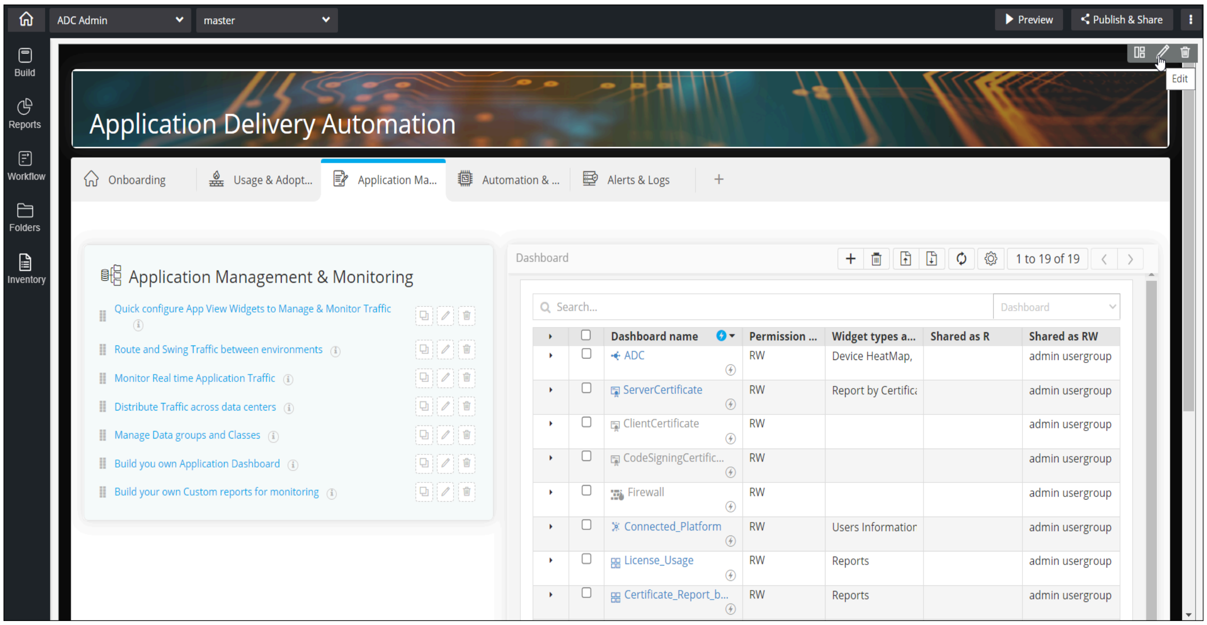
1. Hover your mouse over a tab and click .
2. Drag and drop the tab to a new position.



Modifying page layout

This feature allows you to modify the layout of the entire catalog.

1. Open a page from the Design section.
2. Hover your mouse to the top right corner of the page and click  **(Edit)** icon.



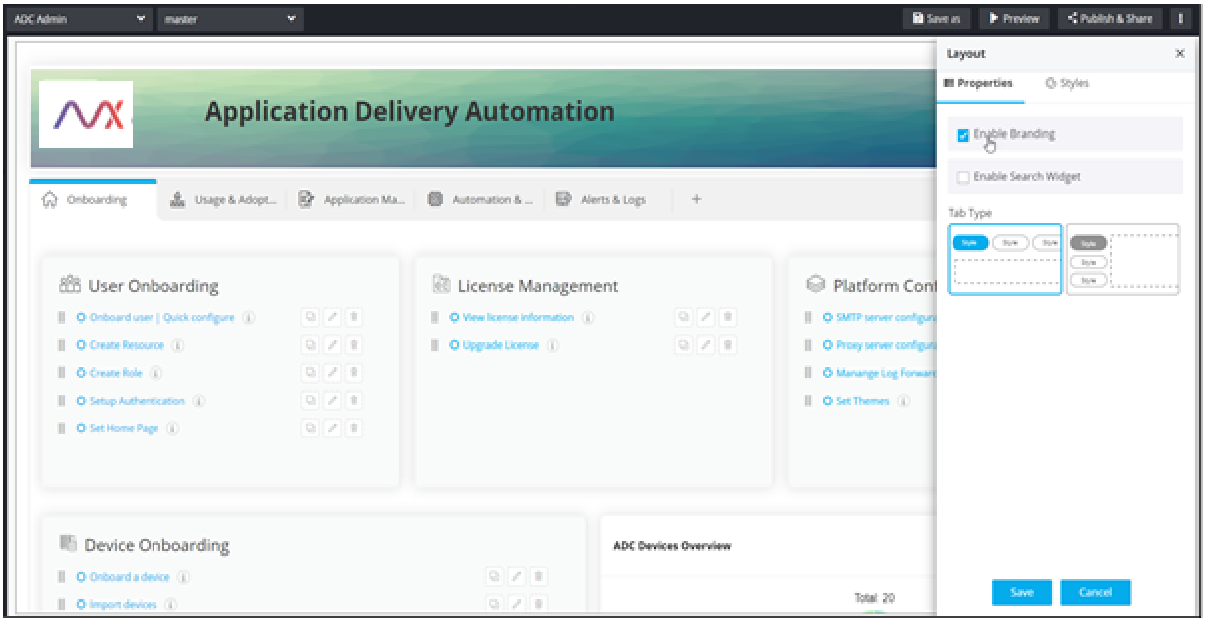
The **Layout** pop-up window is displayed with the **Properties** tab open by default.

> **Note:** The **Properties** tab is displayed only when the Page Layout is selected as Page with Tabs. In the Single Page Layout mode, the Layout pop-up window opens displaying the **Styles** tab.

- Modifying page layout - Properties
- Modifying page layout - Styles

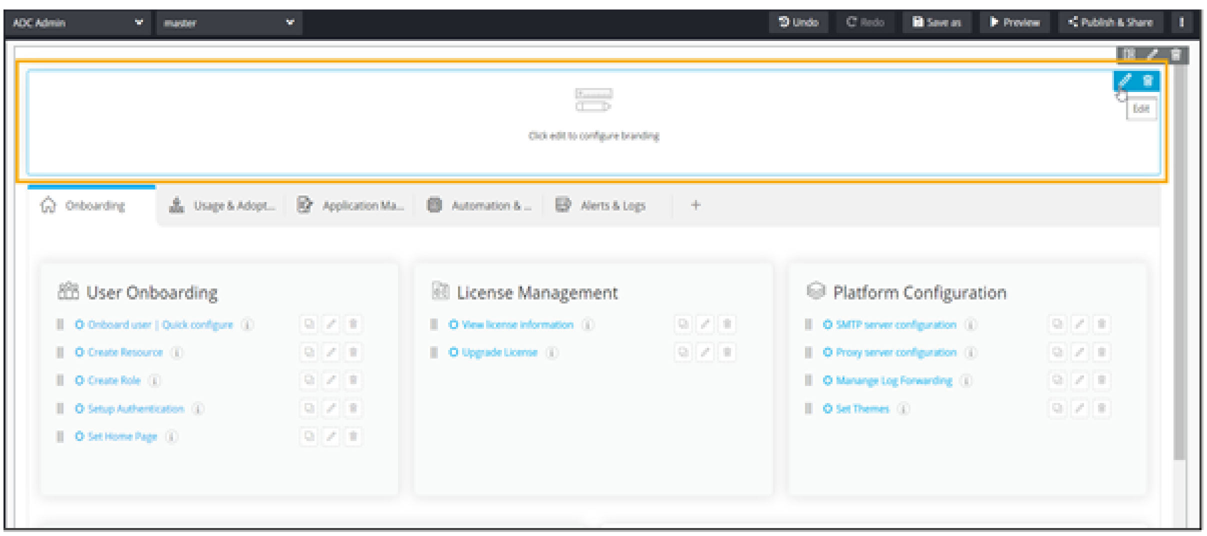
Modifying page layout - Properties

1. To enable Branding, in the **Layout** window, under **Properties**, select the **Enable Branding** checkbox.



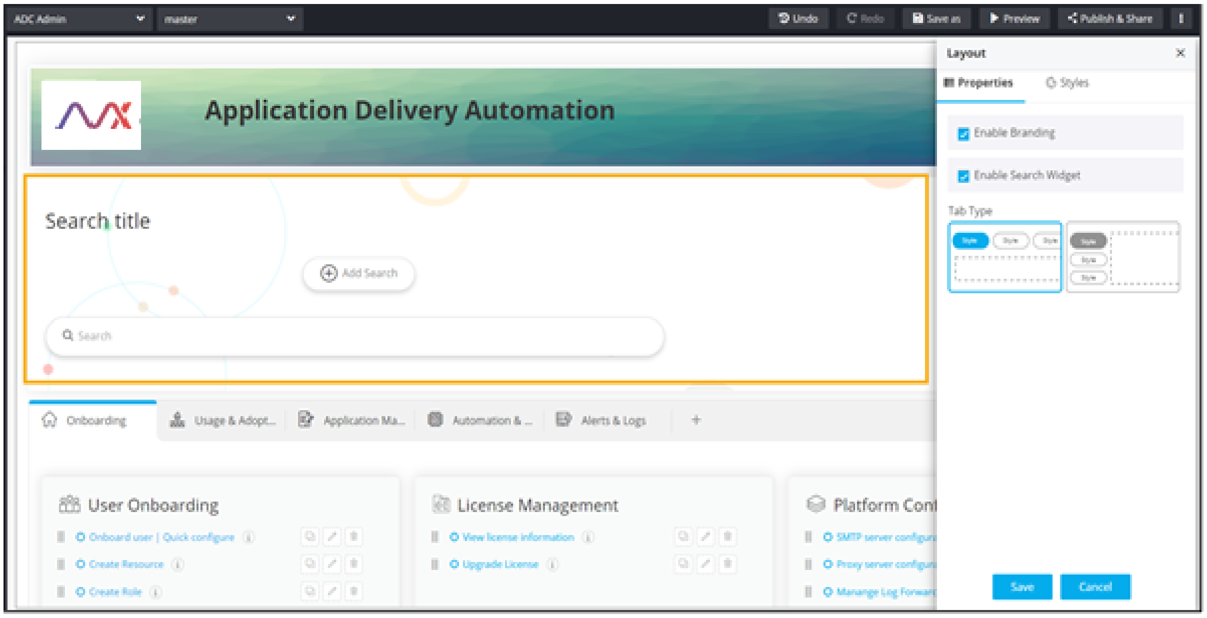
2. Click **Save**.

   Branding component is added to the build area.



3. Configure the Branding component.
4. To enable the search widget, in the **Layout** window, under **Properties**, select the **Enable Search Widget** checkbox.
5. Click **Save**.

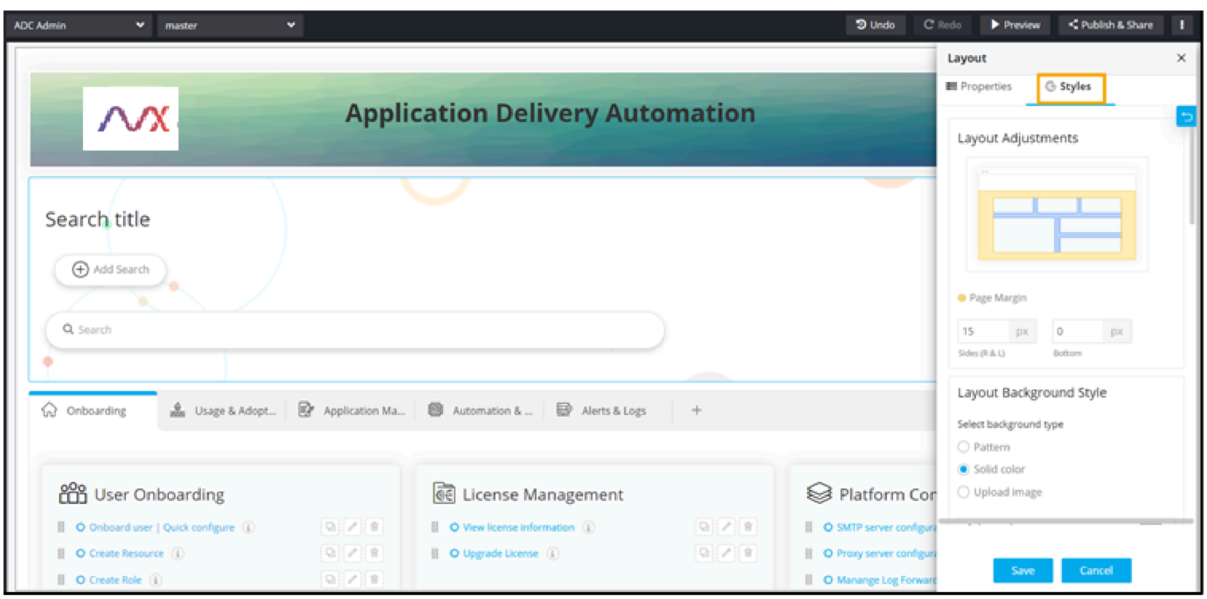
   The Search widget is added to the page.



6. Configure the Search widget.
7. To select the tab orientation, under **Tab Type**, select the required option.

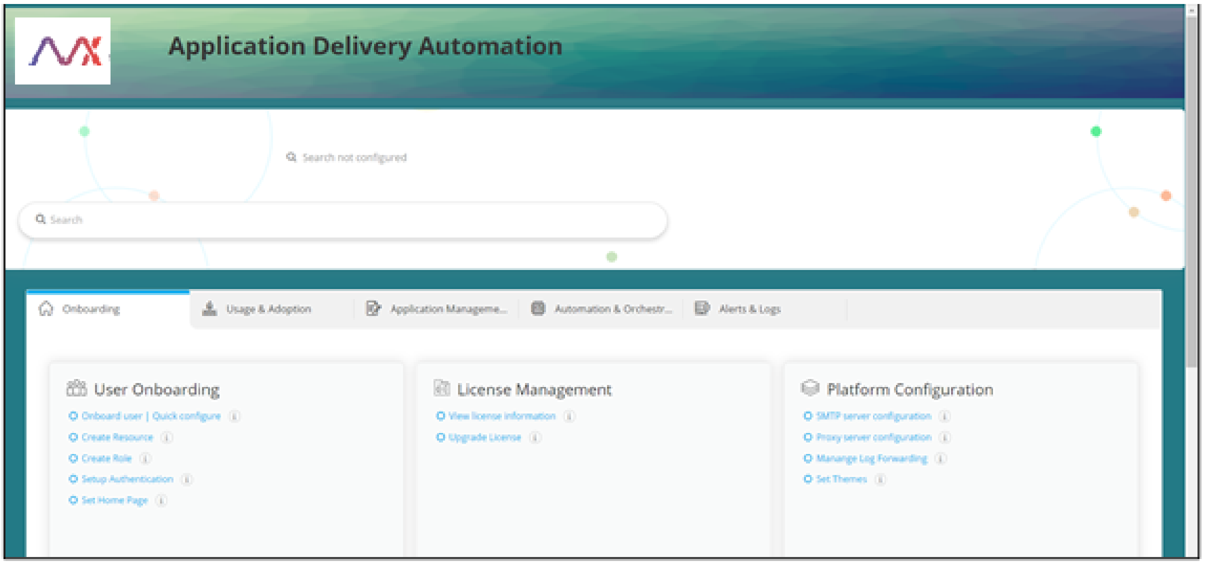
Modifying page layout - Styles

1. For more options to customize the page layout, in the **Layout** window, click **Styles**.



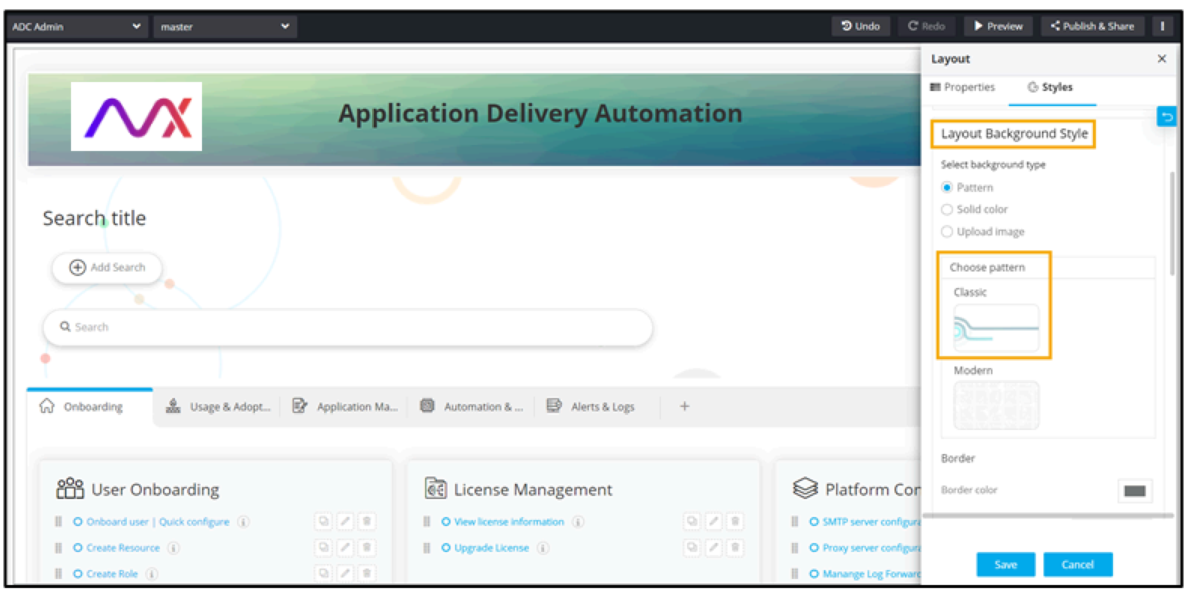
2. Under **Layout Adjustments**, increase/decrease the **Page Margin** at the sides and the bottom.
3. To see the changes reflected on the page, click **Save** and then click **Preview**.

   The **Page Margin** settings have been applied to the left, right, and bottom of the page.

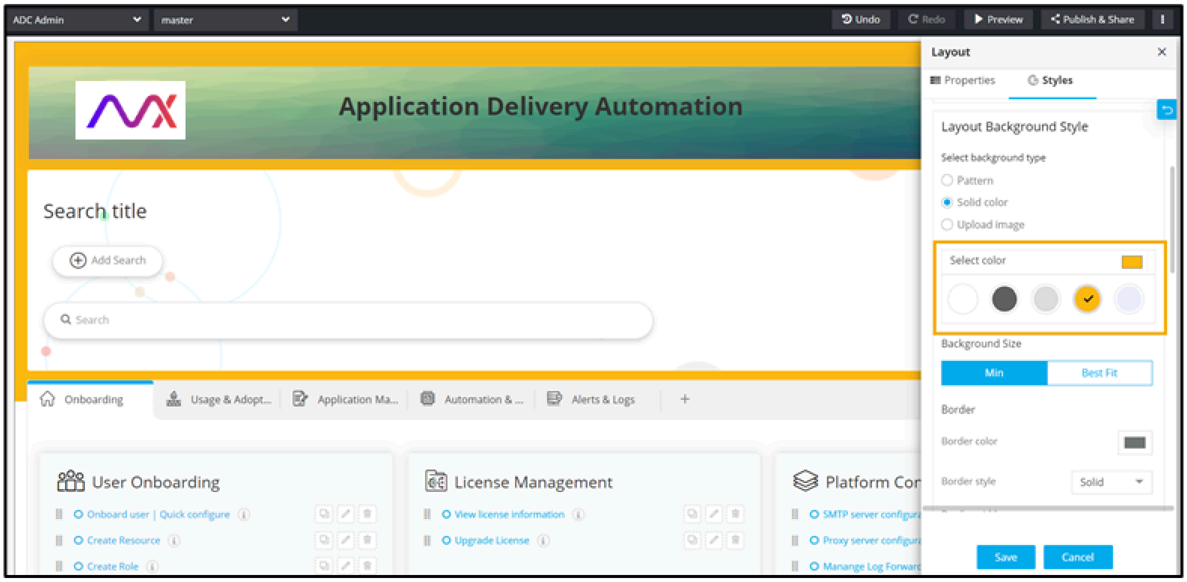


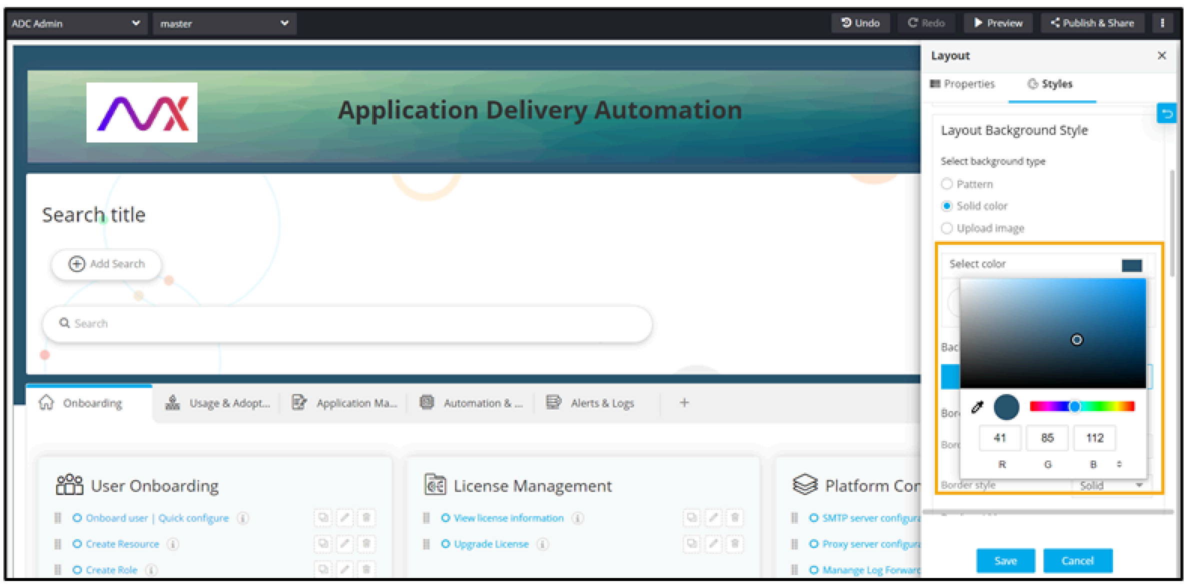
Note: Page margins are not applicable between individual widgets.

4. Under **Layout Background Style**, to change the background, **Select background type** from the available options.
   - **Pattern** - Select the **Pattern** option to choose a pattern from the options displayed.

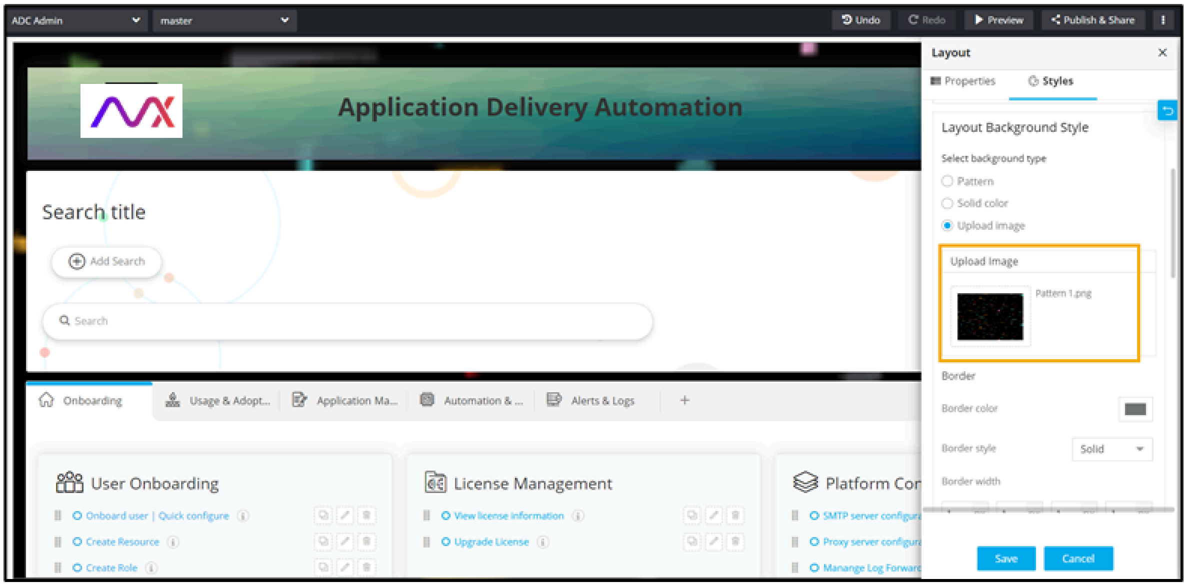


   - **Solid color** - Select the **Solid color** option to choose a solid color either from the options displayed or from the color picker.

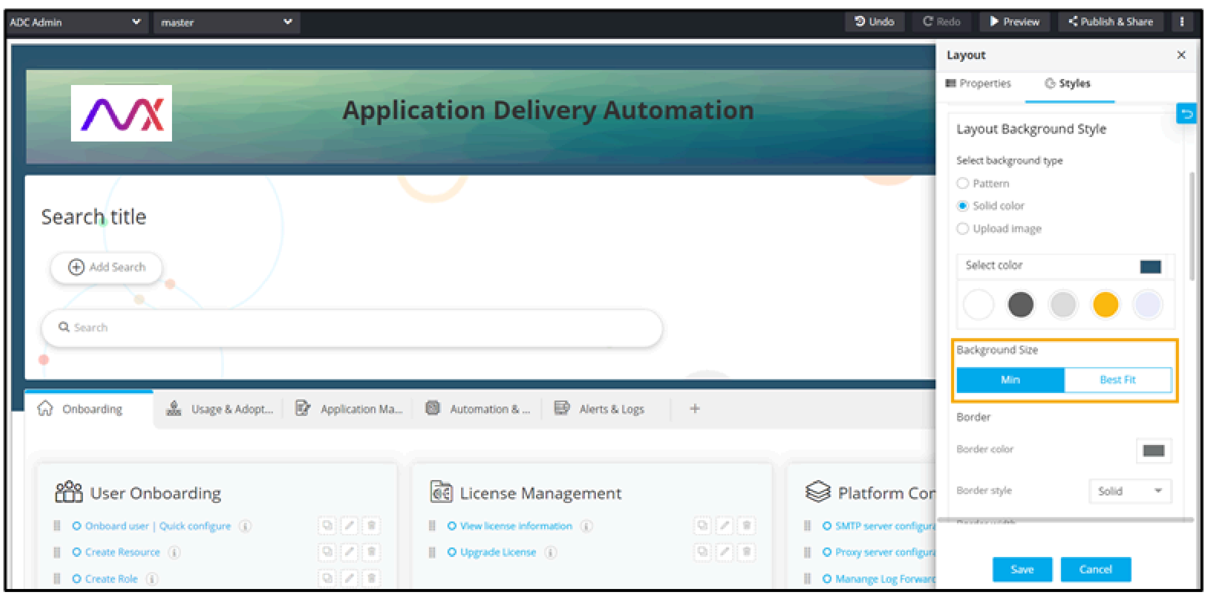

ADC Admin
master
Undo
Redo
Preview
Publish & Share
Application Delivery Automation
Search title
Add Search
Search
Onboarding
Usage & Adopt...
Application Ma...
Automation & ...
Alerts & Logs
User Onboarding
Onboard user | Quick configure
Create Resource
Create Role
License Management
View license information
Upgrade License
Platform Cor
Layout
Properties
Styles
Layout Background Style
Select background type
Pattern
Solid color
Upload image
Select color
Background Size
Min
Best Fit
Border
Border color
Border style
Solid
Save
Cancel


ADC Admin
master
Undo
Redo
Preview
Publish & Share
Application Delivery Automation
Search title
Add Search
Search
Onboarding
Usage & Adopt...
Application Ma...
Automation & ...
Alerts & Logs
User Onboarding
Onboard user | Quick configure
Create Resource
Create Role
License Management
View license information
Upgrade License
Platform Cor
Layout
Properties
Styles
Layout Background Style
Select background type
Pattern
Solid color
Upload image
Select color
41
85
112
R
G
B
Save
Cancel

- **Upload image** - Select the Upload image option to upload an image from your machine as the background image.



5. Under **Background Size**, to set the background color to only a part of the page, select **Min**.

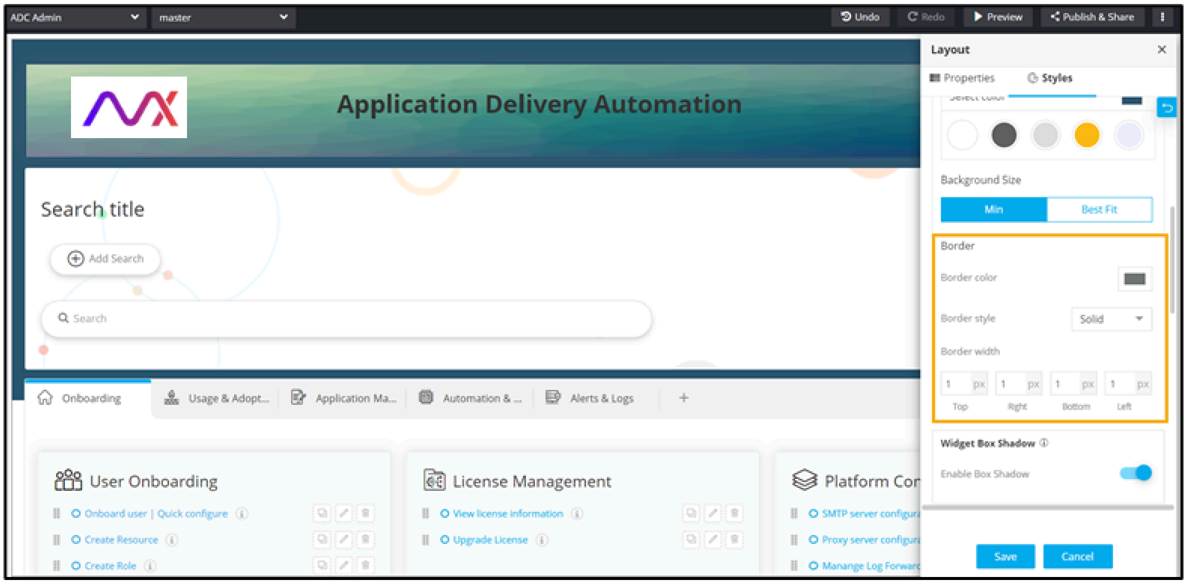


Note: This feature is applicable only when the background type is selected as **Solid color**.

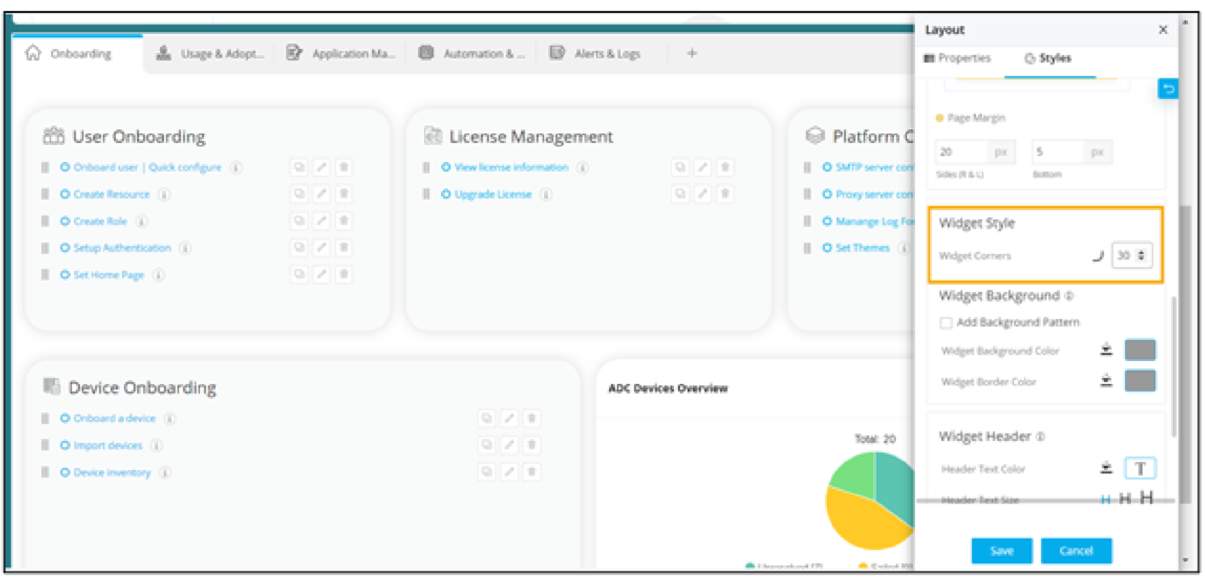
6. Under **Background Size**, to set the background color for the entire page, select **Best Fit**.

Note: This feature is applicable only when the background type is selected as **Solid color**.

7. Under **Border** section, you can customize the following:
   - **Border color -** Select the border color from the color picker. The selected color is applied to the page border.
   - **Border style -** Select the border style from the options displayed in the dropdown.
   - **Border width -** Select the width of the border at the top, right, bottom, and left of the page.



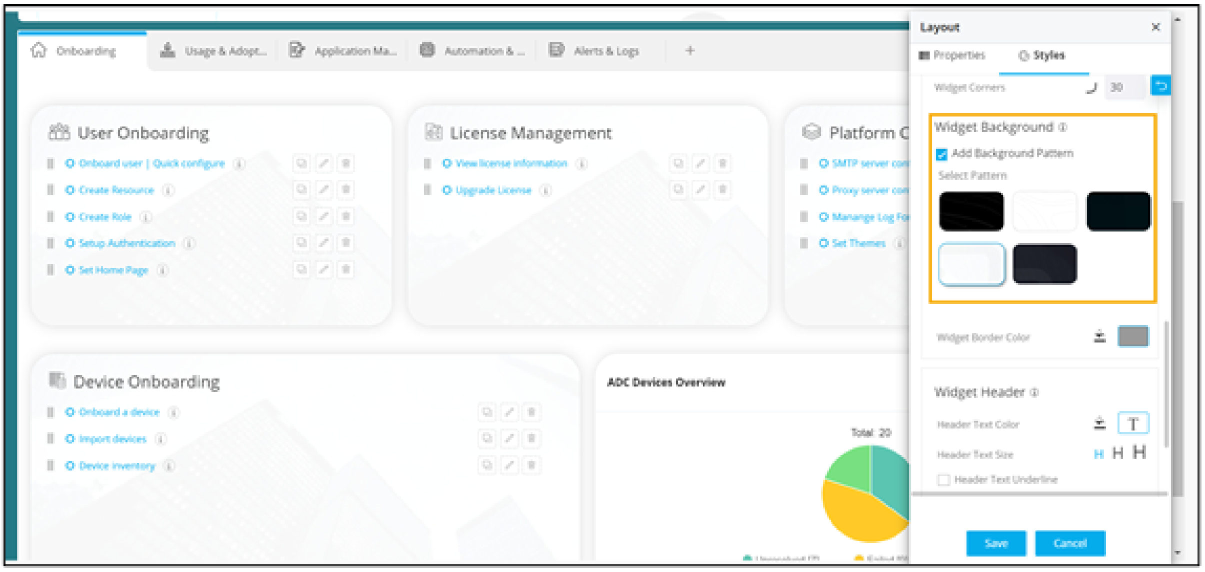
8. Under **Widget Box Shadow**, to remove the shadow for all widgets on the page, turn off the **Enable Box Shadow** toggle.
9. Under **Widget Style**, increase/decrease the curvature of the **Widget corners**.



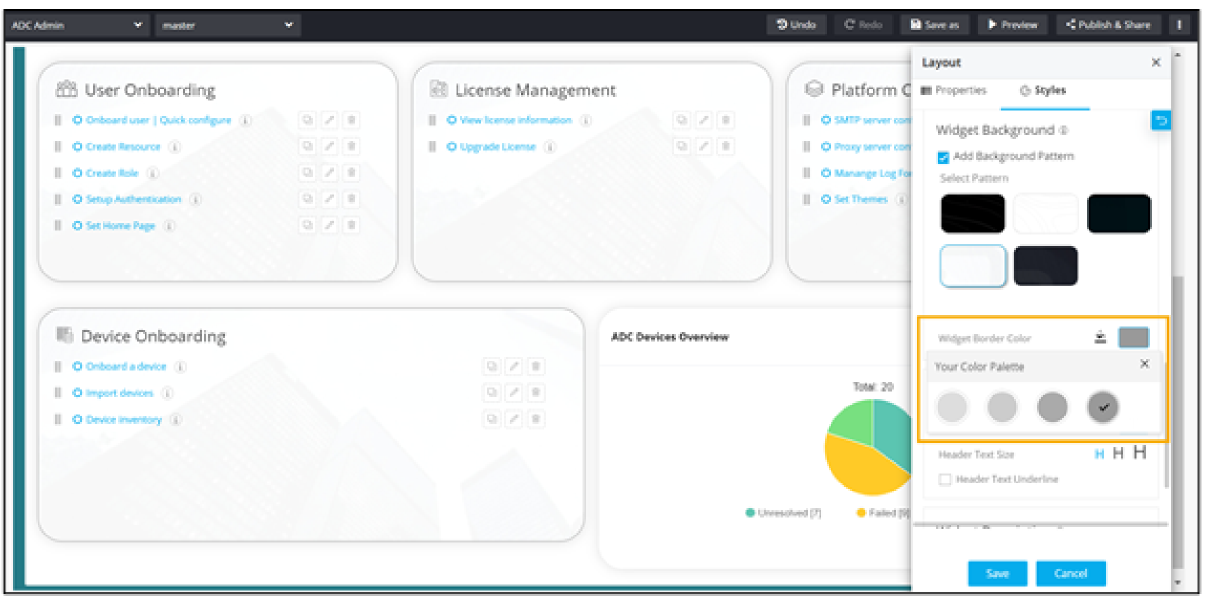
10. Under **Widget Background**, to add a background pattern, select the **Add Background Pattern** checkbox.

11. Select a pattern from the options available.

    The selected pattern is applied to all the widgets on the page.



12. Select the **Widget Border Color** from the color picker or the color palette.



13. Under **Widget Background**, select the **Widget Background Color** from the color picker or the color palette.

Note: The **Widget Background Color** option is displayed only when the **Add Background Pattern** checkbox is **not** selected.

14. Under the **Widget Background** section, select the **Widget Border Color** from the color picker or the color palette.
15. Under the **Widget Header** section, the following options are available:
    - **Header Text Color -** Select the **Header Text Color** from the color picker or the color palette.
    - **Header Text Size -** Select the **Header Text Size** from the available options.
    - **Header Text Underline -** Select the **Header Text Underline** checkbox to underline the Header Text.

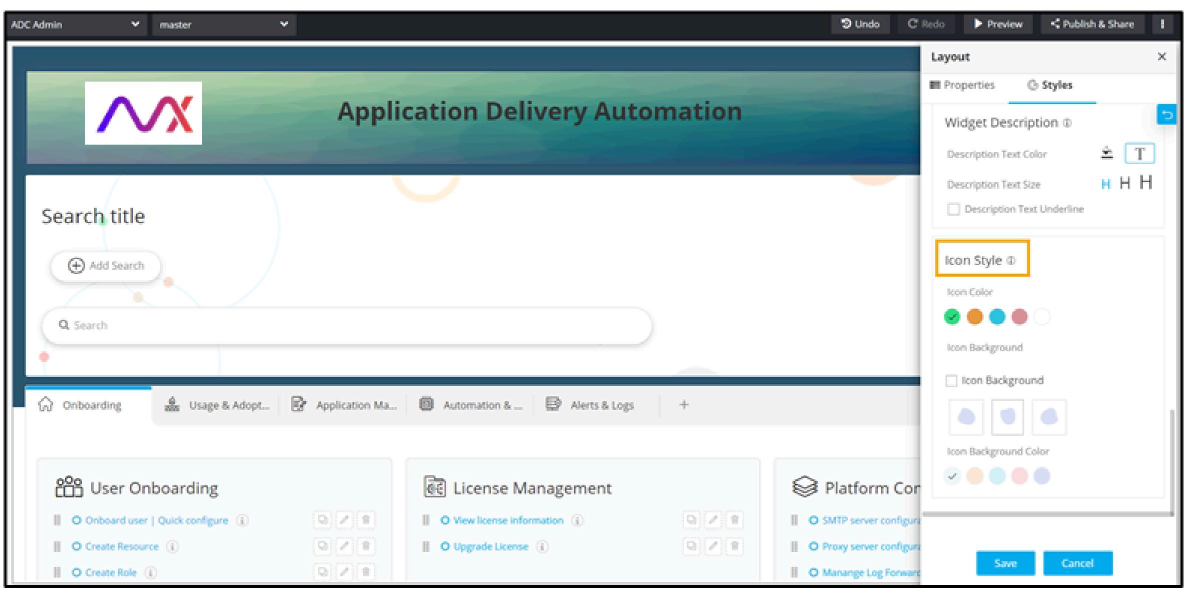
Note: Widget Header style changes are applicable to only those widgets that have a widget header.

16. Under the **Widget Description** section, the following options are available:

   - **Description Text Color -** Select the **Description Text Color** from the color picker or the color palette.
   - **Description Text Size -** Select the **Description Text Size** from the available options.
   - **Description Text Underline** - Select the **Description Text Underline** checkbox to underline the Header Text.

> **Note:** Widget Header style changes are applicable to only those widgets that have a widget header.

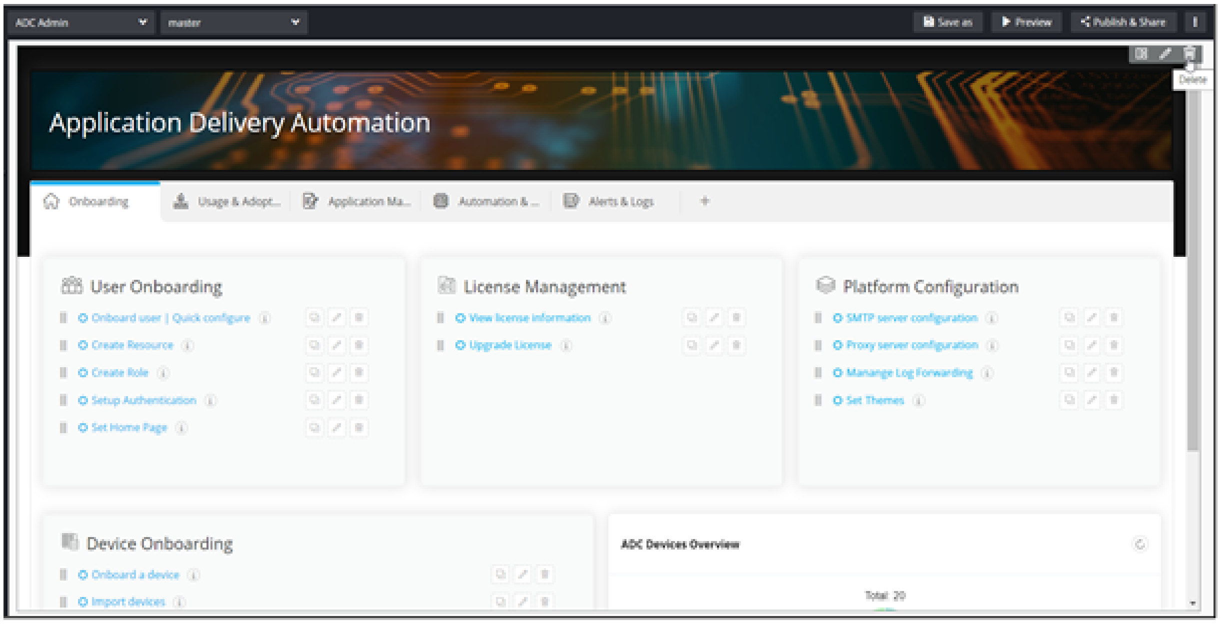
17. Under the **Icon Style** section, choose from the following available options:
    - **Icon Color** - Select the Icon Color from the available options.
    - **Icon Background** - Select the Icon Background from the available options.
    - **Icon Background Color** - Select the Icon Background Color from the available options.



Deleting a Catalog

You can delete the entire catalog in one go.

1. Form the Design section, open the page that is to be deleted.
2. Hover your mouse to the top right corner of the page and click **(Delete)** icon.



3. In the **Confirmation** pop-up window, click **Yes**.

   The catalog is deleted.

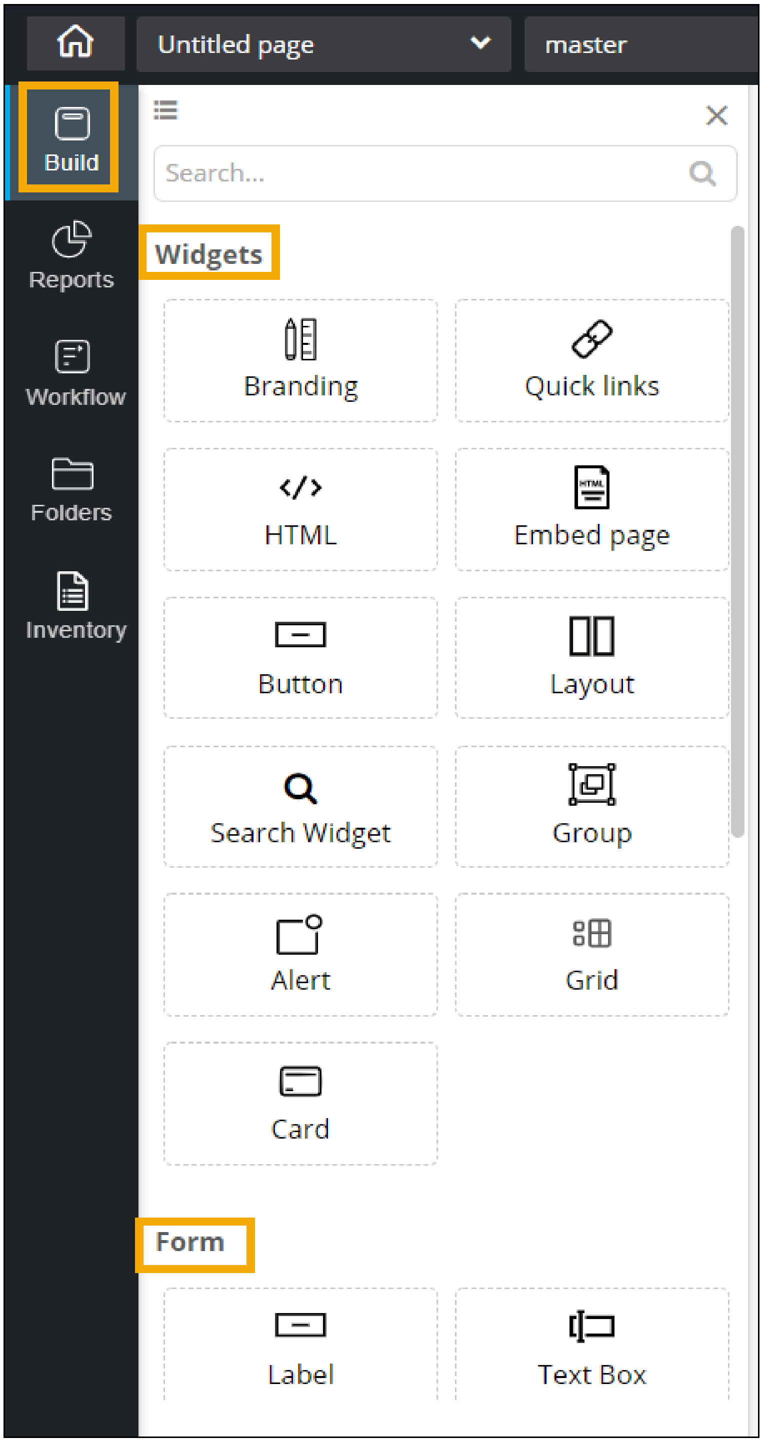
> **Important:** This action cannot be undone.

Build

The Service Catalog allows you to build your custom page by using the Build component. You can simply drag and drop components from this section to design a customized page.

1. Design a new page.
2. From the component panel on the left, click **Build**.

   A list of Page elements is displayed.



Options available and Description

Group	Option	Description
		Allows you to toggle between a list view and grid view of the components.

Group	Option	Description
	Search bar	Allows you to search for a particular component.
Widgets		
	Branding	Allows you to configure branding for your page.
	Quick links	Allows you to add a quick links widget to your page.
	HTML	Allows you to create your own HTML page within the Page.
	Embed page	Allows you to Embed a page into the Page.
	Button	Allows you to add buttons for redirecting to internal/external web pages from the catalog.
	Layout	Allows you to arrange widgets within a layout.
	Search widget	Allows you to add a search widget to the page.
	Group	Allows you to create groups for clubbing similar components/widgets together in the Page.
	Alert	Allows you to add and view Visual Workflow Alerts in the Page.
	Grid	Allows you to add a Grid component to your Page.
	Card	Allows you to add a card component with option to add multiple links, buttons and content.
Form		
	Label	Allows you to add a Label to the form in your Page.
	Text Box	Allows you to add a Text Area to the form in your Page.
	Password	Allows you to add a Password field to the form in your Page.
	Date Time Picker	Allows you to add a Date Time Picker field to the form in your Page.
	Text Area	Allows you to add a Multi-select Box to the form in your Page.
	Radio Button	Allows you to add a Radio button to the form in your Page.
	Checkbox	Allows you to add a Checkbox to the form in your Page.
	Select Box	Allows you to add a Select Box to the form in your Page.

Group	Option	Description
	Multi-select Box	Allows you to add a Multi-select Box to the form in your Page.
	Form Button	Allows you to add a Form Button to the form in your Page.
	Download	Allows you to add a Download field/link to the form in your Page.
	File Upload	Allows you to add a File Upload field to the form in your Page.
	Script Editor	Allows you to add a Script Editor to the form in your Page.
	Tabular	Allows you to add a tabular feature to the form in your Page.
	Form Group	Allows you to define a Form Group to your form in the Page.
	Rich Text Editor	Allows you to add a text editor to the form in your Page.

- Widgets

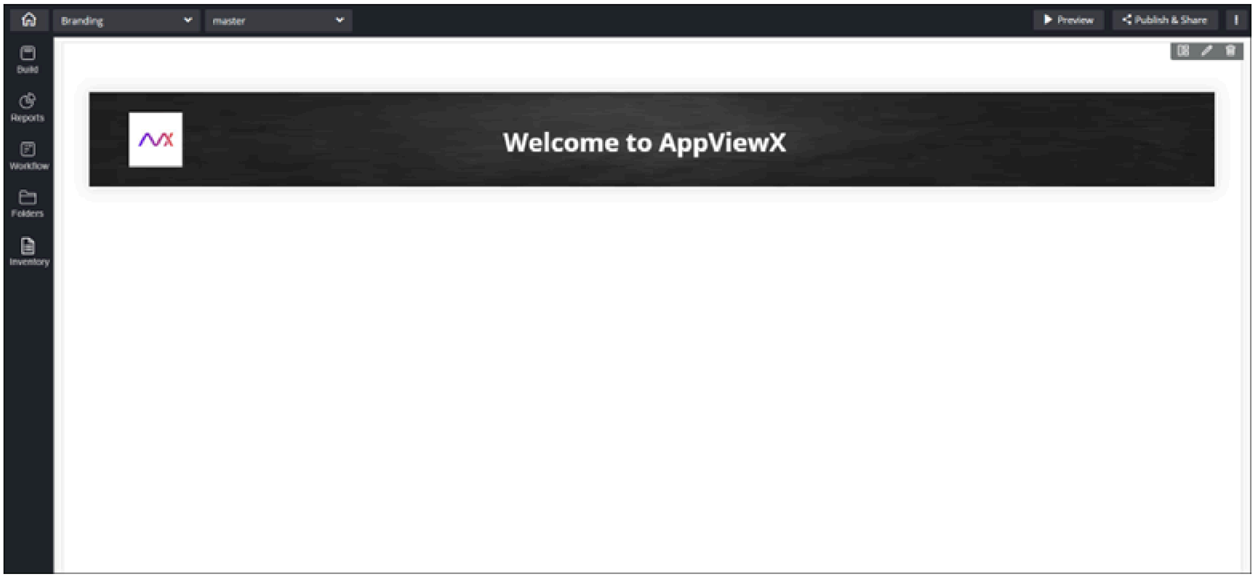
Widgets

You can add different elements to your page in the form of widgets, configure them, and customize them as per your requirement.

- Branding
- Quick links
- HTML
- Embed page
- Button (Action Widget)
- Layout
- Search widget
- Group
- Alert
- Grid
- Card

Branding

You can add branding such as the company logo or any other header on your page using the Branding component.



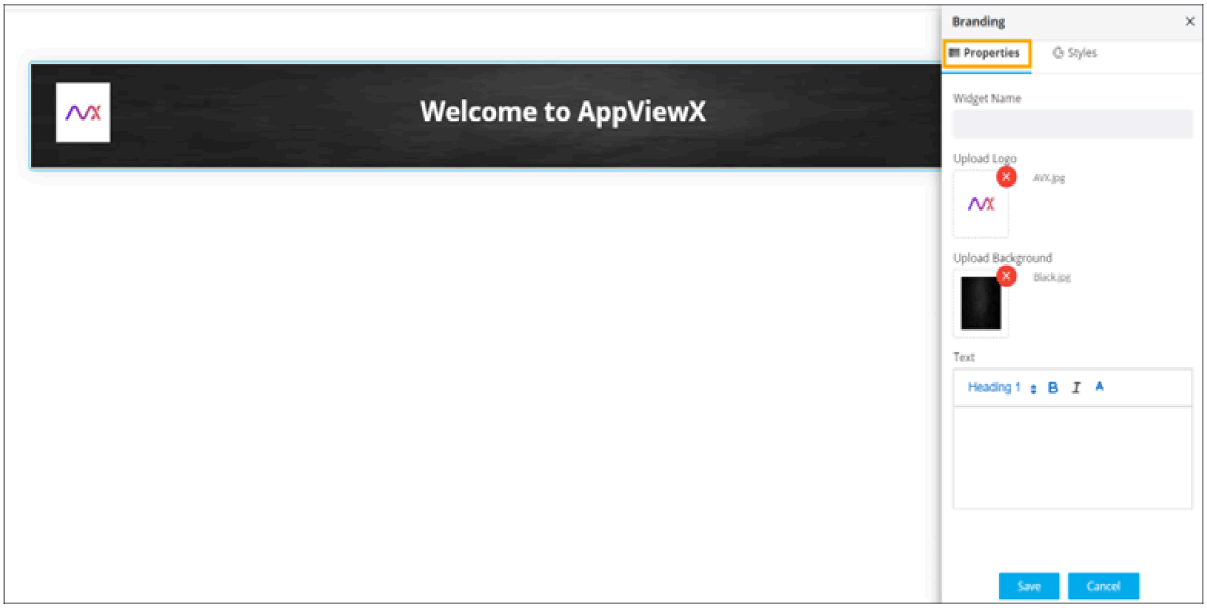
• Configuring Branding - Properties

• Configuring Branding - Styles

Configuring Branding - Properties

1. From the **Build** component in the left menu, under the **Widgets** section, drag and drop the **Branding** widget to any blue highlighted space on the canvas.
2. To configure the Branding widget, hover your mouse over the widget and click **(Edit)** icon.

   The **Branding** pop-up window opens with the **Properties** tab open by default.
3. In the **Branding** window, under **Properties**, enter or select the required fields.

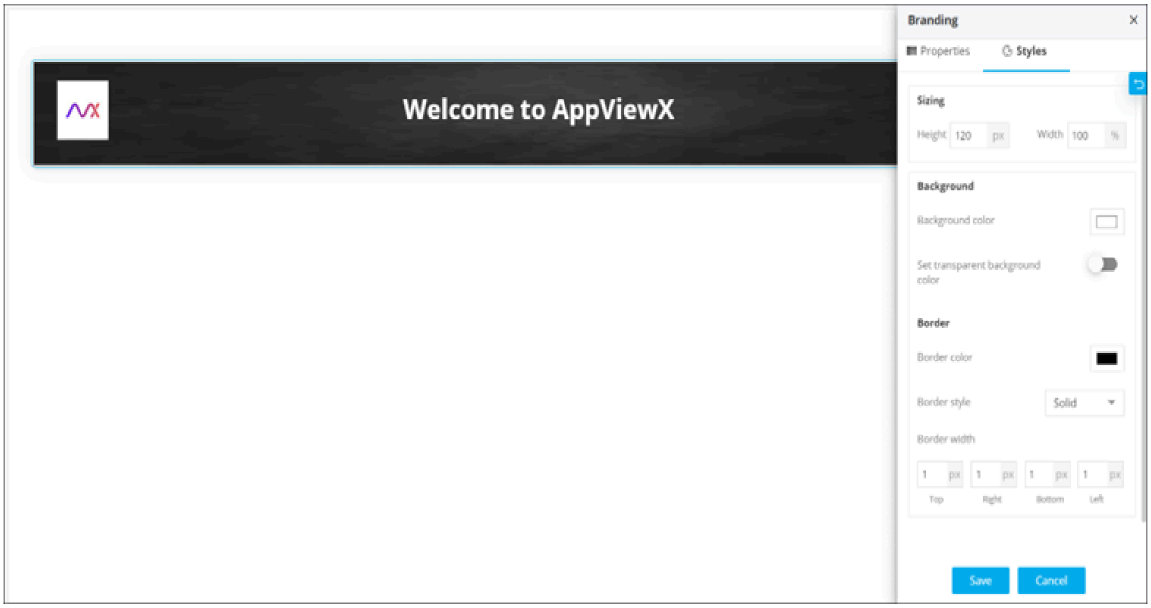


Field descriptions for Properties

Field	Description
Widget Name	Enter a suitable widget name. For example, AppViewX.
Upload Logo	Upload the appropriate logo, if required.
Upload Background	Upload a suitable background.
Text	Enter the text that will be displayed in the Branding widget, next to the logo.

Configuring Branding - Styles

1. For more options to customize the widget, in the **Branding** window, click **Styles**.



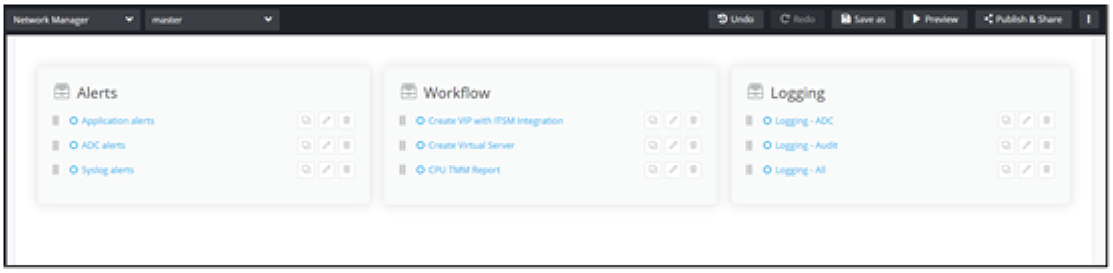
Field descriptions for Styles

Option	Description
Sizing	Allows you to increase/decrease the height and width of the widget.
Background	The following options are available here: • **Background color**: Allows you to select a background color for the branding widget. • **Set transparent background color**: To set the background color as transparent, turn on the toggle.
Border	The following options are available here: • **Border color**: Allows you to select the border color. • **Border style**: Allows you to select the border style from the options available in the dropdown. • **Border width**: Allows you to adjust the width of the border.

2. Click **Save**.
3. Drag and drop the background image and text to move it to any position within the widget and resize the company logo.
4. To restore settings to default, click **(Restore)** icon.

Quick links

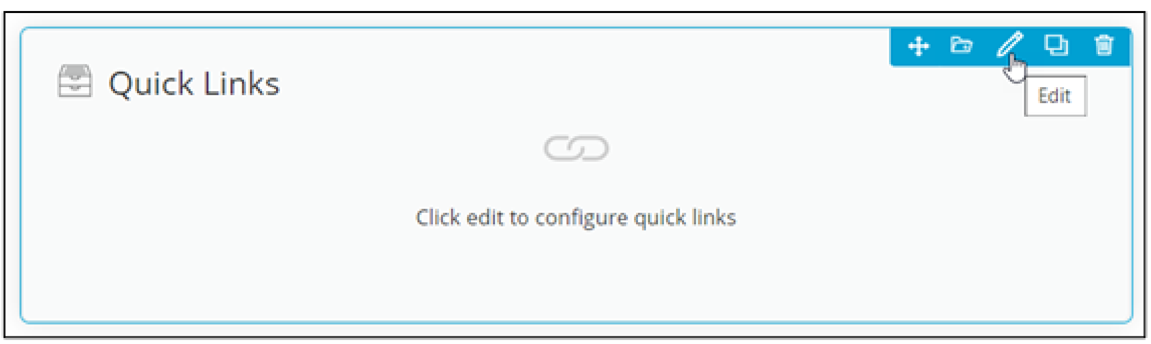
You can add Quick links to your page for redirecting end users to specific features within AppViewX (ADC, Certificates, Reports etc.) and outside AppViewX (Jira etc). Clicking on a Quick link within the widget will redirect you to a new tab.



- Configuring Quick links - Properties
- Adding a Workflow Quicklink
- Adding an Internal Quicklink
- Adding Other/External Quicklinks
- Configuring Quicklinks - Styles
- Editing and Deleting Quicklinks

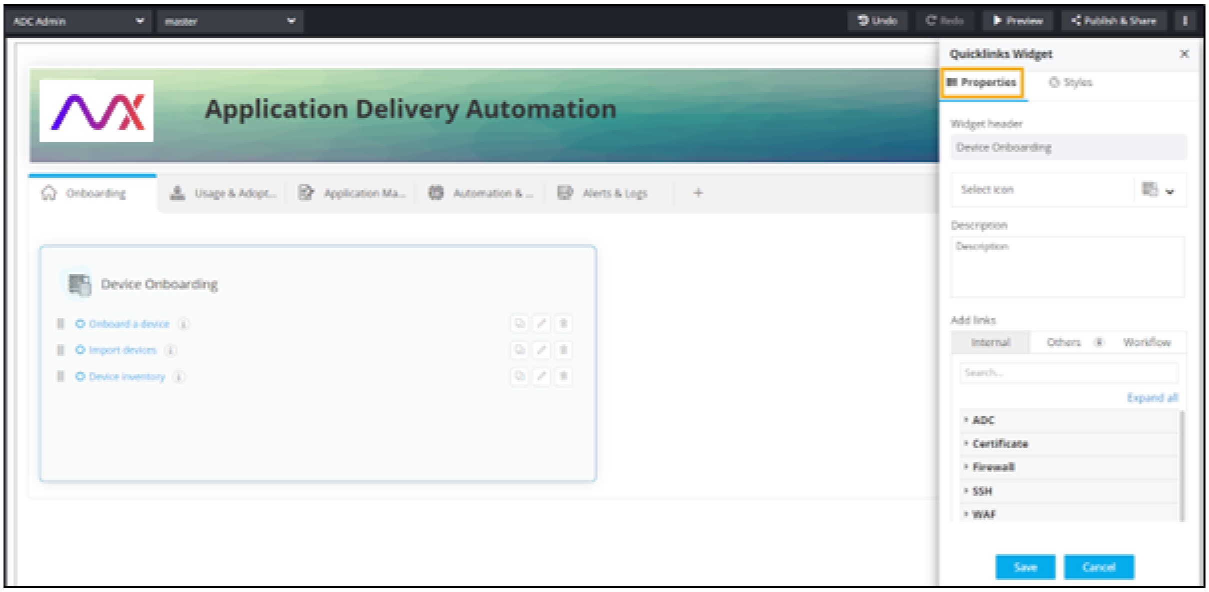
Configuring Quick links - Properties

1. From the **Build** component in the left menu, under the **Widgets** section, drag and drop the **Quick links** widget to any blue highlighted space on the page.
2. To configure the Quick links component, hover your mouse over the widget and click  **(Edit)** icon.



The **Quicklinks Widget** pop-up window opens with the **Properties** tab open by default.

3. In the **Quicklinks Widget** window, under **Properties**, enter or select the required field information.



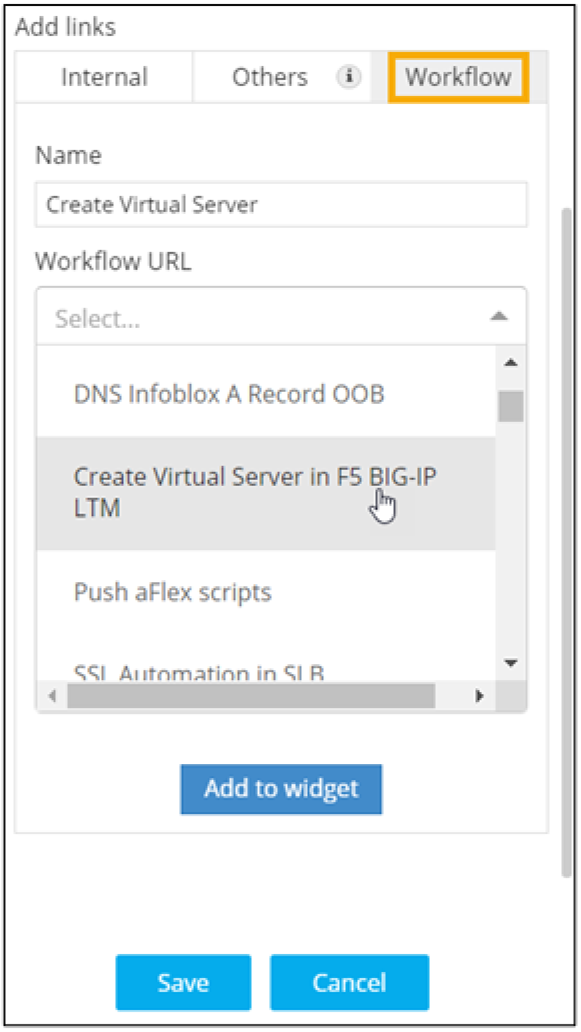
Field descriptions for Properties

Field	Description
Widget Header	Provide a name for the Widget Header. For example: **Quick Links**.
Select icon	Select an icon for the Quick Links widget from the dropdown.
Add Links	You can add three types of links here: • **Internal**: Add an internal link from within AppViewX pages. • **Others**: Add links to other pages within AppViewX or embed external cross origin enabled pages. • **Workflow**: Add a workflow link which redirects to the workflow execution page.

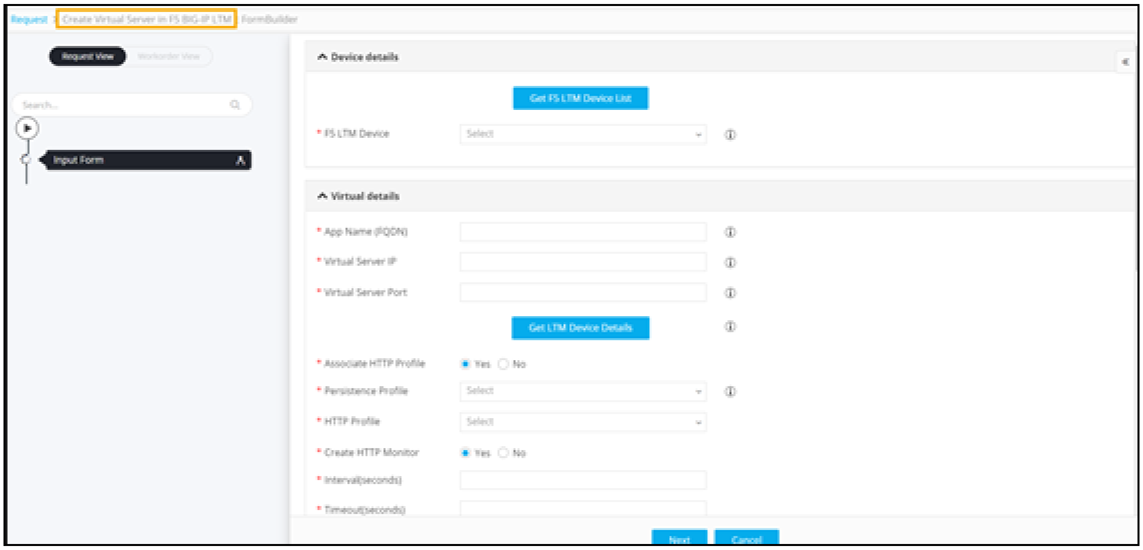
4. Click **Save**.

Adding a Workflow Quicklink

1. To add a link to redirect to the Workflow Request execution page for the selected workflow, under **Add links**, click **Workflow**.
2. Enter the link **Name** and select the **Workflow URL** from the dropdown.



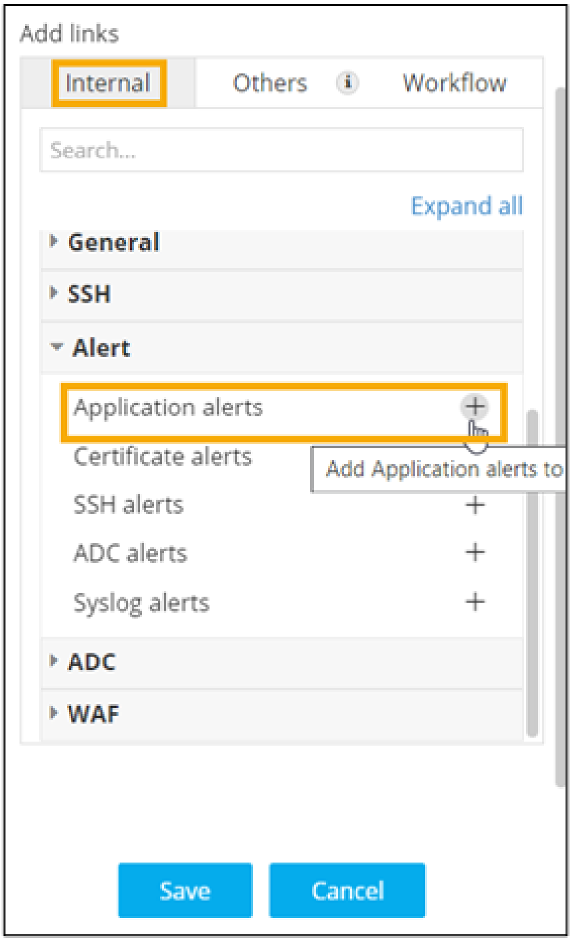
3. To add this workflow URL to the Quick Links widget, click **Add to Widget.**
   Clicking on the quick link will redirect to the workflow request execution page of the selected workflow.



4. Once you have added the required workflow URLs to the widget, click **Save**.
5. For more options to customize the widget, click here.

Adding an Internal Quicklink

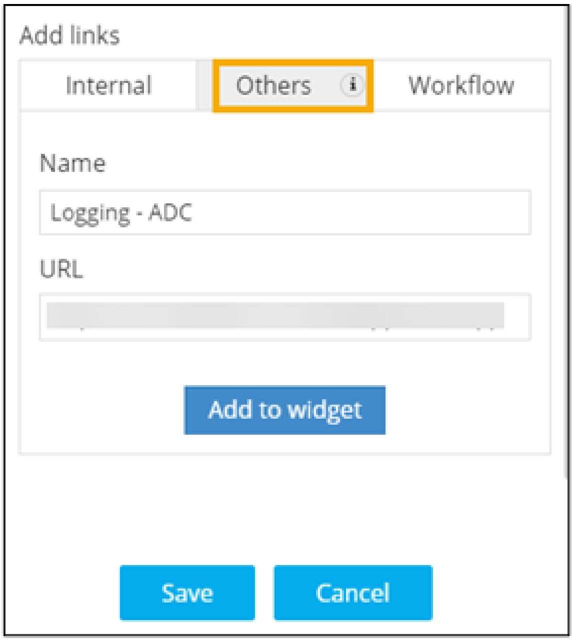
1. To link an internal AppviewX page to the widget, under **Add links**, click **Internal**.
2. To select a link to be added to the widget, click + **(Add)** icon.



3. To add selected links to the widget, click **Save**.
   Clicking on the quick link will redirect to the selected page in a new tab.
4. For more options to customize Quicklinks, click here.

Adding Other/External Quicklinks

1. To embed an external link or a cross origin enabled page URL into the Quick link widget, under **Add links**, click **Others**.
2. Enter the **Name** and **URL** for the link.

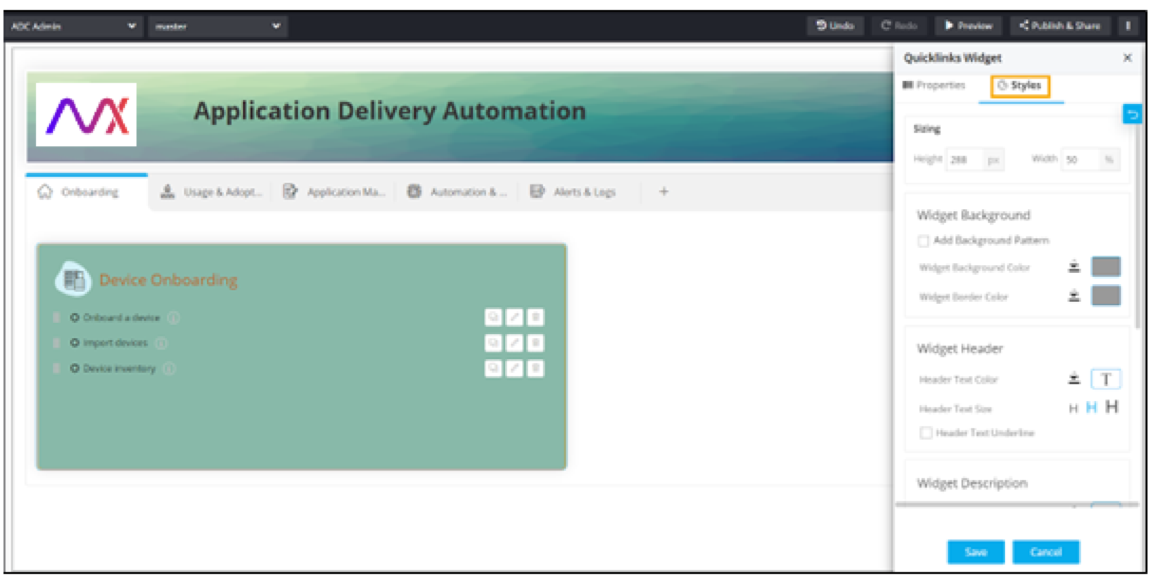


3. To add this external link to your widget, click **Add to Widget**.

   Clicking on the quick link will redirect to the configured external page.
4. For more options to customize Quicklinks, click here.

Configuring Quicklinks - Styles

1. For more options to customize the widget, in the **Quicklinks Widget** window, click **Styles**.



Field descriptions for Styles

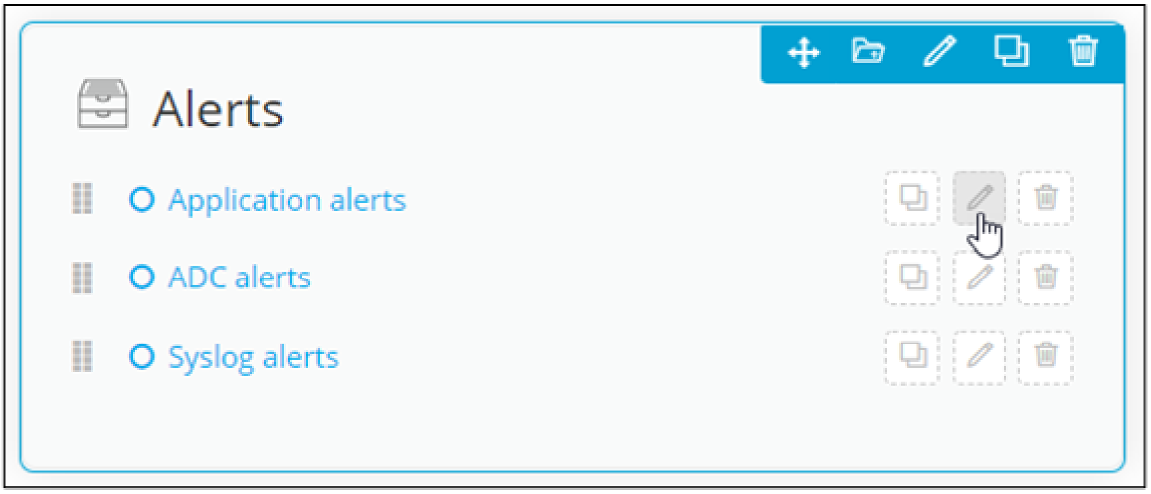
Field	Description
Sizing	Allows you to increase/decrease the height and width of the widget.
Widget Background	The following options are available here: • **Add Background Pattern**: To add a background pattern to the widget, select the **Add Background Pattern** checkbox and select a pattern from the available options. • **Widget Background Color**: To add a background color to the widget, unselect the **Add Background Pattern** checkbox and select the color from the color palette. • **Widget Border Color**: Allows you to change the color of the widget border.
Widget Header	The following options are available here: • **Header Text Color**: Allows you to change the color of the header text. • **Header Text Size**: Allows you to select the size of the header text. • **Header Text Underline**: To underline the header text, select the Header Text Underline checkbox.
Widget Description	The following options are available here: • **Description Text Color**: Allows you to change the color of the label text. • **Description Text Size**: Allows you to select the size of the label text. • **Description Text Underline**: To underline the description text, select the **Description Text Underline** checkbox.
Icon Style	The following options are available here: • **Icon Color** - Select the Icon Color from the available options. • **Icon Background** - Select the Icon Background from the available options. • **Icon Background Color** - Select the Icon Background Color from the available options.

2. To save your settings, click **Save**.
3. To restore the settings to default, click  **(Restore)** icon.

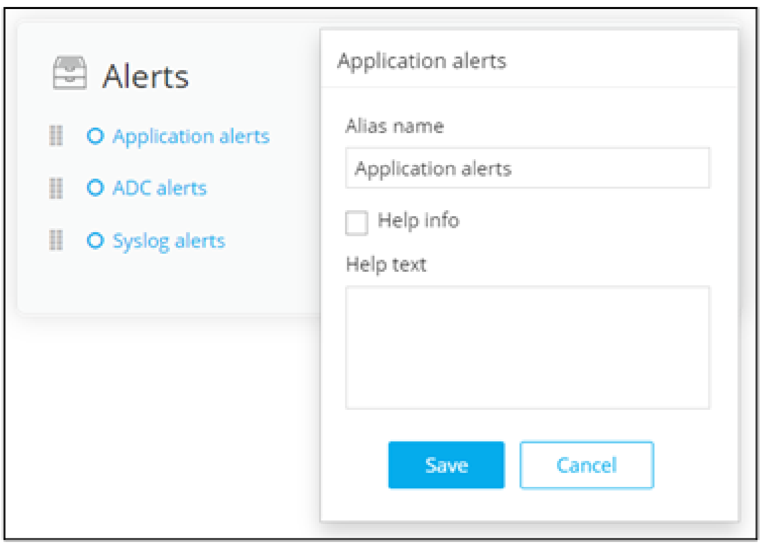
Editing and Deleting Quicklinks

You can edit individual quick links to change the name displayed on the widget, add help text to quick links, and also delete quick links from the widget.

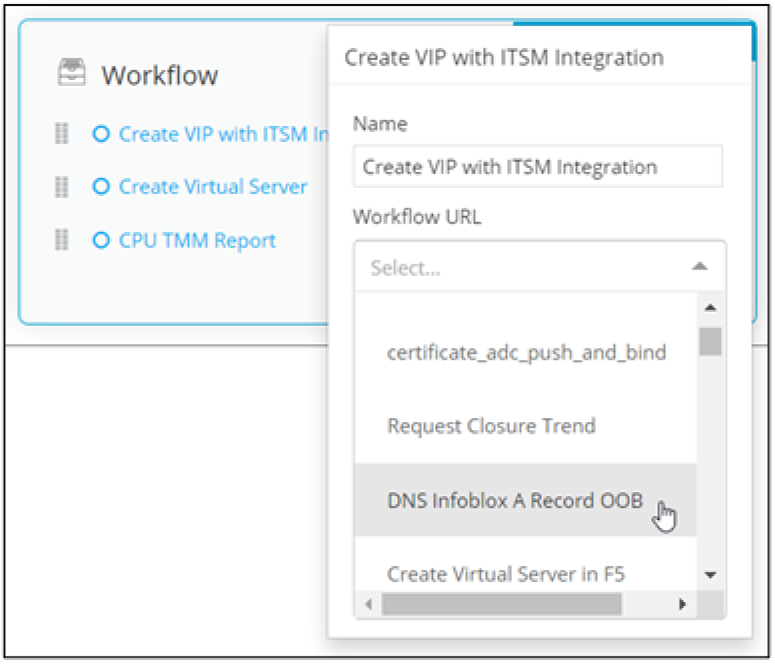
1. To edit an internal or external Quick link click **(Edit)** icon.



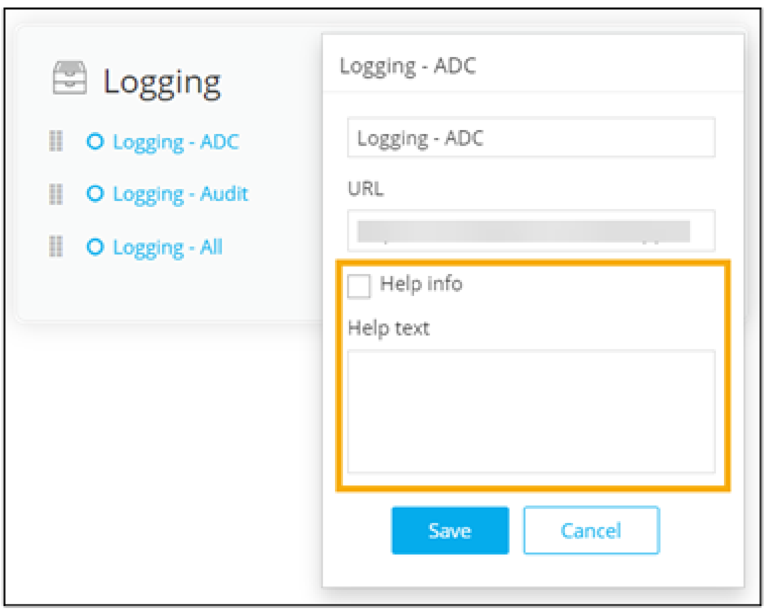
2. Enter the **Alias name** in the quick link pop-up window.



3. To edit the Workflow quick link, click **(Edit)** icon.
4. In the workflow quick link pop-up window, edit the workflow name and select a different workflow URL from the dropdown list.

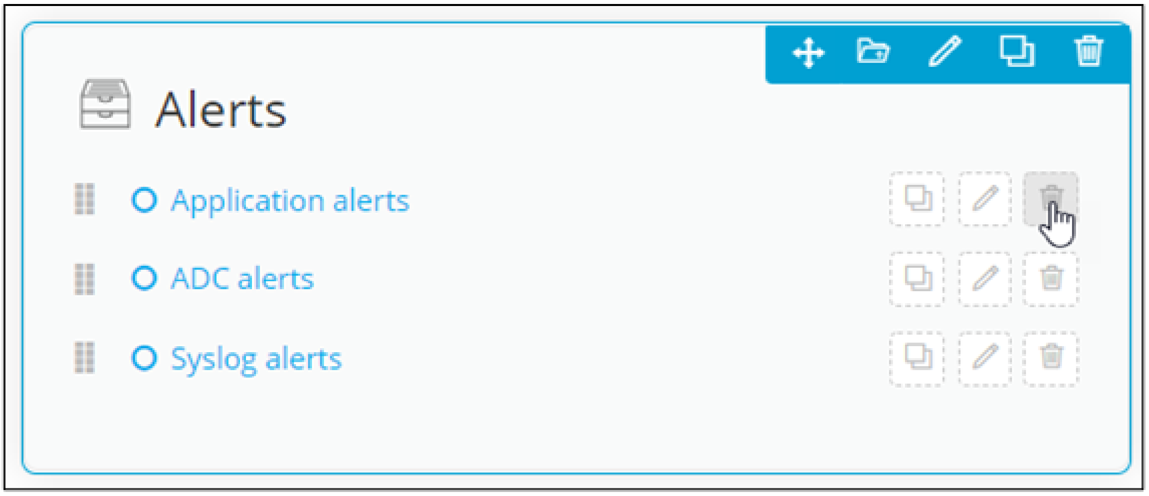


5. To display the help text when you hover your mouse over a quick link, select the **Help info** checkbox and add the text to be displayed.

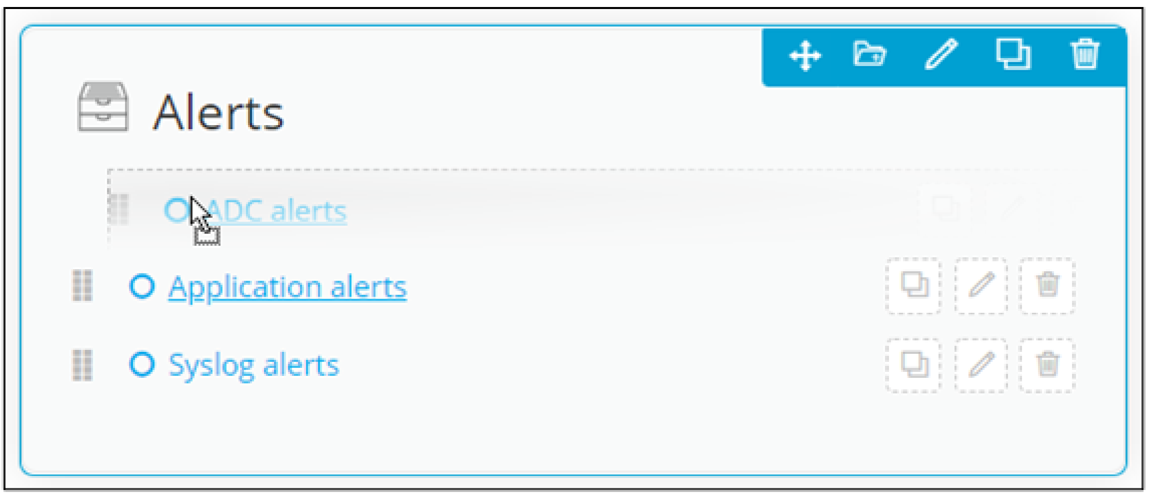


6. Click **Save**.

7. To delete a quick link, click **(Delete)** icon and click **Ok** in the **Confirmation** window.



8. To rearrange the order in which the quick links are displayed in the widget, hold and drag the quick link where required.

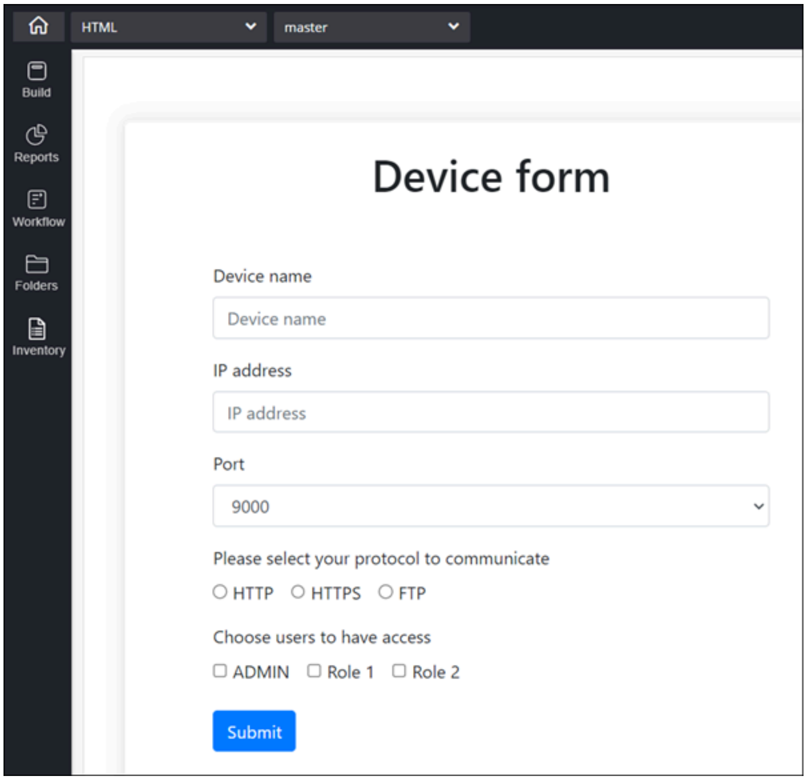


9. To make a copy of a quick link, click **(Clone)** icon.

HTML

You can design customized HTML pages based on specific user needs.

• Create customized HTML page using HTML5 syntax
• Add third party web plugins
• Add own style components and interactive functionalities using CSS and JS
• Use Hooks to design custom HTML pages.



- Configuring HTML Widget - Properties
- Creating custom HTML pages using plugins
- Creating custom HTML pages using sample HTML script
- Creating custom HTML pages using Hook Inventory
- Configuring HTML Widget - Styles

Configuring HTML Widget - Properties

1. From the **Build** component in the left menu, under the **Widgets** section, drag and drop the **HTML** widget to any blue highlighted space on the page.
2. To configure/edit the HTML widget, hover your mouse over the widget and click **(Edit)** icon.

The **Html widget** pop-up window opens with the **Properties** tab open by default. You can now create HTML pages using:

- Third-party Plugins
- Prebuilt Samples
- Hooks.

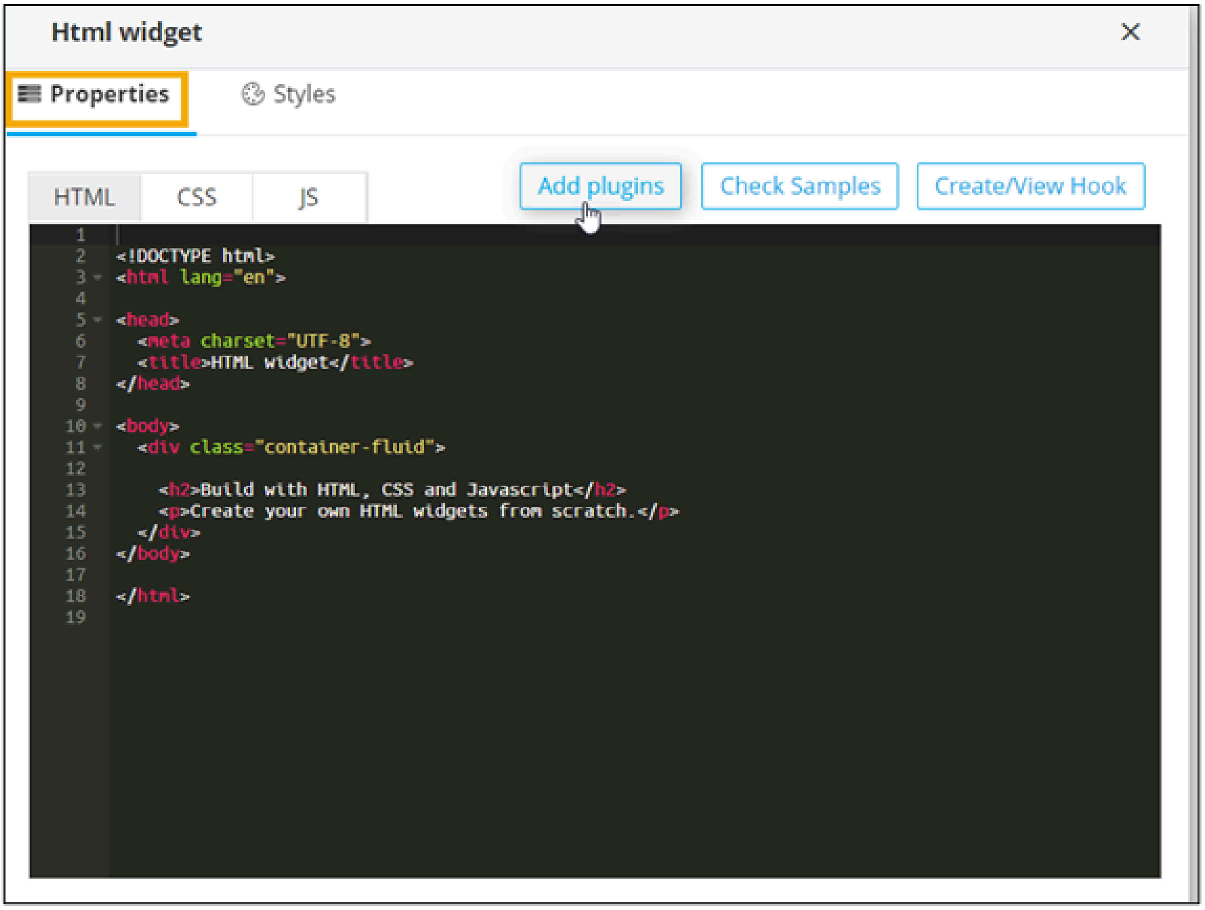
Creating custom HTML pages using plugins

You have the option of adding third party web plugins to the HTML widget. The following plugins are supported natively:

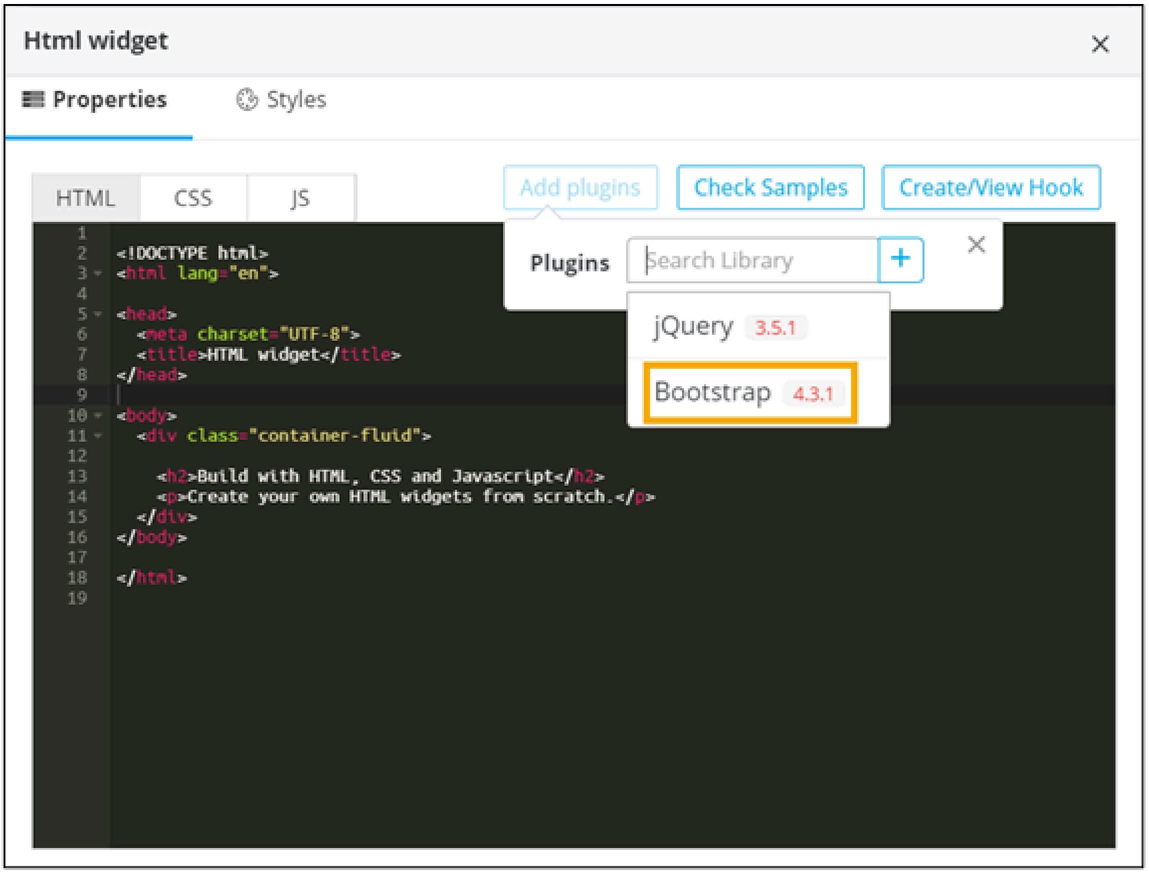
- jQuery
- Bootstrap.

To add plugins to your HTML page:

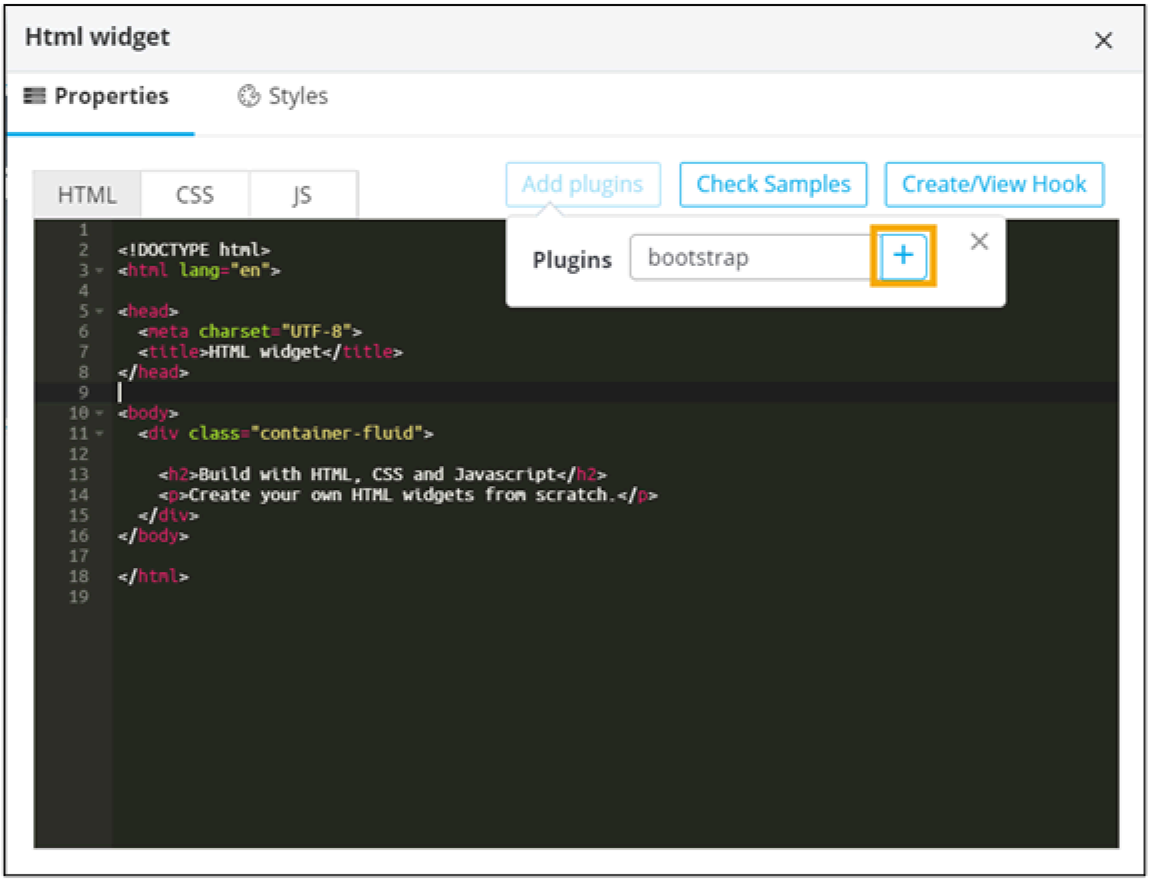
1. In the **HTML widget** window, under **Properties**, click **Add plugins**.



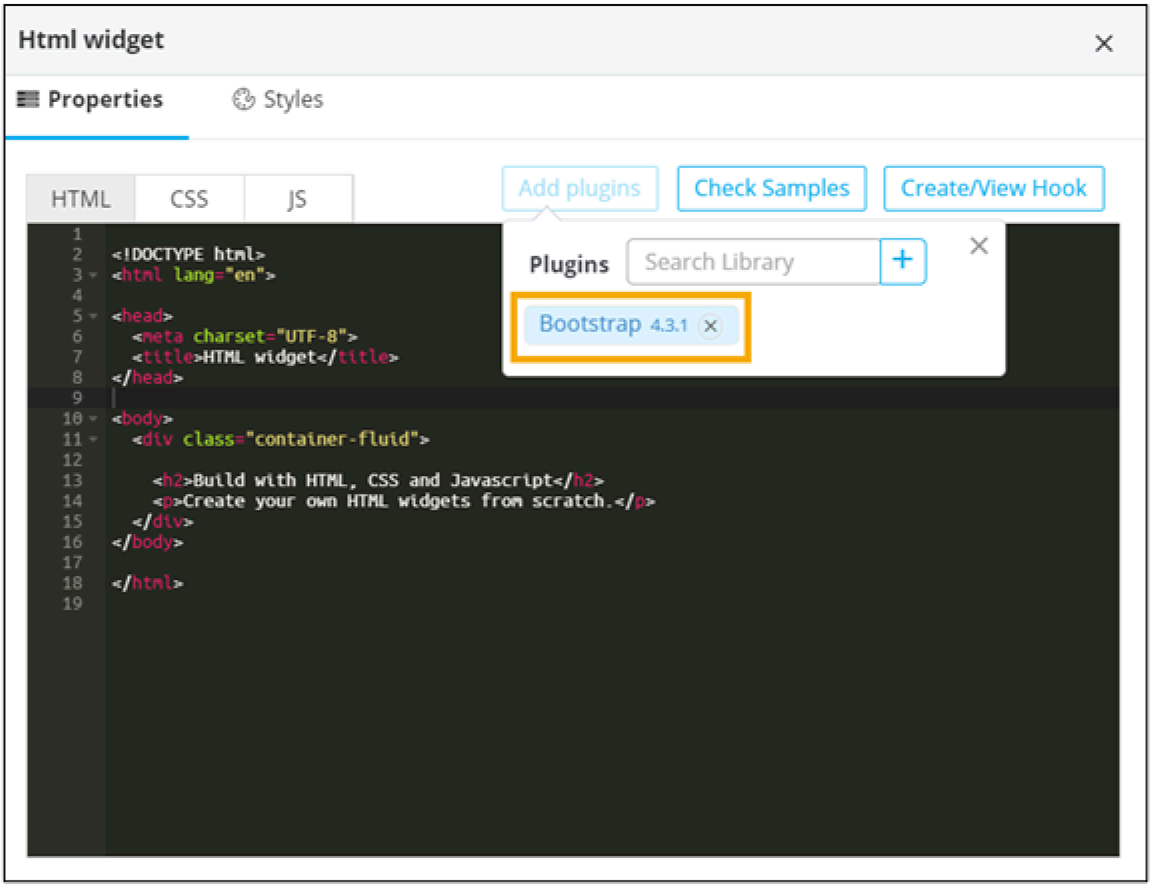
2. Select the library from the available options. For example, **Bootstrap**.



3. To add the selected plugin, click  **(Add)** icon.



4. To see the available design templates, click **Bootstrap**.



The Bootstrap library opens in a new tab. You can customize any of their available templates for your HTML page.

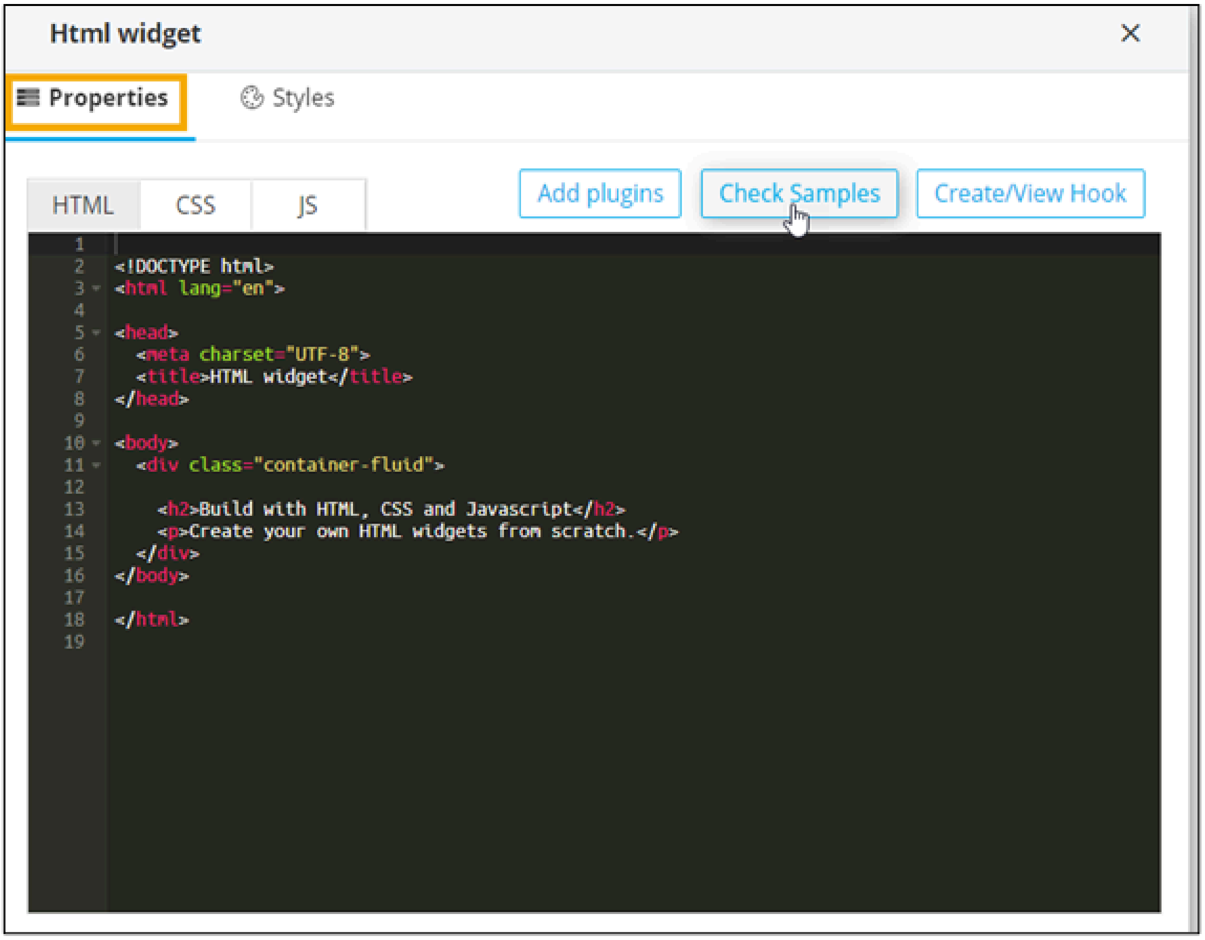
5. To save your selection, click **Save**.

Creating custom HTML pages using sample HTML script

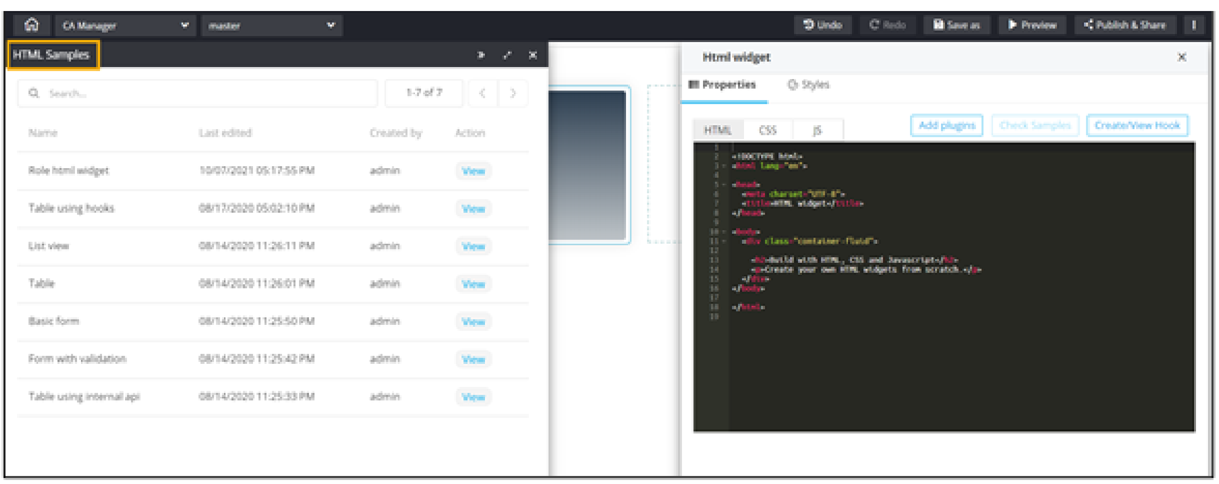
You can also create an HTML widget in your Page using prebuilt HTML scripts provided within the Page Builder.

To create a HTML page using sample script:

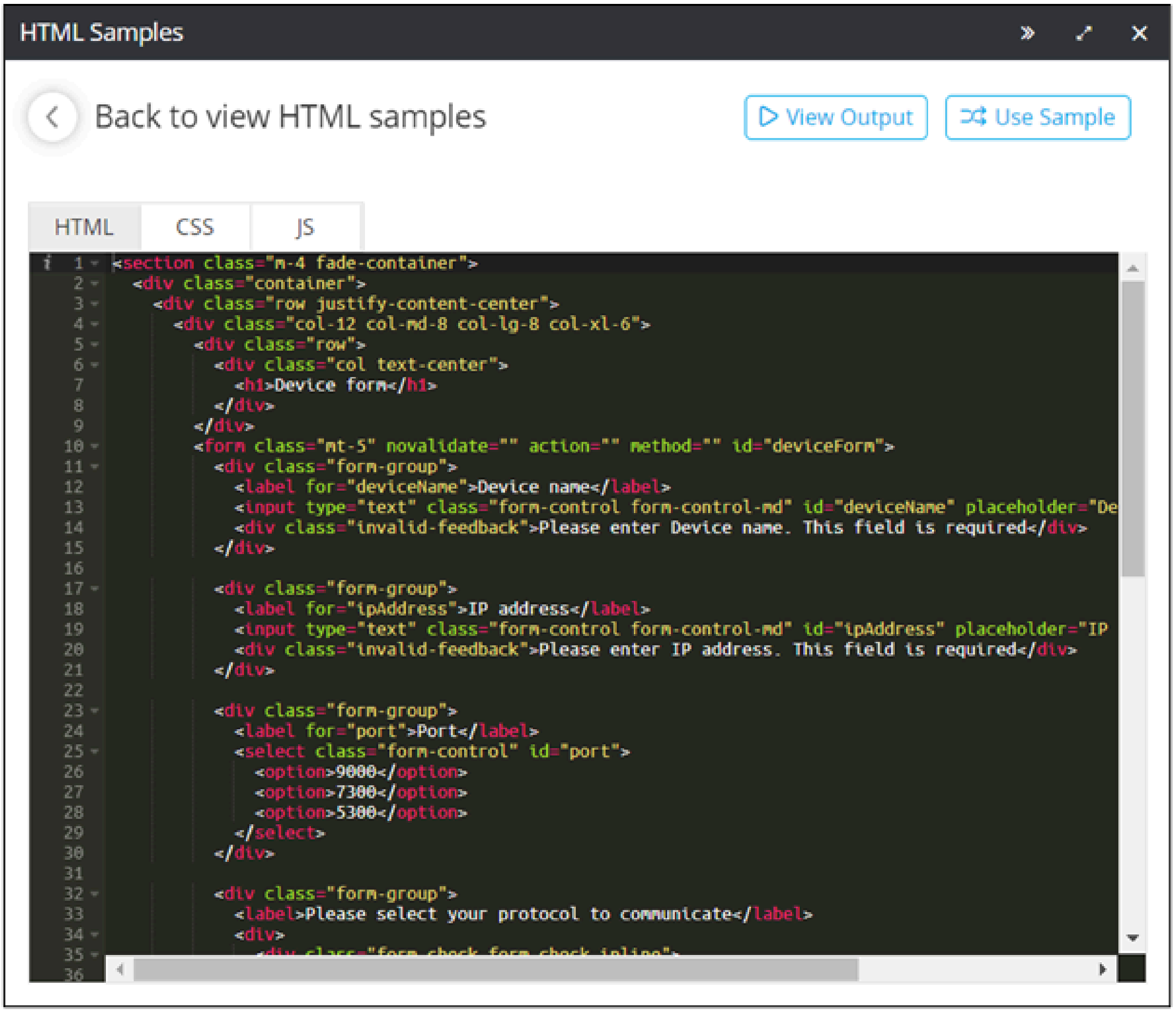
1. In the **HTML widget** window, click **Check Samples**.



A list of **HTML Samples** is displayed on the left side of the screen.

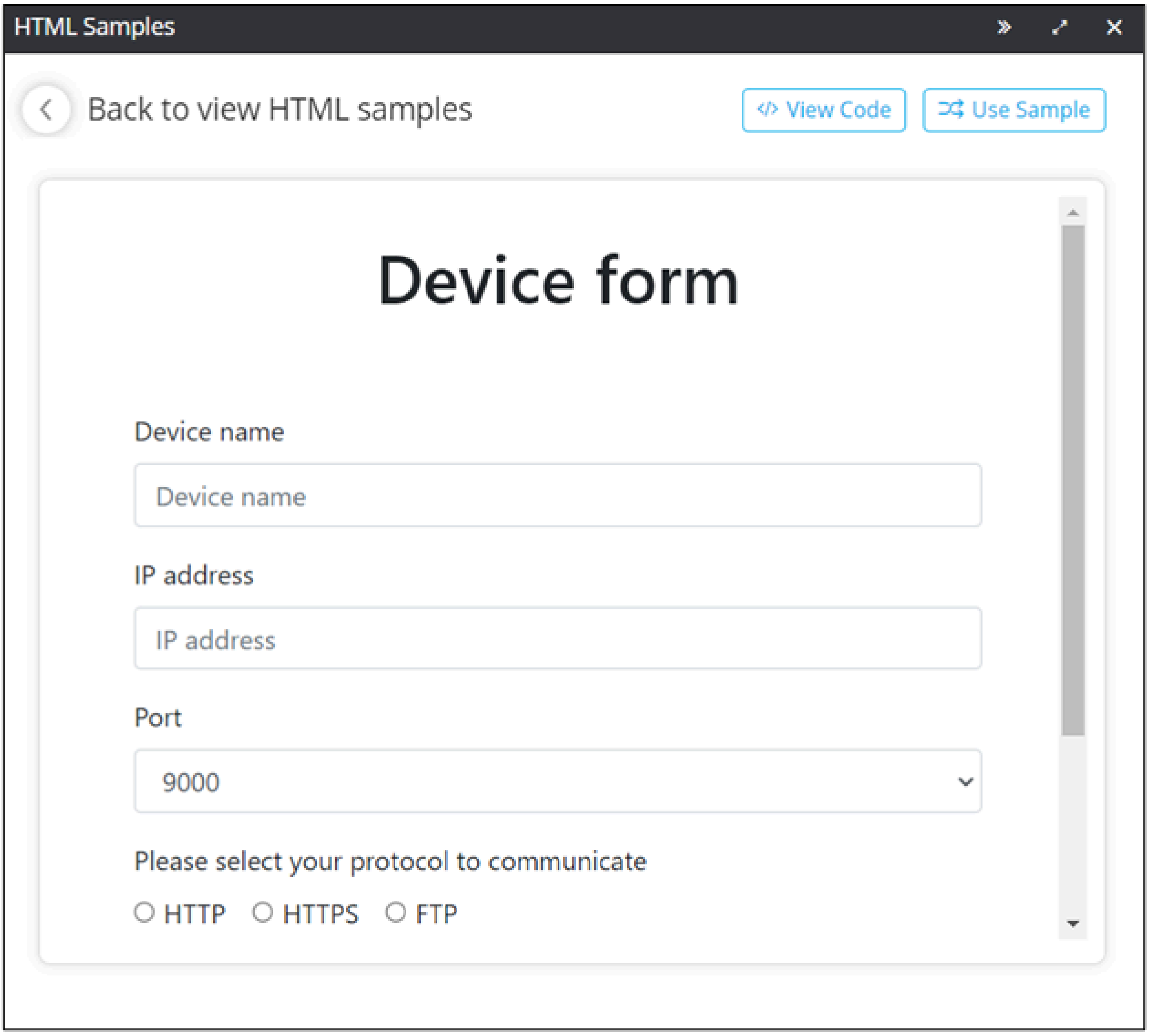


2. To see the details of a sample script, in the **HTML Samples** window, click **View** next to the sample script you wish to view. For example, Basic form.
   The details of the sample script as displayed in the same window.



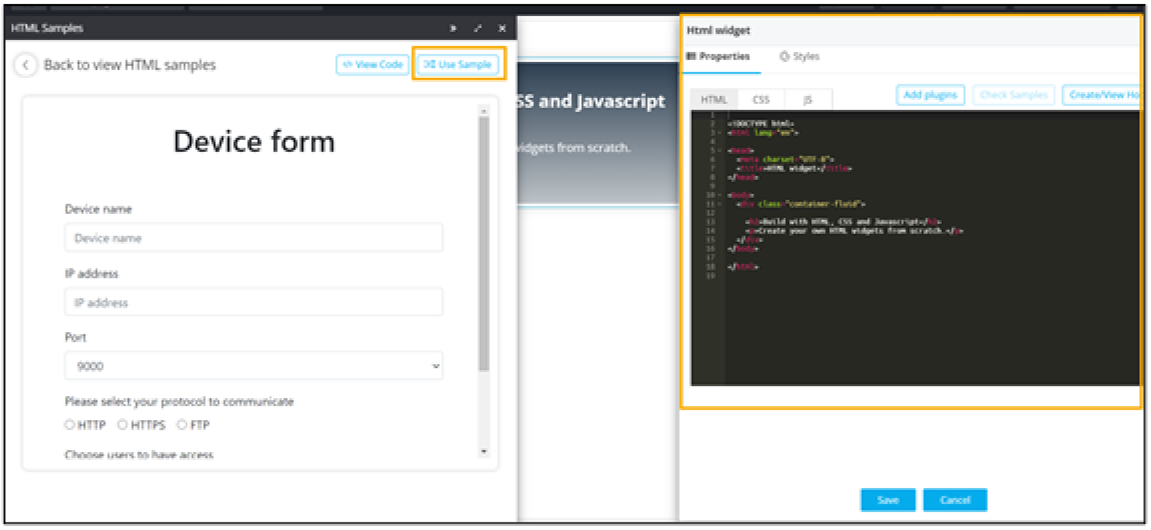
3. To see a preview of the selected HTML page, click **View Output**.

   The selected HTML page is displayed in the **HTML Samples** window.



4. To use this script to build the HTML page, click **Use Sample**.

   The selected sample script is displayed in the **Html widget** window.



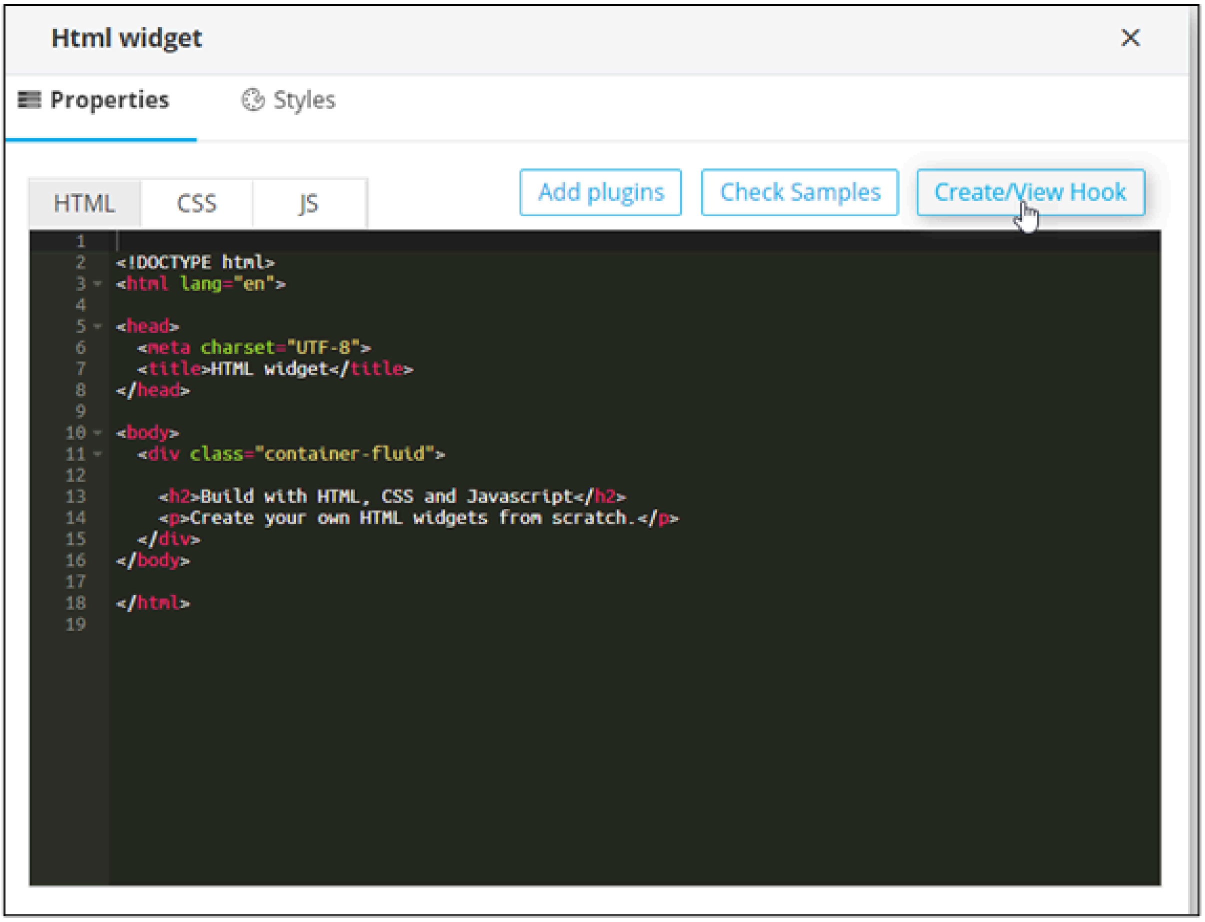
5. To save your selection, click **Save**.

Creating custom HTML pages using Hook Inventory

You can fetch information from a data source using Hooks. There is a provision to select pre existing hooks or create new hooks in the Hooks Inventory and use them to build customized HTML pages.

To use Hooks for building an HTML page:

1. In the **Html widget** window, under **Properties**, click **Create/View Hook**.

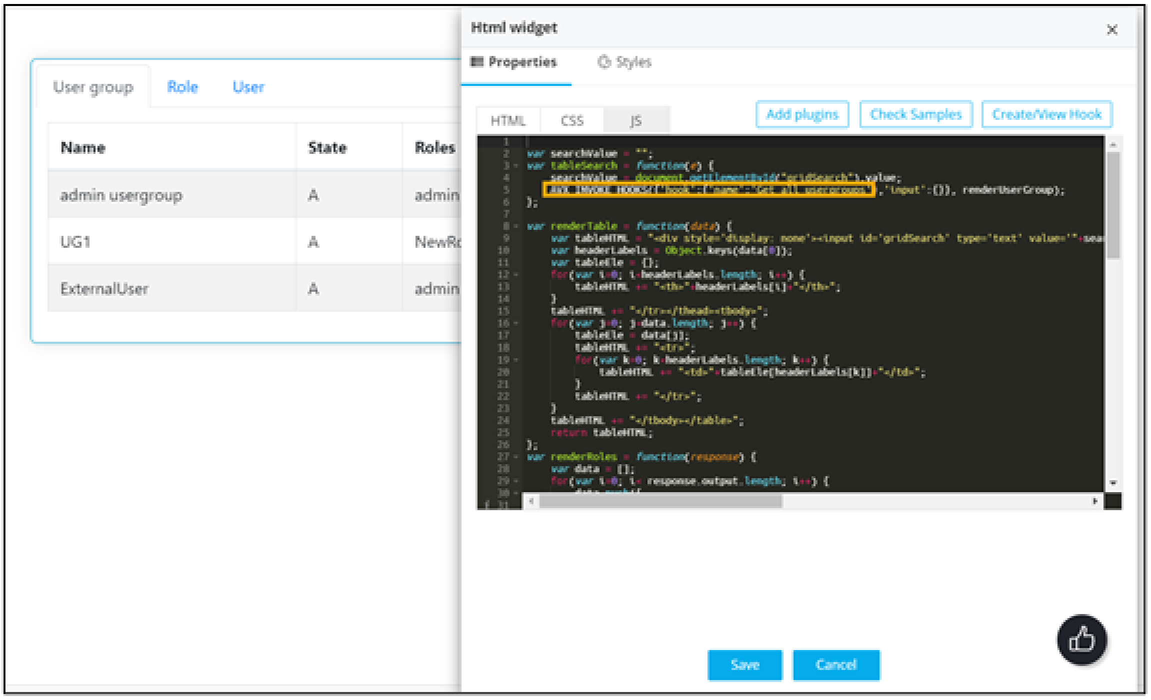


The Hooks Inventory opens in a new tab.

> **Note:** For more information on **Hooks**, refer to the Automation User Guide.

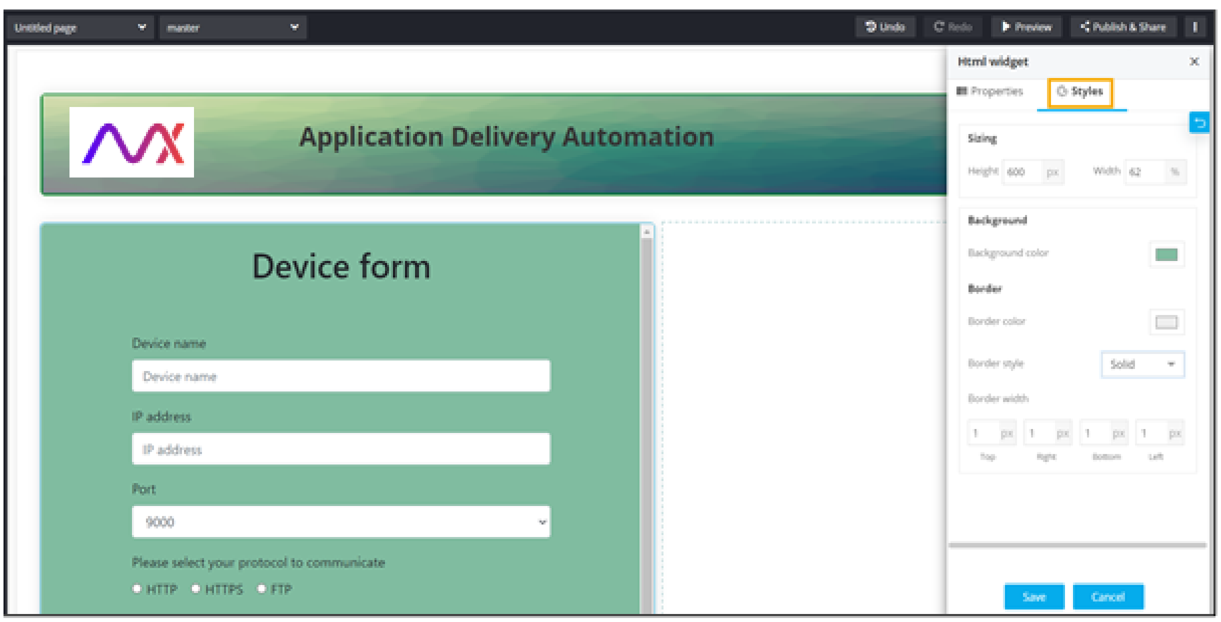
2. To build your customized HTML page, you can either use a preexisting hook from the Hook Inventory or create a new hook.

For example, Table using hooks created using **Get all usergroups** hook.



Configuring HTML Widget - Styles

1. For more options to customize the HTML widget, in the **HTML widget** window, click **Styles**.



Field descriptions for Styles

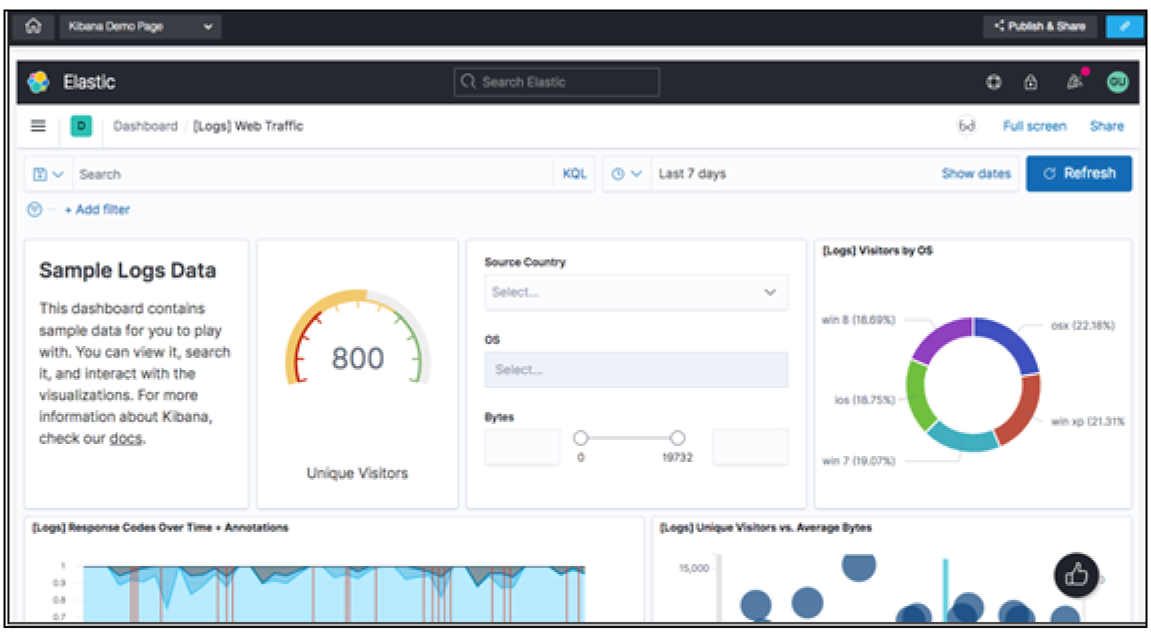
Field	Description
Sizing	Allows you to increase/decrease font **Size** of the field name.
Background	Allows you to define the **Background color** for the HTML widget.
Border	Allows you to define the • **Border color** - Select the color of the HTML widget's border. • **Border style** - Select the border style for the HTML widget from the options available in the dropdown. • **Border width** - Select the thickness of the HTML widget's border.

2. To save your settings, click **Save**.
3. To restore the settings under **Styles** to default, click  **(Restore)** icon.

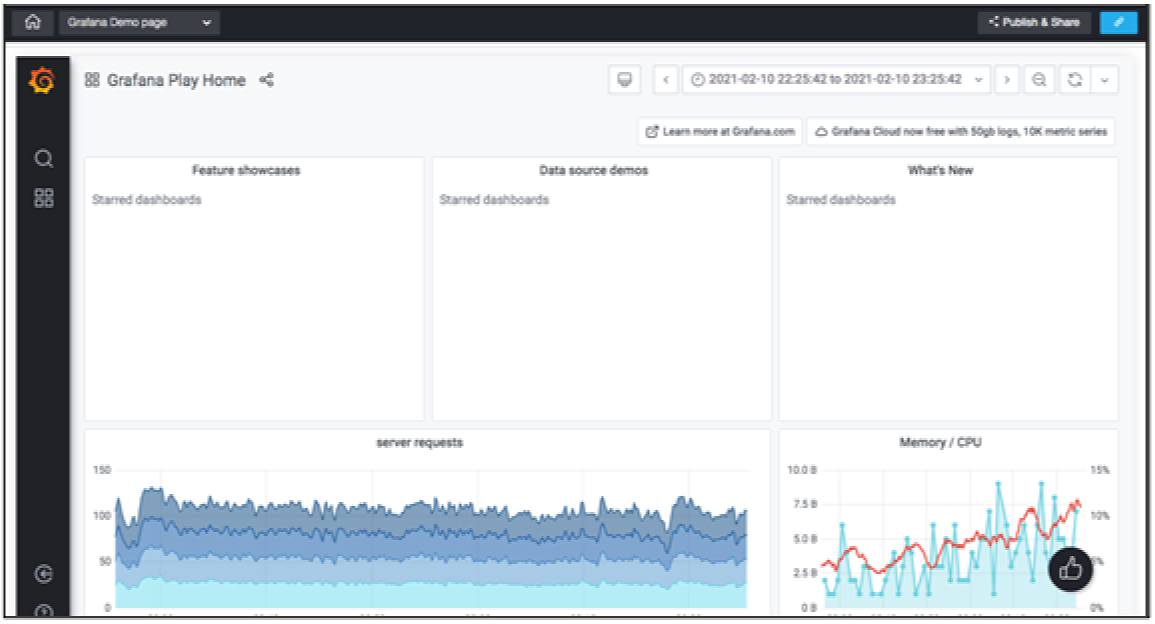
Embed page

You can embed an entire web page inside a single widget in your service catalogs.

For example, embed a Kibana visualization dashboard.



For example, embed a Grafana visualization dashboard.



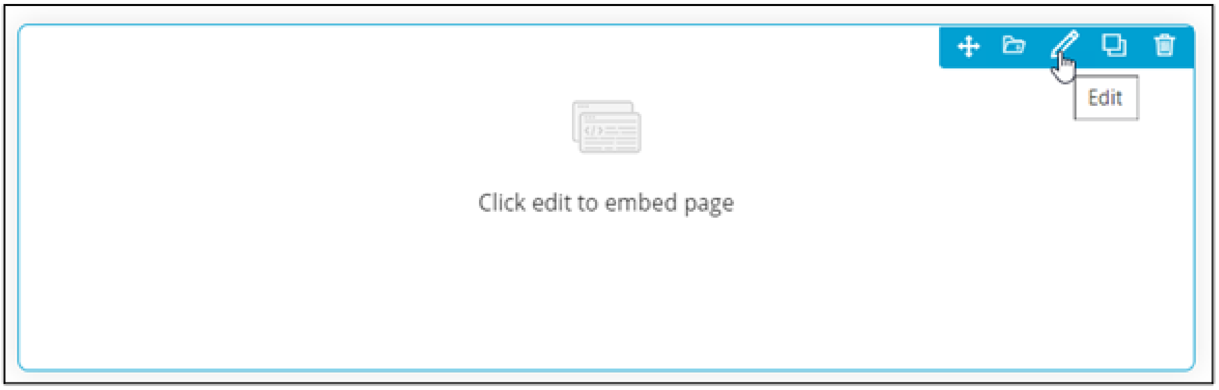
- Embedding a page - Properties
- Embedding an Internal page
- Embedding an External Page
- Embedding a Page - Styles

Embedding a page - Properties

You can either choose from a predefined list of internal pages or embed an external page into the widget. Cross domain data exchange must be possible to embed an external page into your catalog.

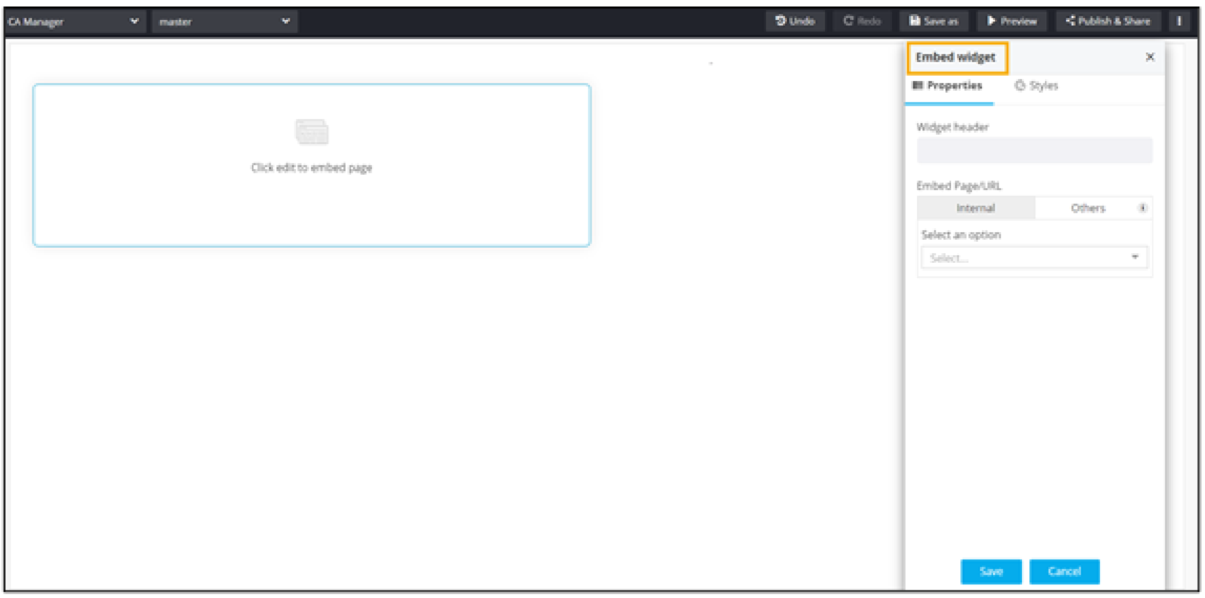
To embed a page:

1. From the **Build** component in the left menu, under the **Widgets** section, drag and drop the **Embed page** widget to any blue highlighted space on the page.
2. Hover your mouse over the widget and click  **(Edit)** icon.



The Embed widget pop-up window opens with the **Properties** tab open by default.

3. In the **Embed widget** window, under **Properties**, enter or select the required field information.



Field descriptions for Properties

Field	Description
Widget Header	Enter a suitable widget header
Embed Page/URL	You have two options for embedding a page in the widget: • **Internal**: Add an internal link from within AppViewX pages. • **External**: Users can embed cross origin enabled pages here.

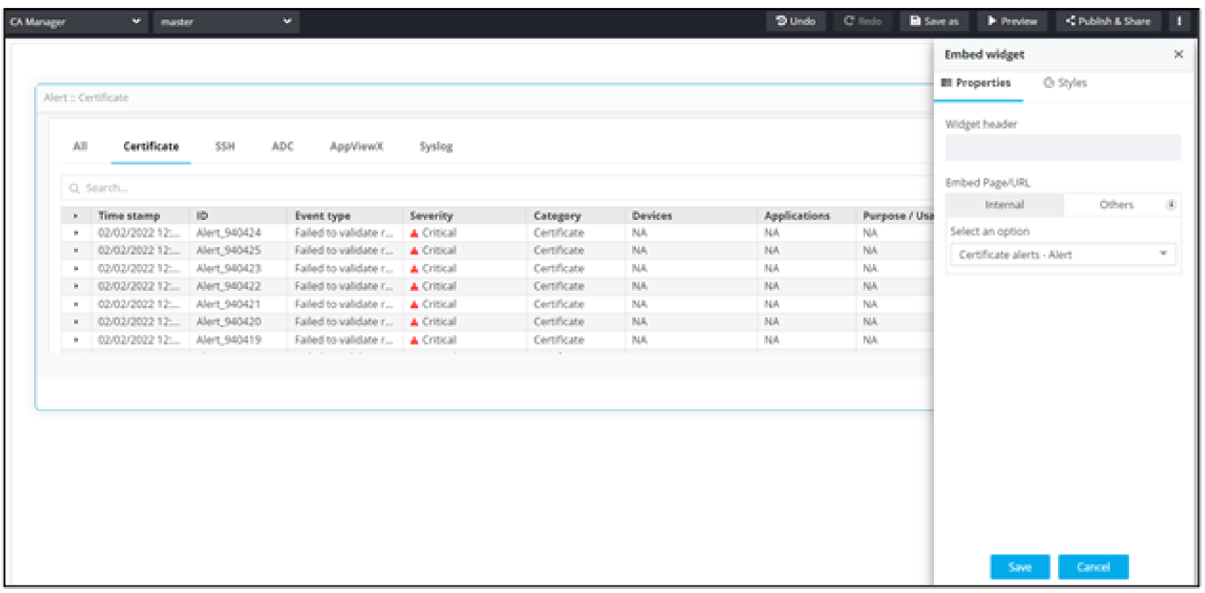
4. Click **Save**.

Embedding an Internal page

1. In the **Embed widget** window, under **Internal**, select an option from the drop down list. For example, **Dashboard - ADC**.

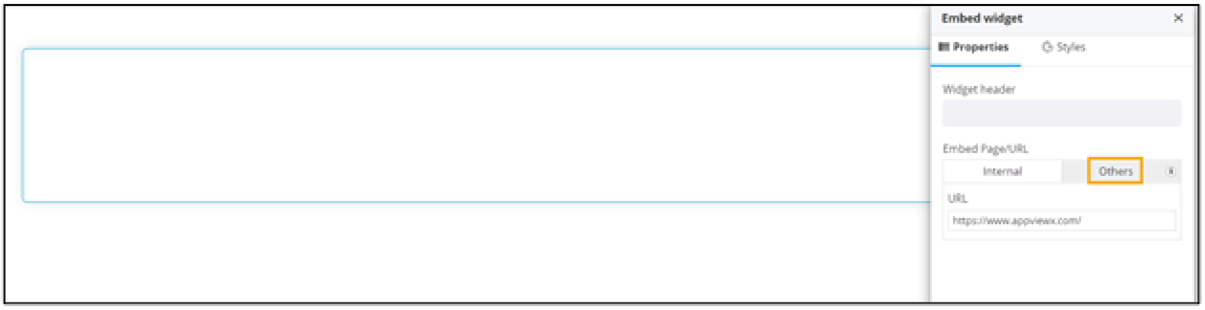
2. To embed this page, click **Save**.

   The selected page is embedded and displayed in your service catalog.



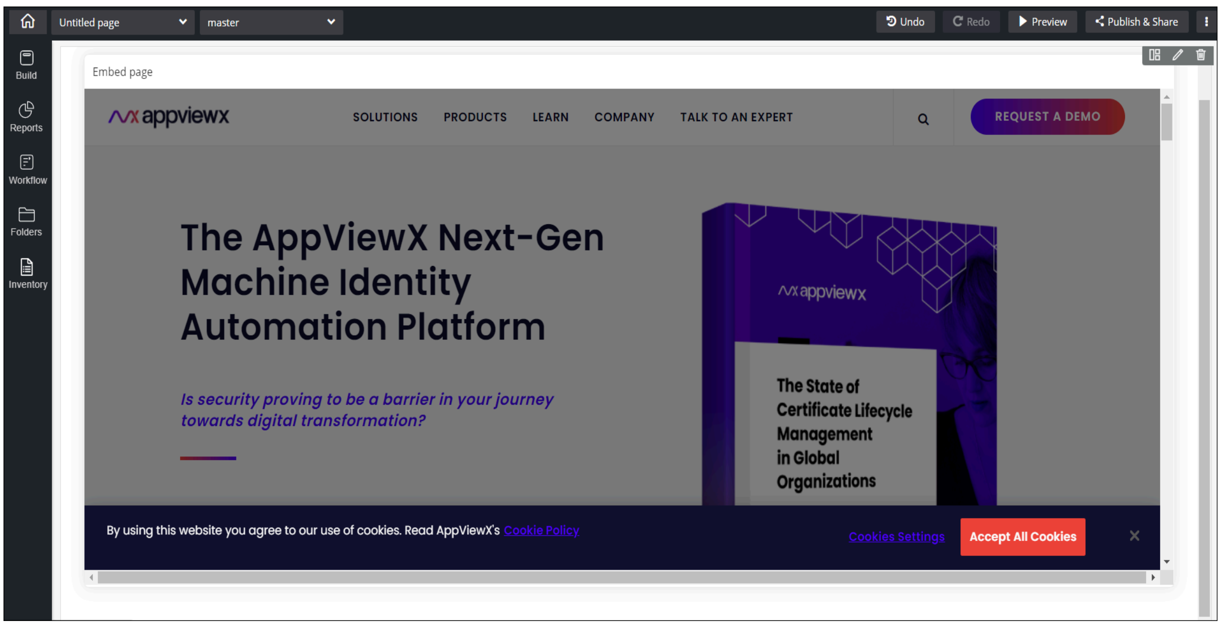
Embedding an External Page

1. In the **Embed widget** window, under **Properties**, click **Others**.



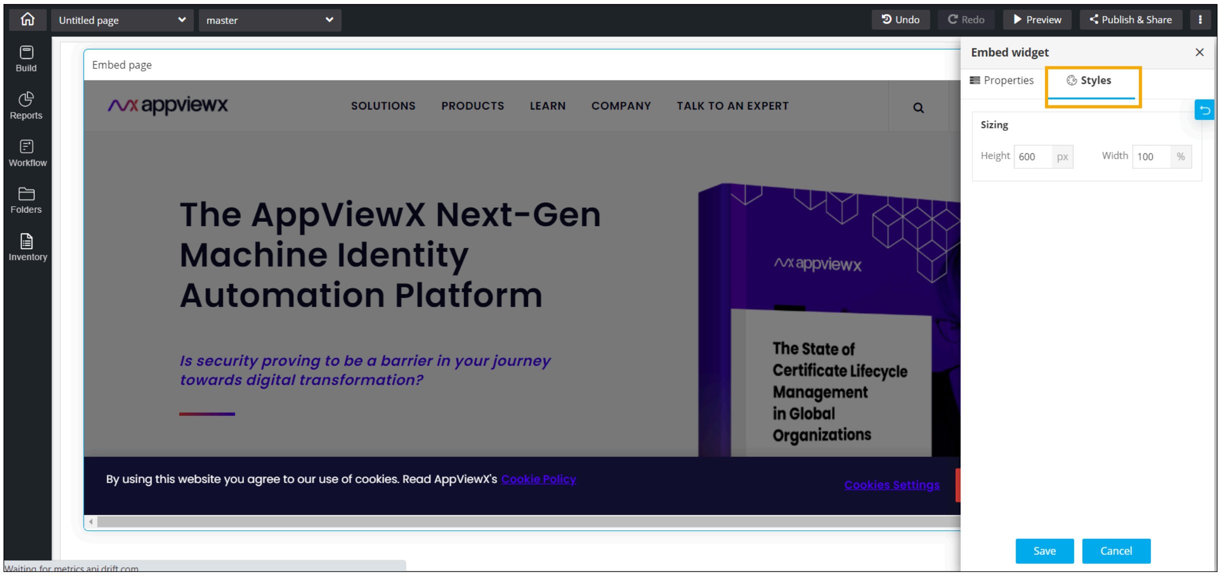
2. Enter the URL of the page that is to be embedded.
3. To embed the page, click **Save**.

   The page is embedded in your service catalog.



Embedding a Page - Styles

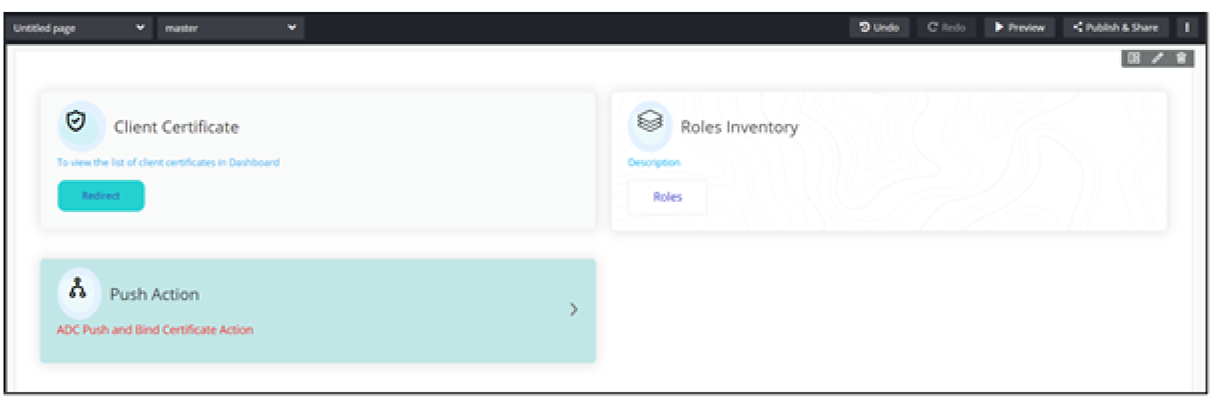
1. For more options to customize the **Embed widget**, in the **Embed widget** window, click **Styles**.



2. Under **Sizing**, you can increase/decrease the size of the widget.
3. To save your settings, click **Save**.
4. To restore the settings under **Styles** to default, click **(Restore)** icon.

Button (Action Widget)

You can add buttons to your service catalog to trigger specific “call to action”.

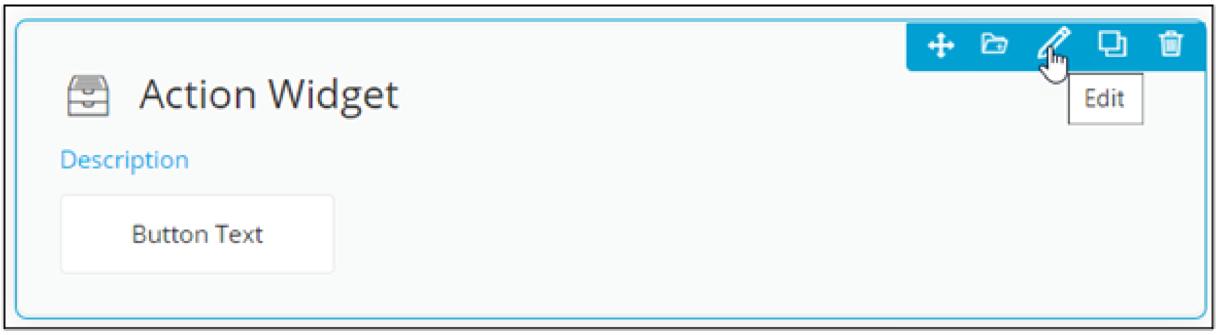


- Configuring the Button/Action Widget - Properties
- Adding an Internal Action Widget Link
- Adding an External Action Widget Link

- [Adding a Workflow Link to the Action Widget]
- [Configuring the Button/Action Widget - Styles]

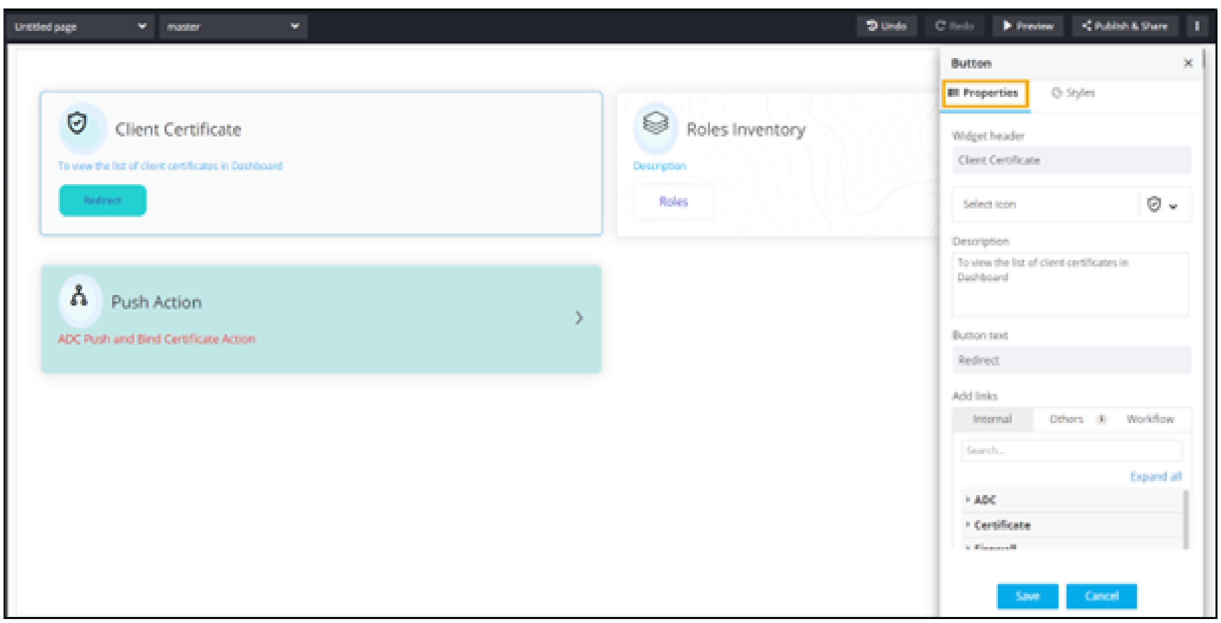
Configuring the Button/Action Widget - Properties

1. From the **Build** component in the left menu, under the **Widgets** section, drag and drop the **Button** widget to any blue highlighted space on the page.
2. To configure the button/action widget, hover your mouse over the widget and click **(Edit)** icon.



The **Button** pop-up window opens.

3. In the Button pop-up window, under **Properties**, enter or select the required fields.



Field descriptions for Properties

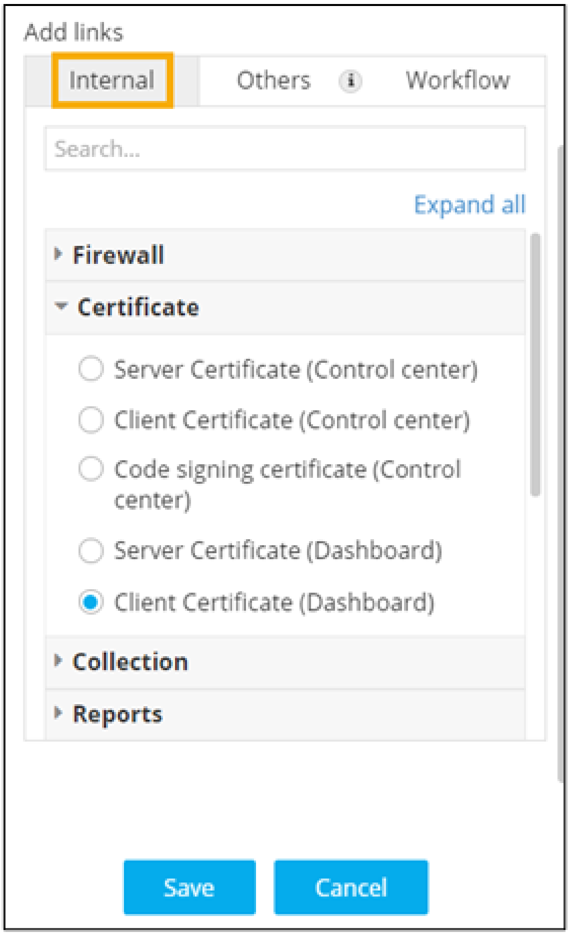
Field	Description
Widget Header	Allows you to enter the widget header name.

Field	Description
Select icon	Allows you to select the icon for the widget from the options available in the dropdown.
Description	Allows you to add a description for the action widget.
Button text	Allows you to add the text that is displayed on the button.
Add links	Allows you to add links to the action widget. The following options are available here: • **Internal**: Add an internal from within AppViewX pages. • **Others**: Add links to other pages within AppViewX or embed external cross origin enabled pages. • **Workflow**: Add a workflow link which redirects to the workflow execution page.

4. Click **Save**.

Adding an Internal Action Widget Link

1. To link an internal AppviewX page to the action widget, under **Add links**, click **Internal**.
2. Select the required internal link from the list.

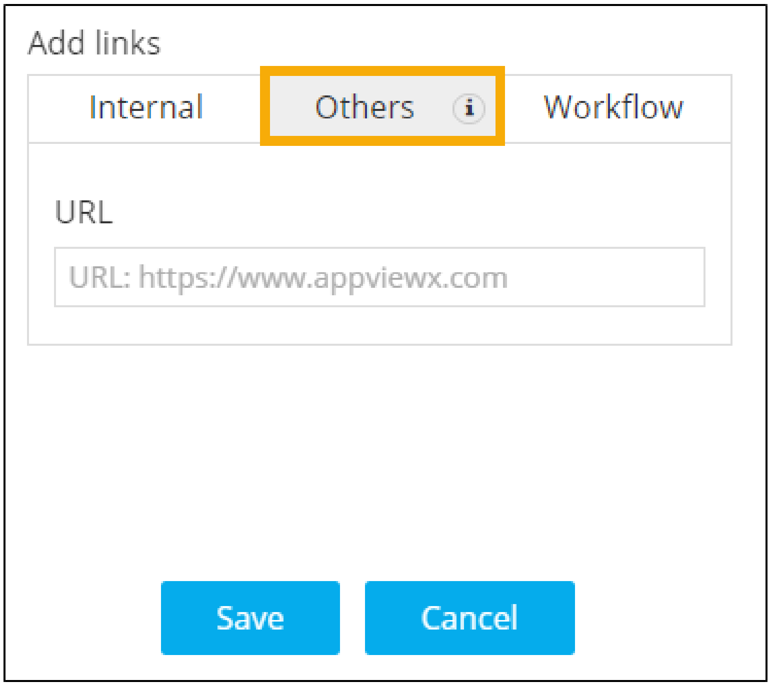


3. Click **Save**.

   Clicking on the button will redirect to the selected link in a new tab.
4. For more options to customize the Action widget, click here.

Adding an External Action Widget Link

1. To embed an external link or a cross origin enabled page URL into the Action widget, under **Add links**, click **Others**.
2. Enter the **Name** and **URL** for the link.

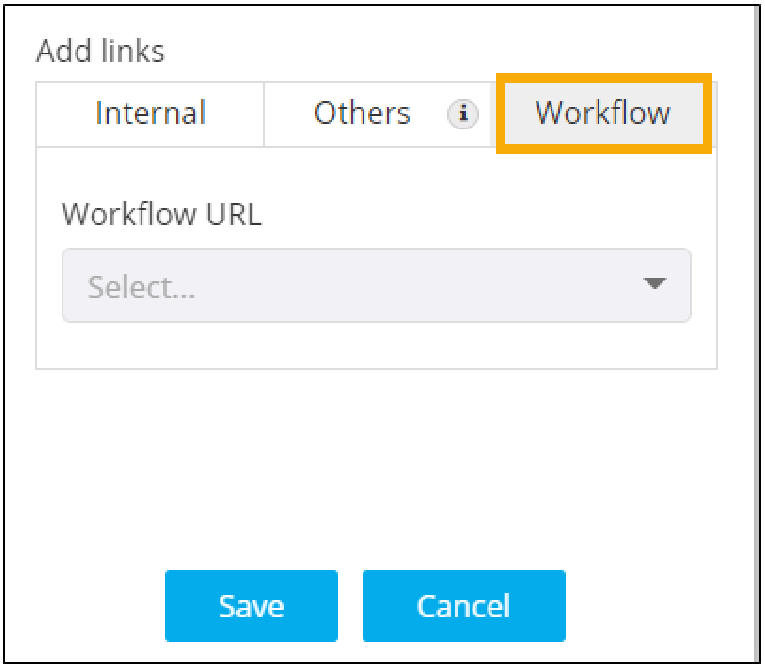


3. Click **Save**.

   Clicking on the button will redirect to the page as per the configured URL.
4. For more options to customize Action widget, click here.

Adding a Workflow Link to the Action Widget

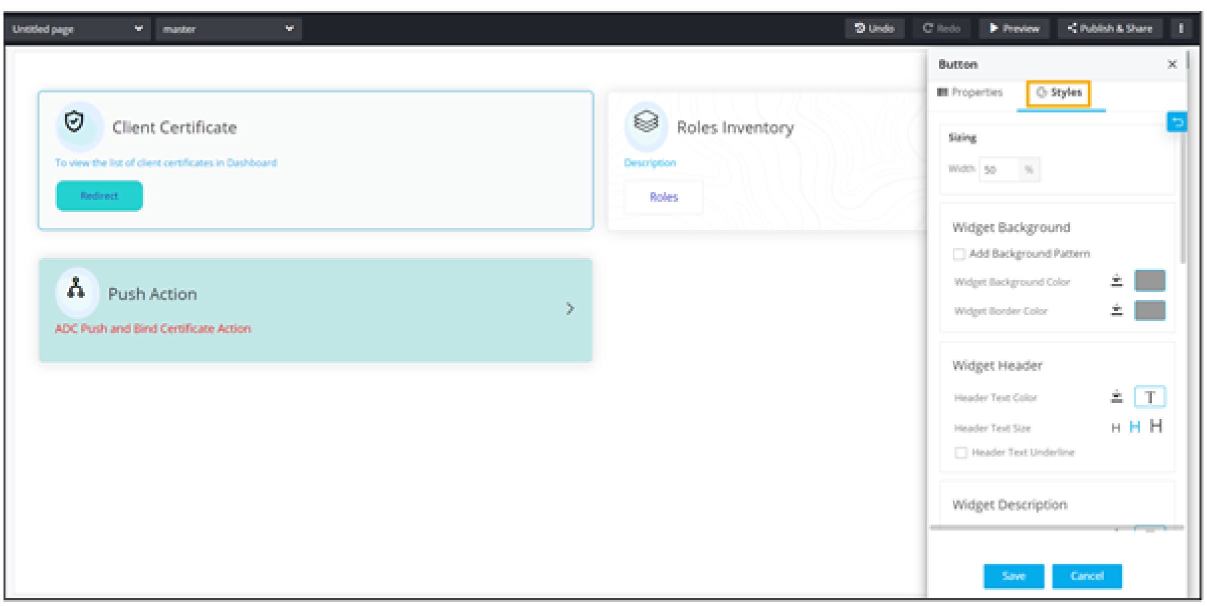
1. To add a link to redirect to the Workflow Request execution page for the selected workflow, under **Add links**, click **Workflow**.
2. Enter the link **Name** and select the **Workflow URL** from the dropdown.



3. Once you have selected the required workflow URL to the action widget, click **Save**.
   Clicking on the button will redirect to the workflow request execution page for the configured workflow URL.
4. For more options to customize Action widget, click here.

Configuring the Button/Action Widget - Styles

1. For more options to customize the action widget, in the **Button** window, click **Styles**.



Field descriptions for Styles

Option	Description
Sizing	Allows you to increase/decrease the width of the action widget.
Widget Background	The following options are available here: • **Add Background Pattern**: To add a background pattern to the widget, select the **Add Background Pattern** checkbox and select a pattern from the available options. • **Widget Background Color**: To add a background color to the widget, unselect the **Add Background Pattern** checkbox and select the color from the color palette. • **Widget Border Color**: Allows you to change the color of the widget border.
Widget Header	The following options are available here: • **Header Text Color**: Allows you to change the color of the header text. • **Header Text Size**: Allows you to select the size of the header text. • **Header Text Underline**: To underline the header text, select the Header Text Underline checkbox.
Widget Description	The following options are available here: • **Description text Color**: Allows you to change the color of the label text. • **Description Text Size**: Allows you to select the size of the label text. • **Description Text Underline**: To underline the description text, select the **Description Text Underline** checkbox.
Widget Type	Allows you to enable/disable the button in the action widget. • To enable the button select the **Enable button** checkbox. • To disable the button, deselect the **Enable button** checkbox. The action widget is displayed in a card view. Click on the widget to redirect to the page that is linked to the widget.
Button Background	This feature is available only when the button is enabled. The following options are available here: • **Button Corners**: Allows you to define the curvature of the button corners. The higher the number, the corners will be curved more. • **Button Alignment**: Allows you to define the placement of the button in the action widget - left, center, and right.

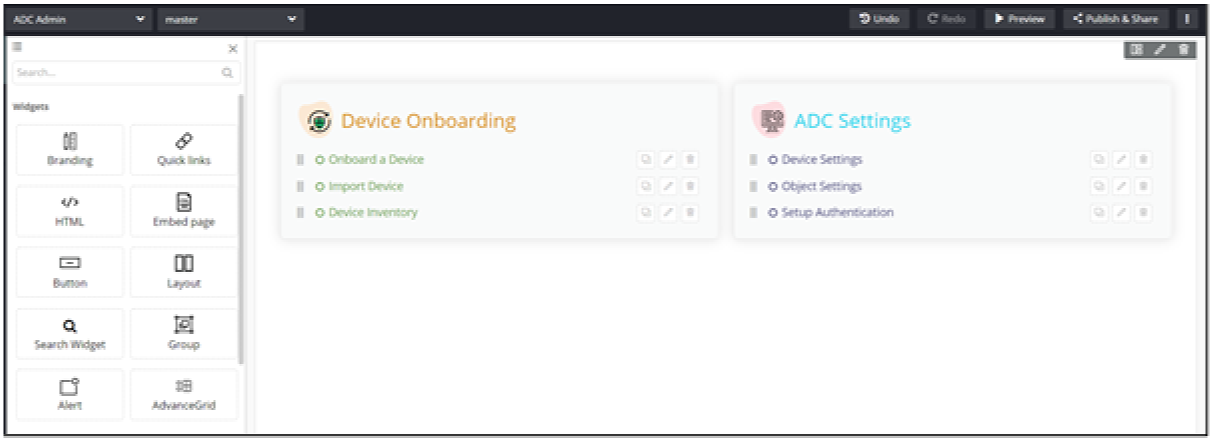
Option	Description
	• **Widget Background Color**: Allows you to select the background color of the widget from the color palette. • **Widget Border Color**: Allows you to select the color of the widget border from the color palette.
Button Text	This feature is available only when the button is enabled. The following options are available here: • **Button Text Alignment**: Allows you to define the placement of the button text inside the button **-** left, center, and right. • **Button Text color**: Allows you to select the button text color from the color palette. • **Text Size**: Allows you to select the size of the text displayed on the button.
Icon Style	The following options are available here: • **Icon Color** - Select the Icon Color from the available options. • **Icon Background** - Select the Icon Background from the available options. • **Icon Background Color -** Select the Icon Background Color from the available options.

2. To save your settings, click **Save**.
3. To restore the settings under **Styles** to default, click  **(Restore)** icon.

Layout

You can customize the layout of your service catalog(s) in the following ways:

- Separate a row into two columns at a time
- Insert more layouts within a layout for additional columns.

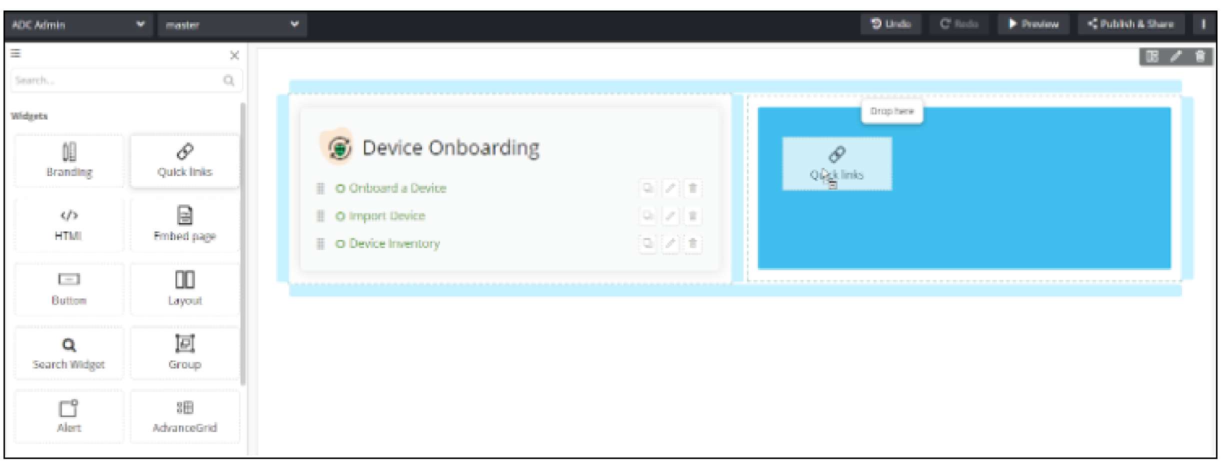


- Adding a Layout component
- Configuring the Layout component

Adding a Layout component

To insert a Layout component to your page:

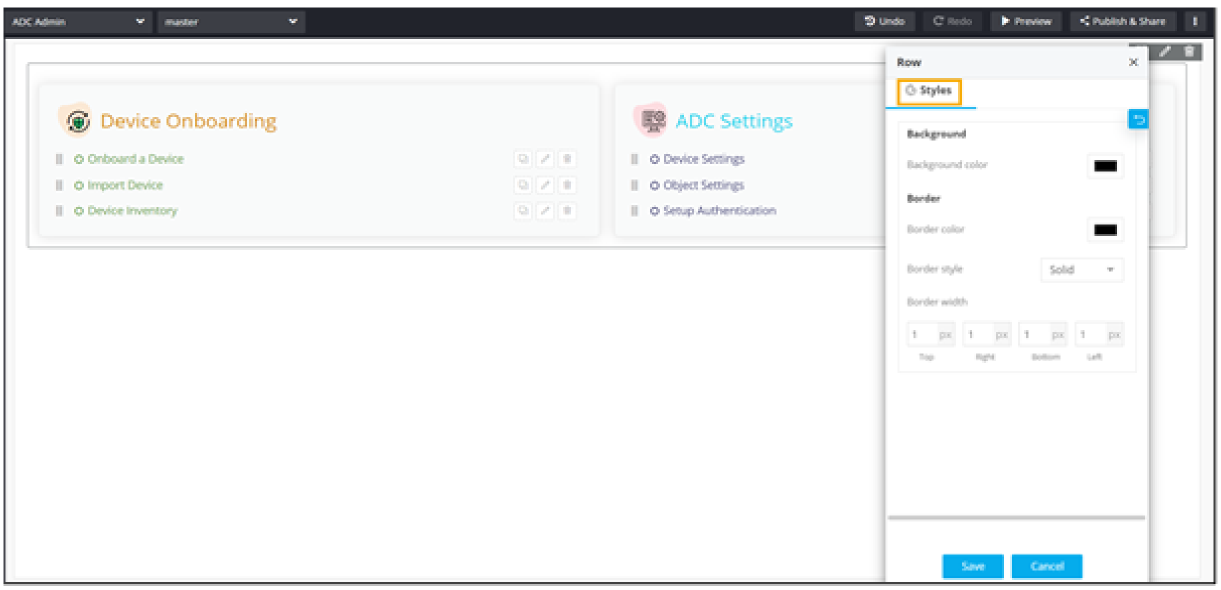
1. Drag and drop the **Layout** component to any blue highlighted space on the page.
2. Drag and drop one or more component(s) into this **Layout** widget as per your requirement.



Configuring the Layout component

1. To configure/edit the Layout component, hover your mouse over the widget and click  **(Edit)** icon.

The **Row** pop-up window opens with the **Styles** tab.



Field descriptions for Styles

Option	Description
Background	Allows you to define the **Background color** for the Layout widget.
Border	Allows you to define the • **Border color** - Select the color of the layout border. • **Border style** - Select the border style for the layout section from the options available in the dropdown. • **Border width** - Select the thickness of the layout border.

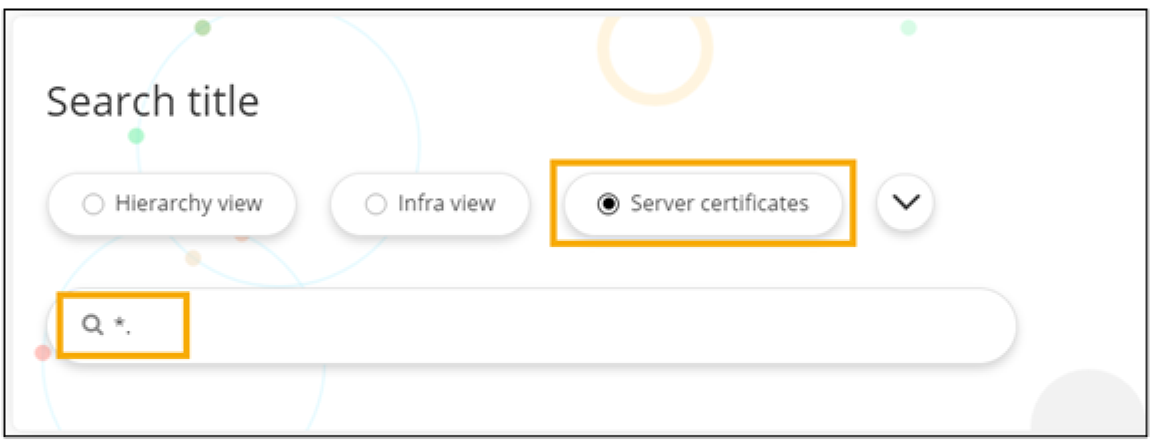
2. To save your settings, click **Save**.

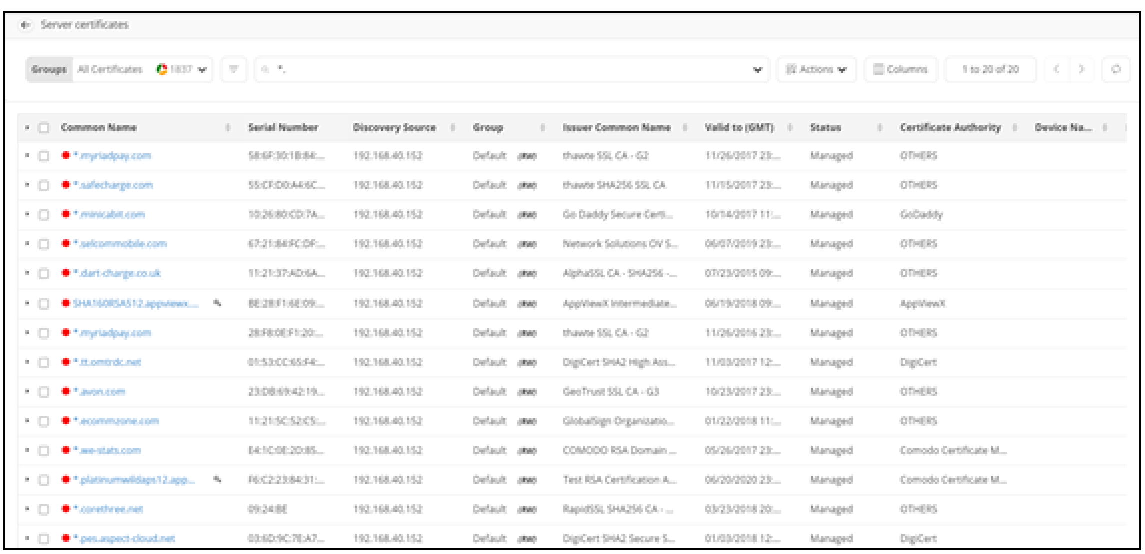
3. To restore the settings under **Styles** to default, click **(Restore)** icon.

Search widget

You can add a search widget to your service catalog to search for applications, security services and certificates based on RBAC. The search widget can be customized based on infra and hierarchy view. Adding a search widget saves time by redirecting to the search results page on the same page.

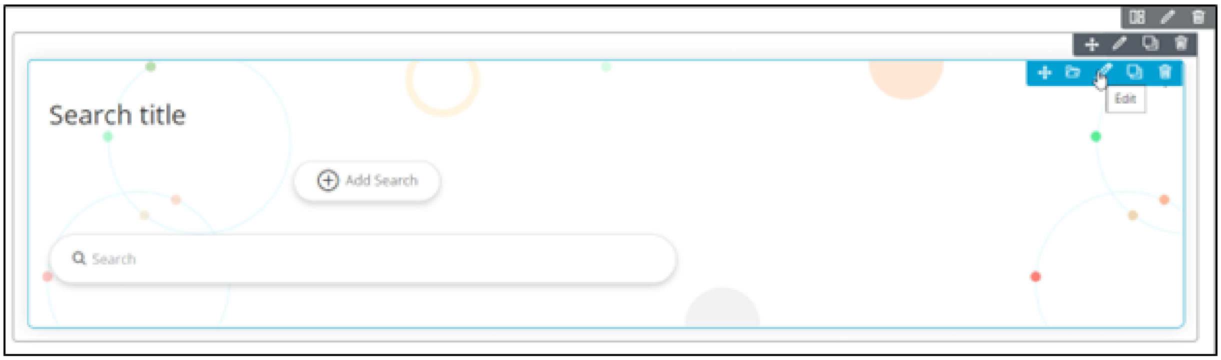
For example, search for certificates by typing *. in the search field to get a list of all server certificates.





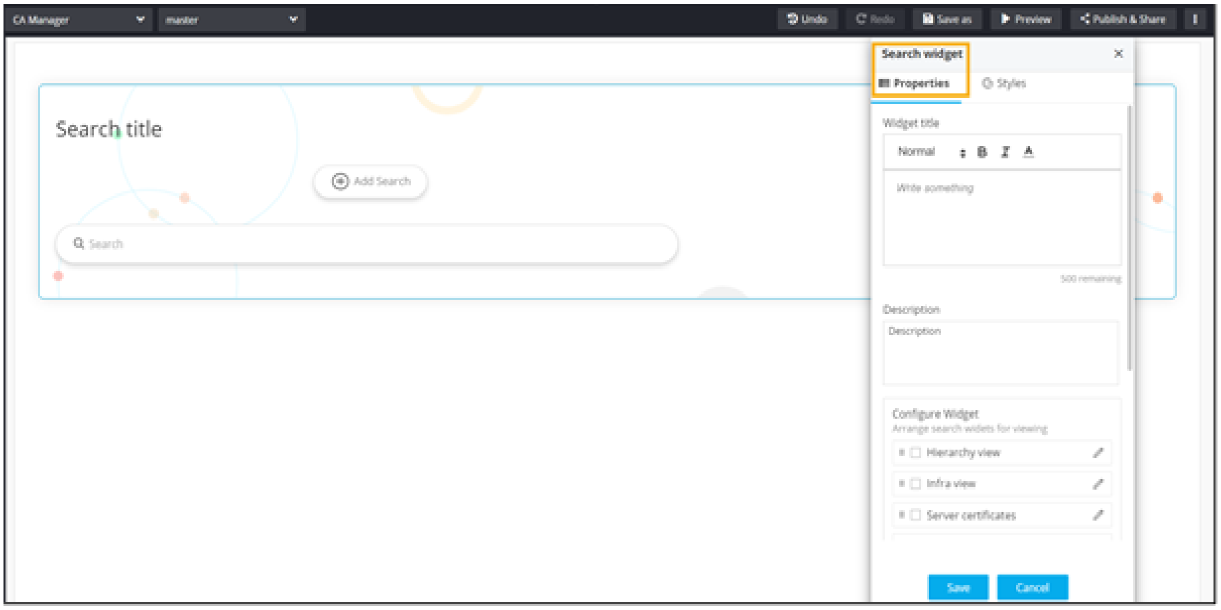
Configuring the Search Widget - Properties

1. From the **Build** component in the left menu, under the **Widgets** section, drag and drop the **Search** widget to any blue highlighted space on the page.
2. To configure/edit the Search Widget component, hover your mouse over the widget and click **(Edit)** icon.



> **Tip:** To configure/edit the Search Widget component, you can also click **Add Search** in the Search widget.

The **Search widget** window opens with the **Properties** tab open by default.



3. In the **Search widget** window, under **Properties**, enter or select the required field information.

   The following table describes the various fields in this tab:

Field descriptions for Properties

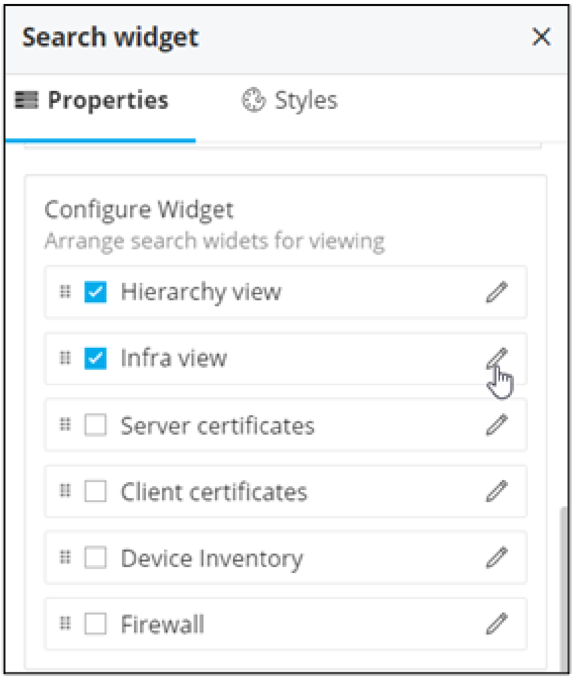
Field	Description
Widget title	Provide a suitable title for the Search widget. **Note:** You have the option to set the font size, color, and thickness in the **Widget title** editor box.
Description	Allows you to add a **Description** for the search widget.
Configure Widget	Select the search parameters that would appear in the search widget by checking the box. **Note:** You have the option to edit the search widgets, add descriptions and tooltips by clicking **(Edit)** icon. For more information, click here.
Widget Layout	Choose a widget layout from the available options.

4. Click **Save**.

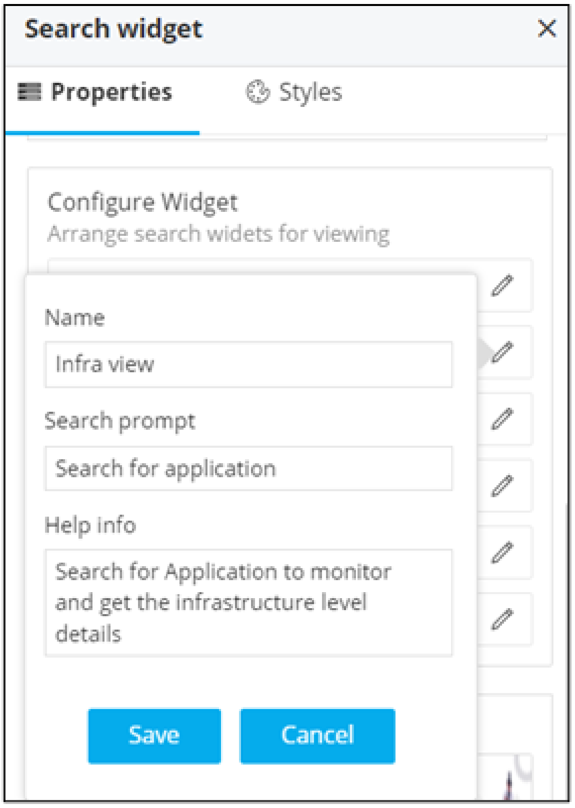
Editing a search widget

To edit a specific search widget:

1. In the **Search widget** window, under **Properties**, click **(Edit)** icon next to the widget to be edited.



2. Enter all the field information.

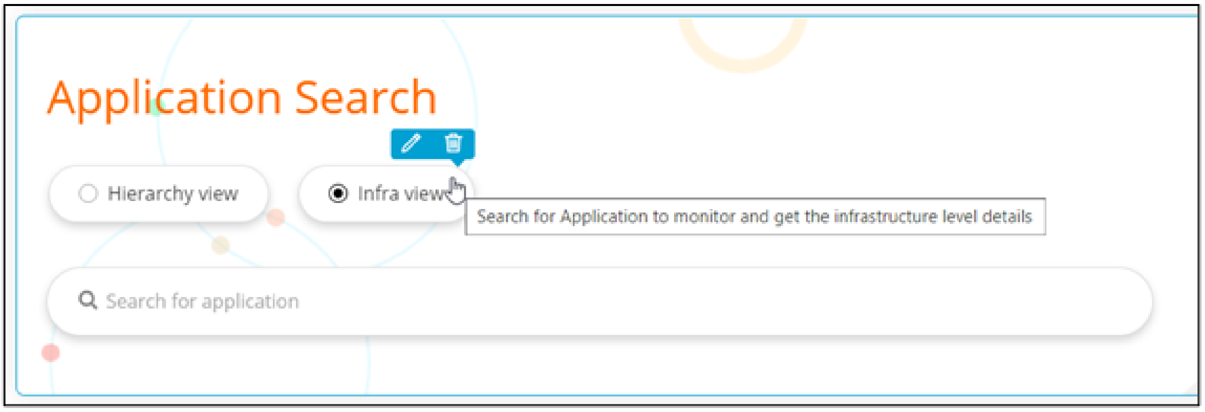


Field descriptions for Properties

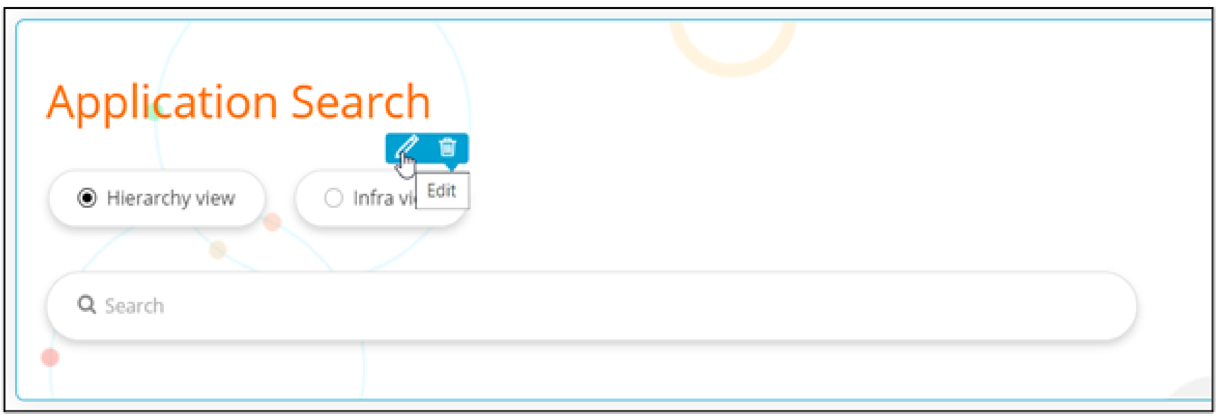
Field	Description
Name	Allows you to change the name of the search widget.
Search prompt	Allow you to add a search prompt in the search bar when a particular search widget is selected.
Help info	Allows you to add a help text to the search widget.

3. Click **Save**.

   The search prompt and help info is updated in the search widget.

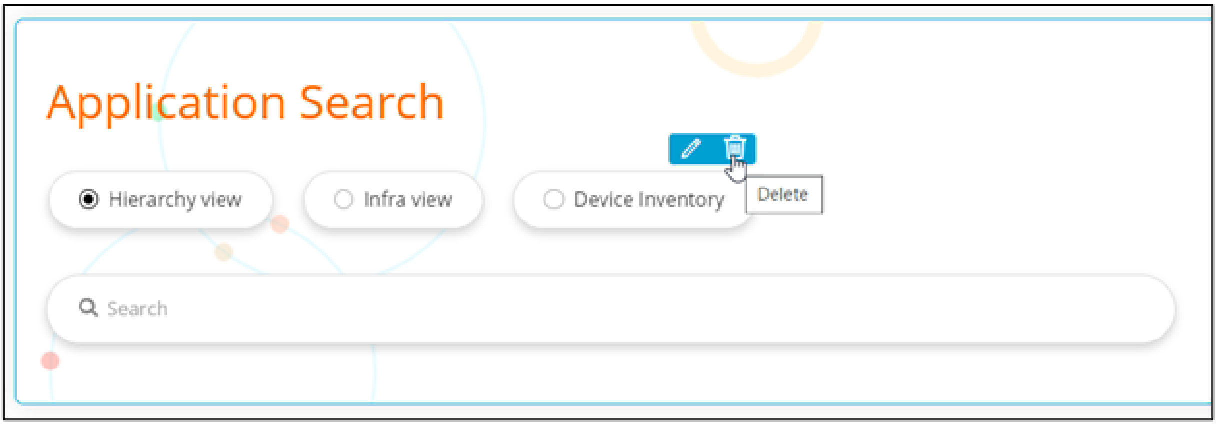


Note: You can also edit the search widget, by hovering your mouse over the search widget and clicking **(Edit)** icon.

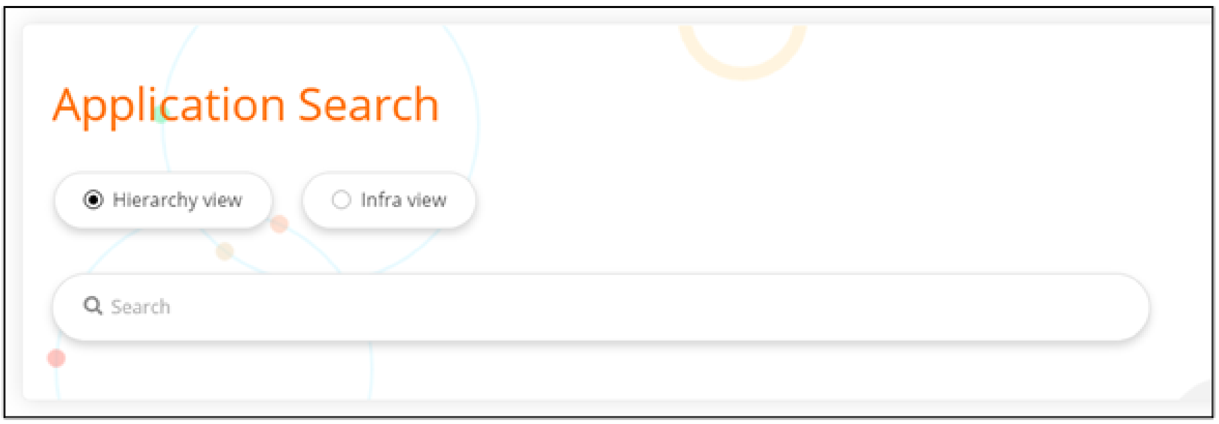


Deleting a single search widget

1. Hover your mouse over the search widget to be deleted.
2. Click **(Delete)** icon.

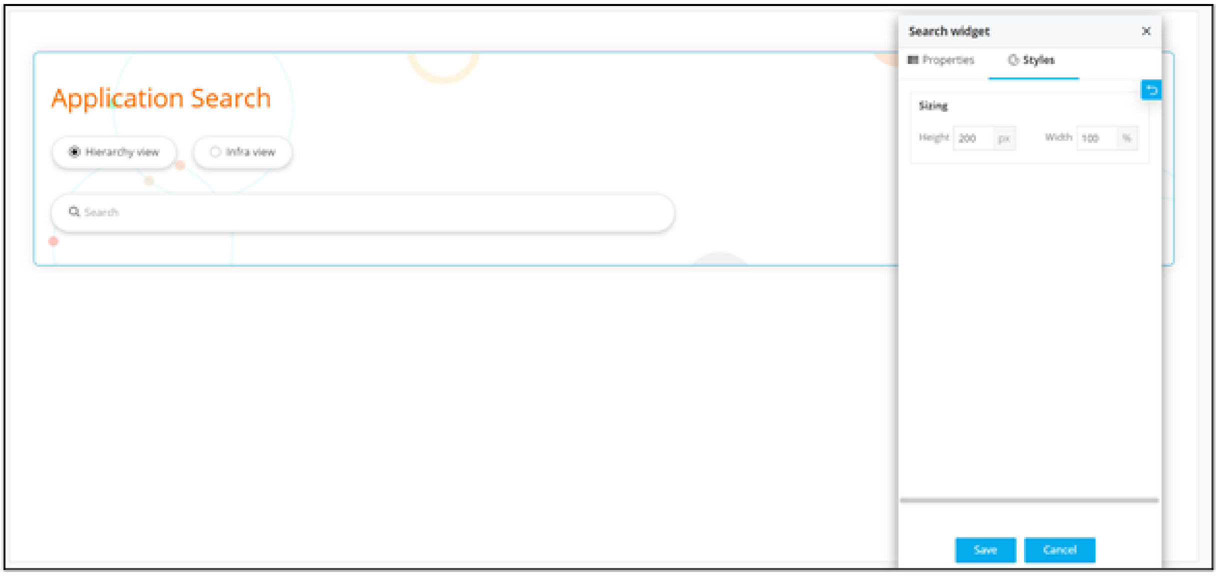


The selected search widget is deleted.



Configuring the Search Widget - Styles

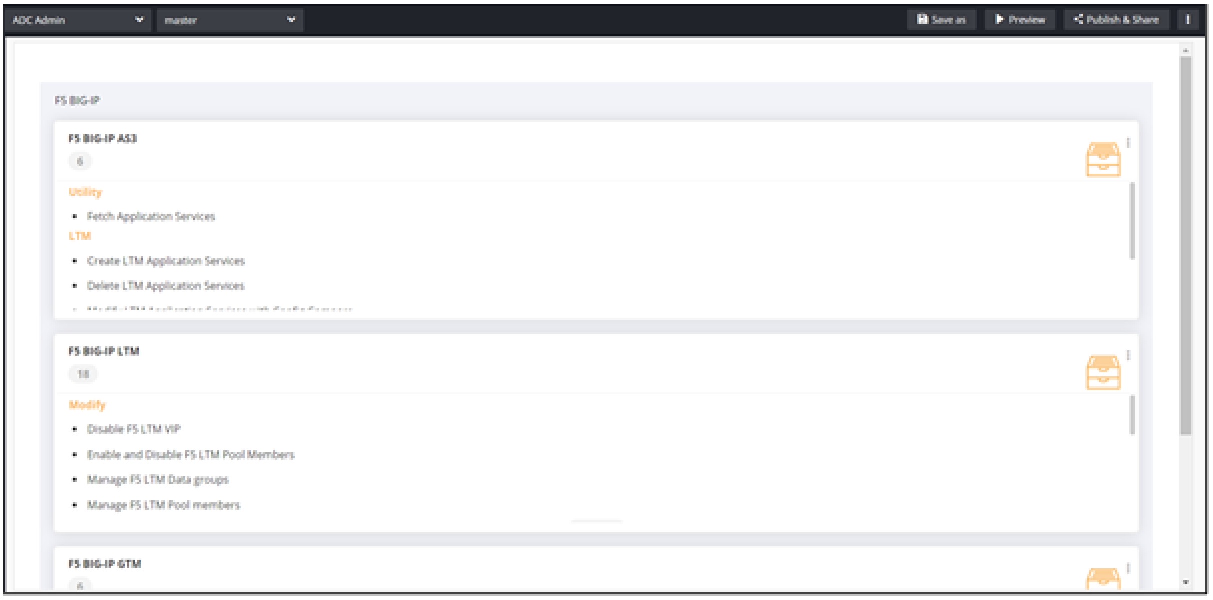
1. For more options to customize the **Search Widget**, in the **Search Widget** window, click **Styles**.



2. Under **Sizing**, you can increase/decrease the height and width of the search widget.
3. To save your settings, click **Save**.
4. To restore the settings under **Styles** to default, click  **(Restore)** icon.

Group

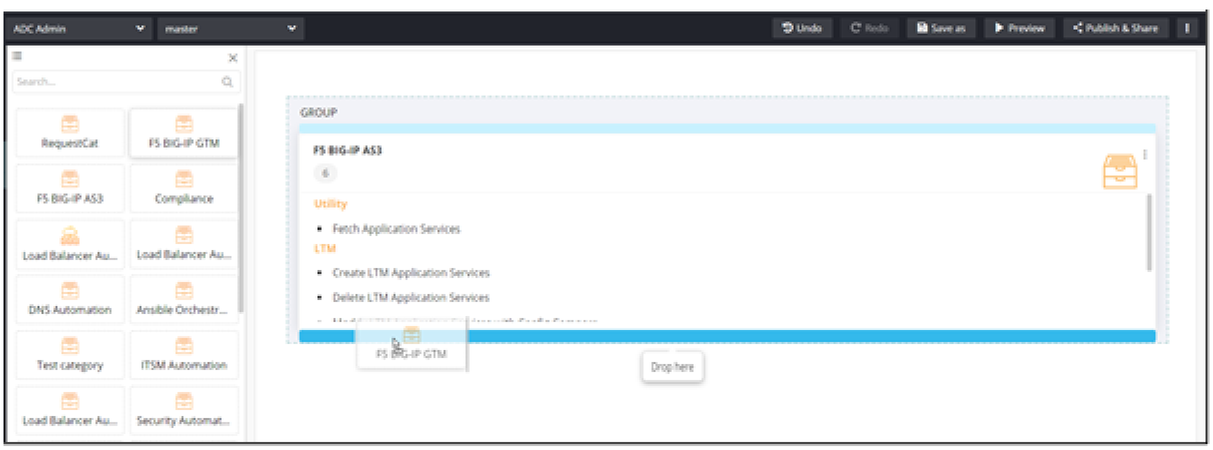
Service catalogs can be customized for logical grouping of multiple page components. You can club together related widgets such as workflow catalogs or reports and arrange them under one group for easy access and visibility.



- Configuring the Group widget - Properties
- Configuring the Group widget - Styles

Configuring the Group widget - Properties

1. From the **Build** component in the left menu, under the **Widgets** section, drag and drop the **Group** widget to any blue highlighted space on the page.
2. Drag and drop related widgets into this **Group** component to view them all in one place.



3. To configure/edit the Group component, click **(Edit)** icon.

The **Group** pop-up window opens with the **Properties** tab open by default.

4. In the **Group** window, under **Properties**, you can change the **Group name**.
5. Click **Save**.
6. To configure individual widgets within the group, hover your mouse over the specific widget and click  **(Edit)** icon.

Configuring the Group widget - Styles

1. For more options to customize the **Group** component, in the **Group** window, click **Styles**.

Field descriptions for Styles

Option	Description
Weight	The **Weight** dropdown list lets you set the font thickness of the field label using one of the following values: • **Default** • **Light** • **Normal** • **Thick**.
Size	Allows you to increase/decrease font **Size** of the field name.
Line height	Allows you to increase/decrease the **Line height** of the field. This means the space between the field and the field border.
Color	Allows you to define the **Color** of the field label.
Align	Allows you to **Align** the field label to the left, center or right side of the field.
Background	Allows you to define the **Background color** for the HTML widget.

Option	Description
Border	Allows you to define the • **Border color** - Select the color of the layout border. • **Border style** - Select the border style for the layout section from the options available in the dropdown. • Border width - Select the thickness of the layout border.

2. To save your settings, click **Save**.

3. To restore the settings under **Styles** to default, click **(Restore)** icon.

Alert

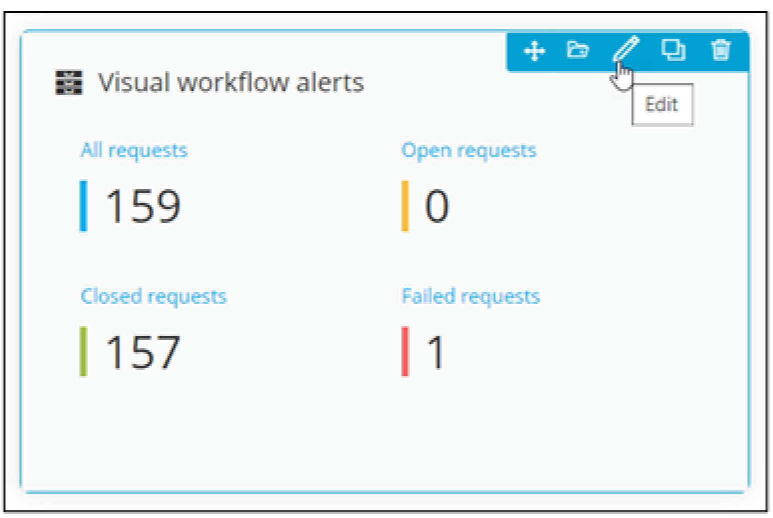
You can add automation and self-service request alerts to your customized service catalogs. This widget displays the status of workflow requests - total requests, open, closed and failed.

- Configuring the Alert widget - Properties
- Configuring the Alert widget - Styles

Configuring the Alert widget - Properties

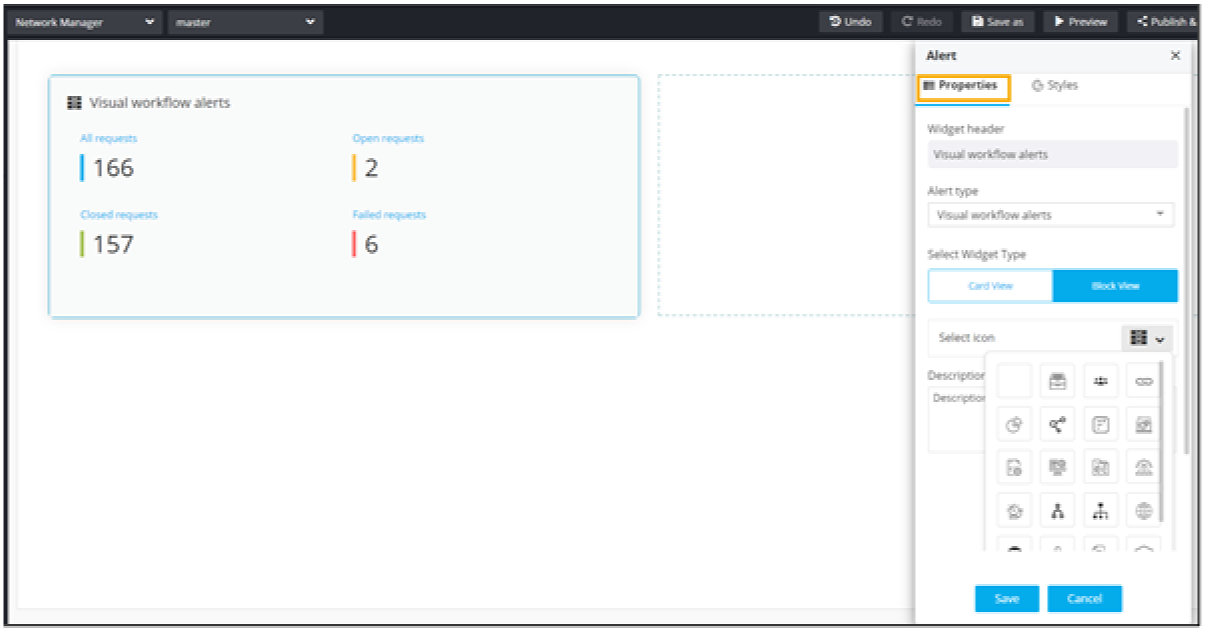
1. From the **Build** component in the left menu, under the **Widgets** section, drag and drop the **Alert** widget to any blue highlighted space on the page.

2. To configure/edit the **Alert** widget, hover your mouse over the widget and click **(Edit)** icon.



The **Alert** widget window opens with the **Properties** tab open by default.

3. In the **Alert** window, under the **Properties** tab, enter or select the required field information.



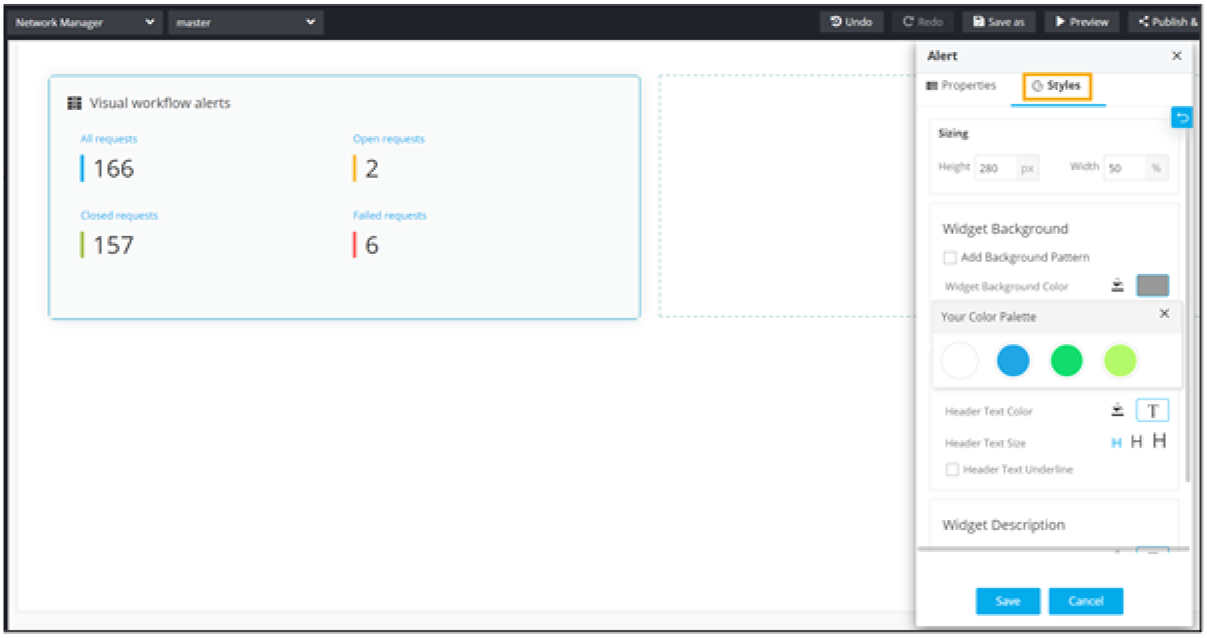
Field descriptions for Properties

Field	Description
Widget header	Allows you to rename the widget header.
Alert Type	Allows you to select the **Alert Type** from the options available in the dropdown.
Select Widget Type	Allows you to switch between Card View and Block View.
Select Widget Icon	Allows you to select an icon for the widget from the options available in the dropdown.
Description	Allows you to add a description for the widget.

4. Click **Save**.

Configuring the Alert widget - Styles

1. For more options to customize the Alert widget, in the **Alert** window, click **Styles**.



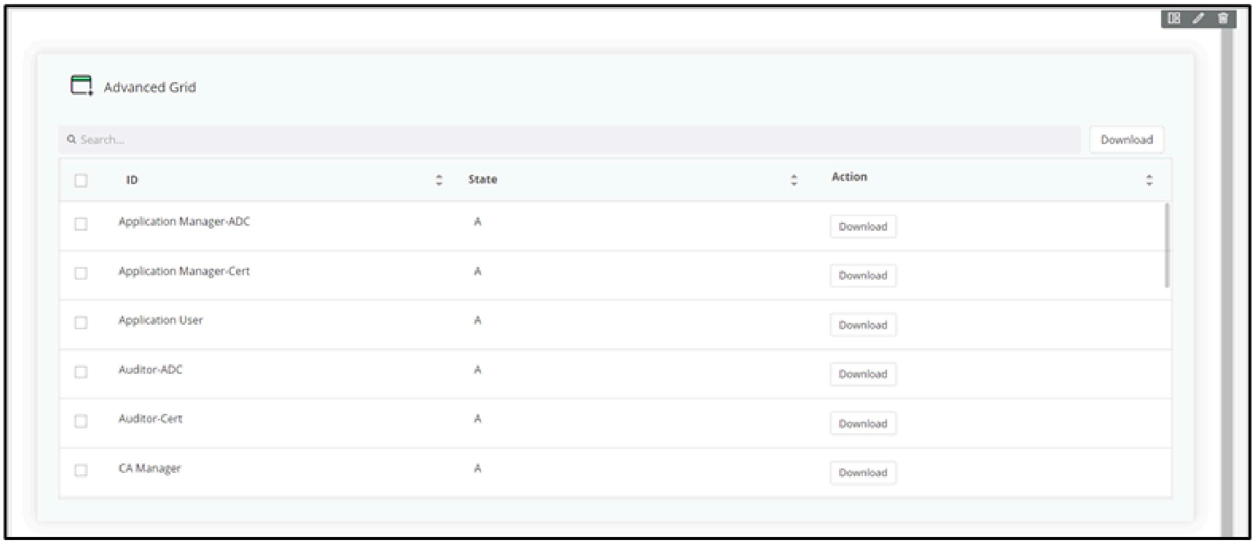
Field descriptions for Styles

Field	Description
Sizing	Allows you to increase/decrease the height and width of the widget.
Widget Background	The following options are available here: • **Add Background Pattern**: To add a background pattern to the widget, select the **Add Background Pattern** checkbox and select a pattern from the available options. • **Widget Background Color**: To add a background color to the widget, deselect the **Add Background Pattern** checkbox and select the color from the color palette. • **Widget Border Color**: Allows you to change the color of the widget border.
Widget Header	The following options are available here: • **Header Text Color**: Allows you to change the color of the header text. • **Header Text Size**: Allows you to select the size of the header text. • **Header Text Underline**: To underline the header text, select the Header Text Underline checkbox.
Widget Description	The following options are available here: • **Label text Color**: Allows you to change the color of the label text. • **Label Text Size**: Allows you to select the size of the label text.

2. To save your settings, click **Save**.
3. To restore the settings under **Styles** to default, click  **(Restore)** icon.

Grid

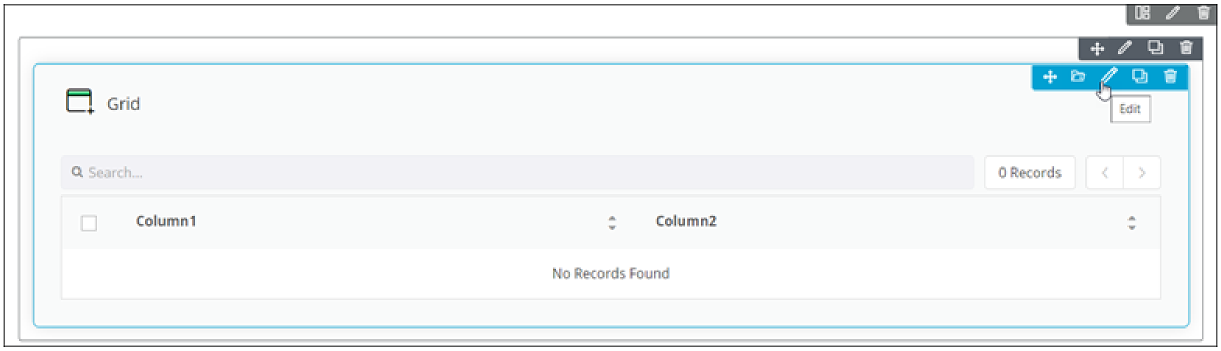
You can add a Grid to your service catalog. This widget allows you to display data in a standard layout with rows and columns.



- Configuring the Grid - Properties
- Configuring Columns in the Grid
- Configuring the Grid - Styles

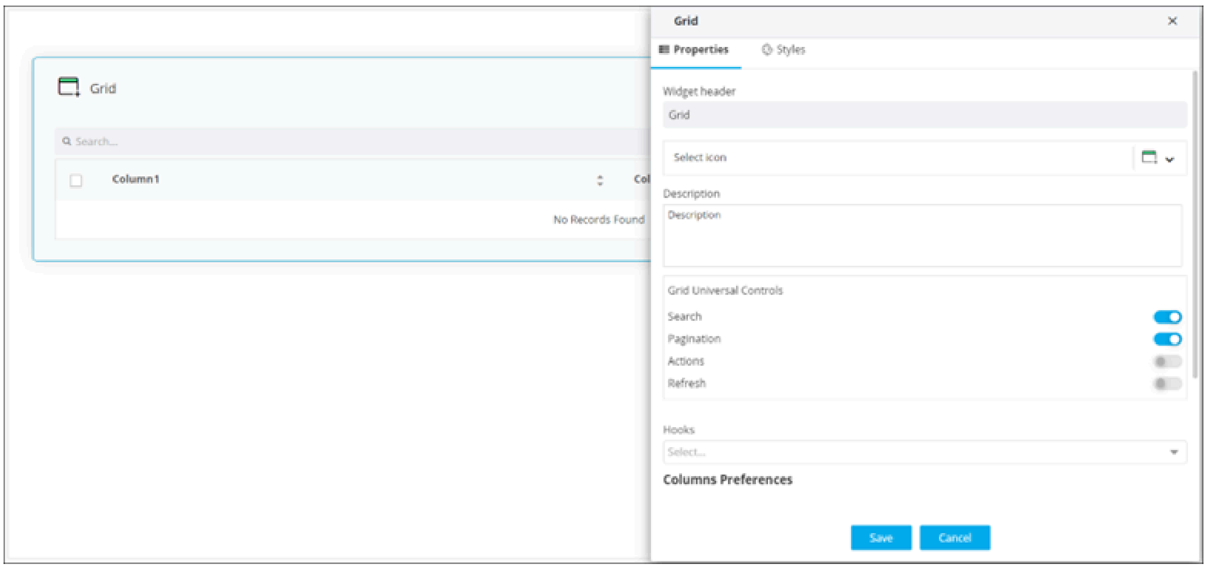
Configuring the Grid - Properties

1. From the **Build** component in the left menu, under the **Widgets** section, drag and drop the **Grid** widget to any blue highlighted space on the page.
2. To configure the **Grid**, hover your mouse over the field and click  **(Edit)** icon.



The **Grid** pop-up window opens with the **Properties** tab open by default.

3. In the **Grid** window, under **Properties**, enter or select the required field information.



Field descriptions for Properties

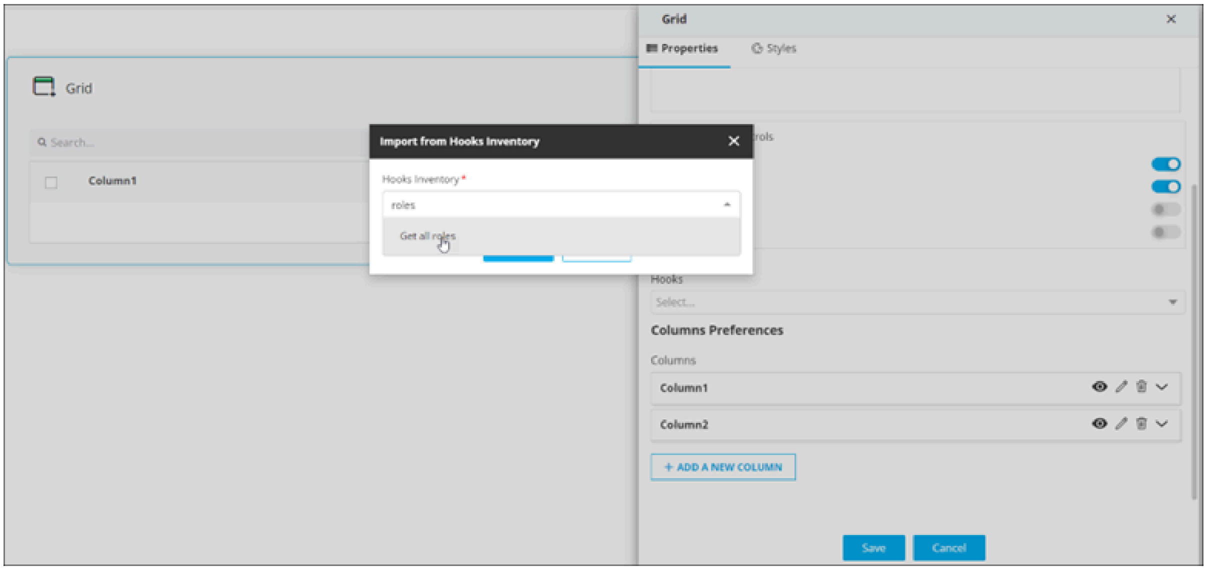
Field	Description
Widget header	Allows you to rename the widget header.
Select icon	Allows you to select an icon for the widget from the options available in the dropdown.
Description	Allows you to add a description for the widget.

Field	Description
Grid Universal Controls	Allows you to enable/disable the following: • **Search**:Turn off the toggle for Search if you don't want the Search bar to be displayed in the grid. Enabling the Search bar allows you to quickly search for a particular item in the grid. • **Pagination**: Turn on the toggle for Pagination to display the number of records that can be viewed on a page. Enabling Pagination allows a page with high volumes of data to load faster. • **Actions**: Turn off the Actions toggle if you don't want the Action buttons to be displayed in the grid. Enabling action buttons such as Add, Edit, and Delete allows you to quickly perform these actions and modify the grid.
Hooks	Allows you to assign an appropriate hook for this field. • Select a hook that is relevant to the form field. • Hooks can be either Script or REST API. • Hooks are used to fetch details against a specific form field from the device and/or database and map them to the respective field on the user interface. **Note:** For more information on Adding Hooks to Form Fields, click here.
Column Preferences	Allows you to configure the columns in the grid.

4. Click **Save**.

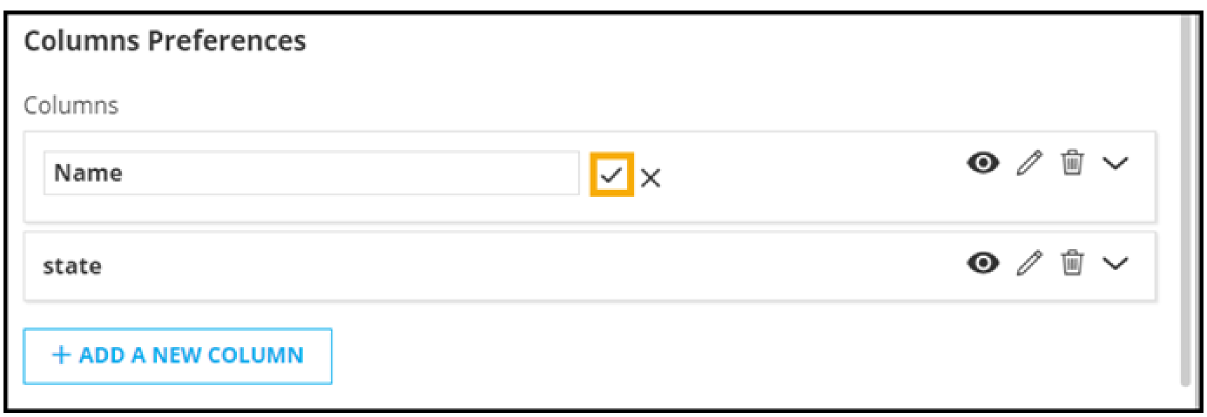
Configuring Columns in the Grid

1. In the **Grid** window, under **Properties**, assign a hook by selecting an option from the dropdown. For example, **Import** the **Get all roles** hook.



> **Note:** For more information on Adding Hooks to Form Fields, click here.

2. Under **Columns Preferences**, to change the name of the first column, click **(Edit)** icon.
3. Type the name of the column as required and click .



4. To configure the **Column Type**, click **(Dropdown)** icon.

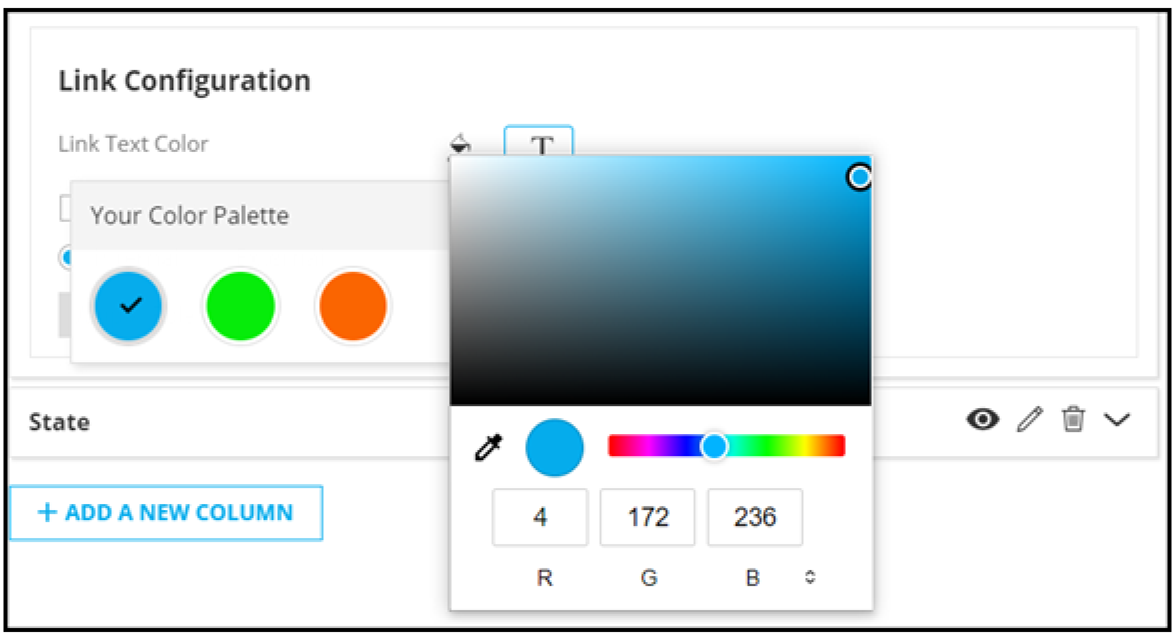
   The following options are available:

- **Plaintext**: Select this option to display the column items as plain text.
- **Link**: Select this option to add links to each column item.
- **Actions**: Select this option to add actions such as Download or Delete to the columns.

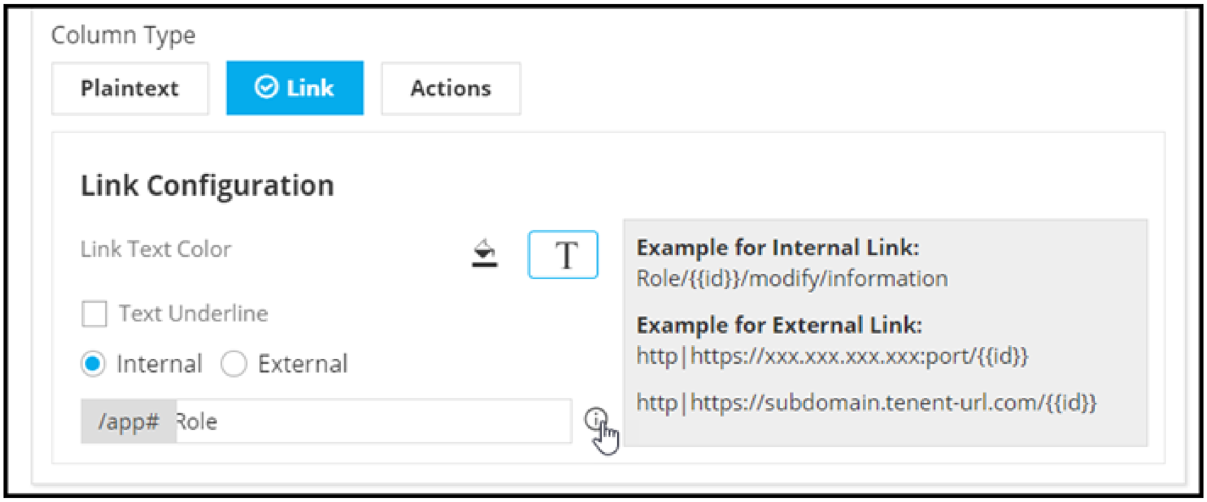
5. To configure links for the column items, under **Column Type**, click **Link**.

   The **Link Configuration** option is displayed.

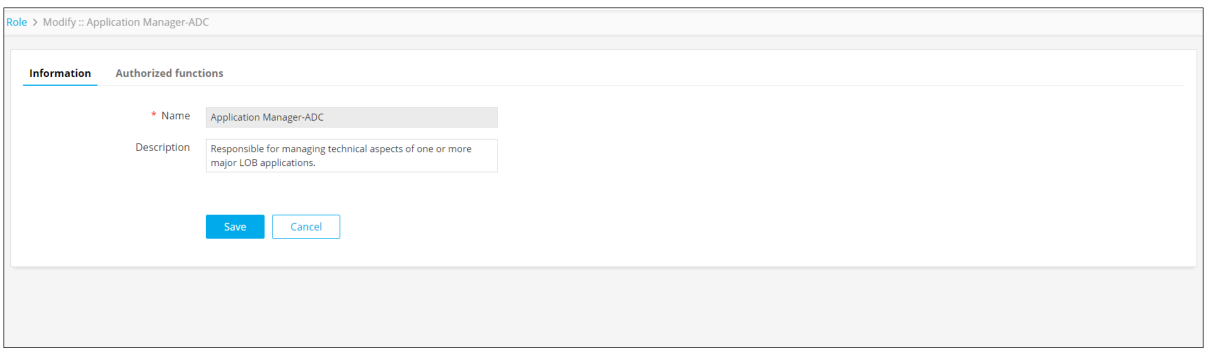
6. Select the color for the **Link Text**.



7. Select the **Text Underline** checkbox to underline the link text.
8. Select the type of link to be assigned to the column fields. For example, **Internal**.

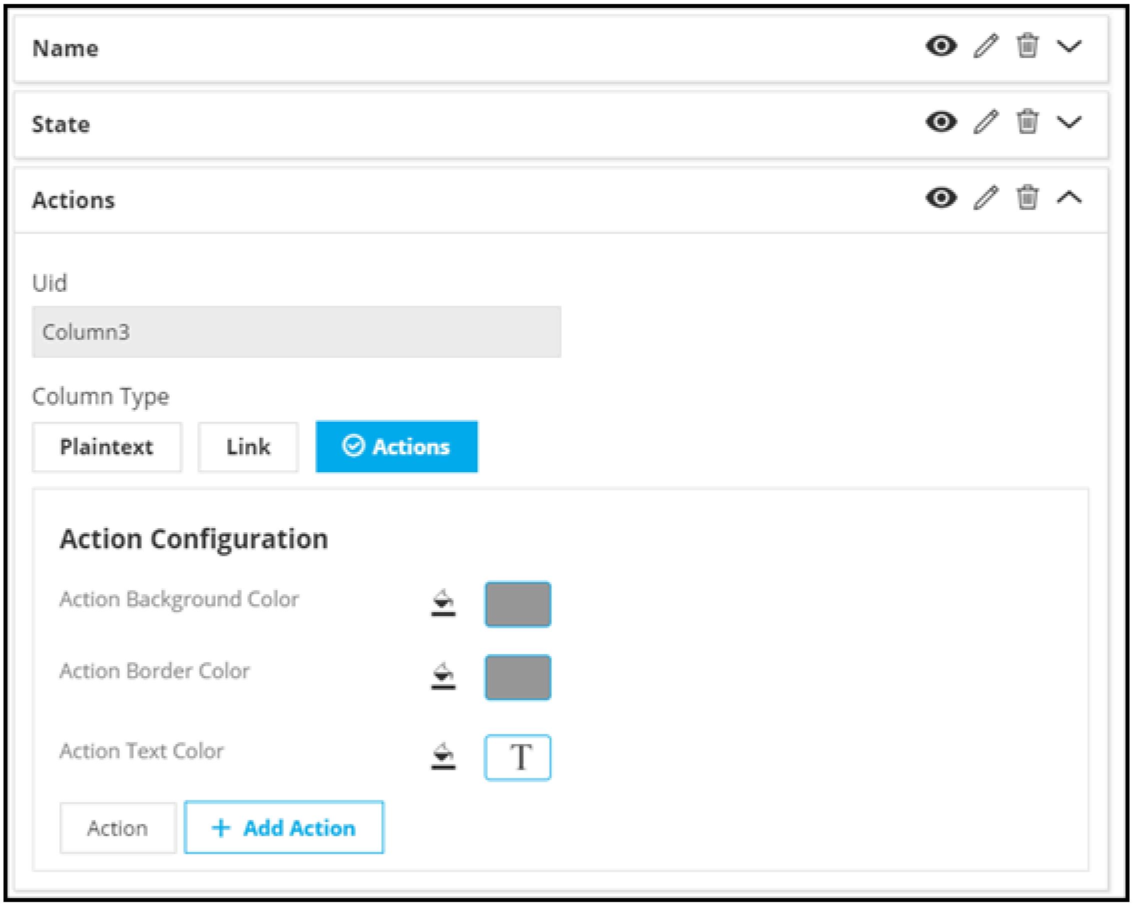


9. Clicking on any role under the **Name** column, will redirect to the linked page.



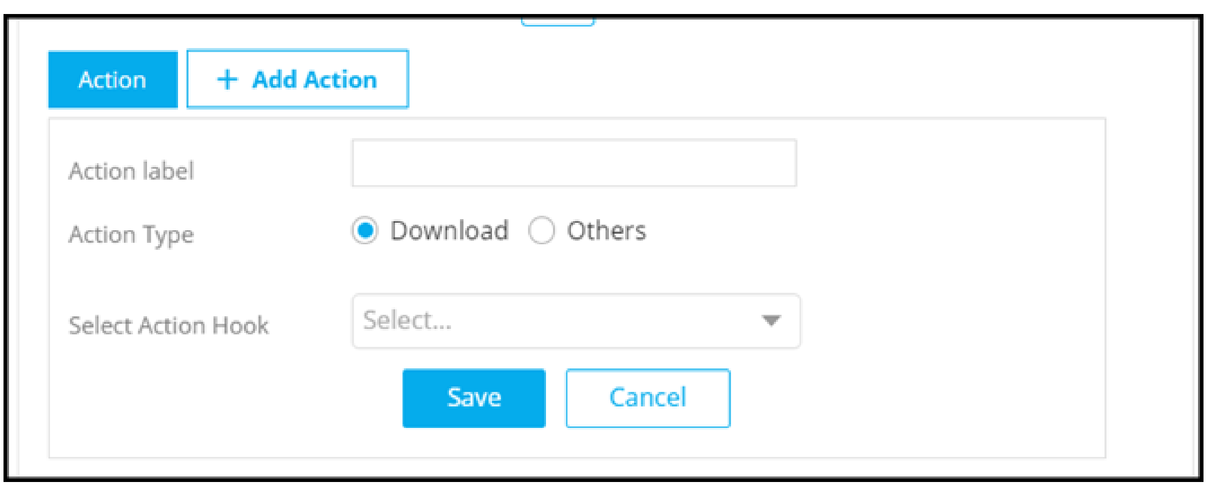
10. To configure an action for a column, under **Column Type**, click **Actions**.

    The **Action Configuration** option is displayed.



11. Select the **Action Background Color**.
12. Select the **Action Border Color**.
13. Select the **Action Text Color**.

14. To add an action for the column field, hover your mouse over **Action** and click **(Edit)** icon. More fields to add the action are displayed as shown.



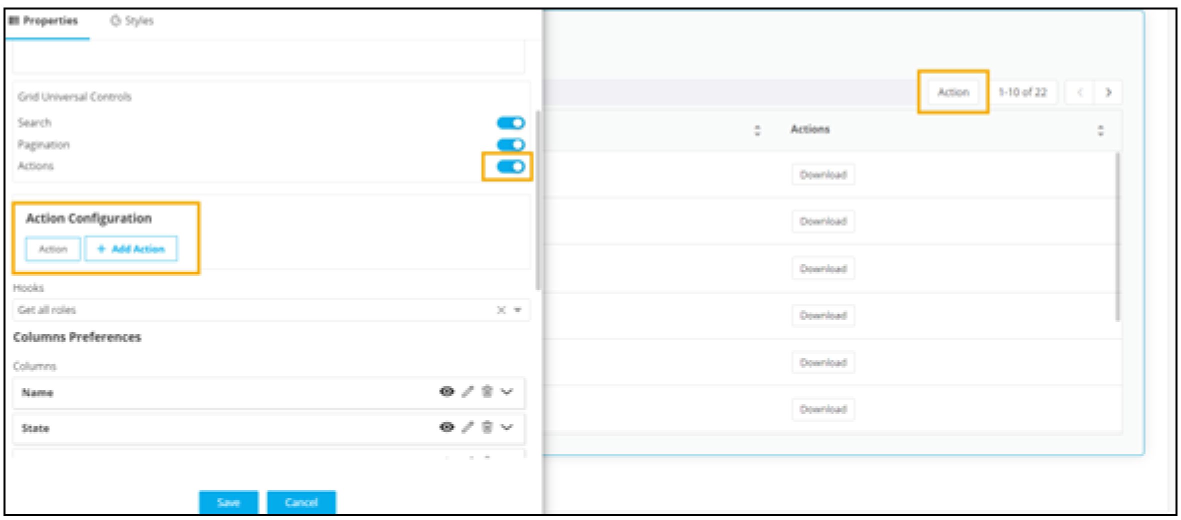
15. Enter the **Action label**. For example: **Download**.
16. Select the **Action Type** as **Download** or **Others**. For example, select Download to download the list of roles to your machine as a .csv file.
17. Select the hook to be assigned to the action. For example: **Role Download**.

> **Note:** For more information on Adding Hooks to Form Fields, click here.

18. Click **Save**.

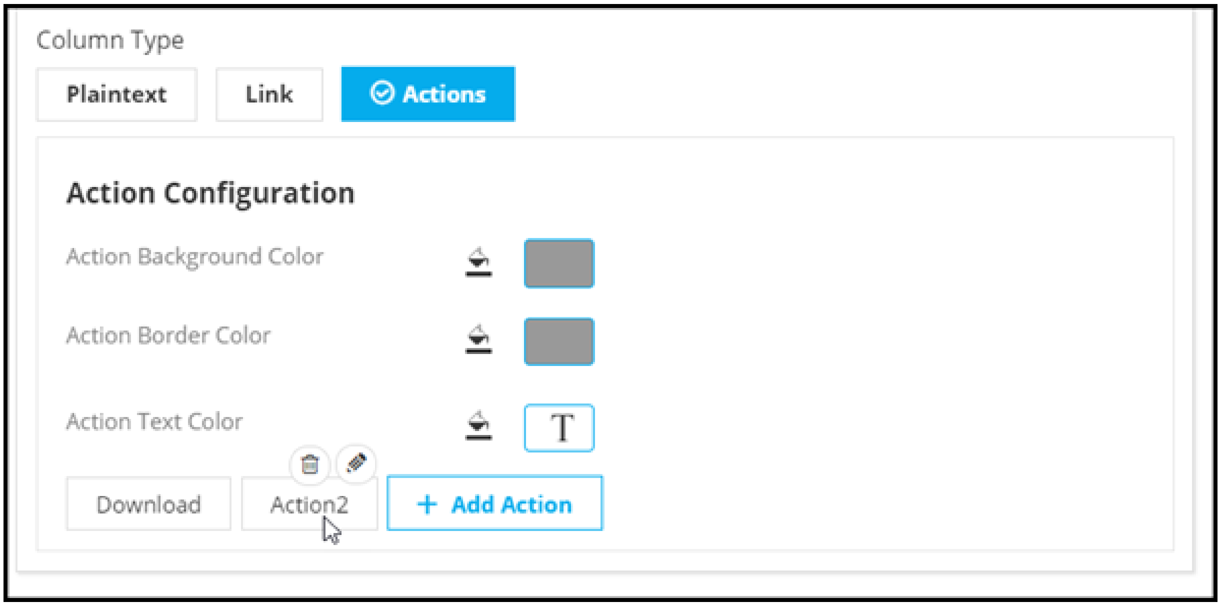
    The configured action is added to the grid.

> **Note:** You can also add actions to the grid by turning on the **Actions** toggle under **Grid Universal Controls**.



19. To add more actions to the grid, under **Action Configuration**, click **Add Action**.

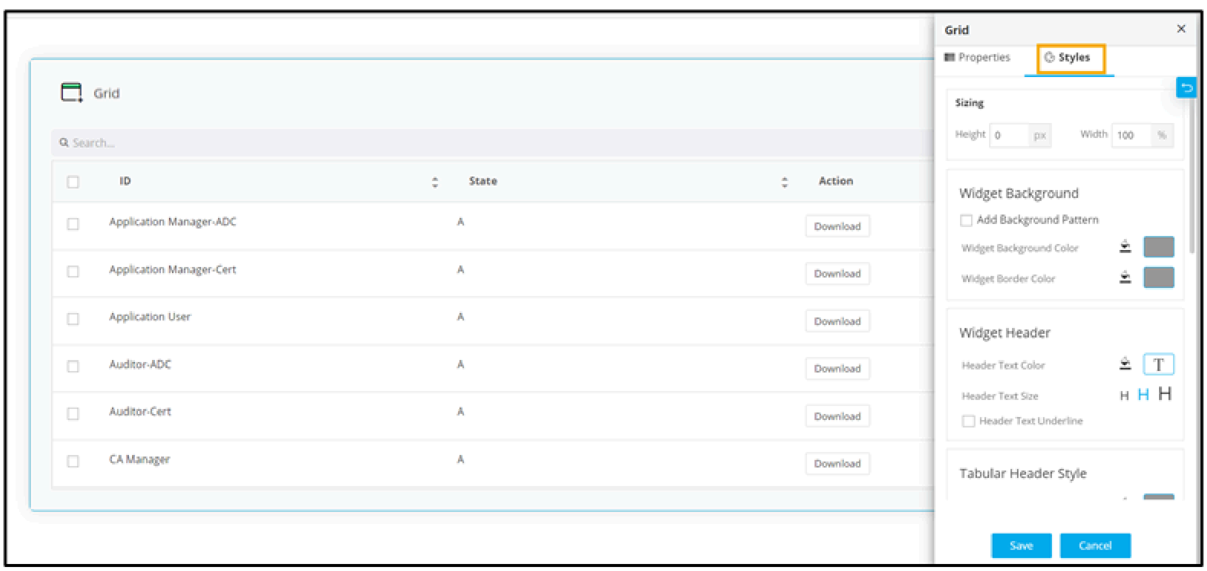
20. For options on configuring this action, hover your mouse over the action and click  **(Edit)** icon.



21. To delete this action, hover your mouse over the action and click  **(Delete)** icon.

Configuring the Grid - Styles

1. For more options to customize the Grid component, in the **Grid** window, click **Styles**.



Field descriptions for Styles

Field	Description
Sizing	Allows you to increase/decrease the height and width of the widget.

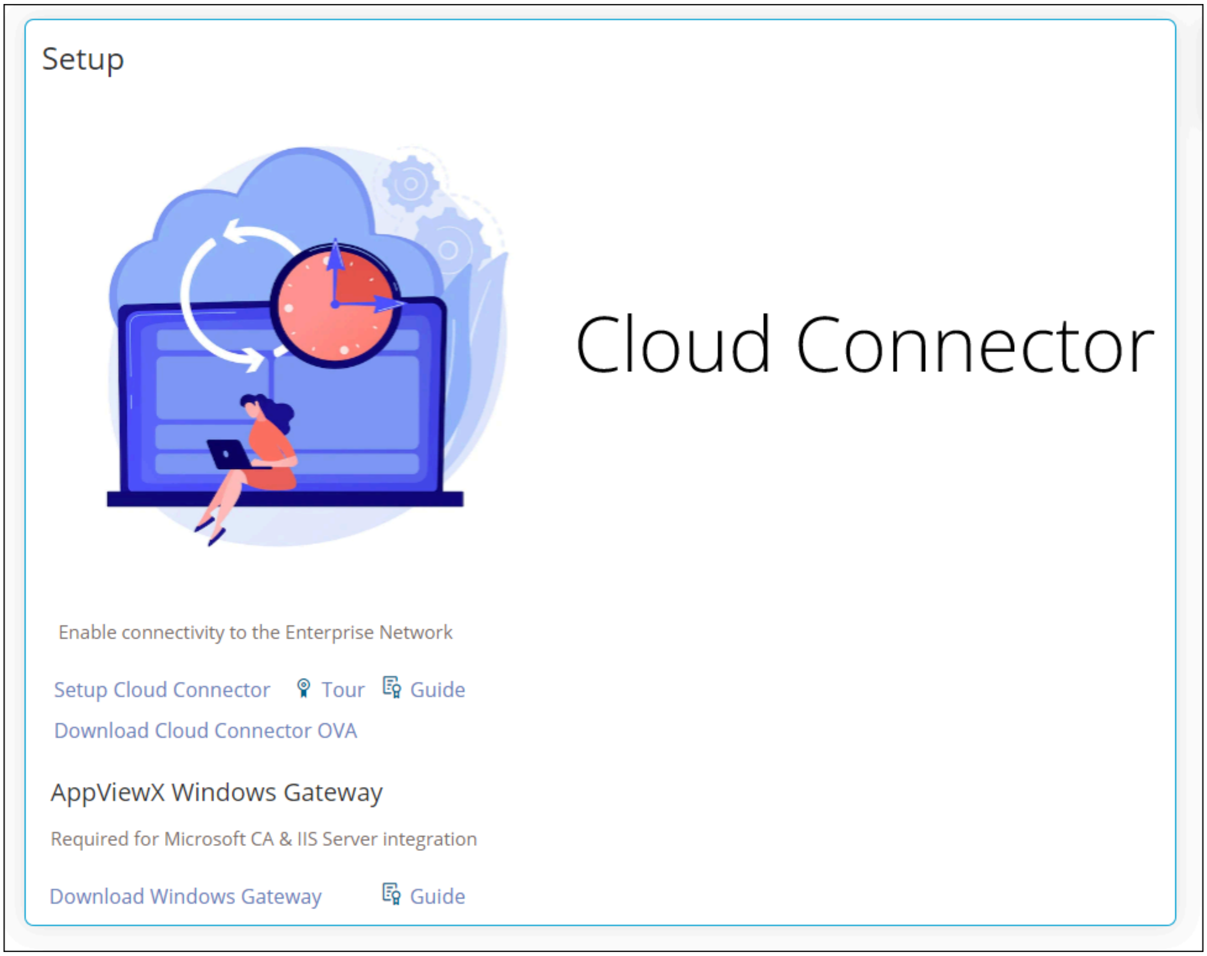
Field	Description
Widget Background	The following options are available here: • **Add Background Pattern**: To add a background pattern to the widget, select the **Add Background Pattern** checkbox and select a pattern from the available options. • **Widget Background Color**: To add a background color to the widget, deselect the **Add Background Pattern** checkbox and select the color from the color palette. • **Widget Border Color**: Allows you to change the color of the widget border.
Widget Header	The following options are available here: • **Header Text Color**: Allows you to change the color of the header text. • **Header Text Size**: Allows you to select the size of the header text. • **Header Text Underline**: To underline the header text, select the Header Text Underline checkbox.
Tabular Header Style	The following options are available here: • **Header Background Color** - Select the background color for the header from the available options. • **Header Text Color** - Select the header text color from the available options. • **Header Text Size** - Select the header text size from the available options. • **Header Text Alignment** - Select the header alignment as left, right or centre.
Tabular Row Style	The following options are available here: • **Row Background Color -** Select the background color for the row from the available options. • **Row Color** - Select the row color from the available options. • **Row Text Size -** Select the row text size from the available options. • **Row Text Alignment** - Select the row alignment as left, right or centre.
Icon Style	The following options are available here: • **Icon Color** - Select the Icon Color from the available options. • **Icon Background** - Select the Icon Background from the available options. • **Icon Background Color -** Select the Icon Background Color from the available options.

2. To save your settings, click **Save**.

3. To restore the settings under **Styles** to default, click  **(Restore)** icon.

Card

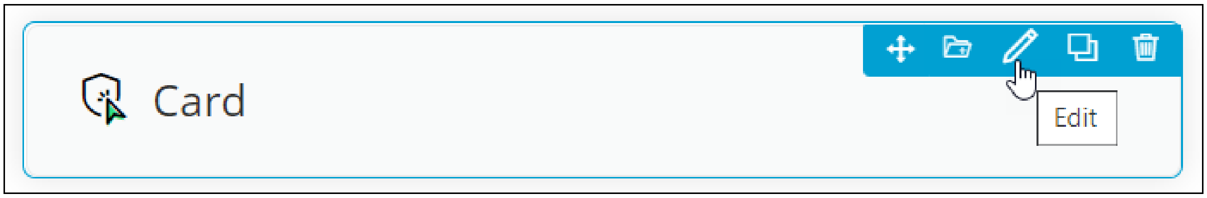
You can add a Card component to your service catalog. This widget allows you to add multiple links, buttons, and content.



- Configuring the Card - Properties
- Configuring Rich Text Editor in the Card Widget
- Configuring Lists in the Card Widget
- Configuring a Button in the Card Widget
- Configuring Grouping in the Card Widget
- Configuring Images in the Card Widget
- Configuring the Card - Styles

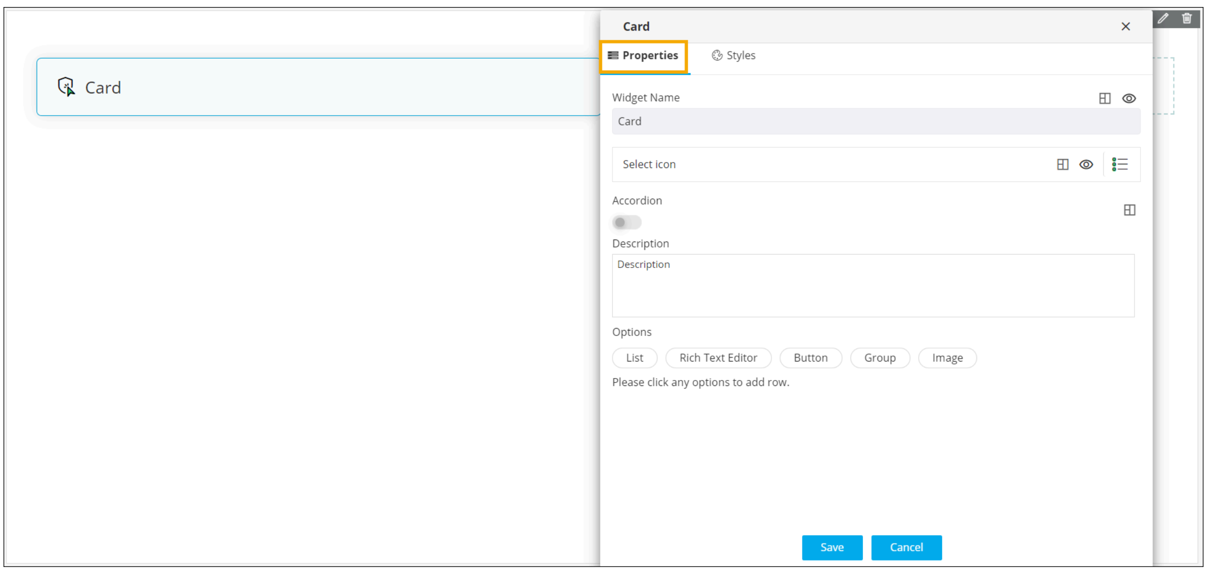
Configuring the Card - Properties

1. From the **Build** component in the left menu, under the **Widgets** section, drag and drop the **Card** widget to any blue highlighted space on the page.
2. To configure the **Card** widget, hover your mouse over the widget and click  **(Edit)** icon.



The **Card** pop-up window opens with the **Properties** tab open by default.

3. In the **Card** widget window, under **Properties**, enter or select the required field information.



Field descriptions for Properties

Field	Description
Widget Name	Allows you to enter the name for the card widget. Styling options available for the Widget name are: • - Allows you to align the widget name to the left, right, center of the widget.  • / - Allows you to show or hide the widget name.
Select icon	Allows you to select an icon for the widget. Styling options available for the icon are: • - Allows you to align the icon to the left, right, or adjacent to the widget name.  • / - Allows you to show or hide the icon.
Accordion	• Turn this toggle on or off to hide or show large volumes of content in the card widget. • Turning the toggle on displays options to select icons to Expand and Collapse the content in the card widget.

<table>
<tr><th>Field</th><th>Description</th></tr>
<tr><td></td><td>• Click to align the Expand and Collapse icons to the left or right of the widget.

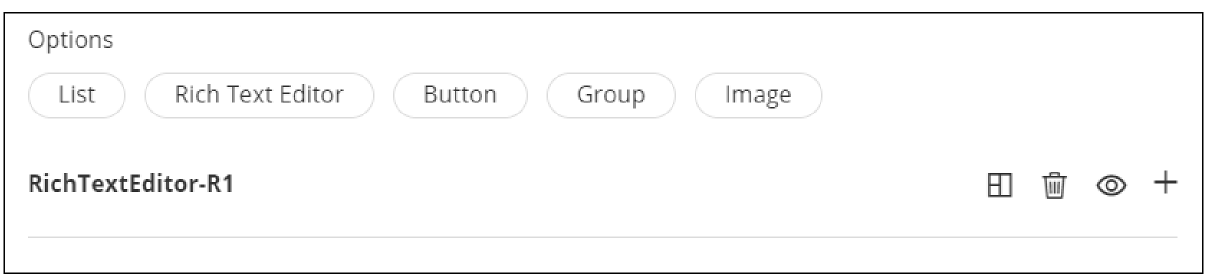
</td></tr>
<tr><td>Description</td><td>Allows you to add a description for the card widget.</td></tr>
<tr><td>Options</td><td>Allows you to add the following components to the card widget:
• List
• Rich Text Editor
• Button
• Group
• Image.</td></tr>
</table>

4. Click **Save**.

Configuring Rich Text Editor in the Card Widget

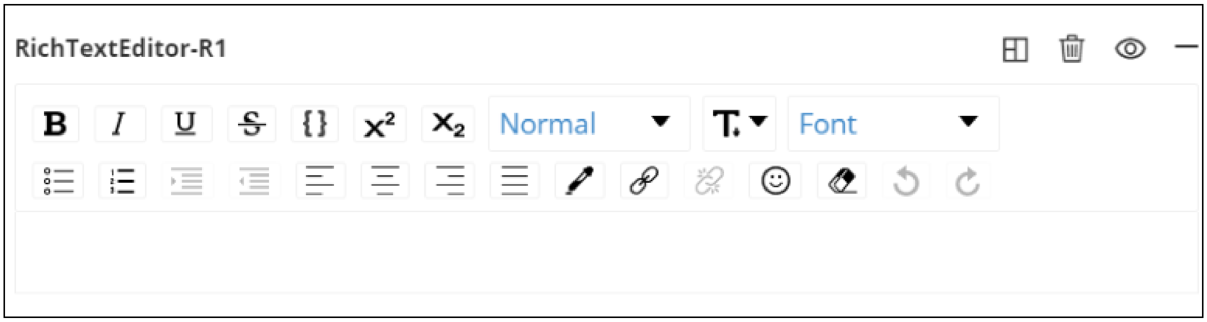
1. In the **Card** window, in the **Properties** tab, under **Options**, click **Rich Text Editor**.

   The **List** row is displayed.



2. To expand the row and add a Rich Text Editor, click **(Add)** icon.

   The **Rich Text Editor** is displayed.



Note: For more information on the Rich Text Editor, click here.

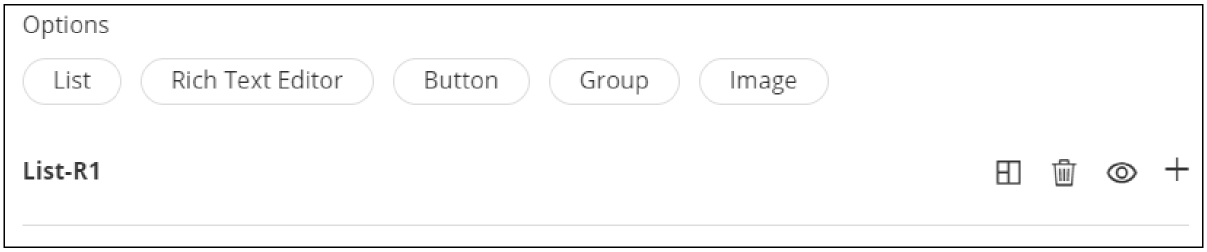
3. To hide the Rich Text Editor, click **(View)** icon. To display it again click **(Hide)** icon.
4. To delete the Rich Text Editor, click **(Delete)** icon.
5. To define the padding, margins, background color, and font size for the Rich Text Editor within the card, click **(Layout)** icon.

6. Once you have configured all the settings, click **Save**.

Configuring Lists in the Card Widget

1. In the **Card** window, in the **Properties** tab, under **Options**, click **List**.

   The **List** row is displayed.



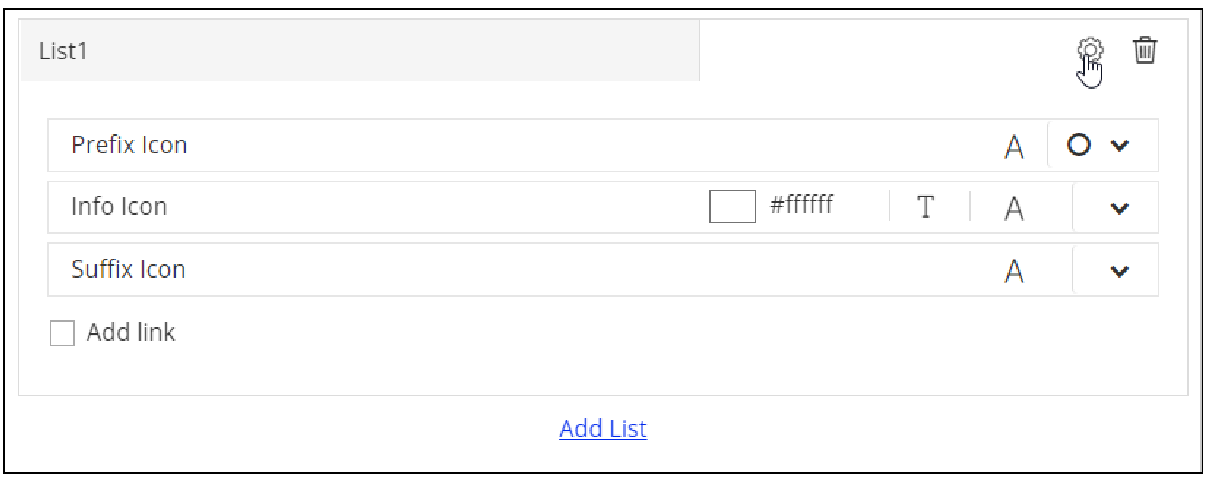
2. To expand the row and add a list, click + **(Add)** icon.

   A new row is displayed with the default name as **List1**. You can rename this by typing over the existing name.

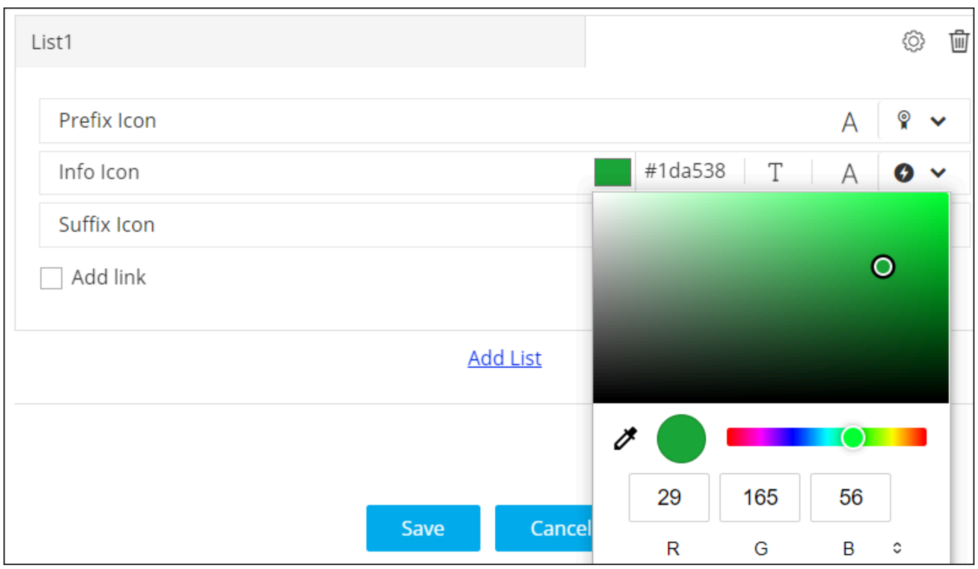
3. In the **List1** row, to configure the list, click **(Settings)** icon.

   The following options are displayed as shown in the screenshot below.

   - **Prefix Icon**
   - **Info Icon**
   - **Suffix Icon**
   - **Add link**.



Options descriptions

Option	Description
Prefix Icon	• To add a prefix icon next to the list name, click and select the required icon from the options available. • To increase or decrease the size of the prefix icon, click and set the icon size.
Info Icon	• To add an Info icon next to the list name, click and select the required icon from the options available. • To increase or decrease the size of the Info icon, click and set the icon size. • To select a color for the info icon, click the color picker box and select the required color.  • To add info text, click and enter the text in the textbox.
Suffix Icon	• To add a suffix icon next to the list name, click and select the required icon from the options available. • To increase or decrease the size of the suffix icon, click and set the icon size.

4. To add a link for the list, select the **Add link** checkbox.
   Option to add **Internal**, **Others** (External), and **Workflow** link (s) is displayed.

> **Note:** For more information on adding **Internal** link(s), click here.

Note: For more information on adding **Others** (External) link(s), click here.

Note: For more information on adding **Workflow** link(s), click here.

5. To hide the List row, click **(View)** icon. To display it again click **(Hide)** icon.

6. To delete the List row, click **(Delete)** icon.

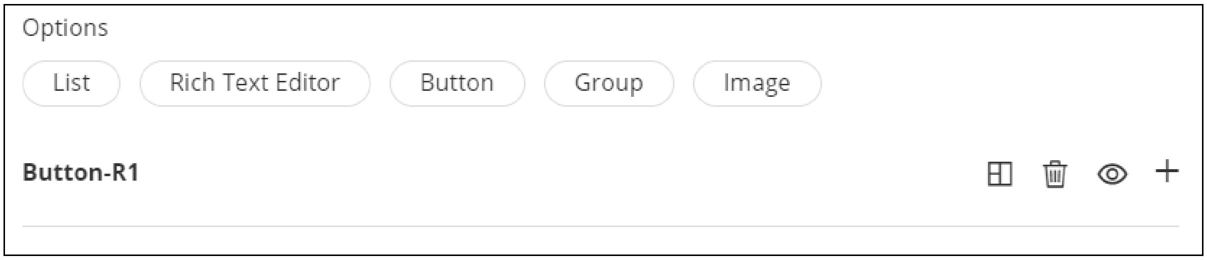
7. To define the padding, margins, background color, and font size for the List within the card, click **(Layout)** icon.

8. Once you have configured all the settings, click **Save**.
9. To add another list, click **Add List**.

Configuring a Button in the Card Widget

1. In the **Card** window, in the **Properties** tab, under **Options**, click **Button**.

   The **Button** row is displayed.

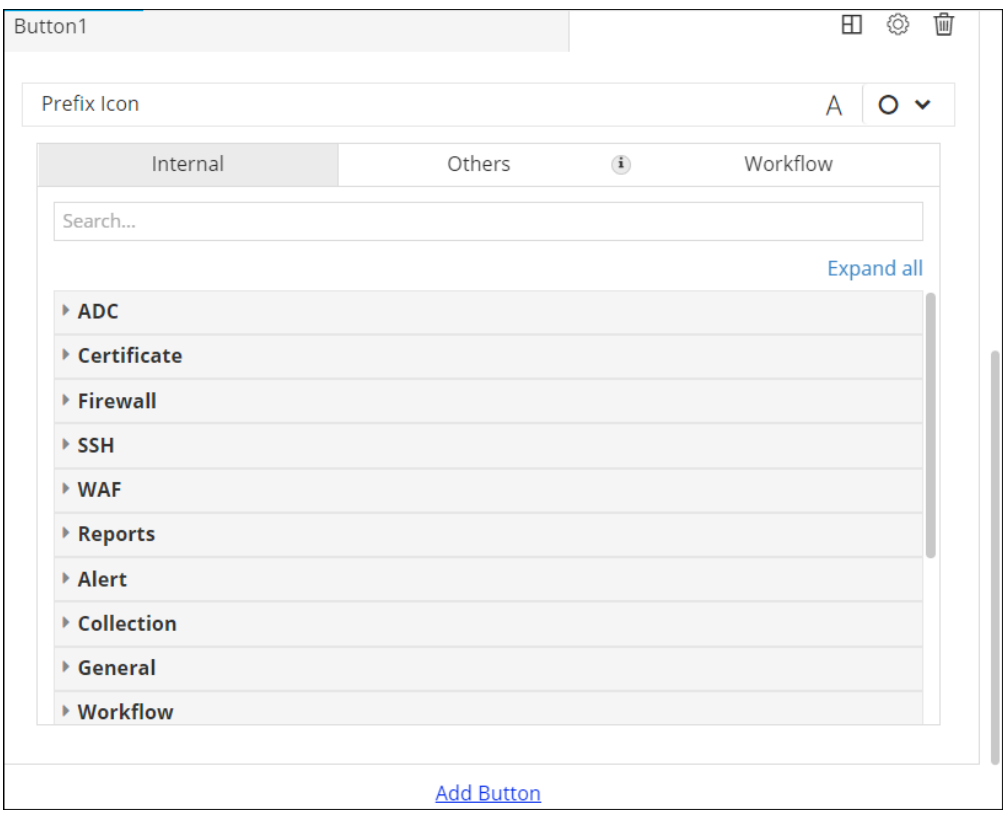


2. To expand the row and add a list, click **(Add)** icon.
   A new row is displayed with the default name as **Button1**. You can rename this by typing over the existing name.

3. In the **Button1** row, to configure the list, click **(Settings)** icon.

   The following options are displayed as shown in the screenshot below:

   - **Prefix Icon** - To add a prefix icon next to the Button click and select an icon from the options available. To change the size of the icon, click .
   - **Add link** - Allows you to add **Internal**, **Others** (External), and **Workflow** link(s) for the Button.

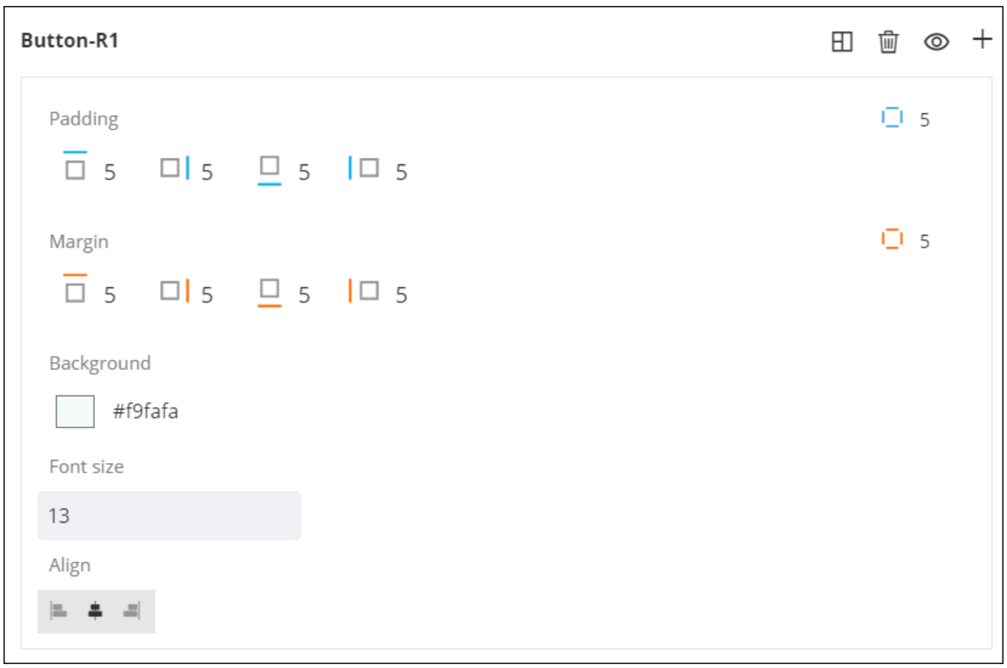


> **Note:** For more information on adding **Internal** link(s) to a Button, click here.

> **Note:** For more information on adding **Others** (External) link(s) to a Button, click here.

> **Note:** For more information on adding **Workflow** link(s) to a Button, click here.

4. To hide the Button row, click **(View)** icon. To display it again click **(Hide)** icon.
5. To delete the Button row, click **(Delete)** icon.
6. To define the padding, margins, background color, font size, and alignment for the Button within the card, click **(Layout)** icon.



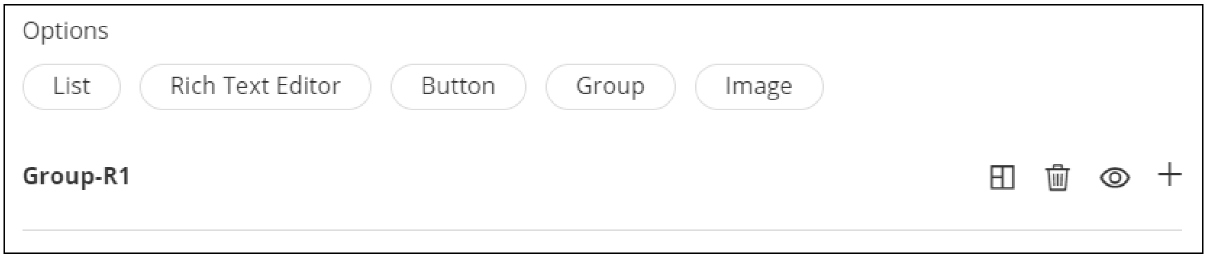
7. Once you have configured all the settings, click **Save**.
8. To add another button click **Add Button**.

Configuring Grouping in the Card Widget

This feature allows you to group card elements such as list(s), rich text editor, and button(s) within a group.

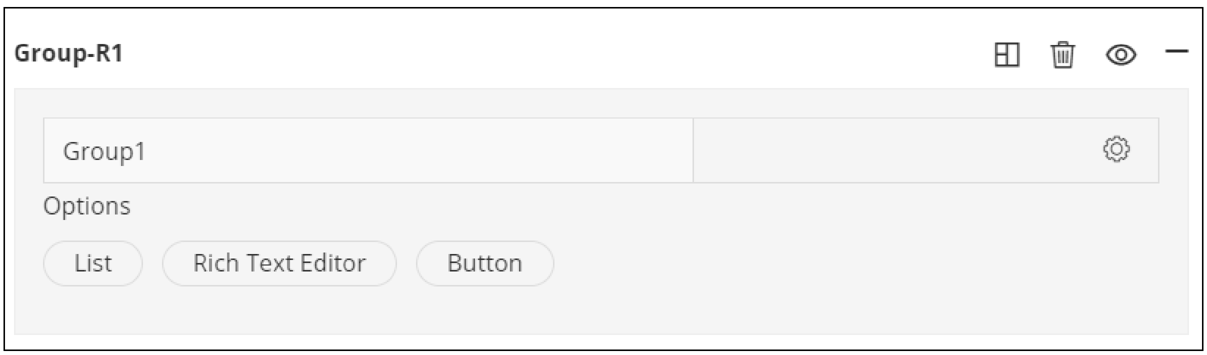
1. In the **Card** window, in the **Properties** tab, under **Options**, click **Group**.

   The **Group** row is displayed.

2. To expand the row and add a list, click + **(Add)** icon.

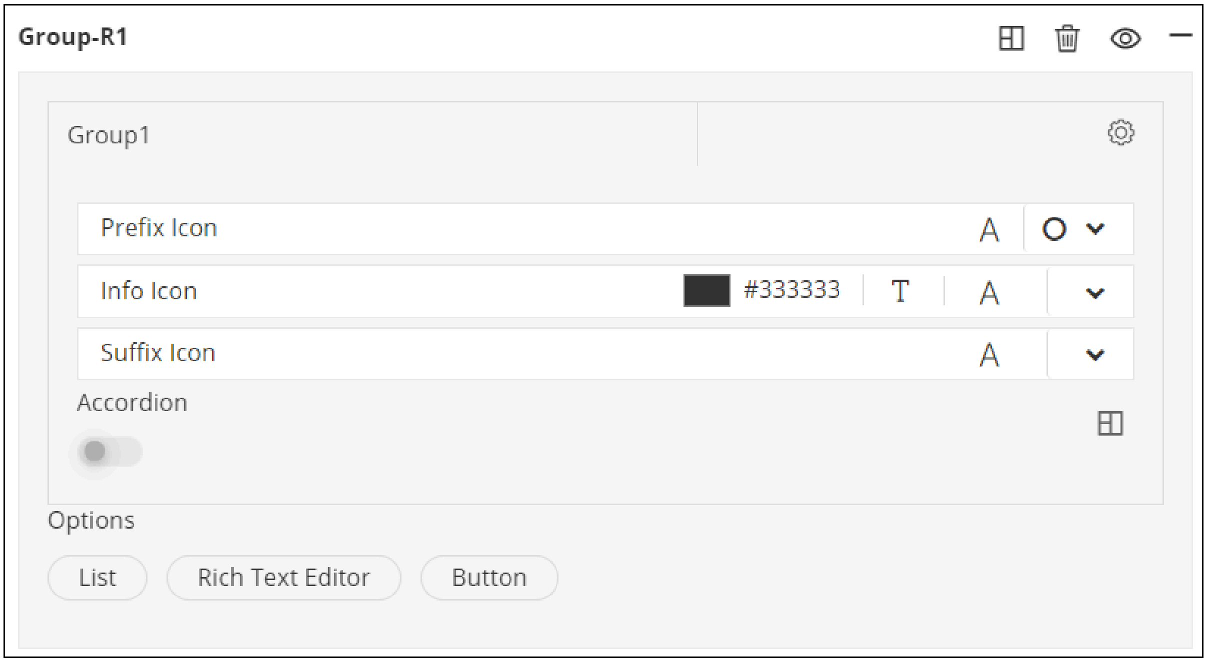
   A new row is displayed with the default name as **Group1**. You can rename this by typing over the existing name.

3. In the **List1** row, to configure the list, click **(Settings)** icon.

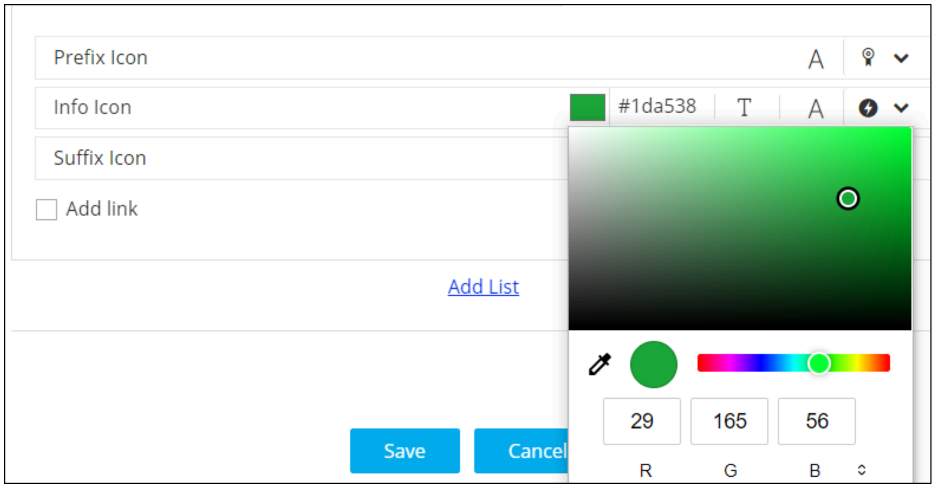
   The following options are displayed as shown in the screenshot below.

   - **Prefix Icon**
   - **Info Icon**
   - **Suffix Icon**
   - **Accordion**.



Options Descriptions

Option	Description
Prefix Icon	• To add a prefix icon next to the list name, click and select the required icon from the options available. • To increase or decrease the size of the prefix icon, click and set the icon size.
Info Icon	• To add an Info icon next to the list name, click and select the required icon from the options available. • To increase or decrease the size of the Info icon, click and set the icon size. • To select a color for the info icon, click the color picker box and select the required color.

Option	Description
	 • To add info text, click and enter the text in the textbox.
Suffix Icon	• To add a suffix icon next to the list name, click and select the required icon from the options available. • To increase or decrease the size of the suffix icon, click and set the icon size.
Accordion	Allows you to hide or show large volumes of content in the card widget by turning this toggle on or off. You can select icons to Expand and Collapse the content in the card widget and also set the placement of these icons.

4. To add a **List**, **Rich Text Editor**, or **Button** in the Group, click on the respective link under **Options**.

Note: For steps on configuring a list, click here.

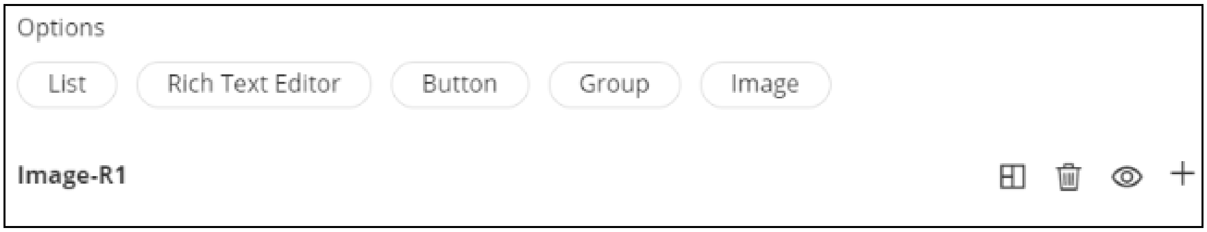
Note: For steps on configuring a Rich Text Editor, click here.

Note: For steps on configuring a Button, click here.

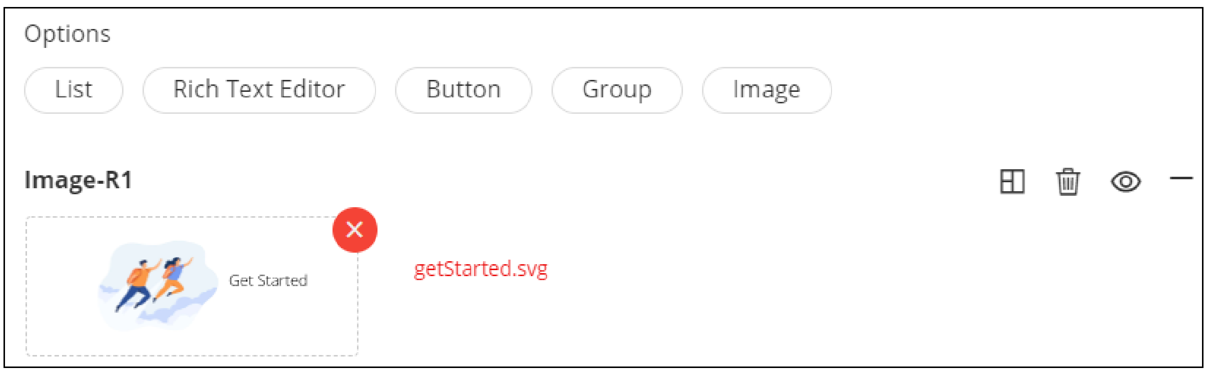
Configuring Images in the Card Widget

1. In the **Card** window, in the **Properties** tab, under **Options**, click **Image**.

   The **Image** row is displayed.



2. To expand the row and add an image, click  **(Add)** icon.
3. To upload an image in the card widget, click  **(Upload)** icon and upload the required image from your computer.



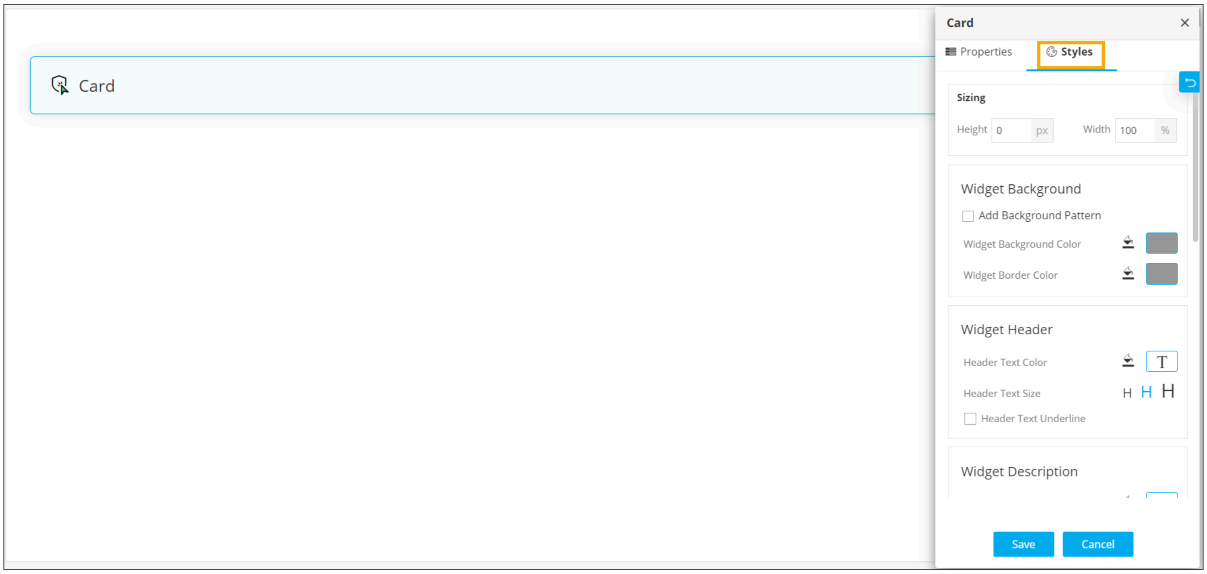
Note: Allowed image formats are: .svg, .jpeg, and .png

4. To delete this image and select another, click  **(Delete)** icon.
5. To add another image, under **Options**, click **Image**.

   A new **Image** row is displayed.
6. Once you have added the required image(s), click **Save**.

Configuring the Card - Styles

1. For more options to customize the Card component, in the **Card** window, click **Styles**.



Field descriptions for Styles

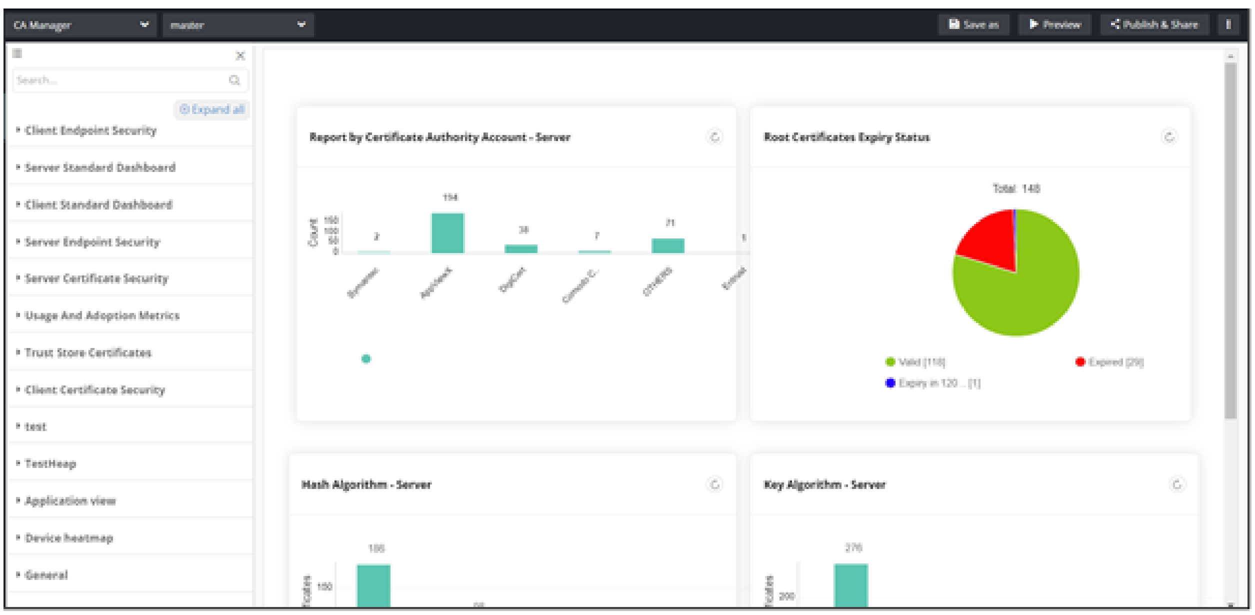
Option	Description
Sizing	Allows you to increase/decrease the height and width of the widget.
Widget Background	The following options are available here: • **Add Background Pattern**: To add a background pattern to the widget, select the **Add Background Pattern** checkbox and select a pattern from the available options. • **Widget Background Color**: To add a background color to the widget, deselect the **Add Background Pattern** checkbox and select the color from the color palette. • **Widget Border Color**: Allows you to change the color of the widget border.
Widget Header	The following options are available here: • **Header Text Color**: Allows you to change the color of the header text. • **Header Text Size**: Allows you to select the size of the header text. • **Header Text Underline**: To underline the header text, select the Header Text Underline checkbox.
Widget Description	The following options are available here: • **Label text Color**: Allows you to change the color of the label text. • **Label Text Size**: Allows you to select the size of the label text.

Option	Description
Icon Style	Allows you define the following style settings: • **Icon Color -** You can select the color for the icon from the available options. • **Icon Background -** You can select the background for the icon from the available options. • **Icon Background Color -** You can select the background color for the icon from the available options.
Widget Layout Style	Allows you to define the padding and margin of the Card widget.

2. To save your settings, click **Save**.
3. To restore the settings under **Styles** to default, click **(Restore)** icon.

Reports

You can access specific reports based on user roles from the Service Catalog, enabling you to view all reports on a single page.

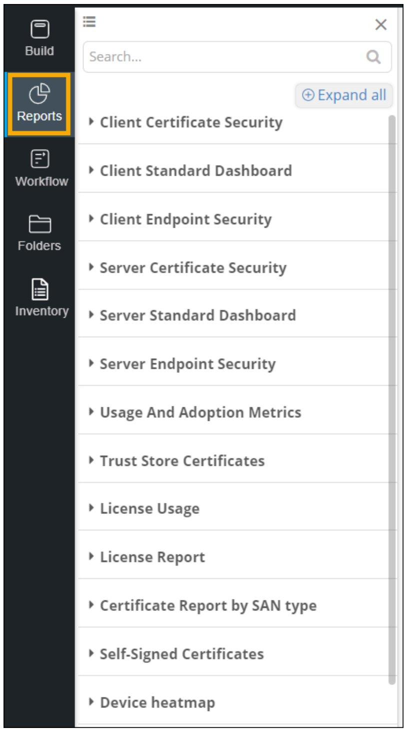


Note: For more information on **Reports**, refer to the Reporting User Guide.

To access Reports:

1. From the component panel on the left, click **Reports.**

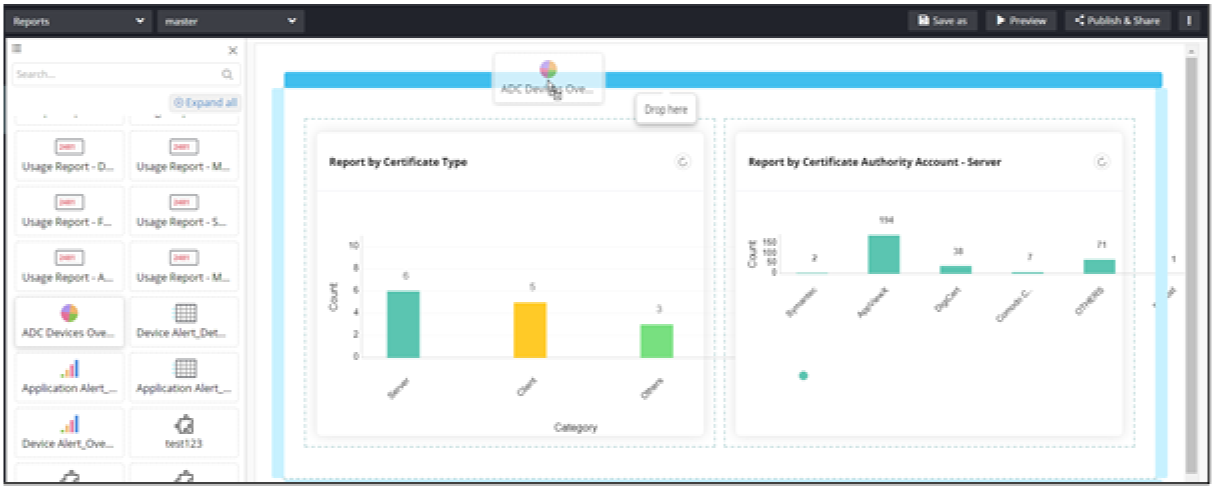
   The **Reports** inventory is displayed.



Options descriptions for Reports

Options	Description
View	Allows you to switch between list view and grid view when the Reports Inventory is expanded.
Search bar	Allow you to search for a specific report.
Collapse All/ Expand All	Allows you to expand/collapse the Reports inventory.

2. To add a report to your page, drag and drop the report to any blue highlighted space in the build area.

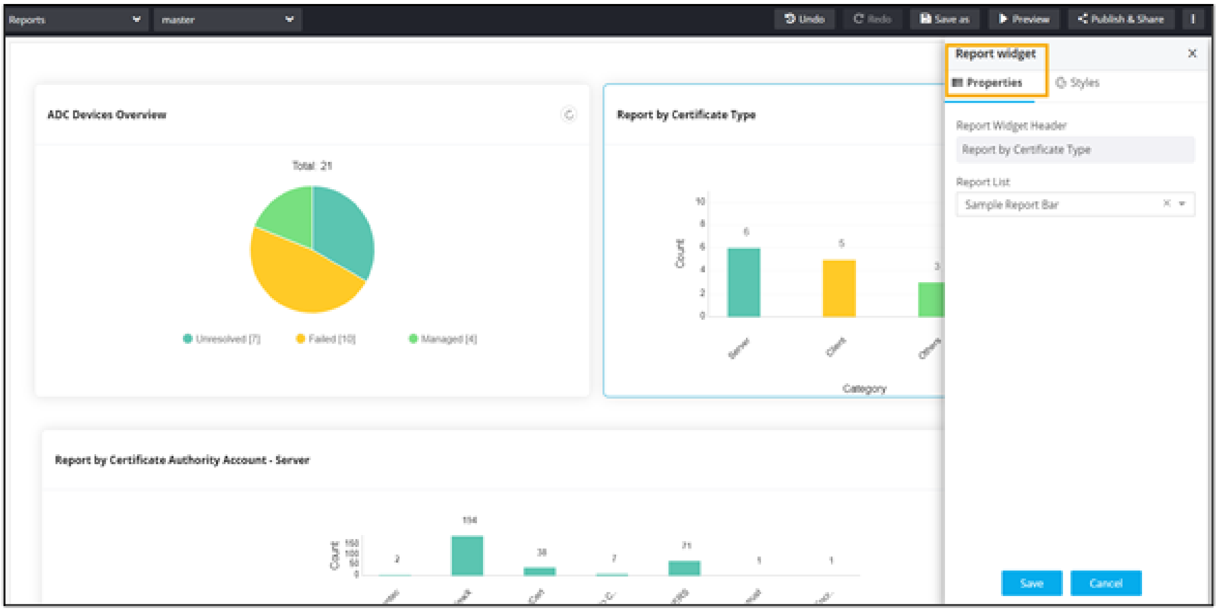


- Configuring Reports - Properties
- Configuring Reports - Styles
- Pinning Application Widgets
- Pinning Device heatmap Widget

Configuring Reports - Properties

1. To configure/edit the **Reports** component, hover your mouse over the widget and click  **(Edit)** icon.

The **Report widget** window opens with the **Properties** tab open by default.



2. In the **Report widget** window, under **Properties**, enter or select the required field information.

Field descriptions for Properties

Field	Description
Report Widget Header	Allows you to provide a name for the Report Widget.
Report List	Allows you to select reports from the options available in the dropdown. **Note:** Selecting a report from this dropdown will replace the existing report on the page.

3. Click **Save**.

Configuring Reports - Styles

1. For more options to customize the **Report widget**, in the **Report widget** window, click **Styles**.

Field descriptions for Styles

Field	Description
Sizing	Allows you to increase/decrease the **Size** of the Report widget.
Spacing	Allows you to set the **Padding** and **Margin** of the Report widget header.
Widget Background	The following options are available here: • **Add Background Pattern**: To add a background pattern to the widget, select the **Add Background Pattern** checkbox and select a pattern from the available options. • **Widget Background Color**: To add a background color to the widget, deselect the **Add Background Pattern** checkbox and select the color from the color palette. • **Widget Border Color**: Allows you to change the color of the widget border.
Widget Header	The following options are available here: • **Header Text Color**: Allows you to change the color of the header text. • **Header Text Size**: Allows you to select the size of the header text. • **Header Text Underline**: To underline the header text, select the Header Text Underline checkbox.

Field	Description
Icon Style	The following options are available here: • **Icon Color** - Select the Icon Color from the available options. • **Icon Background** - Select the Icon Background from the available options. • **Icon Background Color** - Select the Icon Background Color from the available options.

2. To save your settings, click **Save**.
3. To restore the settings under **Styles** to default, click **(Restore)** icon.

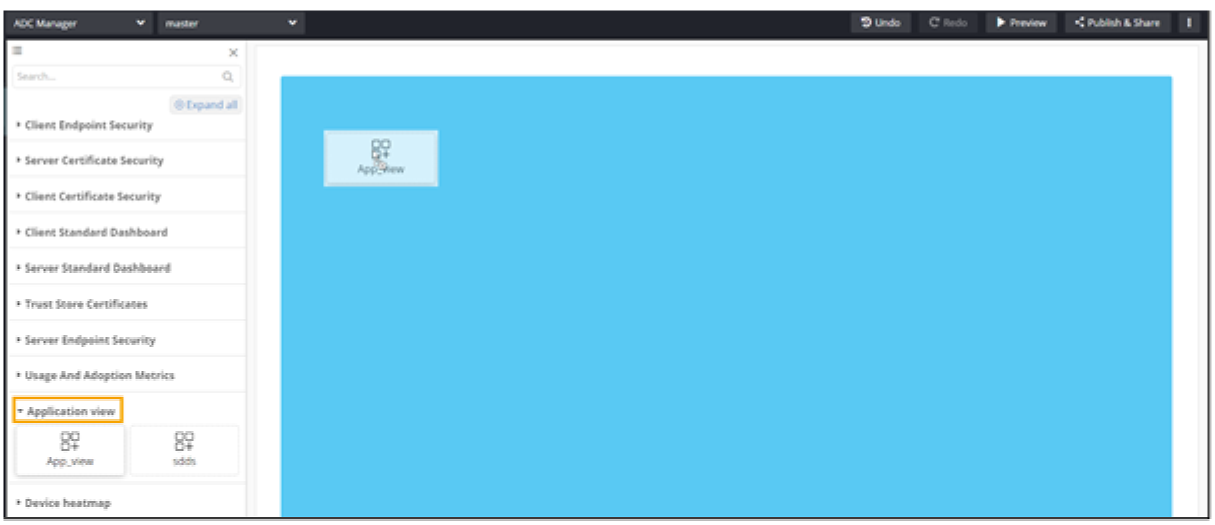
Pinning Application Widgets

Through the Service Catalog, you can pin Application view and Device HeatMap widgets to your page. To be able to pin these widgets to your page, they must be configured on the ADC dashboard first.

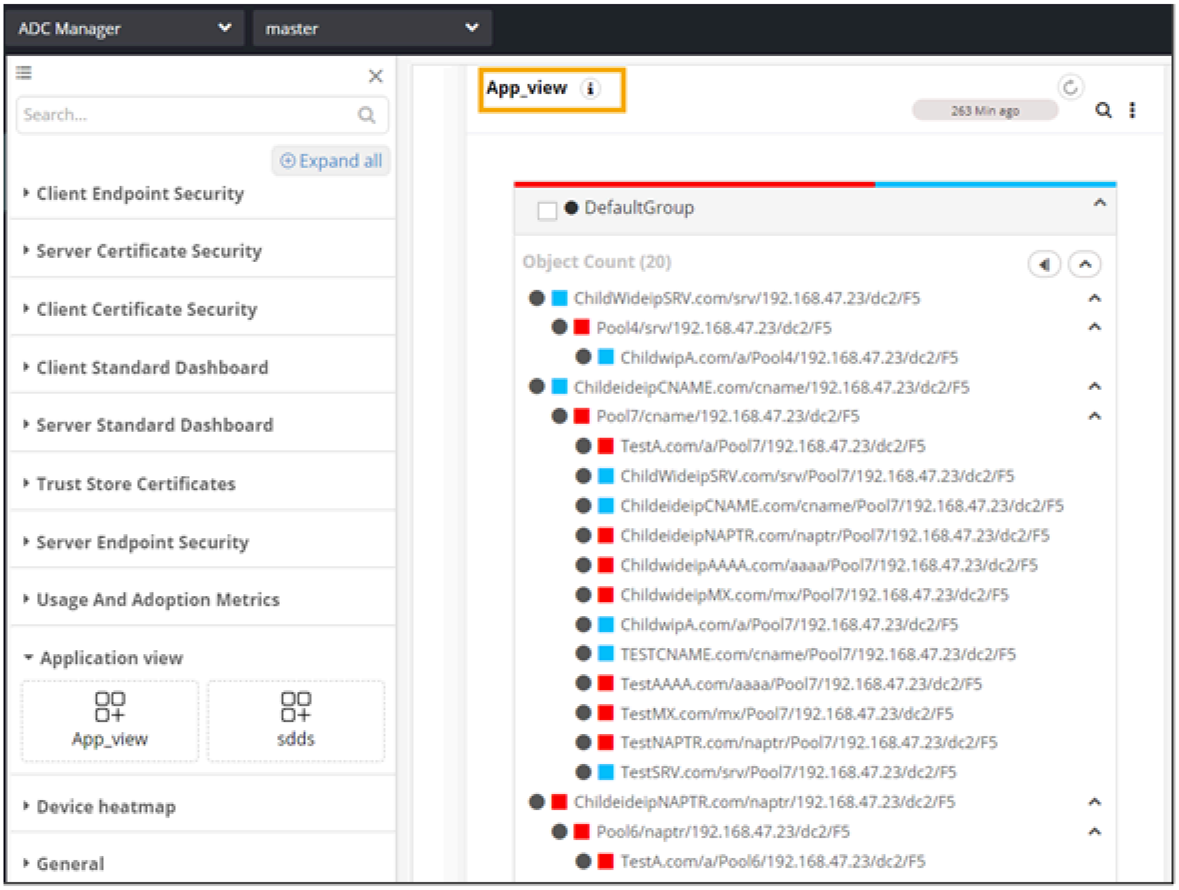
The Application View widget displays all the configured devices in your database. Through the widget you can also perform actions such as enabling/disabling a device, view logs and alerts, backup and restore and so on.

To pin the Application view widget to your page:

1. From the component panel on the left, click **Reports**.
2. Under **Application view**, drag and drop the widget that you want to pin to the page.



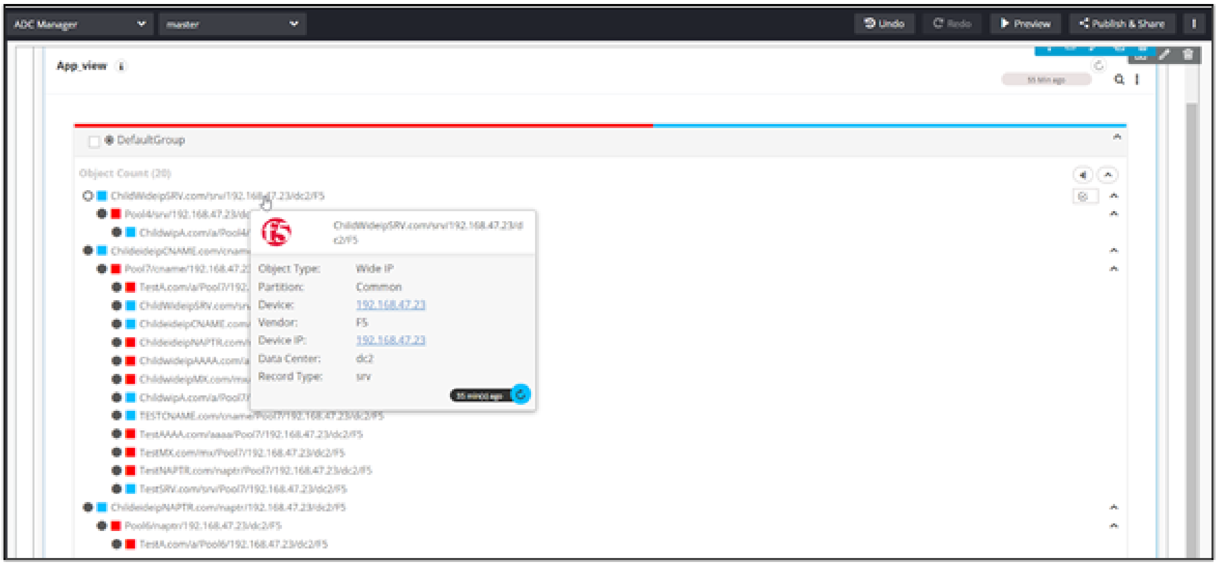
The **App_view** widget displays all the devices configured in the database.



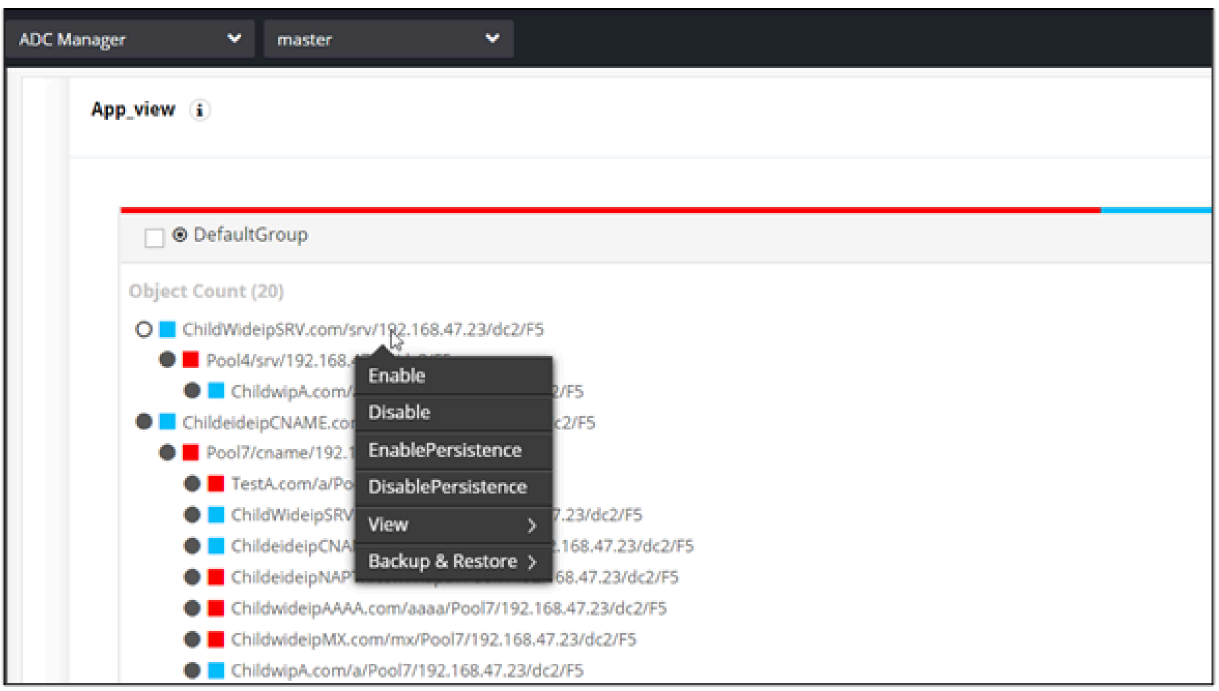
3. To modify the widget, hover your mouse over the top right corner of the widget and click **(Edit)** icon.

> **Note:** For more information on configuring the Report widget, click here.

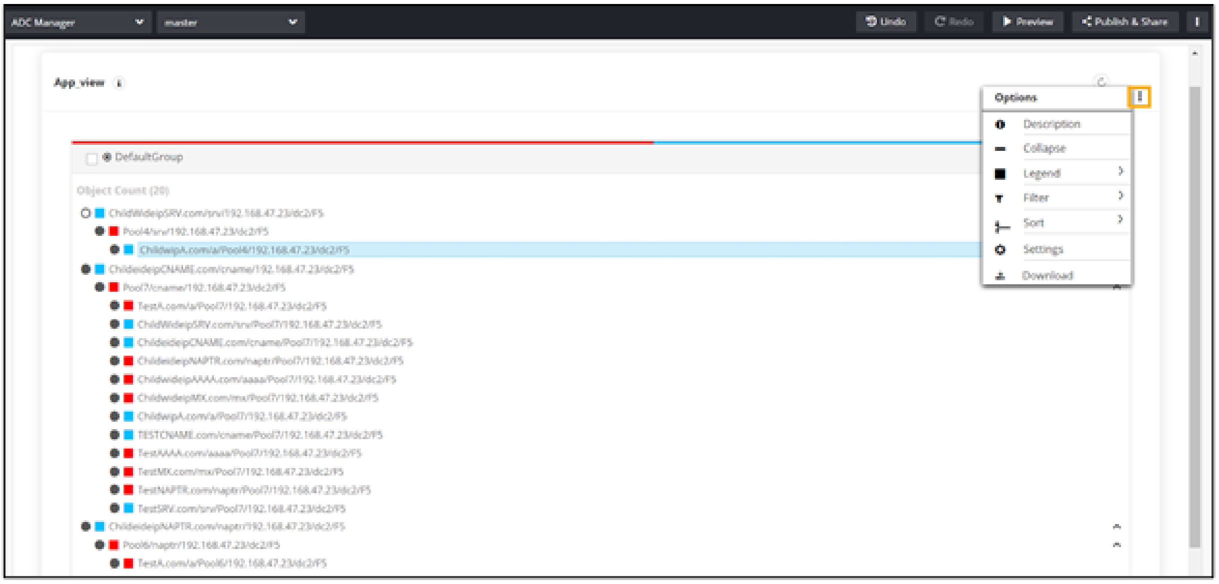
4. To view details about any device, hover your mouse over the device name.



5. To see the list of actions that can be performed on these devices, right-click on a device name.



6. To see the list of available options for the App_view widget, from the top right corner of the widget, click ⋮ .

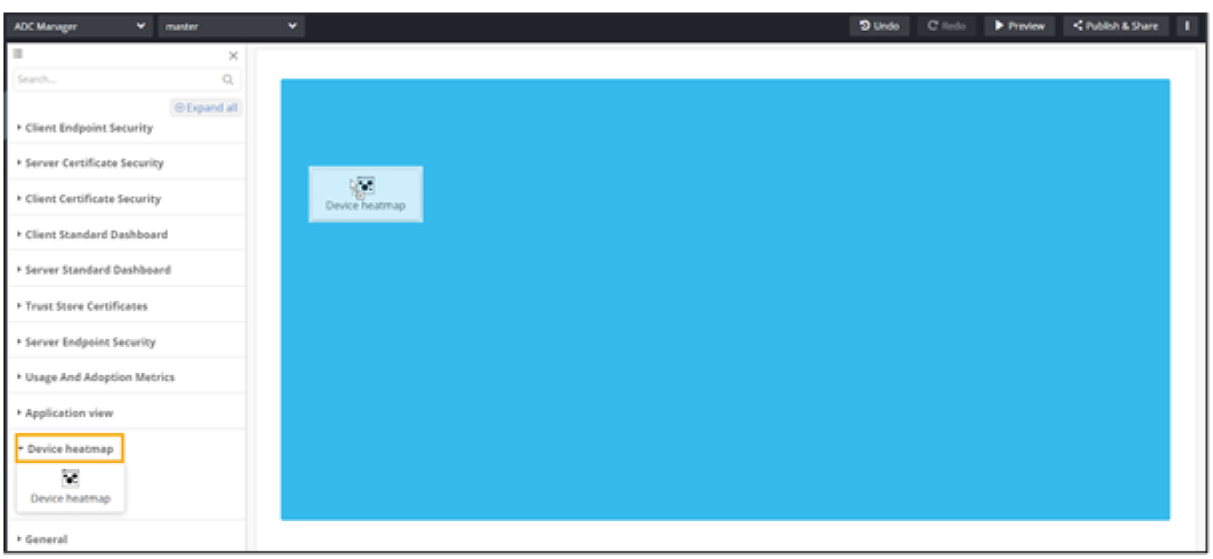


Pinning Device heatmap Widget

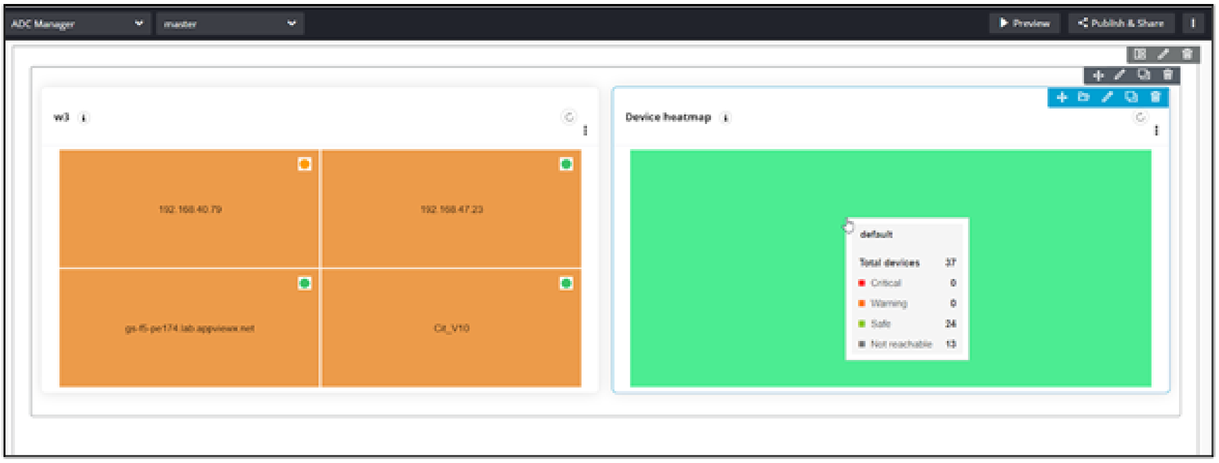
The **Device heatmap** widget shows a color-coded status of your configured devices.

To pin the **Device heatmap** widget to your page:

1. From the component panel on the left, click **Reports**.
2. Under **Device heatmap**, drag and drop the widget that you want to pin to the page.

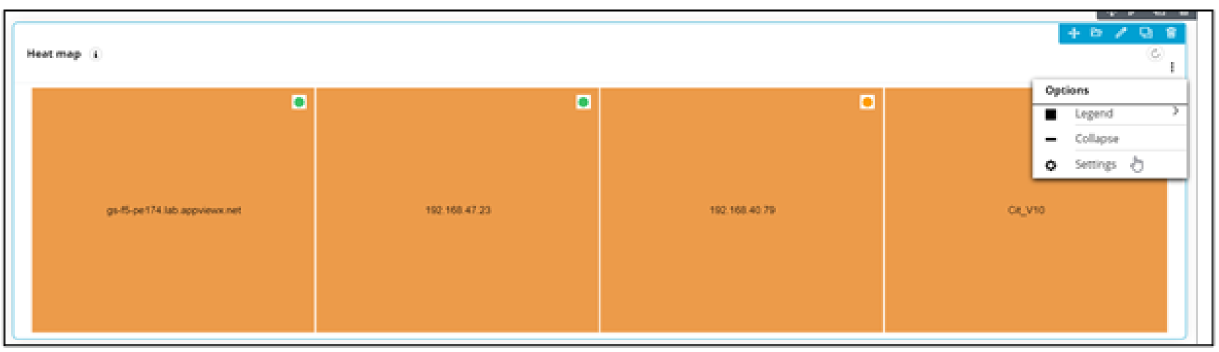


3. To modify the widget, hover your mouse over the top right corner of the widget and click **(Edit)** icon.

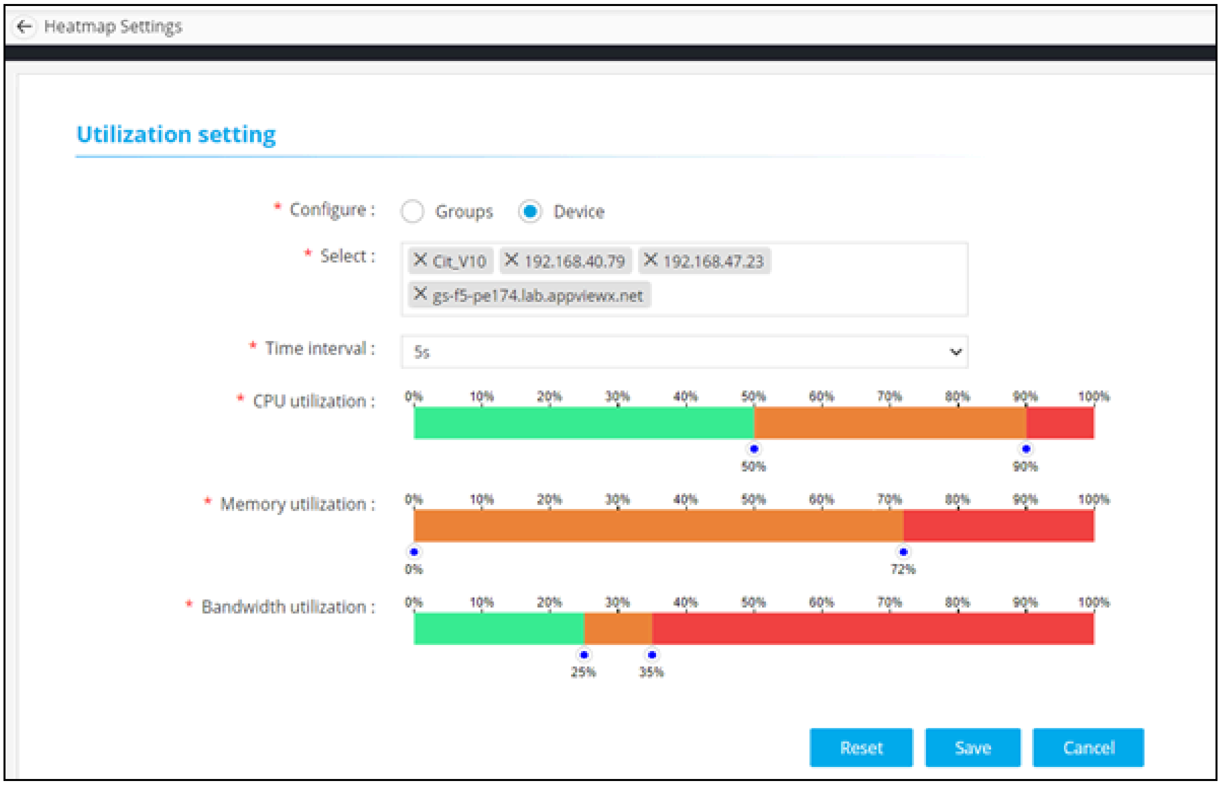


Note: For more information on configuring the Report widget, click here.

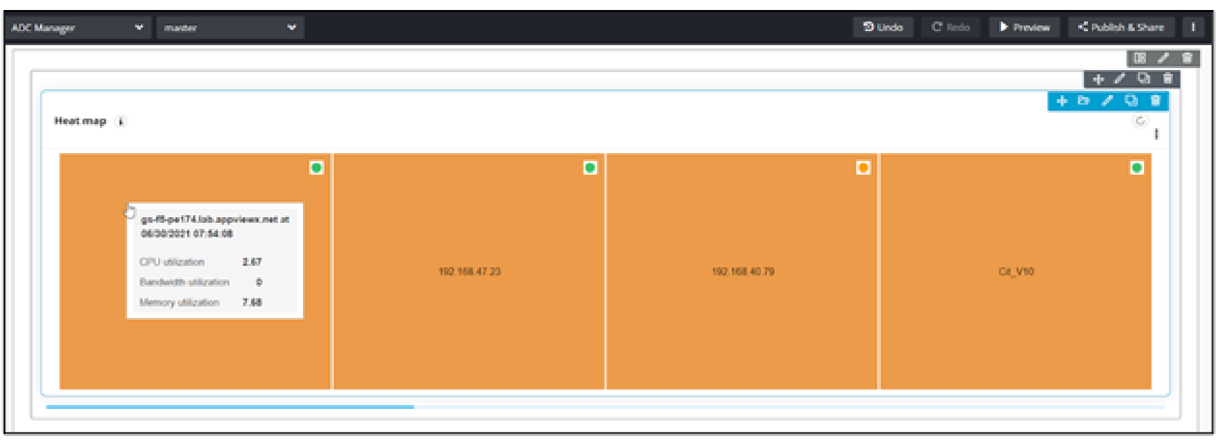
4. To modify the widget settings, from the top right corner of the widget, click .
5. From the options, select **Settings**.



Note: You can configure devices and groups and other settings here.

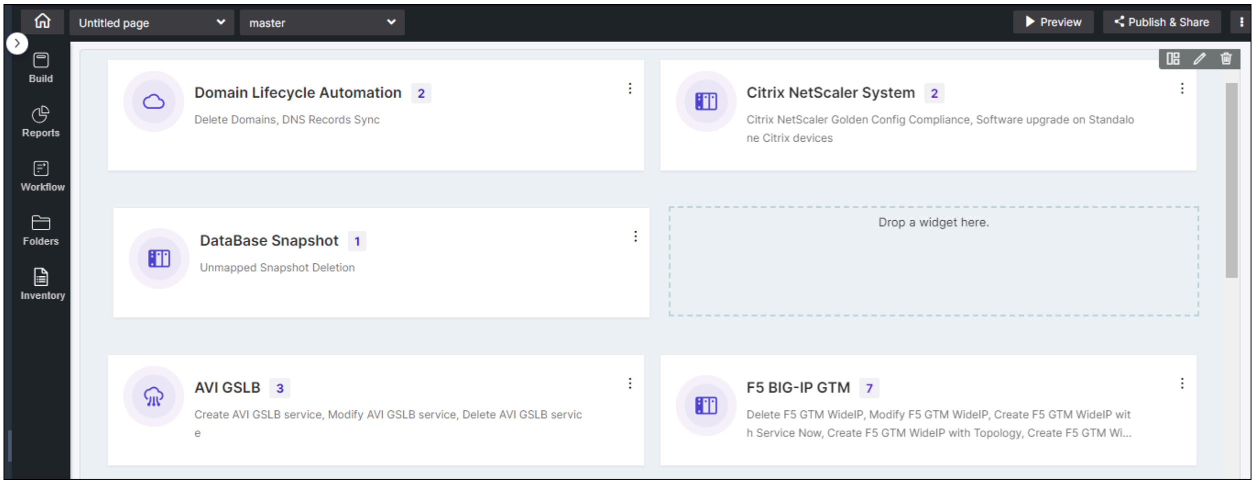


6. For details about a particular device on the heat map, hover your mouse over the device.



Workflow

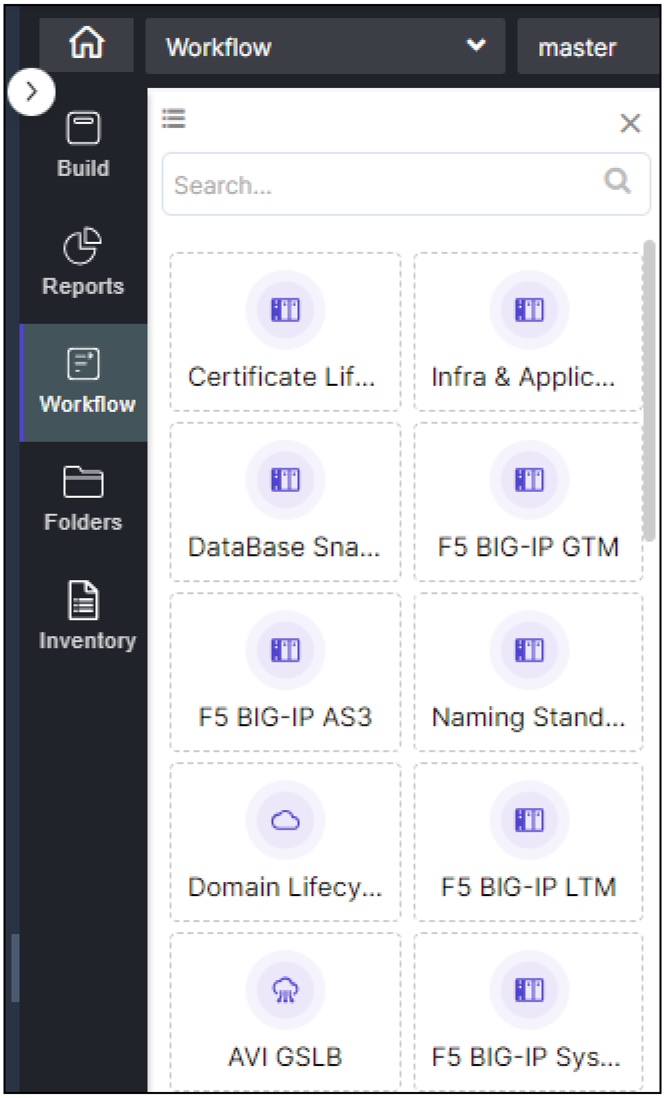
The Workflow component allows you to use the service catalog to execute automation workflows by persona. You can logically select and group workflows such as CLM Automation, Application Delivery and so on.



To access workflow catalogs:

1. From the component panel on the left, click **Workflow**.

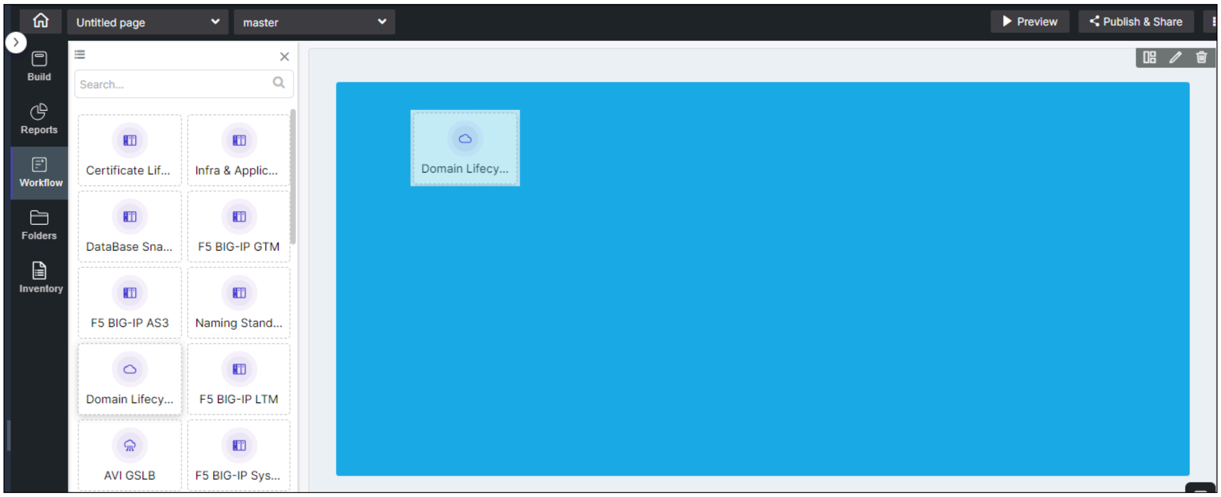
   A list of available workflow catalogs is displayed.



This table describes the options present in the **Workflow** section:

Field	Description
	Allows you to switch between list view and grid view of the Workflow catalogs.
Search bar	Allows you to search for a specific workflow catalog.

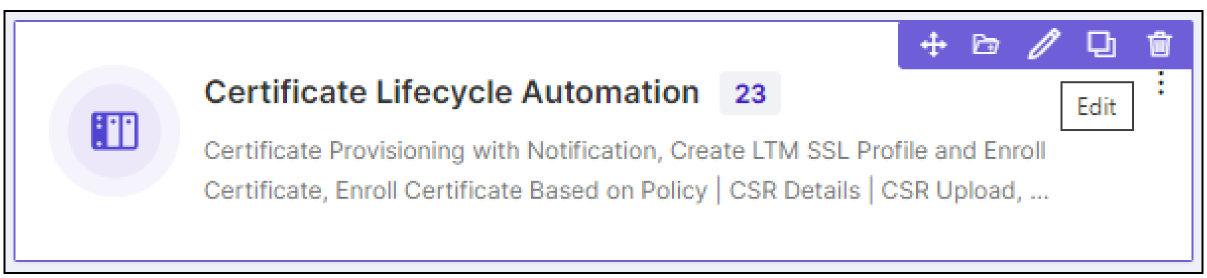
2. To add a workflow catalog to your page, drag and drop the selected workflow to any blue highlighted space in the build area.



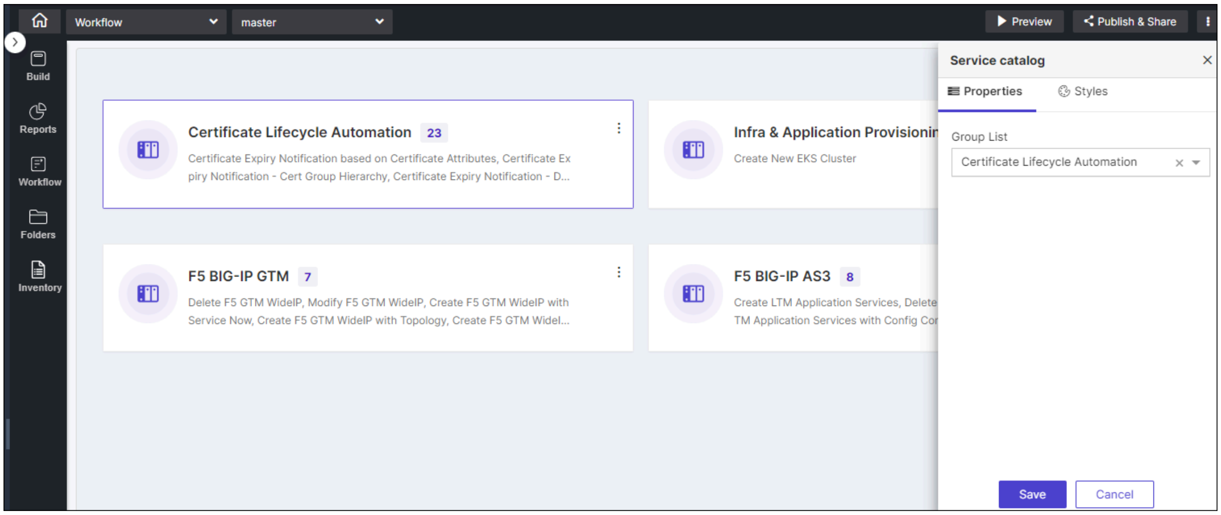
- Configuring Workflow - Properties
- Configuring Workflow - Styles

Configuring Workflow - Properties

1. To configure/edit the **Workflow** component, hover your mouse over the widget and click  **(Edit)** icon.



The **Service Catalog** window opens with the **Properties** tab open by default.



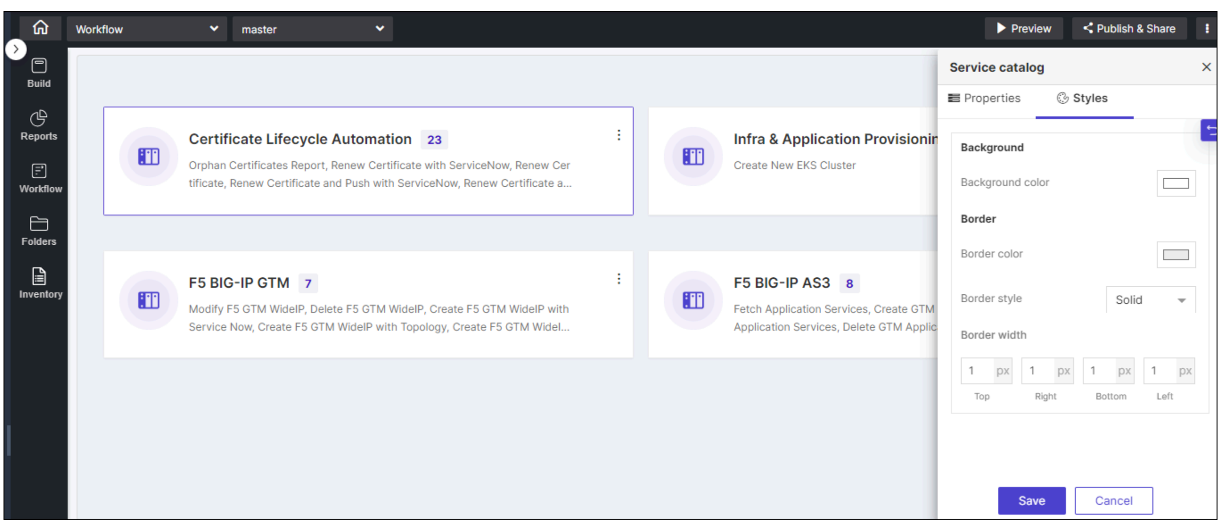
2. In the **Service catalog** window, under **Properties**, select a workflow catalog from the **Group List** dropdown.

> **Note:** The selected catalog will replace the catalog existing on the page.

3. Click **Save**.

Configuring Workflow - Styles

1. For more options to customize the **Workflow** widget, in the **Service catalog** window, click **Styles**.

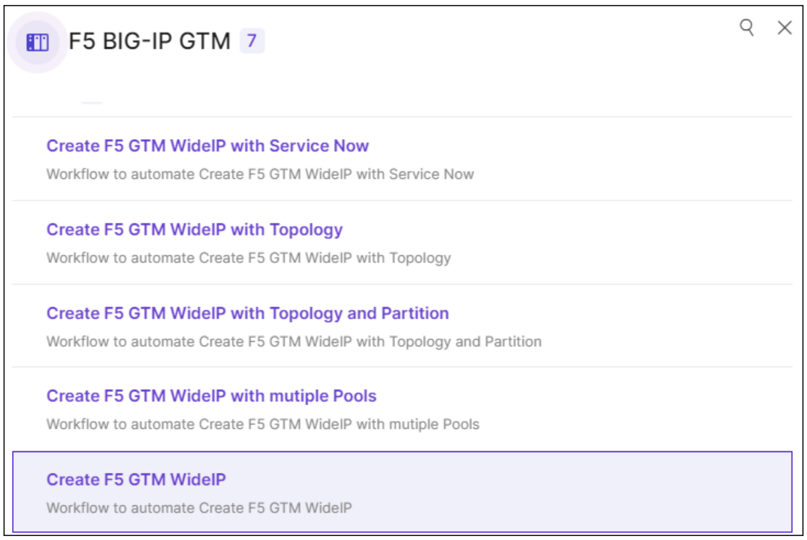


Field descriptions for Styles

Option	Description
Background	Allows you to set the background color of the workflow widget.
Border	Allows you to define the • **Border color** - Set the color of the Workflow widget's border. • **Border style** - Set the border style for the Workflow widget from the options available in the dropdown. • **Border width** - Set the thickness of the Workflow widget's border.

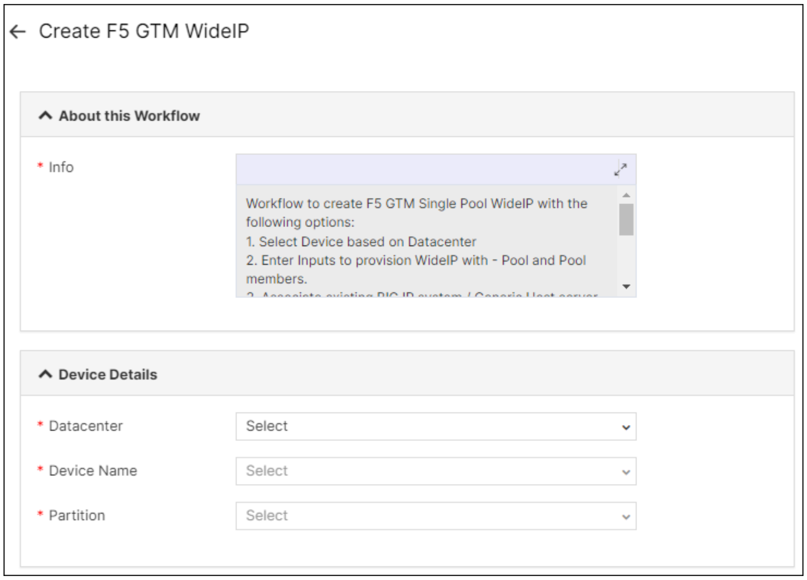
2. To trigger workflows from your page, click **Preview**.

   A preview of the page opens in a new tab.



3. To trigger a workflow locate the workflow in the catalog and click the catalog name.

   The selected workflow is triggered.

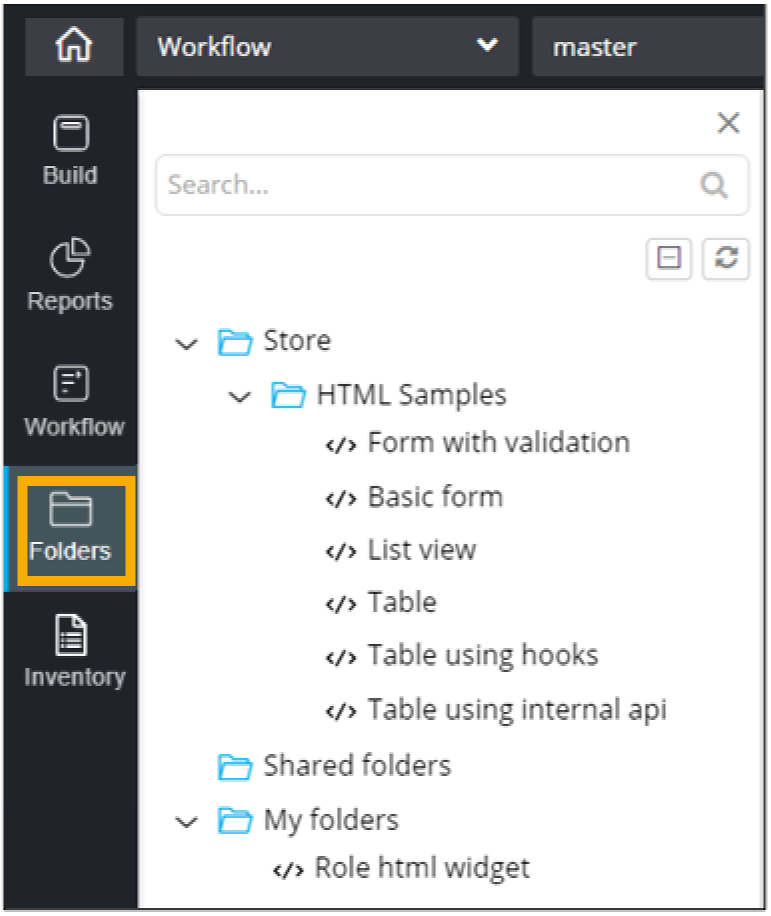


Note: For more information on workflows, refer to the Automation User Guide.

Folders

You can reuse widgets by saving them in folders. The Folders section also provides sample HTML scripts. Folders enable reusability of elements across multiple users thereby saving time.

The **Folders** can be accessed from the component panel on the left.



This table describes the options present in this section:

Options descriptions

Option	Description
Search bar	Allows you to search for a specific folder, widget or sample HTML script by typing keyword(s) in the search field.
/	Allows you to toggle between expanded and collapsed views of the folders.
	Refresh the folders inventory.

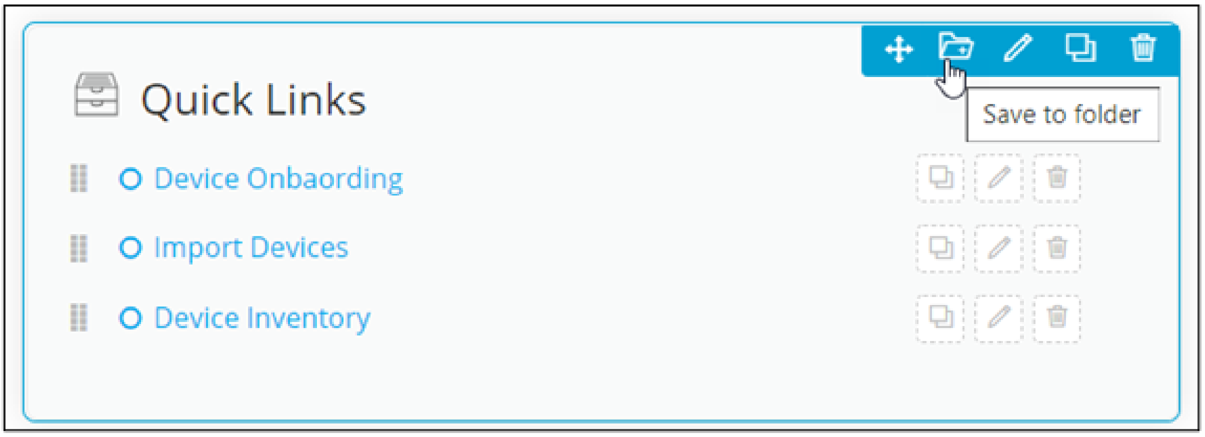
- Saving a widget in Folders
- Deleting Folders

Saving a widget in Folders

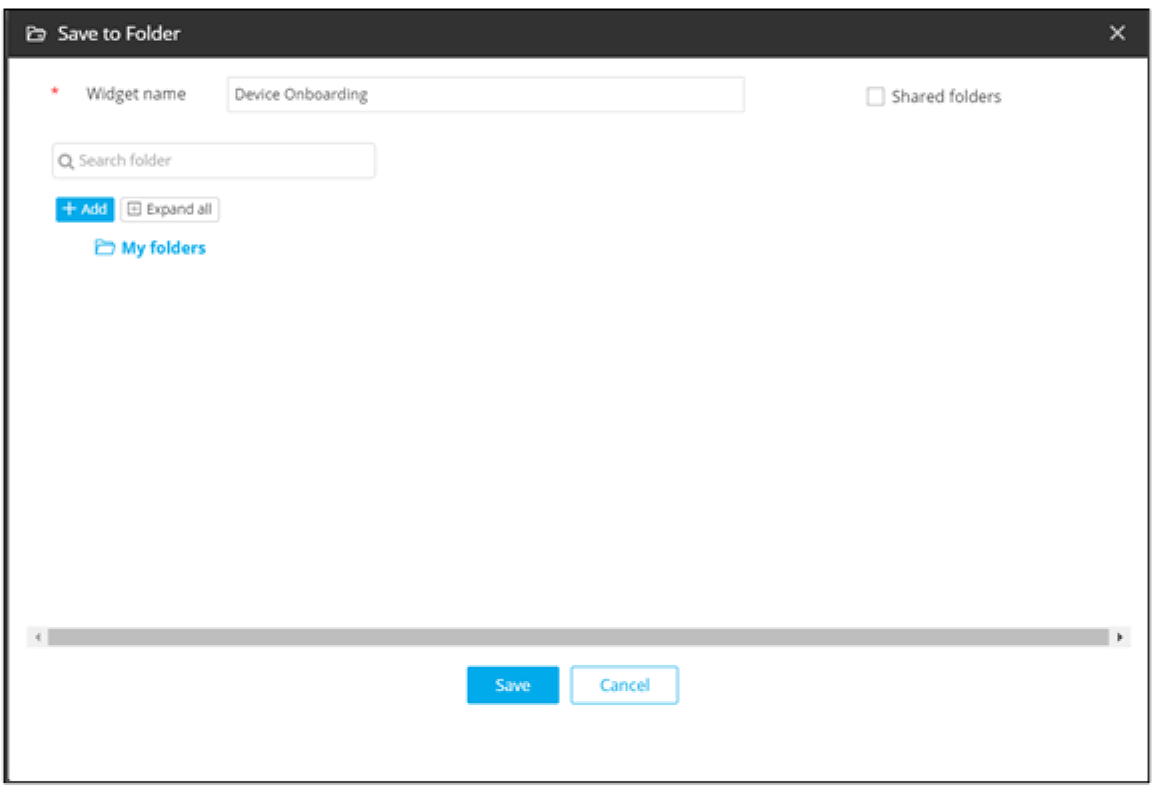
You can save one or more page/widget elements into a custom folder.

To save a widget to a folder:

1. From the top right corner of the widget to be saved, click  **(Folder)** icon.



2. In the **Save to Folder** window, enter or select the field information.



Field descriptions for Save to Folder

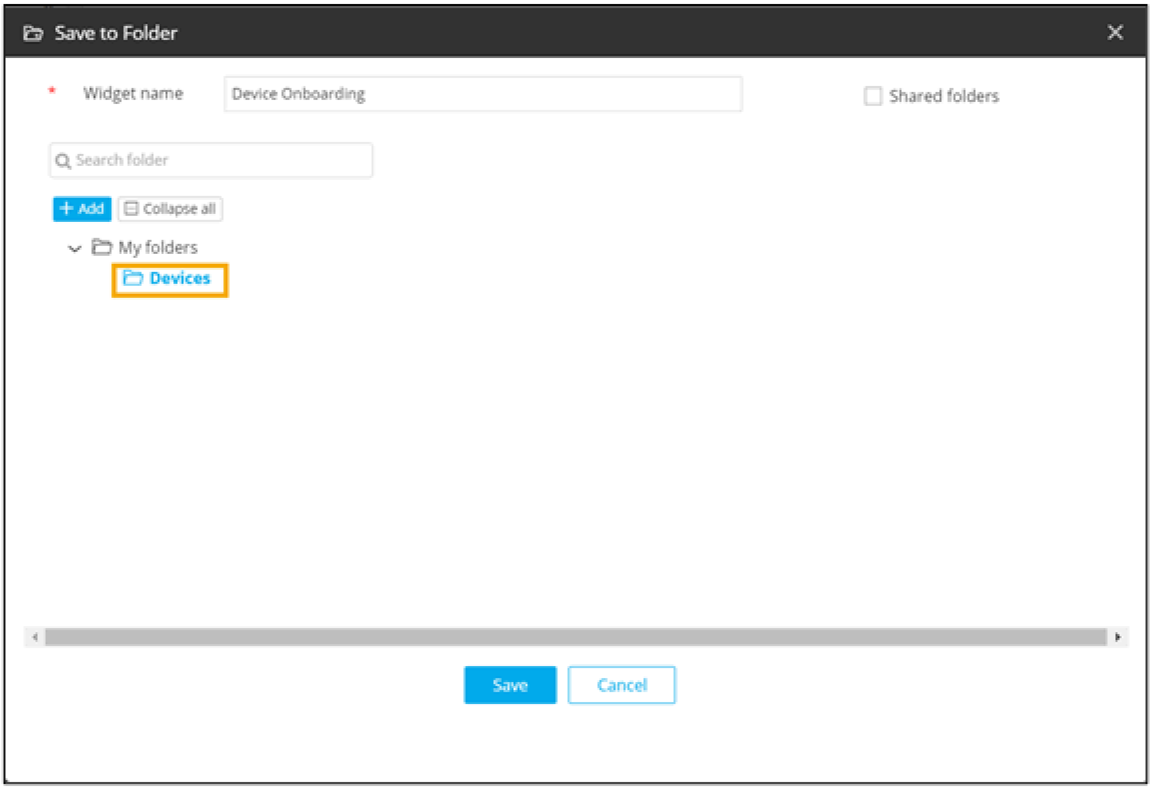
Field	Description
***Widget name**	Enter a suitable widget name.
Search bar	Allows users to search for a particular folder.
Add	Allows users to add new folders and sub-folders.
Expand All	Allows users to expand and collapse folders.
My folders	Shows sub-folders within My folders.
Shared folders	Allows users to save a widget in a shared folder.
**: Mandatory fields*	

- Saving a widget in My Folders
- Saving a widget in Shared Folders

Saving a widget in My Folders

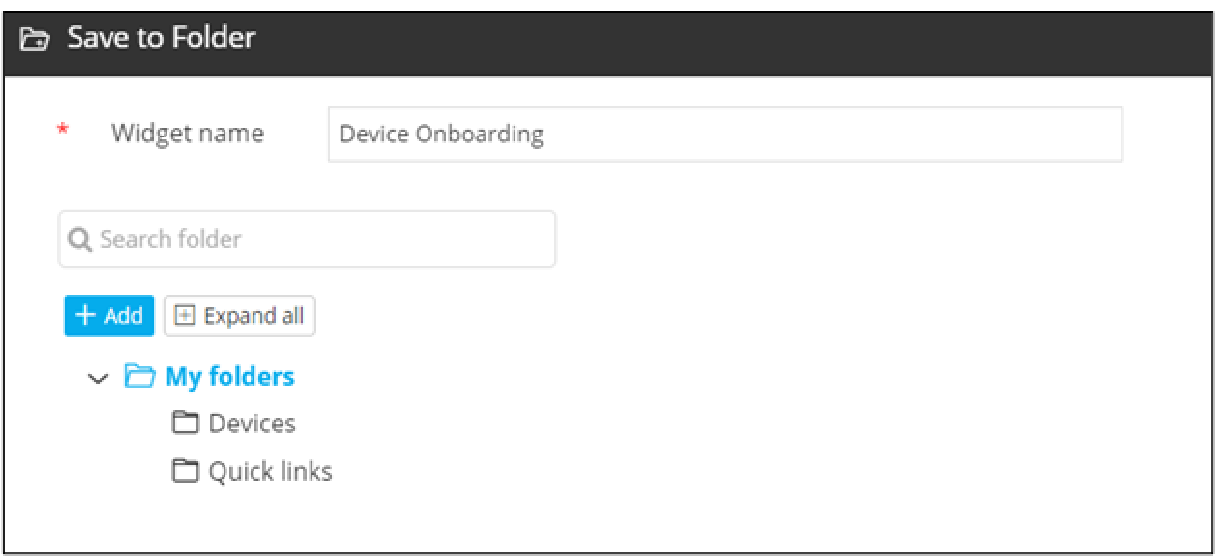
To save a widget to be reused only by the user who created the widget:

1. In the **Save to Folder** window, under **My folders**, click on the folder in which you want the widget to be saved.



2. To create a new folder for saving the widget, click **Add**.
3. Enter the folder name.

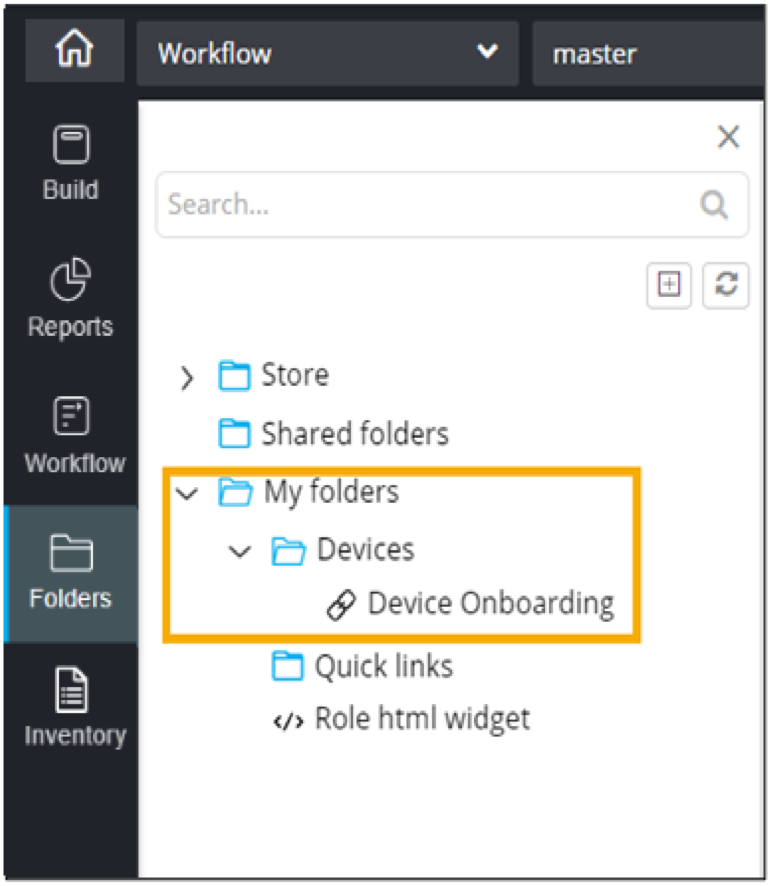
   A new folder is created.



4. Select the folder in which you want to save the widget.

5. Click **Save**.

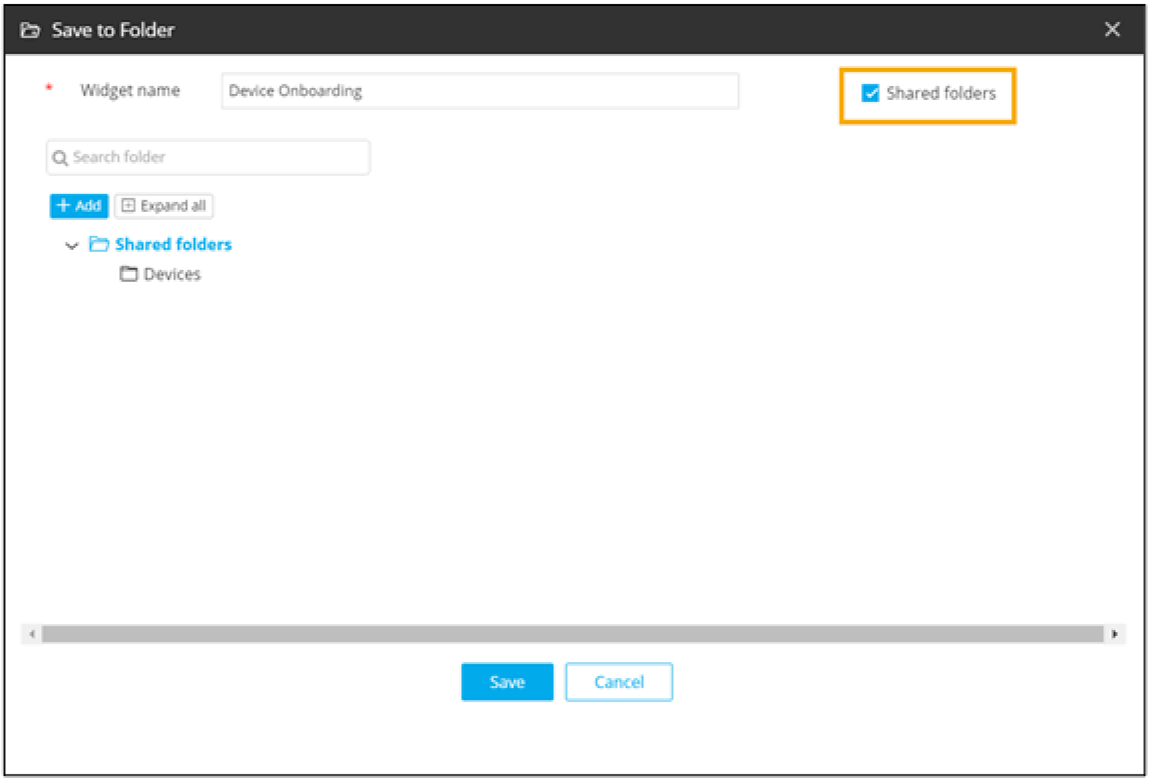
   The widget is saved in the selected folder.



Saving a widget in Shared Folders

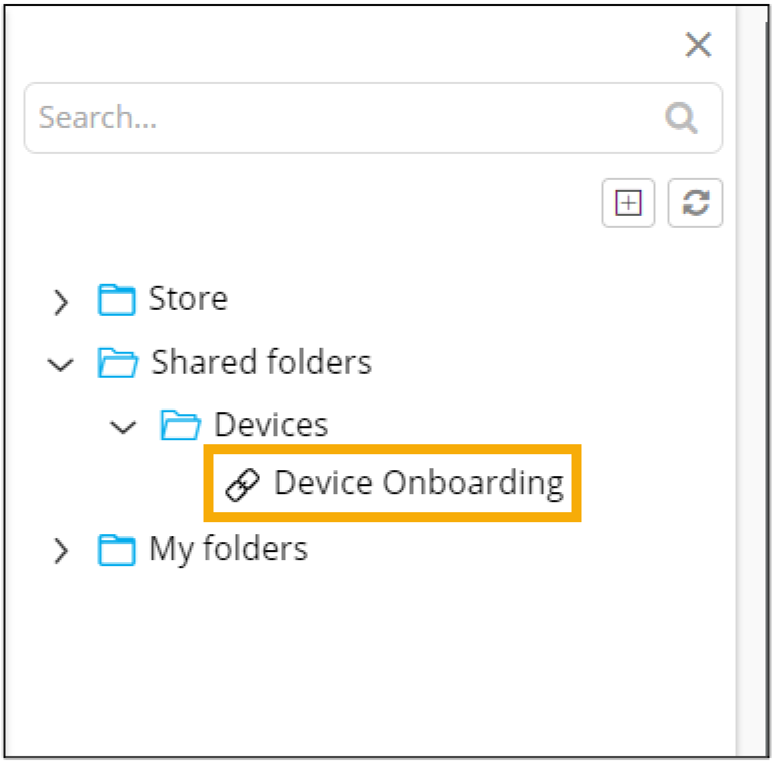
To save a widget that can be shared with other users:

1. In the **Save to Folder** window, select the **Shared Folders** checkbox.



2. Click on the shared folder where you want the widget to be saved. For example, Devices.
3. To create a new folder under the Shared folders, click **Add**.
4. Select the required folder and click **Save**.

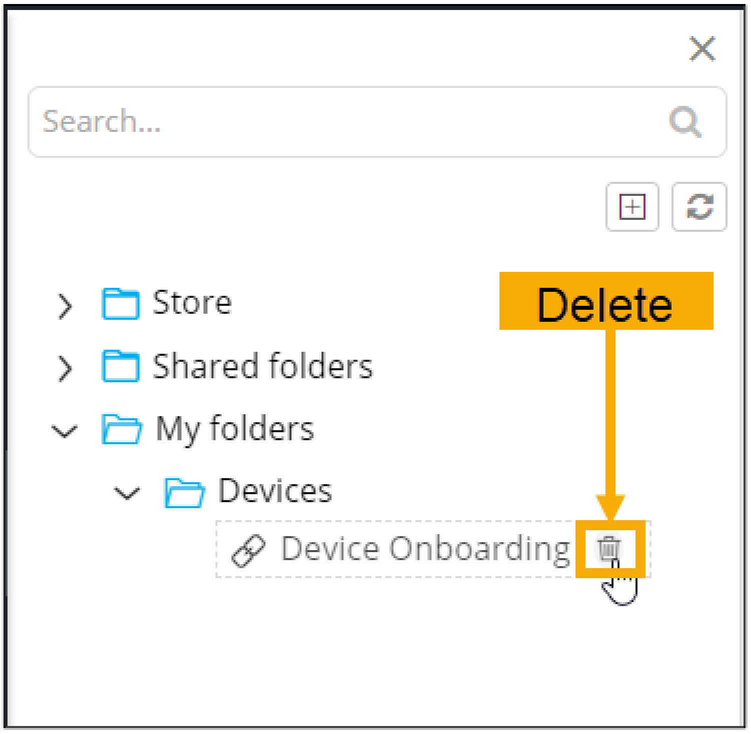
The widget is saved in the selected shared folder.



Deleting Folders

To delete a folder from My Folder or Shared Folders:

1. From the component panel on the left, click **Folders**.
2. Click **(Delete)** icon next to the folder name.



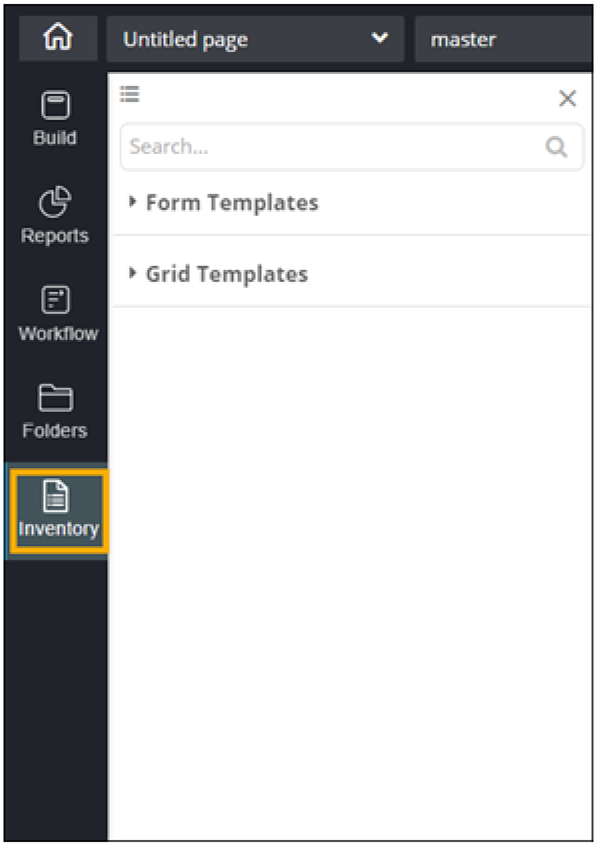
3. In the **Confirmation** window, click **Yes**.

   The folder is deleted.

Inventory

Service Catalog has an inventory of pre-built form and grid templates that can be used out of the box for quick self-servicing or customized as per requirement. Forms enable easy self-servicing capabilities to the Line of Business (Application owners, NetOps and so on) via an intuitive GUI-based interface, thus abstracting the end user from underlying infrastructure configurations.

The **Inventory** can be accessed from the component panel on the left.



Options available with descriptions in Inventory

Option	Description
	Allows you to switch between a list view and grid view of the forms.
Search	Allows you to search for a specific OOB or custom form.
Form Templates	This folder contains a set of prebuilt out of the box (OOB) forms that you can add to your page for
Grid Templates	This folder contains a set of prebuilt out of the box (OOB) grid templates that you can add to your p servicing.

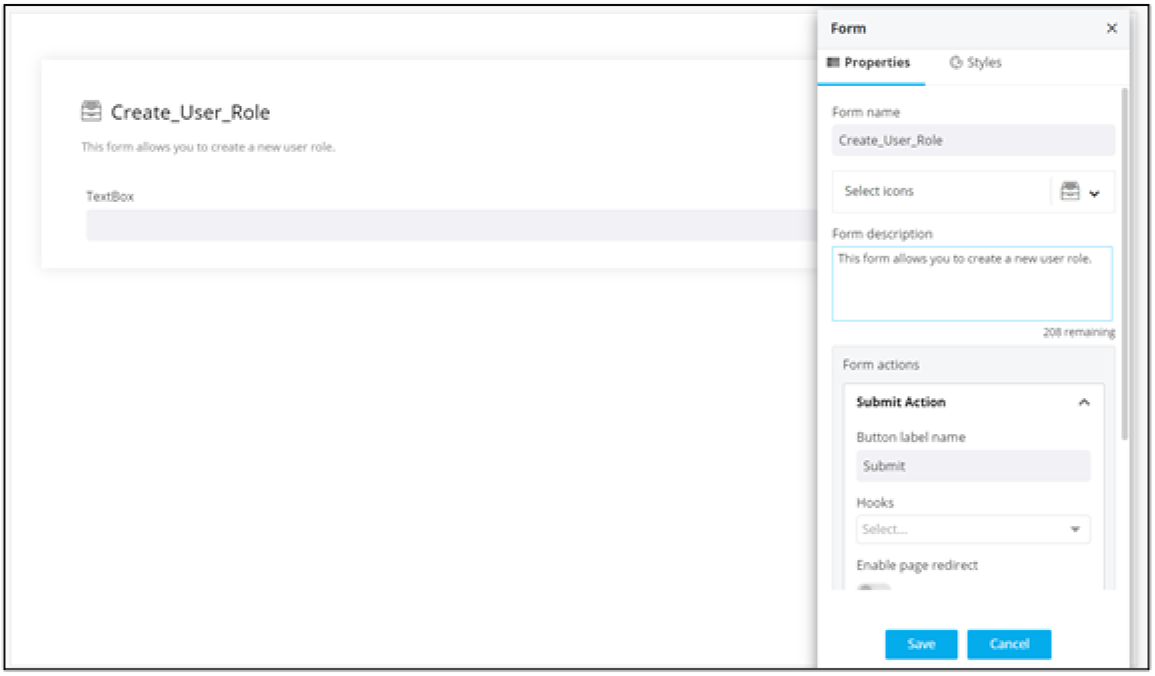
- Getting started with designing a Custom Form
- Using Form Templates
- Designing a new Form
- Enabling Auto trigger
- Saving Forms to Custom Folder
- Form Field - Label
- Form Field - Text Box

- Form Field - Password
- Form Field - Date Time Picker
- Form Field - Text Area
- Form Field - Radio Button
- Form Field - Checkbox
- Form Field - Select Box
- Form Field - Multi-select Box
- Form Field - Form Button
- Form Field - Download
- Form Field - File Upload
- Form Field - Script Editor
- Form Field - Tabular
- Form Field - Rich Text Editor

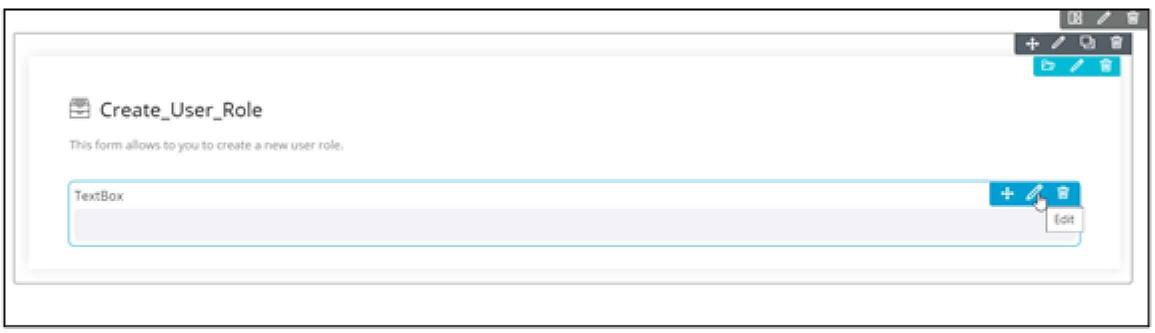
Getting started with designing a Custom Form

This example will help you understand how to design a basic form for creating a new role.

1. Design a new page.
2. Under **Build** > **Form**, drag and drop the **Text Box** form field to the design space.
3. To change the form name, hover your mouse over the form widget and from the top right corner, click **(Edit)** icon.
4. In the **Form** pop-up window, under **Properties**, change the **Form name** to **Create_User_Role**.
5. In the **Form** pop-up window, under **Properties**, add the **Form description**.



6. Click **Save**.
7. To rename the **Text Box** field, hover your mouse over the field and click **(Edit)** icon.



8. In the **Text Box** pop-up window, under **Properties**, enter or select the following field information:

Field descriptions for Properties

Field	Description
***Label name**	Enter a valid **Label name** for the form field. For example: **Role name**.
***Field ID**	Enter a unique Field ID.
Mandatory	Turn on the toggle to make this field mandatory.
**: Mandatory fields*	

9. Click **Save**.
10. To add another Text Box field to the same form, drag and drop the **Text Box** widget within the existing form space.

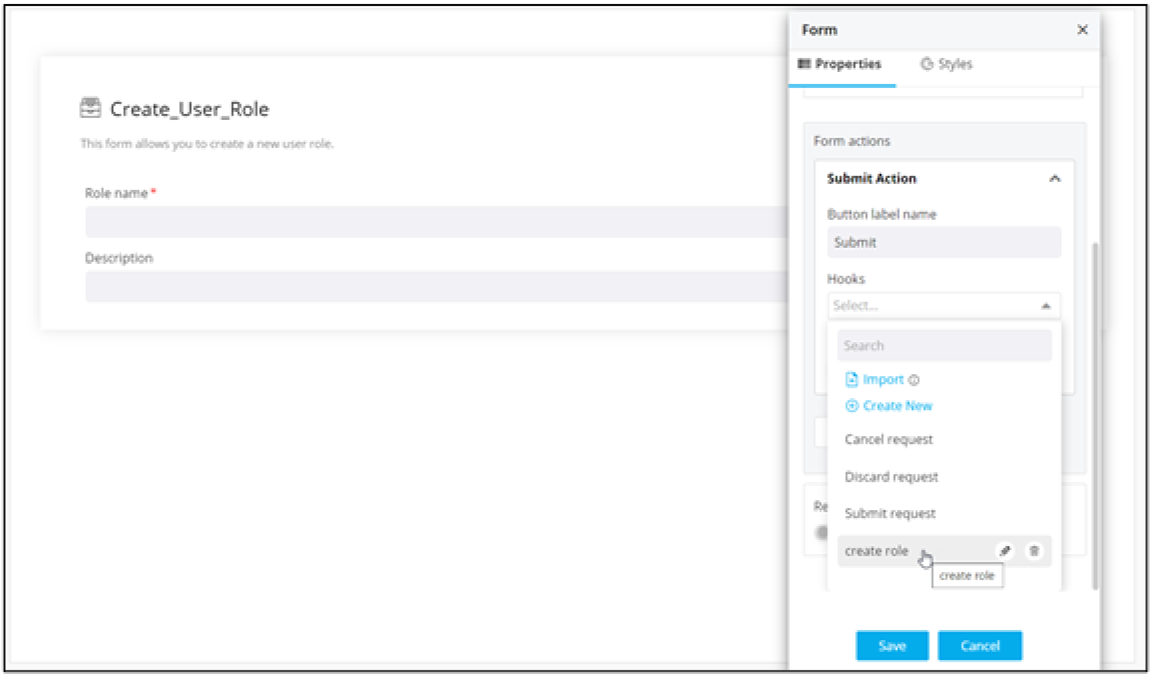
11. To configure this **Text Box** field, hover your mouse over the Text Box field and click **(Edit)** icon.
12. In the **Text Box** pop-up window, under **Properties**, enter or select the following field information:

Field descriptions for Properties

Field	Description
***Label name**	Enter a valid label name for this field. For example, Role name.
***Field ID**	Enter a unique Field ID.
**: Mandatory fields*	

13. Click **Save**.
14. To add a Submit button to the form, hover your mouse over the form and from the top right corner of the form, click **(Edit)** icon.

15. In the **Form** pop-up window, under **Properties,** scroll down to **Form actions**.
16. Under **Form actions**, select, import, or create a new hook for the Submit form action.

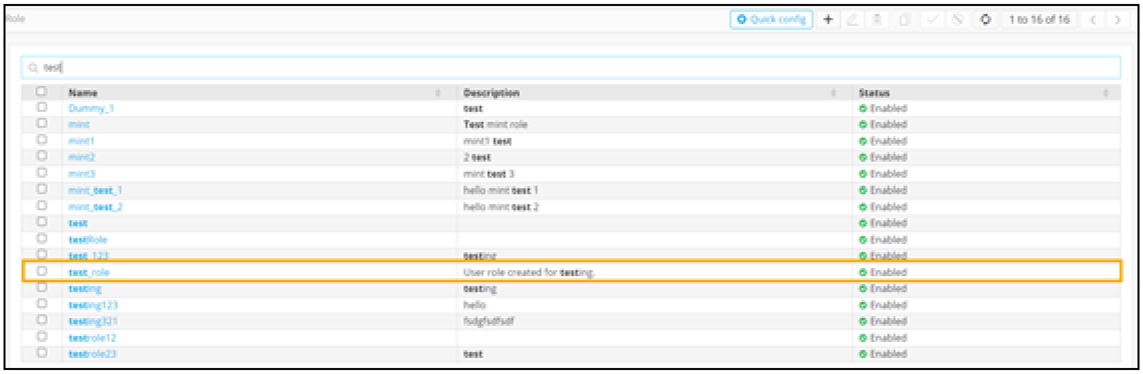


17. Click **Save**.
18. To use this form to create a new user role, click **Preview**.

The Preview page opens in a new tab.

19. Enter the **Role name** and **Description** for the new user.
20. Click **Submit**.

The new user role is created and added to the list of Roles.



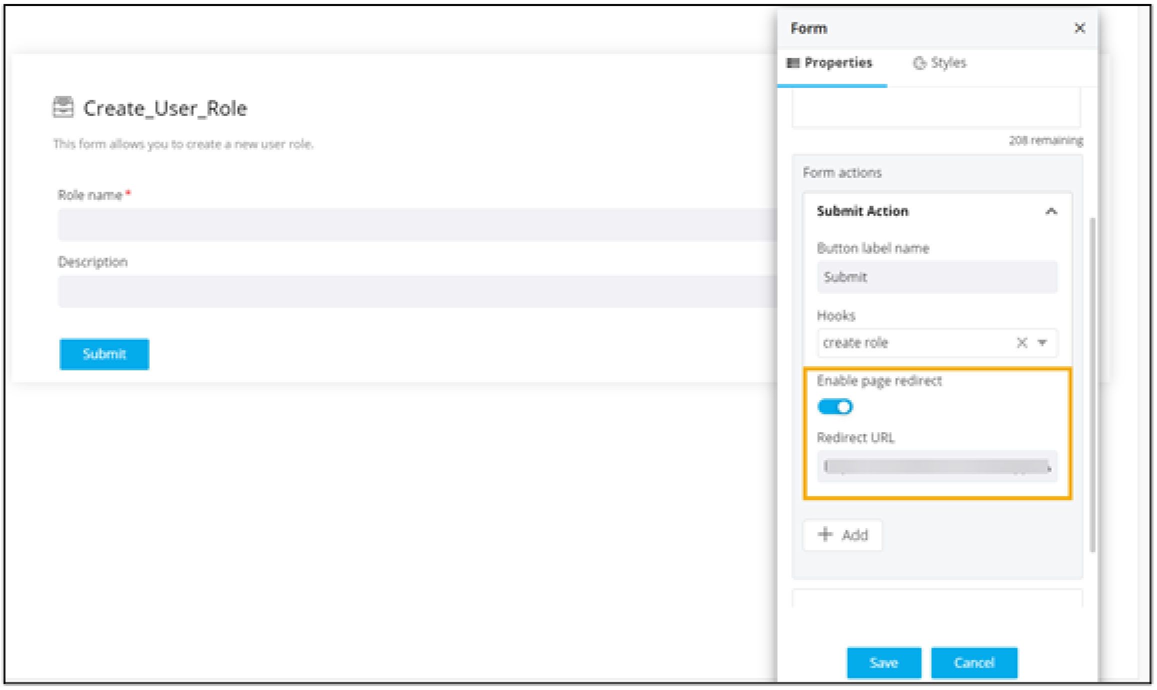
- Redirecting to a Page after Submit

Redirecting to a Page after Submit

When designing a form, there is a provision to define the form action to redirect to another page in a new tab.

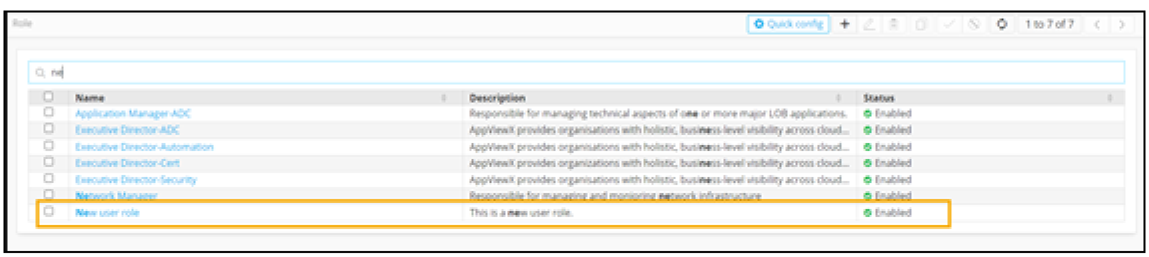
To add redirect action to the Submit form:

1. Hover over the form widget and click **(Edit)** icon.
2. In the **Form** pop-up window, under **Properties**, scroll down to **Form actions**.
3. Under Form actions, turn on the **Enable page redirect** toggle.
4. In the **Redirect URL** field, enter the URL of the page to be redirected to.



5. Click **Save**.
6. To create a new role and redirect to the required page, click **Preview**.
7. Enter the **Role name** and **Description**.
8. Click **Submit**.

   You will be redirected to the **Role** page in the same tab.

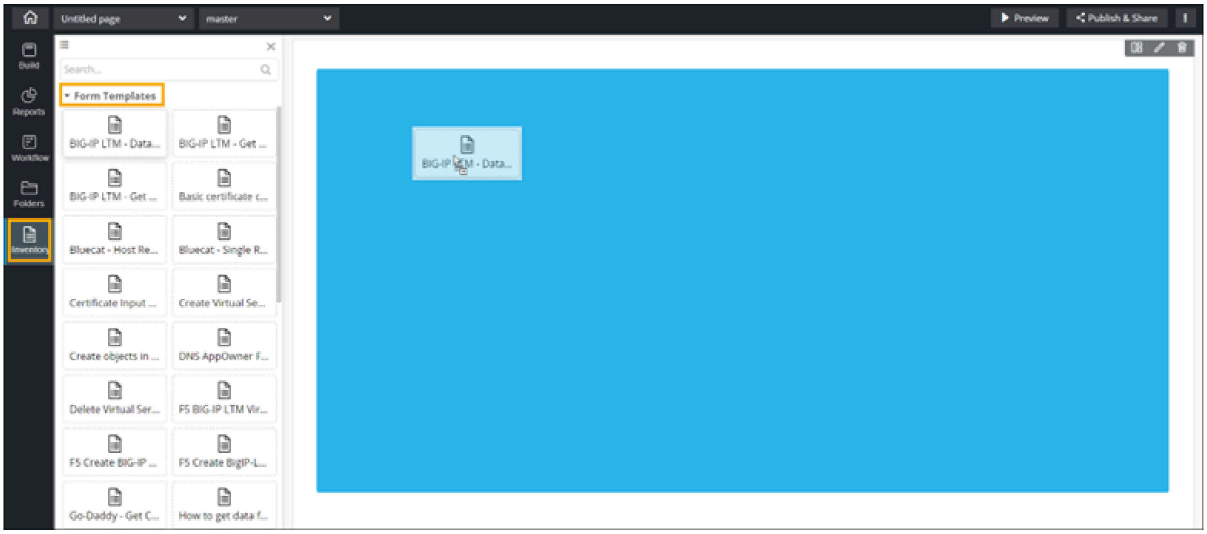


Using Form Templates

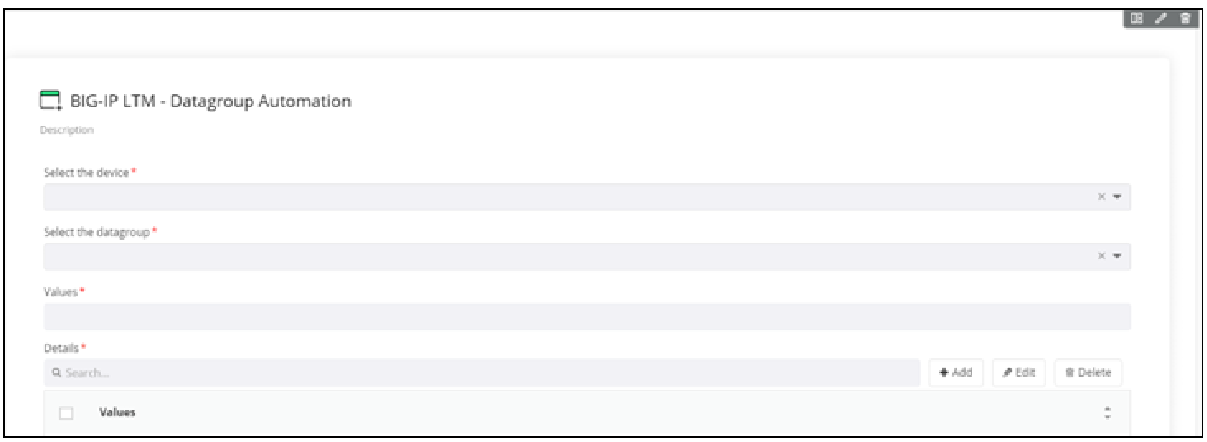
The **Inventory** section contains a set of prebuilt out of the box (OOB) Form Templates that you can add to your page for quick self-servicing.

To use a Form Template:

1. Design a page.
2. From the **Inventory** section, expand **Form Templates**.
3. Drag and drop the required form template to the canvas.



The selected form template is added to the canvas. It can be used as is or modified as per your requirement.



Designing a new Form

You can also design a form for your page using the different form field palettes available in the Design module under the **Build** section. Add individual form fields as required to the canvas and build your own form from scratch.

The following form fields are available:

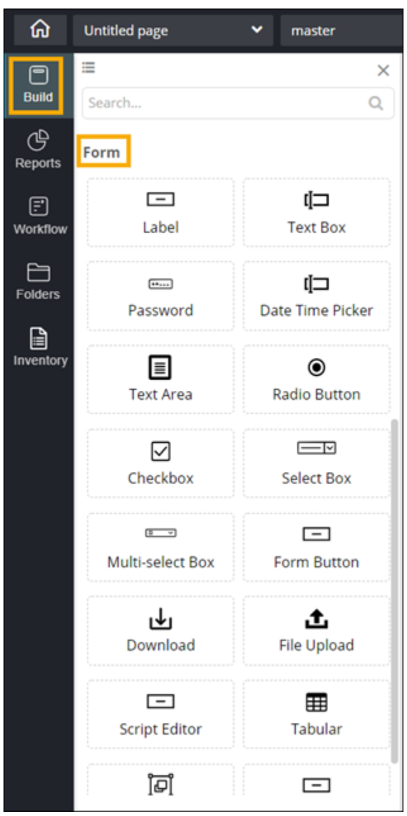
- Date Time Picker
- Download
- File Upload
- Text Area

- Text Box
- Label
- Password
- Ace Editor
- Form Button
- Checkbox
- Multi-select Box
- Select Box
- Radio Button
- Form Group
- Tabular.

To access these form fields:

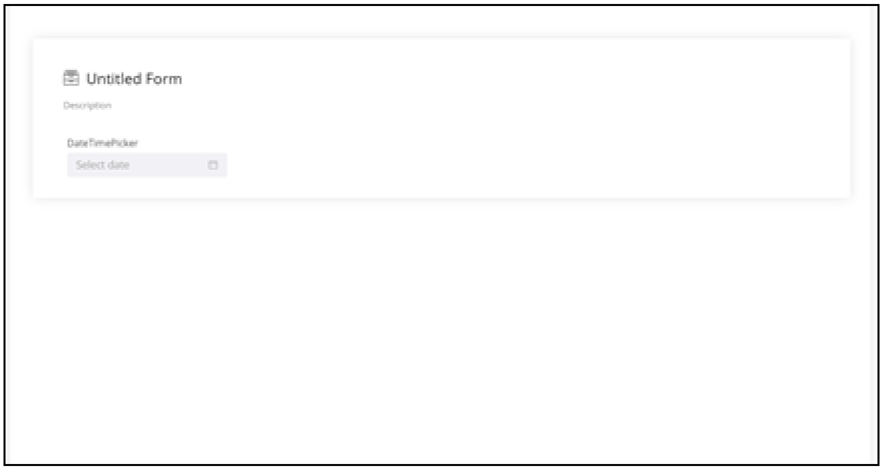
1. From the menu on the left, click **Build**.

   A list of form components are displayed under **Form**.

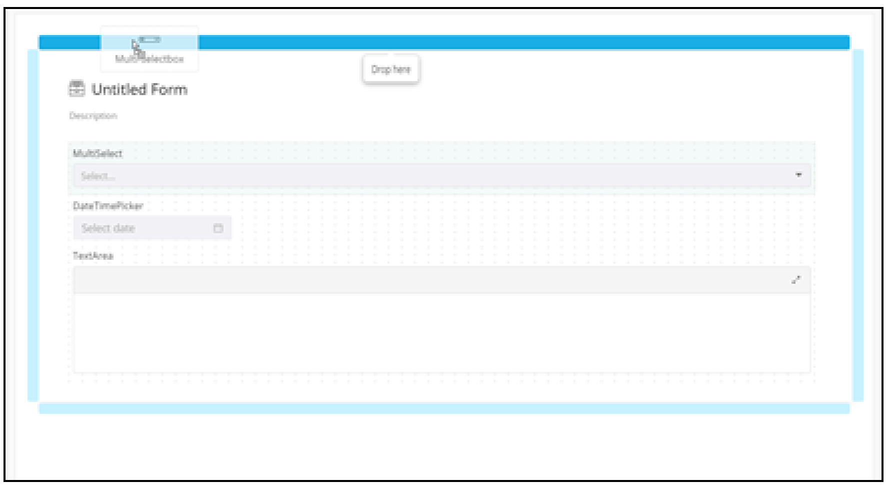
   

2. To create a form, drag and drop a form field to the canvas.

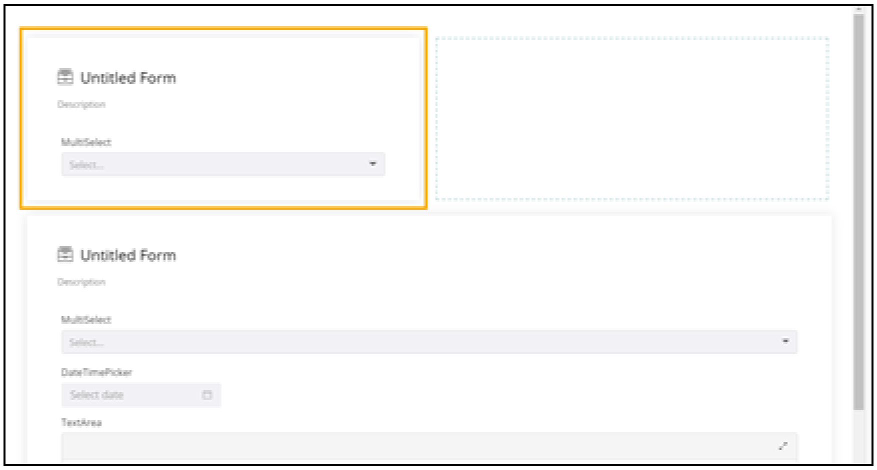
   A form is created with the selected form field.



3. To add another form field(s) to the same form, drag and drop the required form field(s) in the form area.
4. To create a new form on the same page, drag and drop a form field into any blue highlighted space in the canvas.



A new form with the selected form field is created.



Note: For more information on configuring the form properties such as Form name and description, adding hooks, click here.

Note: For more information on customizing the form features such as border, size, background color and so on, click here.

- Configuring the Form - Properties
- Configuring Form Actions
- Assigning Hooks to Form Fields
- Configuring the Form - Styles
- Modifying Individual Form Fields

Configuring the Form - Properties

1. To customize the form properties, hover your mouse over the top right corner of the form.
2. Click **(Edit)** icon.

   The **Form** pop-up window opens.

Field descriptions for Properties

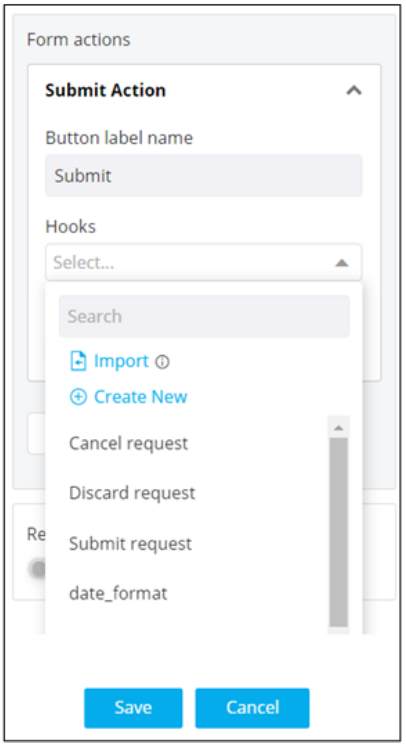
Field	Description
Form name	Enter a valid name for the form.
Select Icons	Select an icon for the form from the available options in the dropdown.
Form Description	Enter a short description for the form.
Select script to run on form load	Select a script from the options available in the dropdown. This script automatically fetches the values for all the fields in the form when the form loads.
Form Actions	Add action buttons to the form such as Next, Save, Submit and assign hooks. **Note:** For more information on Form Actions, click here.
Reset Action	Add a Reset button to the form. Enabling the toggle displays the **Button label name** field where you can provide the name for the button.

Configuring Form Actions

This feature allows you to add buttons to your form such as Next, Save, Submit, Reset, and assign hooks.

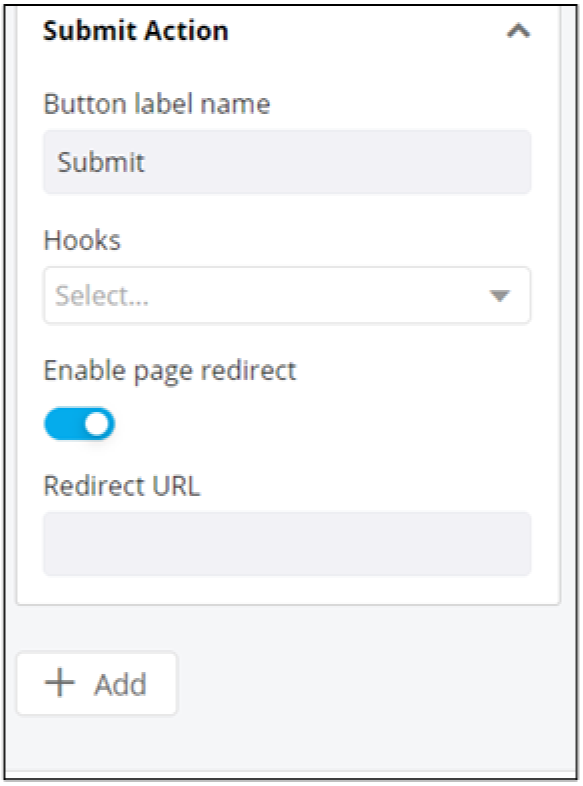
To add a Submit action to the form:

1. Under **Button label name**, enter a valid name for the button. For example, **Submit**.
2. Under **Hooks**, select the appropriate hook from the available options in the dropdown or import/create/modify a hook.



> **Note:** For more information on assigning hooks to form fields, click here.

3. To redirect to a page after clicking the Submit button, turn on the **Enable page redirect** toggle.
   For example, if you want to create a new user or role and after submitting the form you want to be redirected to the Role page, you can enable this option.
   Once the page redirect is enabled, the **Redirect URL** field is displayed.



4. Add the Redirect URL and click **Add** to add the Submit action to the form.

   Submit button is added to the form.

Assigning Hooks to Form Fields

You can assign hooks to the form fields or to the form action buttons (eg. Submit) to fetch details against a specific form field from the device and/or database and map them to the respective field on the user interface.

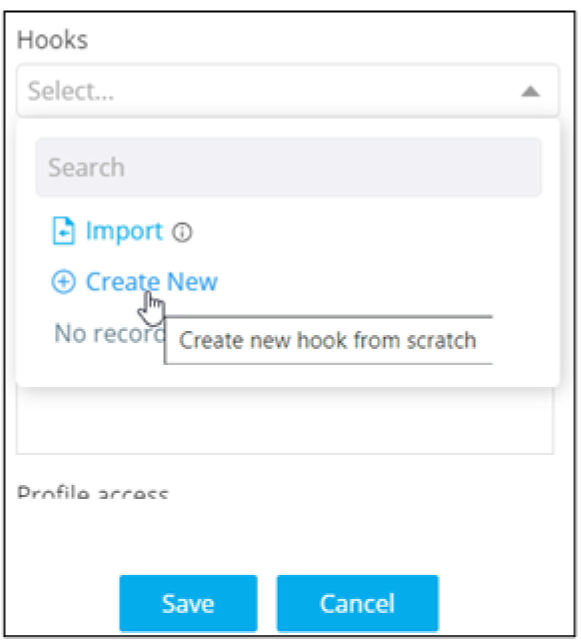
1. To assign a hook to a form field, hover over the form field and click **(Edit)** icon.
2. In the field's pop-up window, under **Properties**, scroll down to **Hooks**.

   The following options are available under Hooks:
   - **Select an existing hook:** You can select an existing hook from the available options and assign it to the form field to fetch data.
   - **Import:** You can import a hook from the Hooks Inventory and assign it to the form field.
   - **Create New:** You can create a new hook and assign it to the form field.
   - **Modify:** You can modify an existing hook from the options available in the dropdown and assign it to the form field.

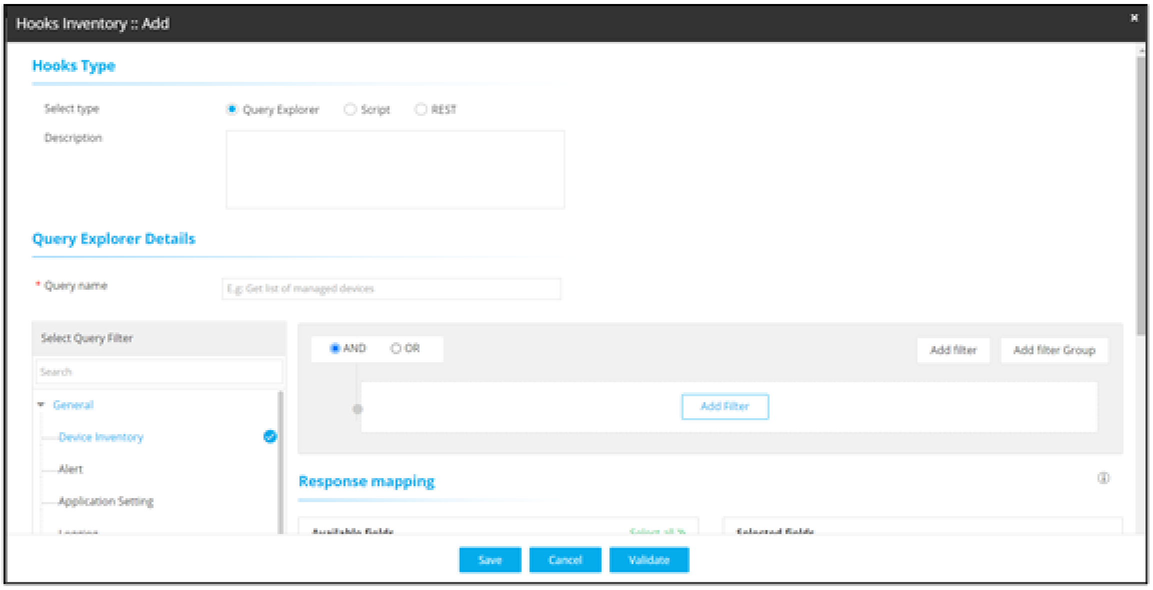
- Creating a New Hook
- Importing Hooks from Inventory
- Modifying a Hook

Creating a New Hook

1. Under the **Hooks**, click **Create New**.



The **Hooks Inventory :: Add** window opens.



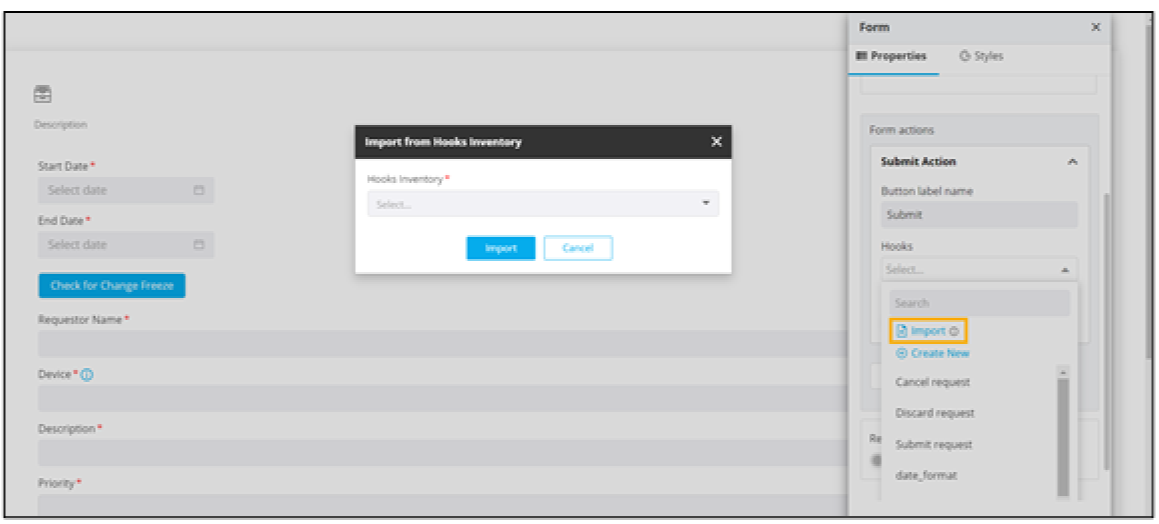
2. Create a new hook as required and click **Save**.

> **Note:** For more information on Hooks, refer to the Automation User Guide.

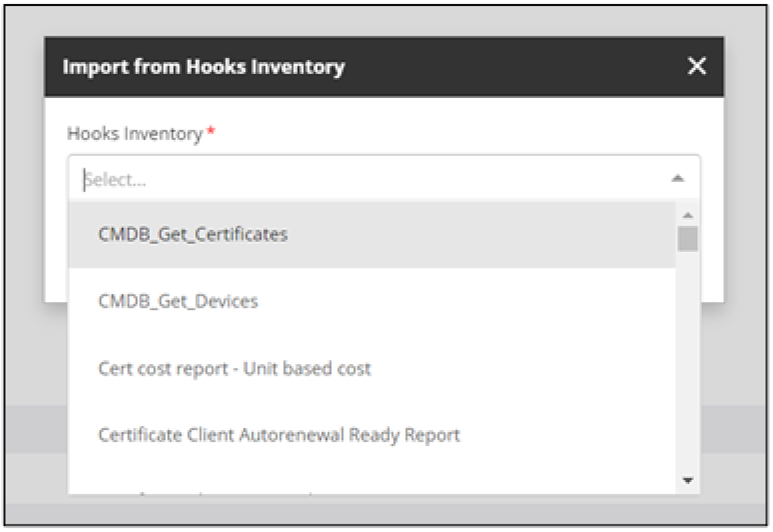
Importing Hooks from Inventory

1. Under the **Hooks** section, click **Import**.

   The **Import from Hooks Inventory** window opens with the list of hooks available in the inventory displayed in the dropdown.

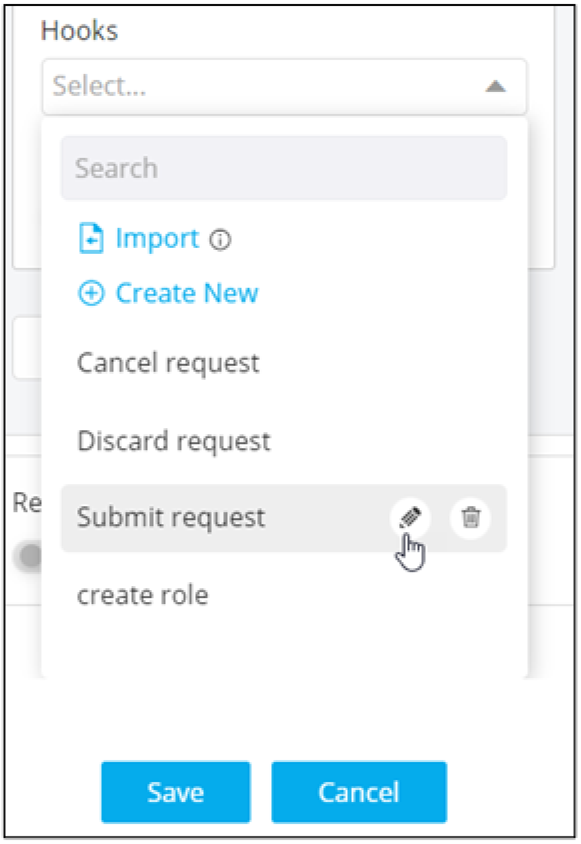


2. Select the required hook from the list and click **Import**.

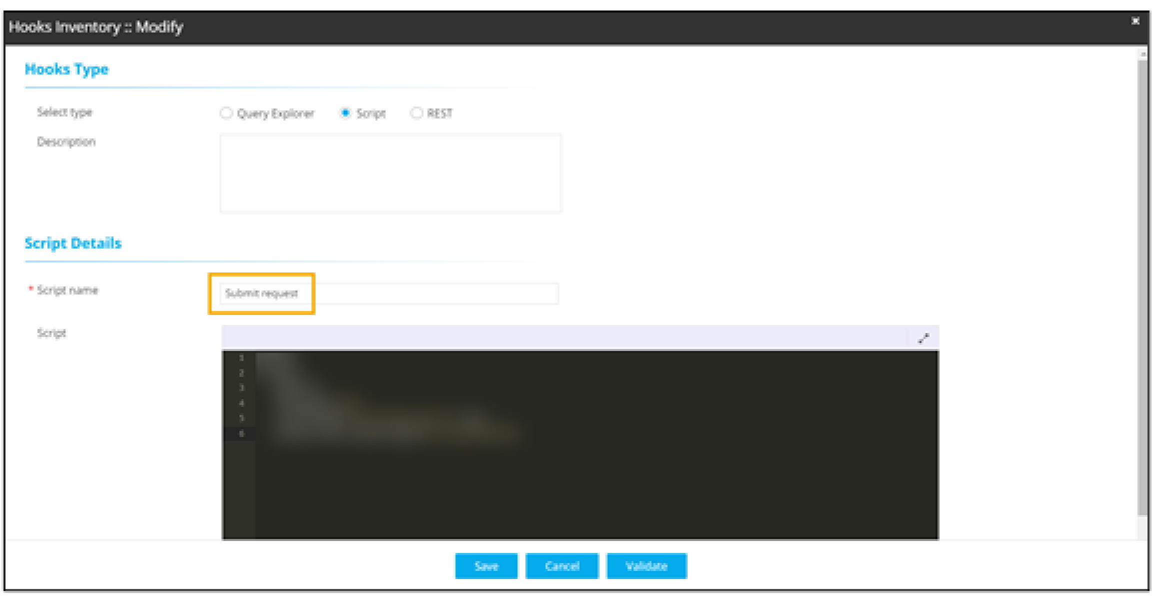


Modifying a Hook

1. In the **Form** window, under the **Hooks** section, hover your mouse over the hook that is to be modified.
2. Click  **(Edit)** icon.



The **Hooks Inventory :: Modify** window opens. You can modify the hook and assign it to the form field.



Note: For more information on Hooks, refer to the Automatic User Guide.

Configuring the Form - Styles

1. For more options to customize the Form, click **Styles**.

Field descriptions for Styles

Field	Description
Weight	Allows you to define the weight of the Form name.
Size	Allows you to define the size of the Form name.
Line Height	Allows you to adjust the space between the Form name, description and the form border.
Color	Allows you to define the color for the Form name.
Align	Allows you to align the Form name to the left, center or right side of the form.
Background	Allows you to define the background color for the form.
Border	Allows you to define the border color, border style, and border width.

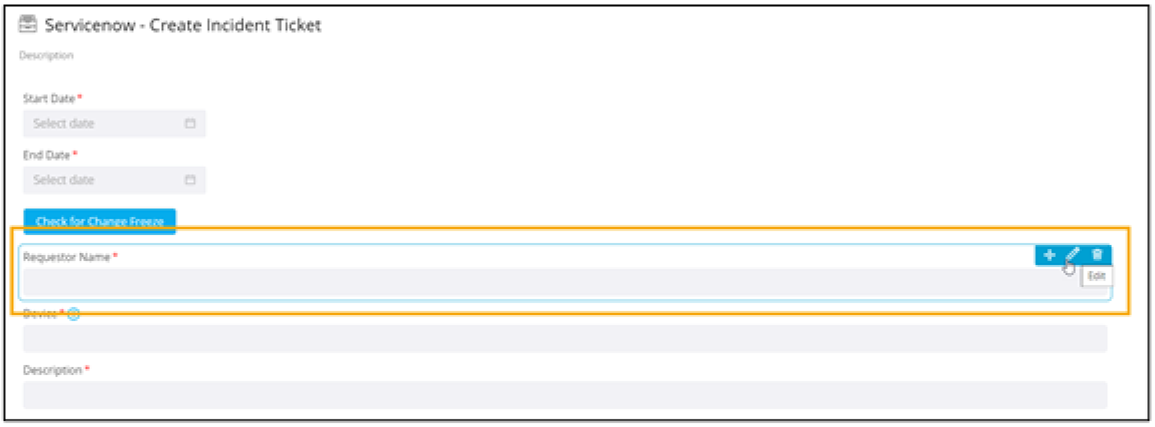
2. To save your settings, click **Save**.
3. To restore the settings under **Styles** to default, click **(Restore)** icon.

Modifying Individual Form Fields

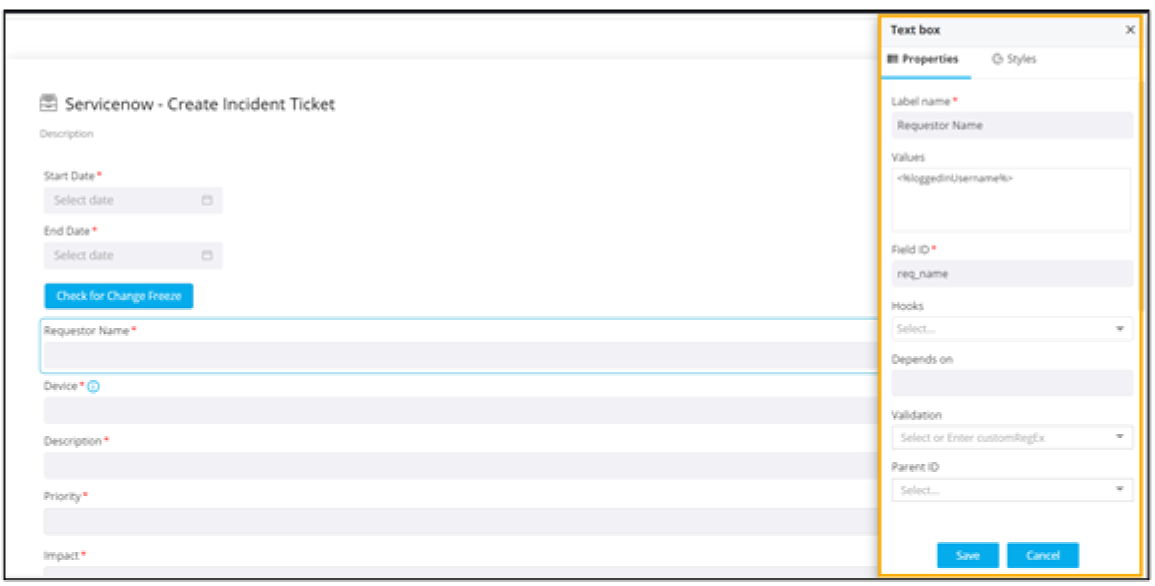
You can also modify, add, or delete individual form fields in an OOB form.

To edit a form field within an OOB form:

1. Hover your mouse over the form field you wish to modify and click **(Edit)** icon.



A pop-up window for the form field is displayed with the **Properties** tab open by default.



2. Modify the form **Properties** and **Style** elements and click **Save**.

Enabling Auto trigger

This feature allows you to define an auto trigger for a form field. Enabling auto trigger fetches data automatically to populate the form field(s). In case Auto trigger is disabled, a icon is displayed next to the form field. Clicking the icon will fetch the data from the database and populate the form field.

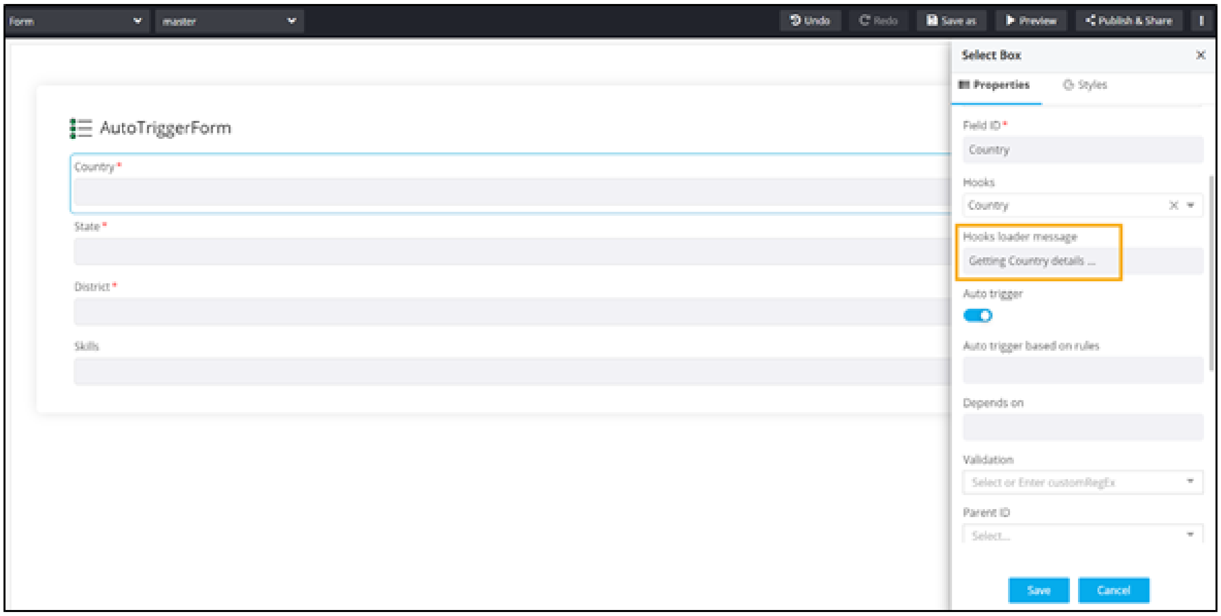
To enable auto trigger for a form field:

1. Design a form.
2. To edit a form field, hover your mouse over the field and click **(Edit)** icon.
3. In the form field window, under the **Properties** tab, define an appropriate hook for this field.

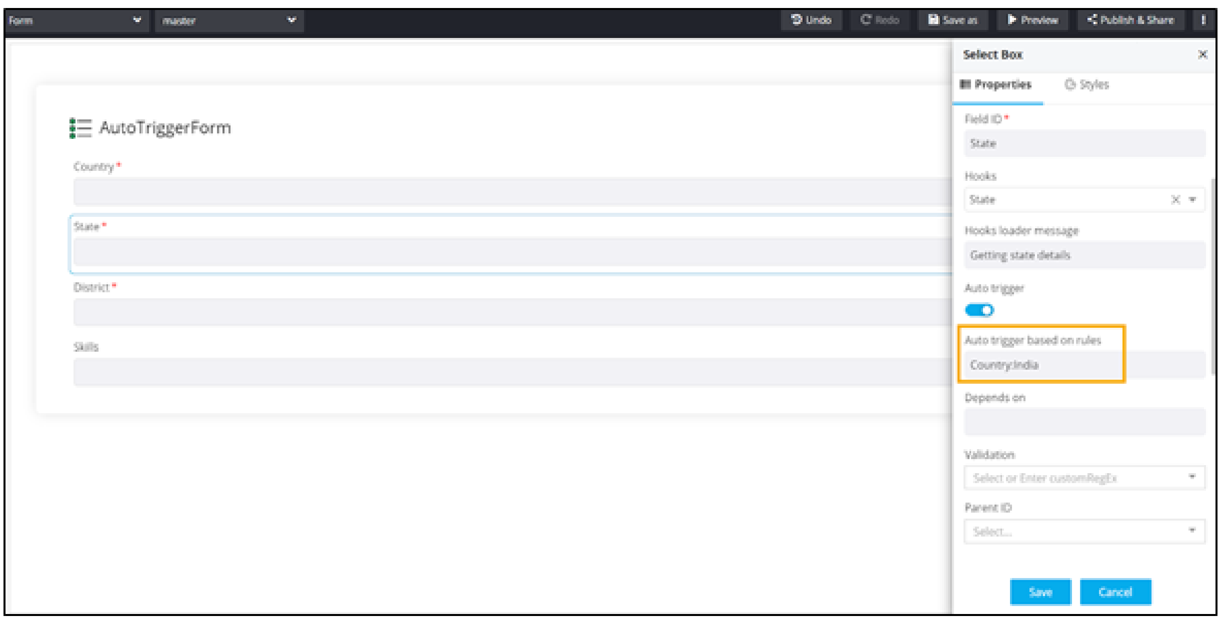
Once the hook is defined, the **Hooks loader message** field is displayed.

> **Note:** For information on assigning hooks, click here.

4. Enter a message in the **Hooks loader message** field. This message is displayed while the form field is loading the field values.



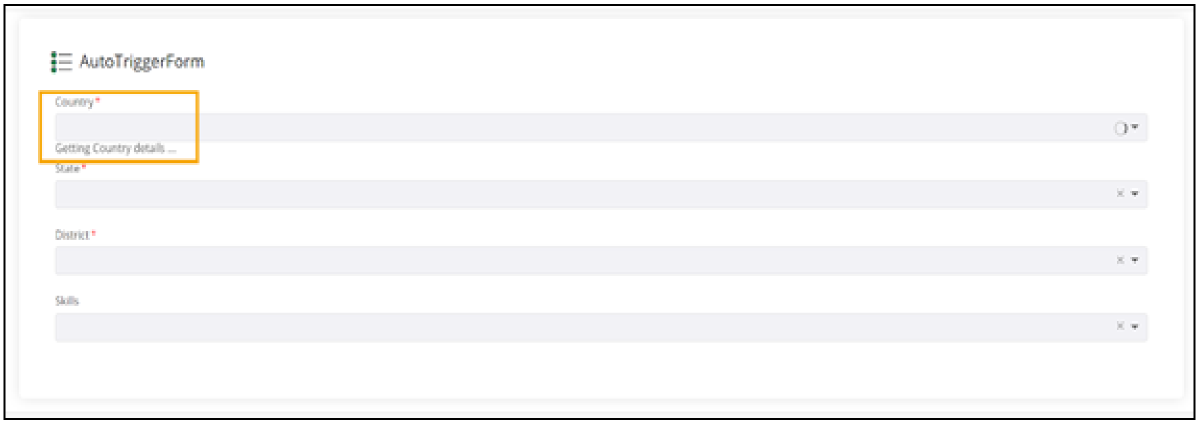
5. To enable **Auto trigger**, turn on the toggle.
6. In the **Auto trigger based on rules** field, define a rule for populating the next field in the form.



Note: This field is displayed only after a hook is assigned to the form field. Rules cannot be defined for the first field in the form.

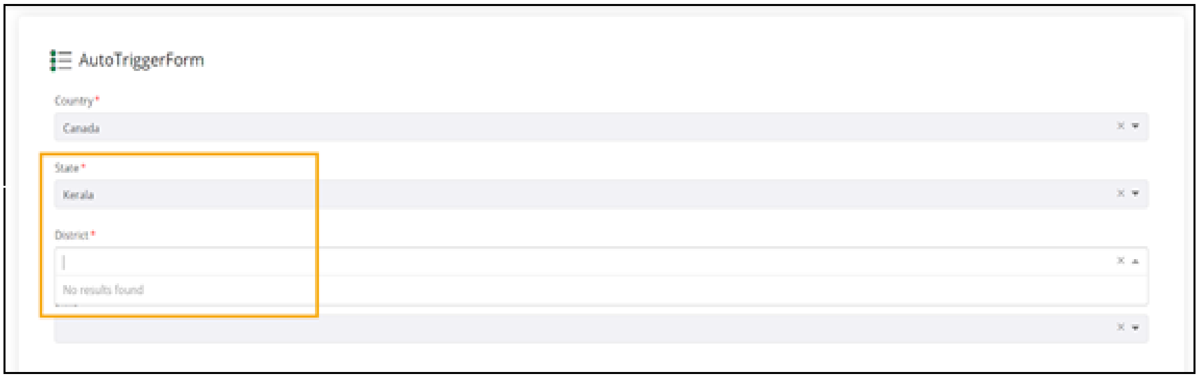
7. Define hooks for the other fields and enable **Auto trigger** for form fields, where required.
8. Add a Hooks loader message and define rules for required fields.
9. Click **Preview**.

- A preview of the form opens in a new tab and the data for the first field is fetched automatically. The Hook loader message is displayed while the values are being fetched.



- The values in the next form field are fetched automatically once a selection is made in the previous form field based on the rules defined.

- Values in the next form field are not fetched/displayed if a selection other than the defined rule is made. For example, the rule “State:TamilNadu” has been defined for the District field. So, if any value other than TamilNadu is selected in the State field, then no results will be displayed in the District field.



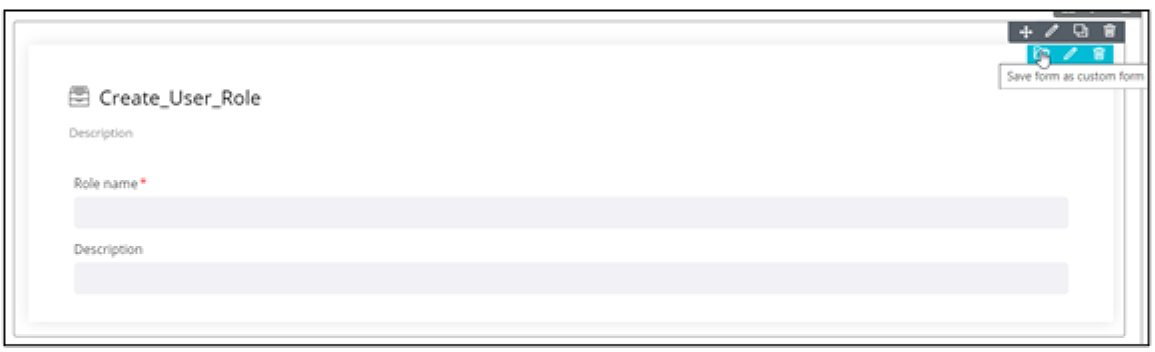
- Once the value TamilNadu is selected in the State field, the values for the District field will be fetched automatically.

Saving Forms to Custom Folder

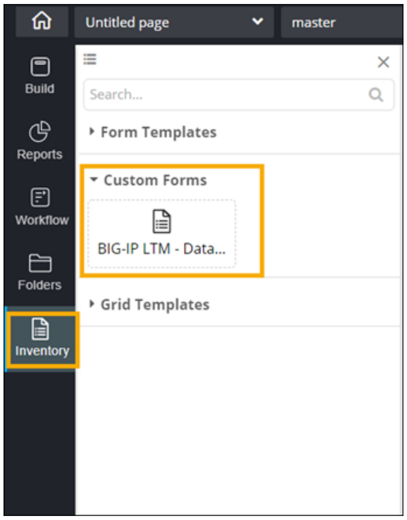
Any form created from scratch or a customized form template can be saved to the Custom folder to be reused later.

To save a form in the custom folder:

1. Hover your mouse over the form and from the top right corner, click  **(Folder)** icon.



2. In the **Confirmation** pop-up window, click **Yes**.
   The form is added to the **Custom** folder in the **Inventory** section.



3. To reuse a custom form, simply drag and drop the required form from the **Custom** folder into the canvas.

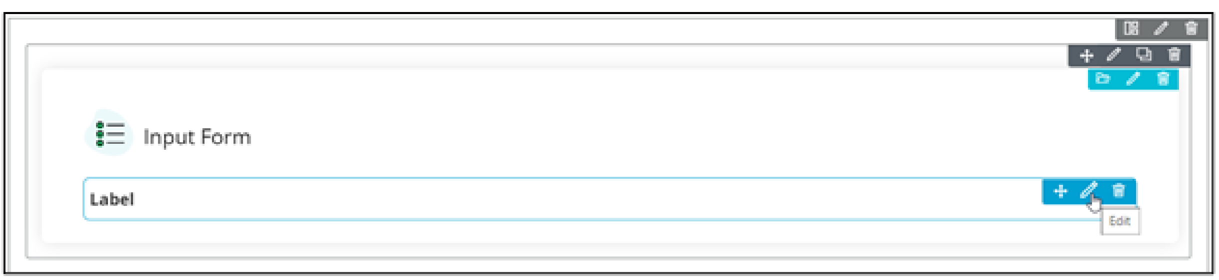
Form Field - Label

The Label field allows you to add a simple text or description to the form.

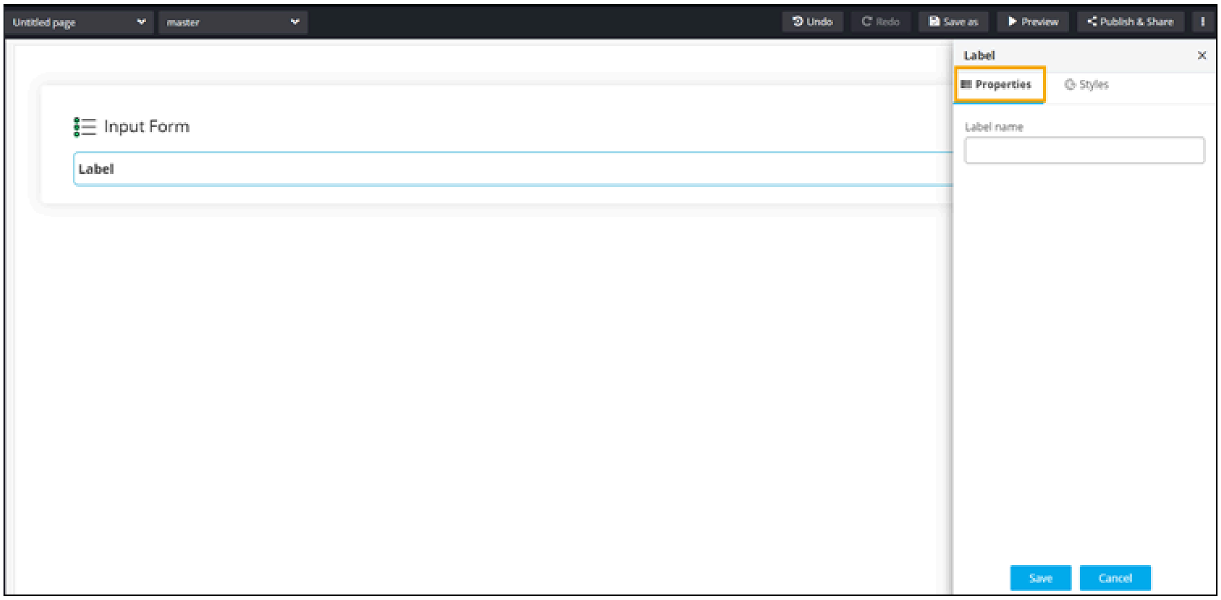
- Configuring the Label field - Properties
- Configuring the Label field - Styles

Configuring the Label field - Properties

1. To add the **Label** field to your form, drag and drop the **Label** component to the design area.
2. To configure the Label widget, hover your mouse over the widget and click  **(Edit)** icon.



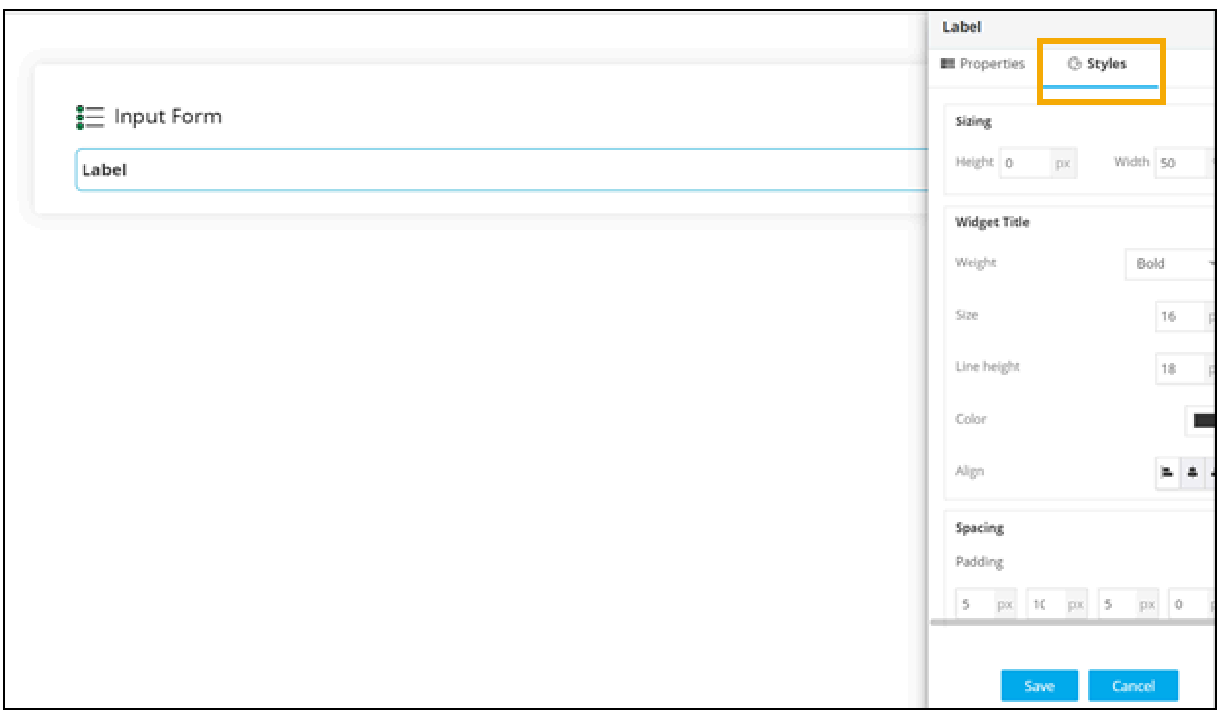
The **Label** pop-up window opens.



3. In the **Label** window, under **Properties**, enter a valid **Label name**.
4. To save your configurations, click **Save**.

Configuring the Label field - Styles

1. For more options to customize the Label, in the **Label** window, click **Styles**.



Field descriptions for Styles

Field	Description
Sizing	Allows you to increase/decrease font **Size** of the field name.
Weight	The **Weight** dropdown list lets you set the font thickness of the field label using one of the following values: • **Default** • **Light** • **Normal** • **Thick**.
Size	Allows you to increase/decrease font size of the field name.
Line Height	Allows you to increase/decrease the line height of the field. This means the space between the field and the field border.
Color	Allows you to define the color of the field name.
Align	Allows you to align the field name to the left, center or right side of the field.
Field Size	Allows you to define the size of the form field as • **Small** • **Medium** • **Large**.

2. To save your settings, click **Save**.
3. To restore the settings under **Styles** to default, click **(Restore)** icon.

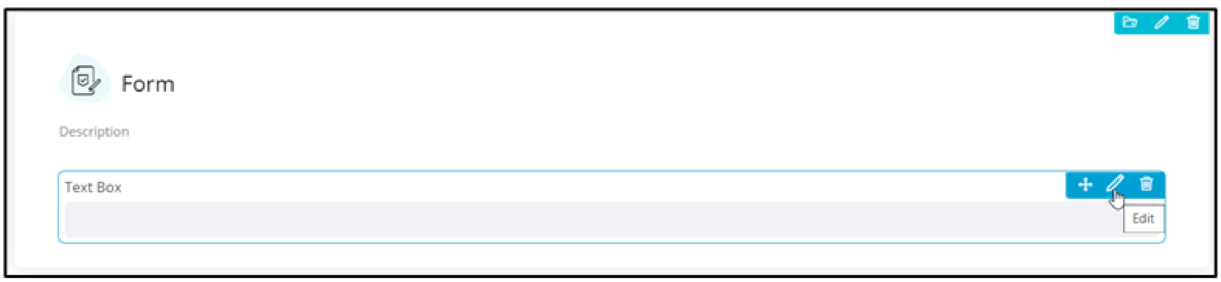
Form Field - Text Box

The Text Box field allows you to add a text box to your form where the user can enter text information.

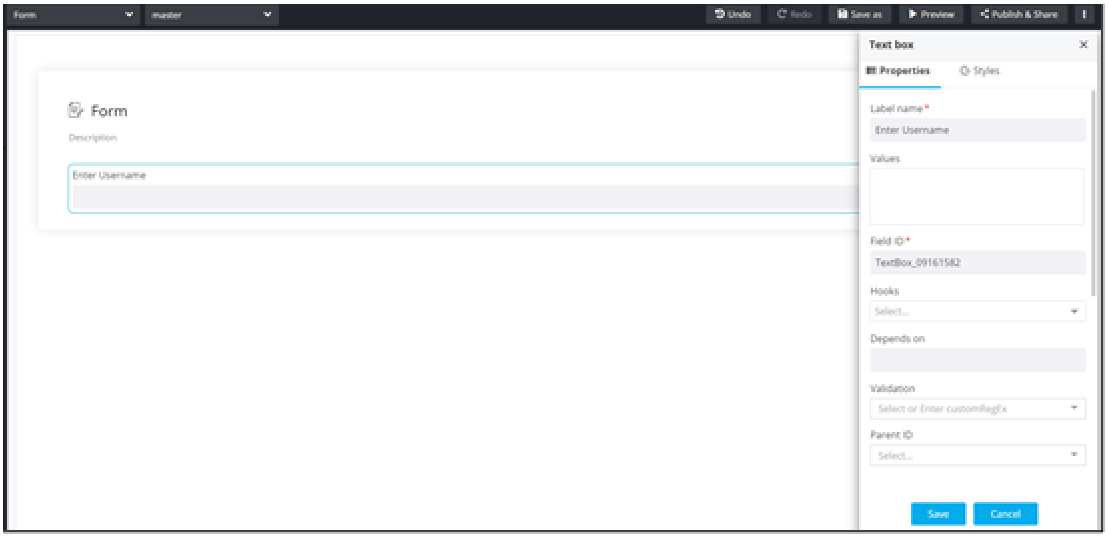
- Configuring the Text Box field - Properties
- Configuring the Text Box field - Styles

Configuring the Text Box field - Properties

1. To add the **Text Box** field to your form, drag and drop the **Text Box** component to the design area.
2. To configure the **Text Box** form field, hover your mouse over the field and click **(Edit)** icon.



The**Text Box**pop-up window opens.
3. In the**Text Box** window, under **Properties**, enter or select the required fields.



Field descriptions for Properties

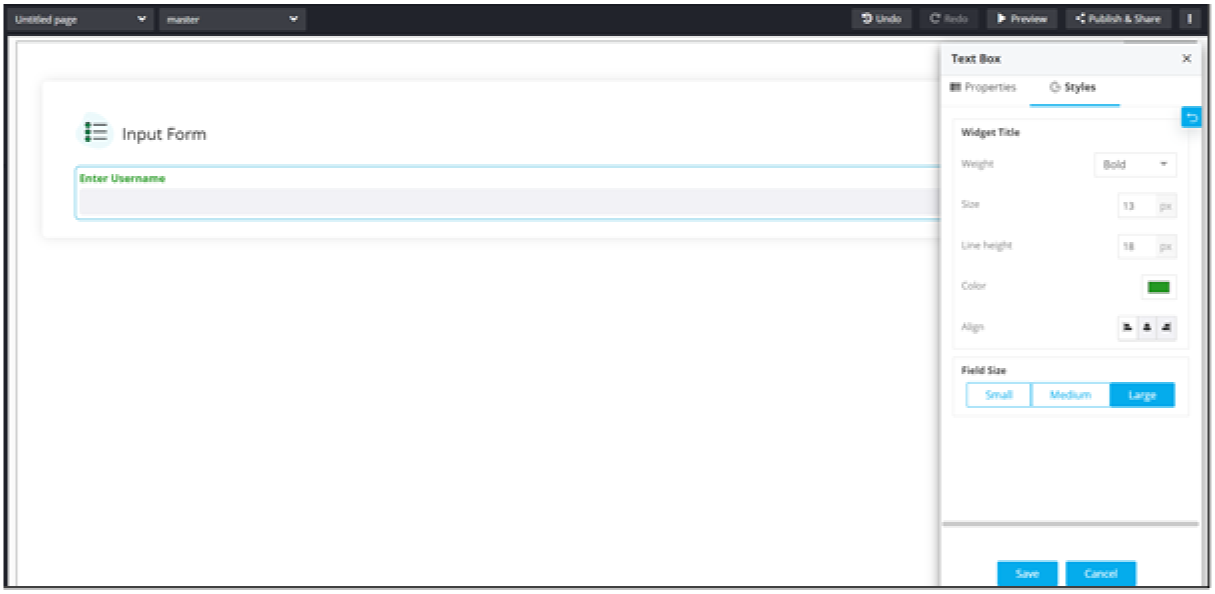
Field	Description
Label name	Enter a valid **Label name** for the form field. For example, Select Date and Time.
Values	Allows you to define field **Values** relevant to a form field. • Values can be delimited by a comma. • Characters not allowed: '&' and '|'.
Field ID	Allows you to assign a unique **Field ID** to refer to the field name. • Allowed characters: '-' and '_' • Character Length: Min (1), Max (50).
Hooks	Allows you to assign an appropriate hook for this field. For example, a hook to fetch the current date. • Select a hook that is relevant to the form field. • Hooks can be either Script or REST API. • Hooks are used to fetch details against a specific form field from the device and/or database and map them to the respective field on the user interface. **Note:** For more information on Assigning Hooks to Form Fields, click here.

Field	Description
Depends on	Allows you to define dependencies between form fields based on 'Field ID' and 'Values'. • Operators used: '&' and '|' • Max character: 550 • Format: [Field Id1: value1 & Field Id2: value1 & Field Id3: value3], [Field Id1: value1 | Field Id2: value1].
Validation	Allows you to select or define a custom regex pattern against the specific form field. • Select from the predefined regex patterns defined in the regex inventory. • Validation ensures that you capture the right information while creating the request.
Parent ID	Allows you to assign a **Parent ID** for fields in a table. • Select the relevant tabular component (Field ID) against which the form field must be mapped to. **Note:** This is used only when defining a tabular element in the form.
Help	Allows you to add **Help** text/tooltip for the field.
Profile Access	Allows you to select the profile from the dropdown to restrict form field display within a template based on the relevant profile.
Mandatory	Allows you to define this field as **Mandatory** or otherwise. If marked as mandatory, an asterisk (*) is displayed next to the field name in the form.
Read only	Allows you to define this field as **Read Only** or otherwise. If marked as Read Only, a red crossed circle is displayed when you hover your mouse over the read only field and does not allow you to enter any information.
Global variable	Allows you to define this field as a **Global Variable**.

4. Click **Save**.

Configuring the Text Box field - Styles

1. For more options to customize the **Text Box**, in the **Date Time Picker** window, click **Styles**.



Field descriptions for Styles

Field	Description
Weight	The **Weight** dropdown list lets you set the font thickness of the field label using one of the following values: • **Default** • **Light** • **Normal** • **Thick**.
Size	Allows you to increase/decrease font size of the field name.
Line Height	Allows you to increase/decrease the line height of the field. This means the space between the field and the field border.
Color	Allows you to define the color of the field name.
Align	Allows you to align the field name to the left, center or right side of the field.

Field	Description
Field Size	Allows you to define the size of the form field as • **Small** • **Medium** • **Large**.

2. To save your settings, click **Save**.
3. To restore the settings under **Styles** to default, click **(Restore)** icon.

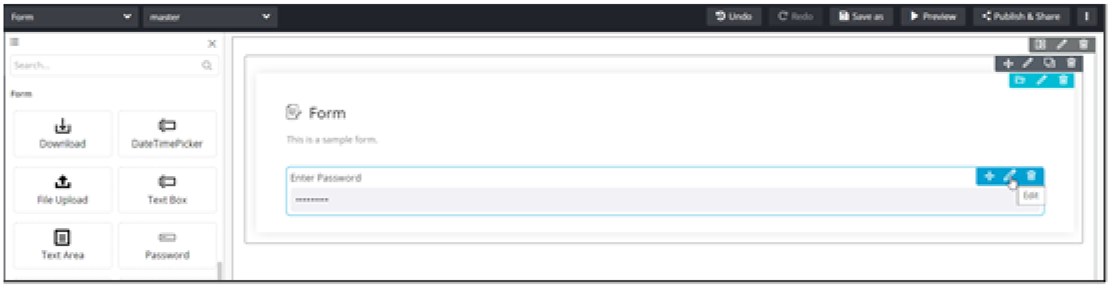
Form Field - Password

The Password field allows you to add a special text field that does not display text when the user types in it. Instead, the text typed by the user is displayed as an asterisk or a bullet to maintain confidentiality of sensitive information.

- Configuring the Password field - Properties
- Configuring the Password field - Styles

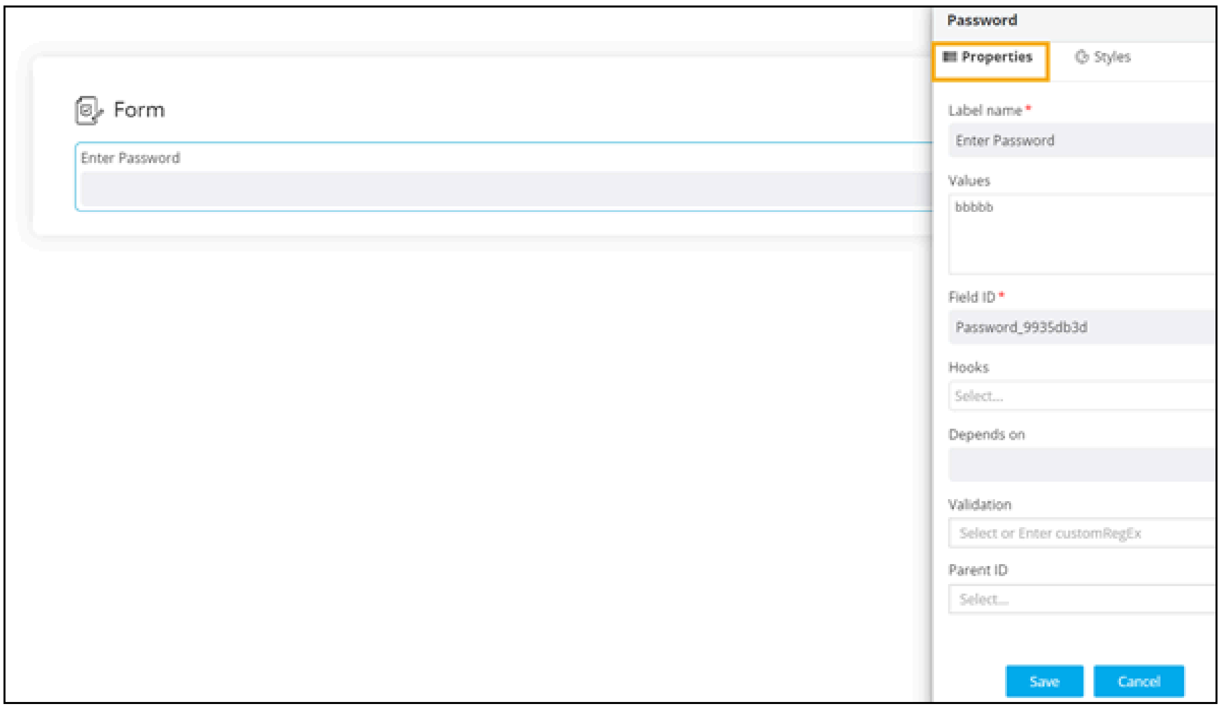
Configuring the Password field - Properties

1. To add the Password field to your form, drag and drop the **Password** component to the design area.
2. To configure the Password field, hover your mouse over the field and click **(Edit)** icon.



The**Password**pop-up window opens.

3. In the**Password** window, under **Properties**, enter or select the required fields.



Field descriptions for Properties

Field	Description
Label name*	Enter a valid **Label name** for the form field. For example, Select Date and Time.
Values	Allows you to define field **Values** relevant to a form field. • Values can be delimited by a comma. • Characters not allowed: '&' and '|'.
Field ID*	Allows you to assign a unique **Field ID** to refer to the field name. • Allowed characters: '-' and '_' • Character Length: Min (1), Max (50).

Field	Description
Hooks	Allows you to assign an appropriate hook for this field. For example, a hook to fetch the current date. • Select a hook that is relevant to the form field. • Hooks can be either Script or REST API. • Hooks are used to fetch details against a specific form field from the device and/or database and map them to the respective field on the user interface. **Note:** For more information on Assigning Hooks to Form Fields, click here.
Depends on	Allows you to define dependencies between form fields based on 'Field ID' and 'Values'. • Operators used: '&' and '|' • Max character: 550 • Format: [Field Id1: value1 & Field Id2: value1 & Field Id3: value3], [Field Id1: value1 | Field Id2: value1].
Validation	Allows you to select or define a custom regex pattern against the specific form field. • Select from the predefined regex patterns defined in the regex inventory. • Validation ensures that you capture the right information while creating the request.
Parent ID	Allows you to assign a **Parent ID** for fields in a table. • Select the relevant tabular component (Field ID) against which the form field must be mapped to. **Note:** This is used only when defining a tabular element in the form.
Help	Allows you to add **Help** text/tooltip for the field.
Profile Access	Allows you to select the profile from the dropdown to restrict form field display within a template based on the relevant profile.
Mandatory	Allows you to define this field as **Mandatory** or otherwise. If marked as mandatory, an asterisk (*) is displayed next to the field name in the form.
Read only	Allows you to define this field as **Read Only** or otherwise. If marked as Read Only, a red crossed circle is displayed when you hover your mouse over the read only field and does not allow you to enter any information.

Field	Description
Global variable	Allows you to define this field as a **Global Variable**.
**: Mandatory fields*	

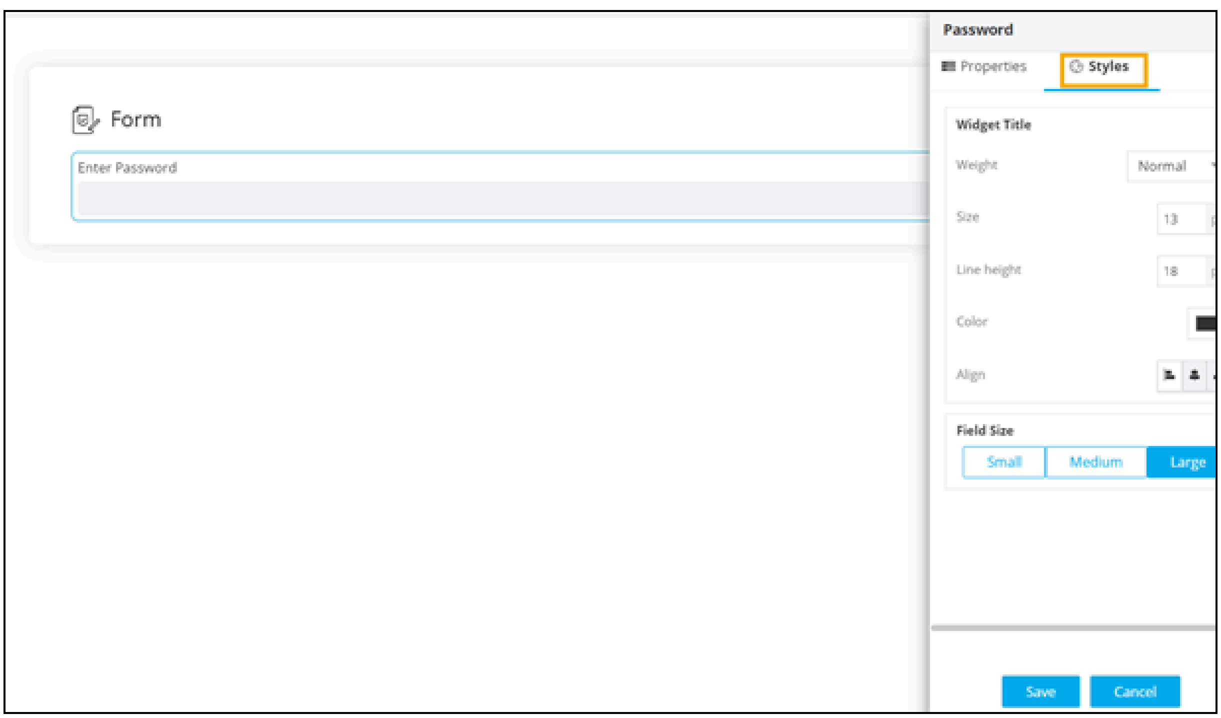
4. Click **Save**.
5. To make the password visible, click  **(View)** icon.

   The password is visible.

6. To hide it again, click  **(Hide)** icon.

Configuring the Password field - Styles

1. For more options to customize the Password, in the **Password** window, click **Styles**.



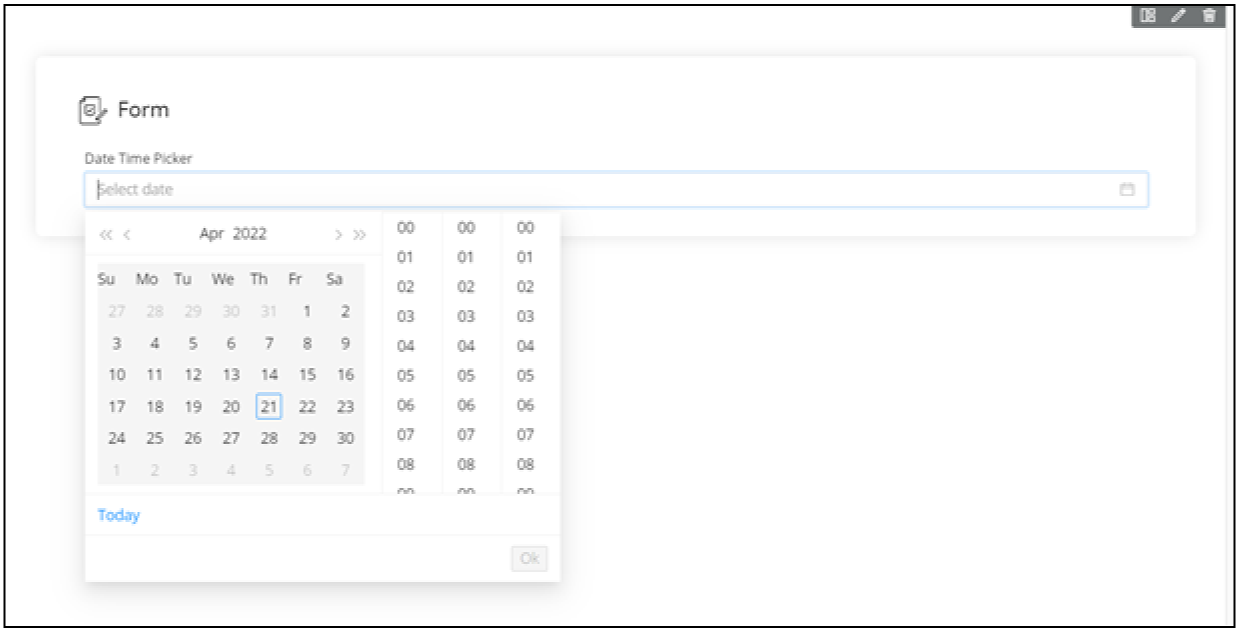
Field descriptions for Styles

Field	Description
Weight	The **Weight** dropdown list lets you set the font thickness of the field label using one of the following values: • **Default** • **Light** • **Normal** • **Thick**.
Size	Allows you to increase/decrease font size of the field name.
Line Height	Allows you to increase/decrease the line height of the field. This means the space between the field and the field border.
Color	Allows you to define the color of the field name.
Align	Allows you to align the field name to the left, center or right side of the field.
Field Size	Allows you to define the size of the form field as • **Small** • **Medium** • **Large**.

2. To save your settings, click **Save**.
3. To restore the settings under **Styles** to default, click  **(Restore)** icon.

Form Field - Date Time Picker

The Date and Time picker field enables you to enter date and time information and retrieve information based on the inputs provided.



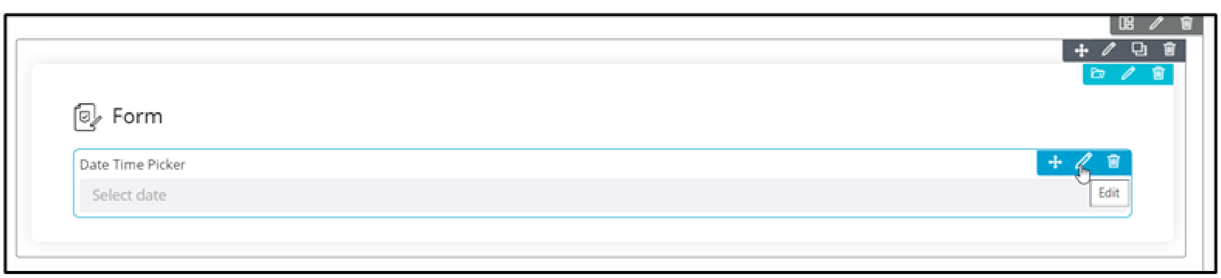
1. Click the **Date Time Picker** field.
2. Select the date from the pop-out calendar.
3. Select the time from the columns.

• Configuring the Date Time Picker - Properties

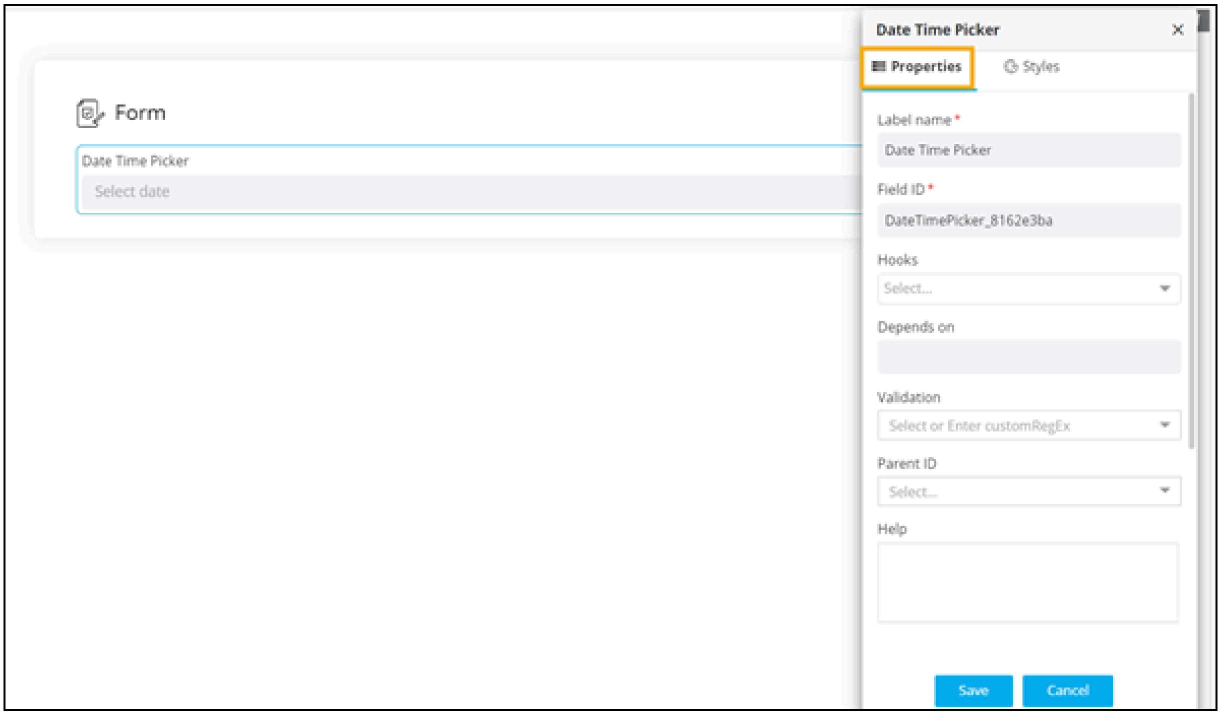
• Configuring the Date Time Picker - Styles

Configuring the Date Time Picker - Properties

1. To add the Date Time Picker to your form, drag and drop the **Date Time Picker** component to the design area.
2. To configure the **Date Time Picker** form field, hover your mouse over the field and click **(Edit)** icon.



The **Date Time Picker** pop-up window opens.



3. In the **Date Time Picker** window, under **Properties**, enter or select the required fields.

Field descriptions for Properties

Field	Description
Label name*	Enter a valid **Label name** for the form field. For example, Select Date and Time.
Field ID*	Allows you to assign a unique **Field ID** to refer to the field name. • Allowed characters: '-' and '_' • Character Length: Min (1), Max (50).
Hooks	Allows you to assign an appropriate hook for this field. For example, a hook to fetch the current date. • Select a hook that is relevant to the form field. • Hooks can be either Script or REST API. • Hooks are used to fetch details against a specific form field from the device and/or database and map them to the respective field on the user interface. **Note:** For more information on Assigning Hooks to Form Fields, click here.

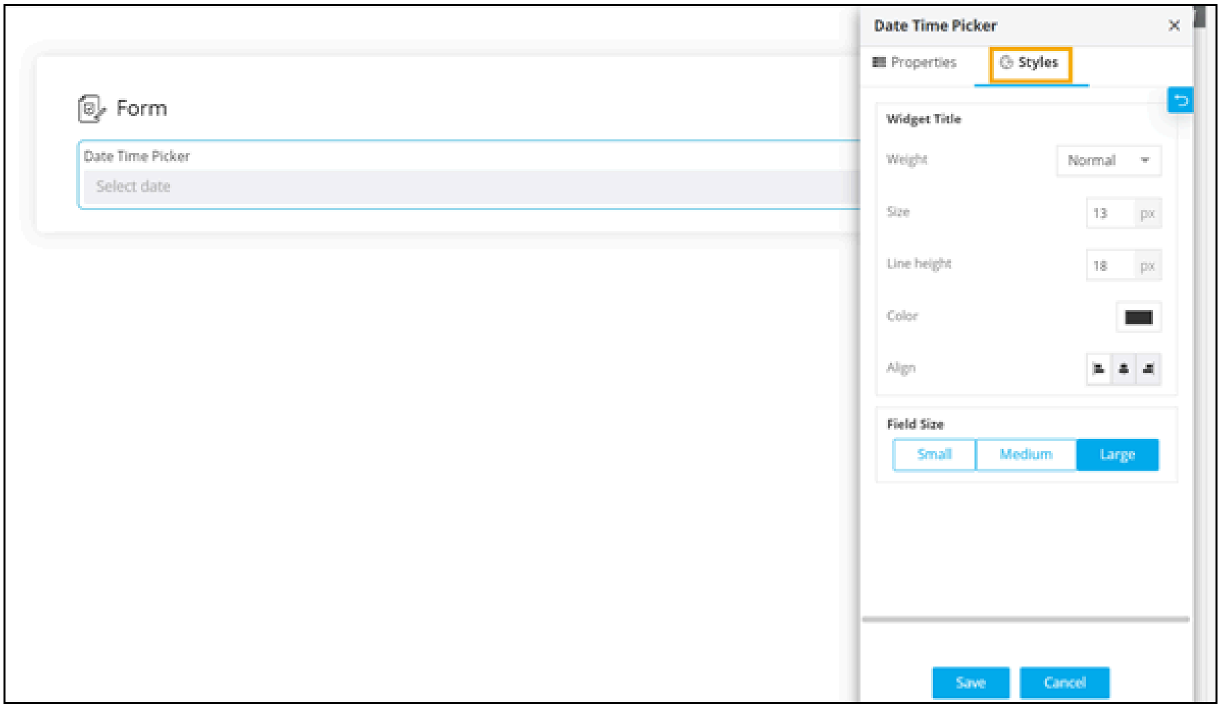
Field	Description
Depends on	Allows you to define dependencies between form fields based on 'Field ID' and 'Values'. • Operators used: '&' and '|' • Max character: 550 • Format: [Field Id1: value1 & Field Id2: value1 & Field Id3: value3], [Field Id1: value1 | Field Id2: value1].
Validation	Allows you to select or define a custom regex pattern against the specific form field. • Select from the predefined regex patterns defined in the regex inventory. • Validation ensures that you capture the right information while creating the request.
Parent ID	Allows you to assign a **Parent ID** for fields in a table. • Select the relevant tabular component (Field ID) against which the form field must be mapped to. **Note:** This is used only when defining a tabular element in the form.
Help	Allows you to add **Help** text/tooltip for the field.
Profile Access	Allows you to select the profile from the dropdown to restrict form field display within a template based on the relevant profile.
Mandatory	Allows you to define this field as **Mandatory** or otherwise. If marked as mandatory, an asterisk (*) is displayed next to the field name in the form.
Read only	Allows you to define this field as **Read Only** or otherwise. If marked as Read Only, a red crossed circle is displayed when you hover your mouse over the read only field and does not allow you to enter any information.
Global variable	Allows you to define this field as a **Global Variable**.
**: Mandatory fields*	

4. Click **Save**.
5. Click the **Date Time Picker** field.

6. Select the date from the pop-out calendar.
7. Select the time from the columns.

Configuring the Date Time Picker - Styles

1. For more options to customize the **Date Time Picker**, in the **Date Time Picker** window, click **Styles**.



Field descriptions for Styles

Field	Description
Weight	The **Weight** dropdown list lets you set the font thickness of the field label using one of the following values: • **Default** • **Light** • **Normal** • **Thick**.
Size	Allows you to increase/decrease font size of the field name.
Line Height	Allows you to increase/decrease the line height of the field. This means the space between the field and the field border.
Color	Allows you to define the color of the field name.

Field	Description
Align	Allows you to align the field name to the left, center or right side of the field.
Field Size	Allows you to define the size of the form field as • **Small** • **Medium** • **Large**.

2. To save your settings, click **Save**.
3. To restore the settings under **Styles** to default, click **(Restore)** icon.

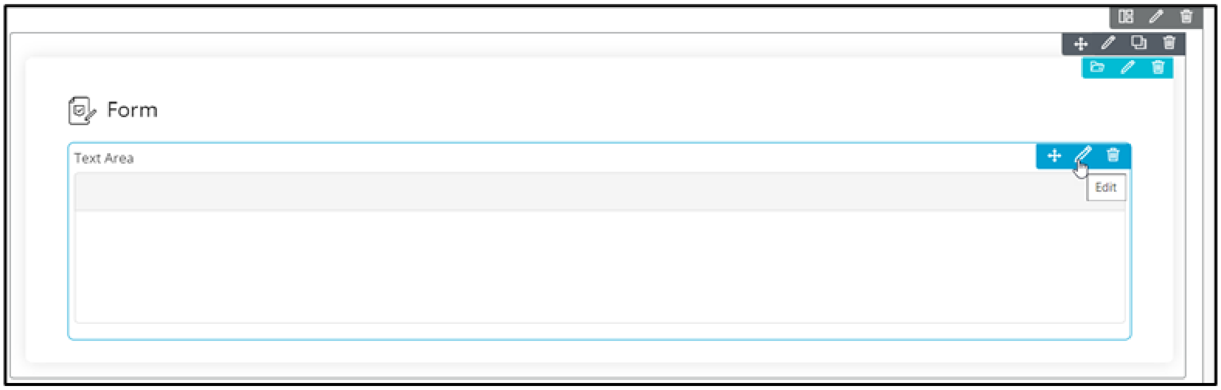
Form Field - Text Area

You can add a Text Area field to the form on your page. This form field allows you to add a multi-line text input area to the page. This can be used to collect user inputs such as comments of feedback or add a description for a form field.

- Configuring the Text Area field - Properties
- Configuring the Text Area field - Styles

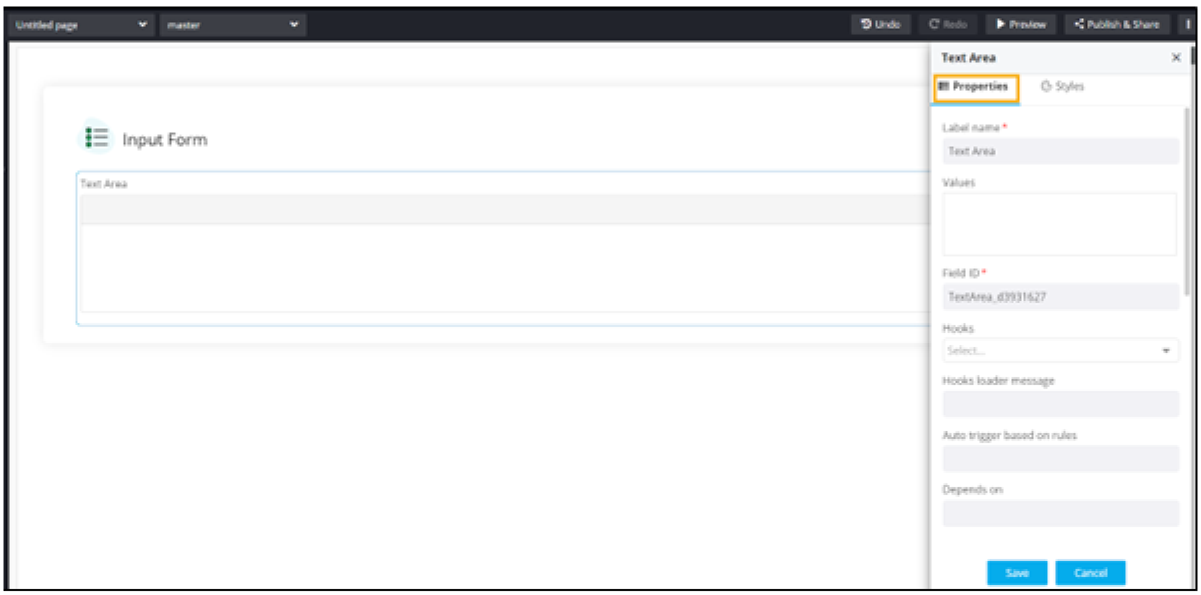
Configuring the Text Area field - Properties

1. To add the Text Area field to your form, drag and drop the **Text Area** component to the design area.
2. To configure the **Text Area** field, hover your mouse over the field and click **(Edit)** icon.



The **Text Area**pop-up window opens.

3. In the **Text Area** window, under **Properties**, enter or select the required fields.



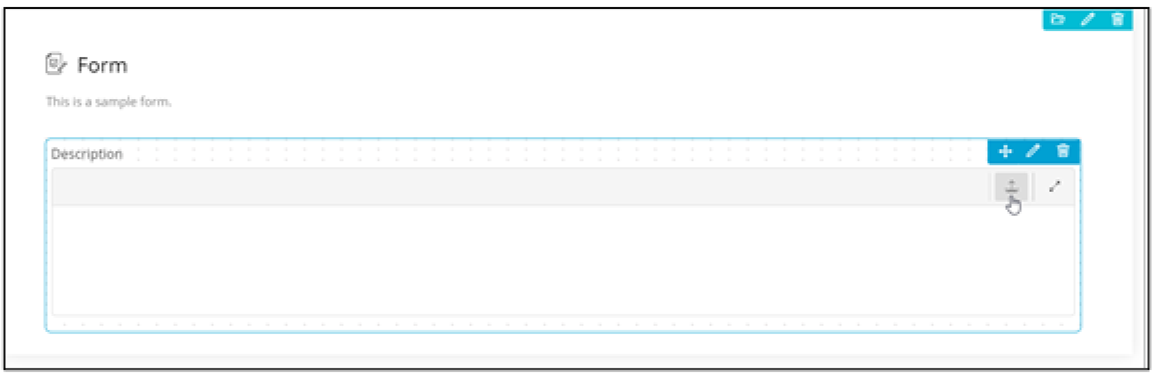
Field descriptions for Properties

Field	Description
Label name	Enter a valid **Label name** for the form field. For example, Select Date and Time.
Values	Allows you to define field **Values** relevant to a form field. • Values can be delimited by a comma. • Characters not allowed: '&' and '|'.
Field ID	Allows you to assign a unique **Field ID** to refer to the field name. • Allowed characters: '-' and '_' • Character Length: Min (1), Max (50).

Field	Description
Hooks	Allows you to assign an appropriate hook for this field. For example, a hook to fetch the current date. • Select a hook that is relevant to the form field. • Hooks can be either Script or REST API. • Hooks are used to fetch details against a specific form field from the device and/or database and map them to the respective field on the user interface. **Note:** For more information on Assigning Hooks to Form Fields, click here.
Depends on	Allows you to define dependencies between form fields based on 'Field ID' and 'Values'. • Operators used: '&' and '|' • Max character: 550 • Format: [Field Id1: value1 & Field Id2: value1 & Field Id3: value3], [Field Id1: value1 | Field Id2: value1].
Validation	Allows you to select or define a custom regex pattern against the specific form field. • Select from the predefined regex patterns defined in the regex inventory. • Validation ensures that you capture the right information while creating the request.
Parent ID	Allows you to assign a **Parent ID** for fields in a table. • Select the relevant tabular component (Field ID) against which the form field must be mapped to. **Note:** This is used only when defining a tabular element in the form.
Help	Allows you to add **Help** text/tooltip for the field.
Profile Access	Allows you to select the profile from the dropdown to restrict form field display within a template based on the relevant profile.
Mandatory	Allows you to define this field as **Mandatory** or otherwise. If marked as mandatory, an asterisk (*) is displayed next to the field name in the form.

Field	Description
Read only	Allows you to define this field as **Read Only** or otherwise. If marked as Read Only, a red crossed circle is displayed when you hover your mouse over the read only field and does not allow you to enter any information.
Global variable	Allows you to define this field as a **Global Variable**.
Hidden	Allows you to hide the field in the form.
File upload	Allows you to add a **File upload** icon **(Upload)** icon in the Text Area.

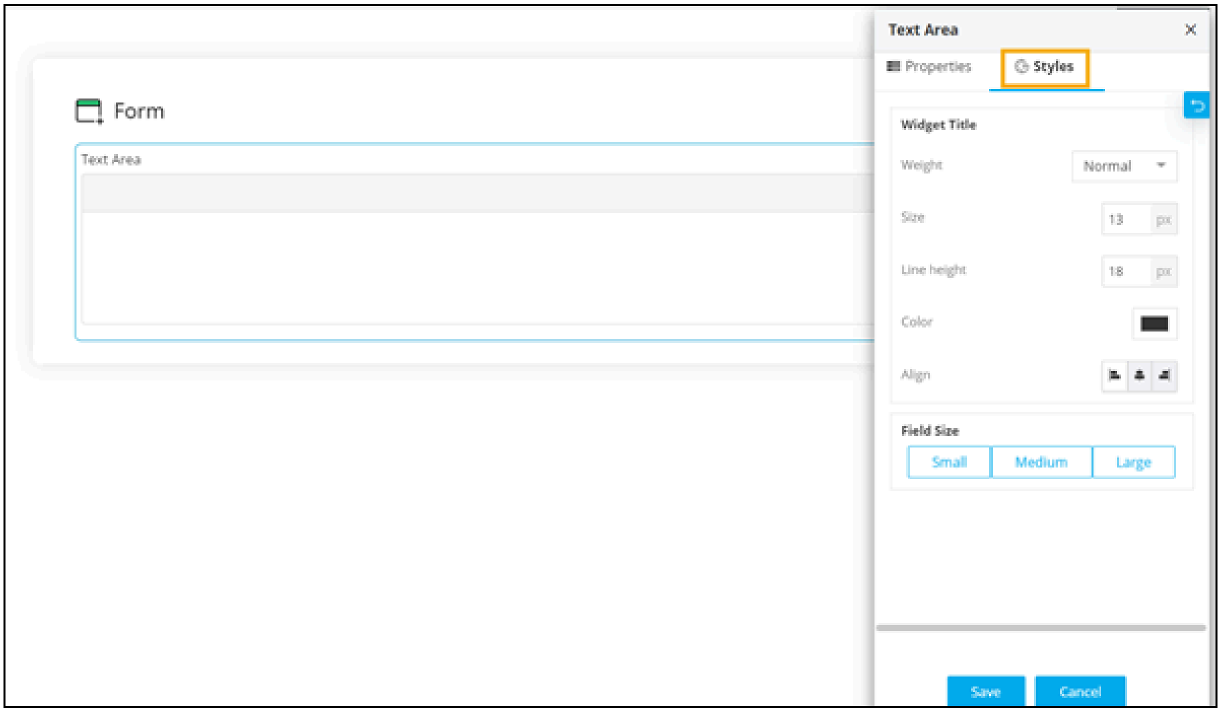
4. To add a file upload icon, in the **Text Area** window, under **Properties**, turn on the **File upload** toggle.
5. To upload a file, in the text area, click **(Upload)** icon.



6. To maximize the text area field, from the top right corner of the field, click **(Maximize)** icon.

Configuring the Text Area field - Styles

1. For more options to customize the Text Area, in the **Text Area**window, click **Styles**.



Field descriptions for Styles

Field	Description
Weight	The **Weight** dropdown list lets you set the font thickness of the field label using one of the following values: • **Default** • **Light** • **Normal** • **Thick**.
Size	Allows you to increase/decrease font size of the field name.
Line Height	Allows you to increase/decrease the line height of the field. This means the space between the field and the field border.
Color	Allows you to define the color of the field name.
Align	Allows you to align the field name to the left, center or right side of the field.

Field	Description
Field Size	Allows you to define the size of the form field as • **Small** • **Medium** • **Large**.

2. To save your settings, click **Save**.
3. To restore the settings under **Styles** to default, click **(Restore)** icon.

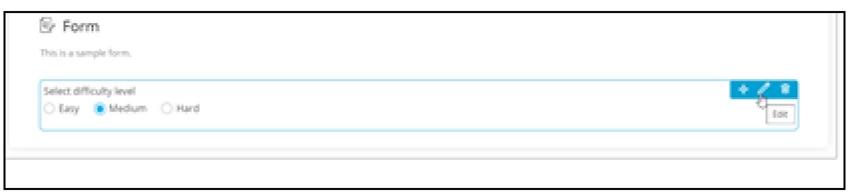
Form Field - Radio Button

The Radio Button field allows you to select one option from the predefined set of options.

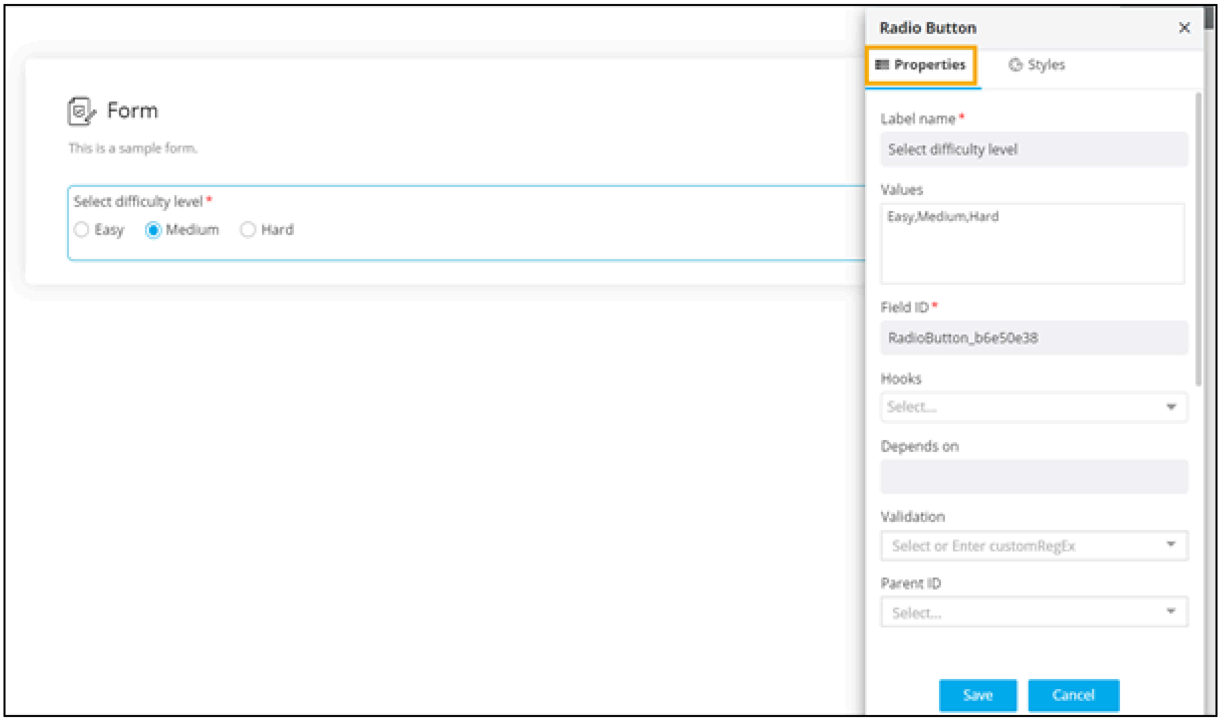
- Configuring the Radio Button field - Properties
- Configuring the Radio Button field - Styles

Configuring the Radio Button field - Properties

1. To add the Radio Button field to your form, drag and drop the **Radio Button** component to the design area.
2. To configure the Radio Button, hover your mouse over the field and click **(Edit)** icon.



   The**Radio Button**pop-up window opens.
3. In the**Radio Button** window, under **Properties**, enter or select the required fields.



Field descriptions for Properties

Field	Description
Label name*	Enter a valid **Label name** for the form field. For example, Select Date and Time.
Values	Allows you to define field **Values** relevant to a form field. • Values can be delimited by a comma. • Characters not allowed: ‘&’ and ‘|’.
Field ID*	Allows you to assign a unique **Field ID** to refer to the field name. • Allowed characters: ‘-‘ and ‘_’ • Character Length: Min (1), Max (50).
Hooks	Allows you to assign an appropriate hook for this field. For example, a hook to fetch the current date. • Select a hook that is relevant to the form field. • Hooks can be either Script or REST API. • Hooks are used to fetch details against a specific form field from the device and/or database and map them to the respective field on the user interface.

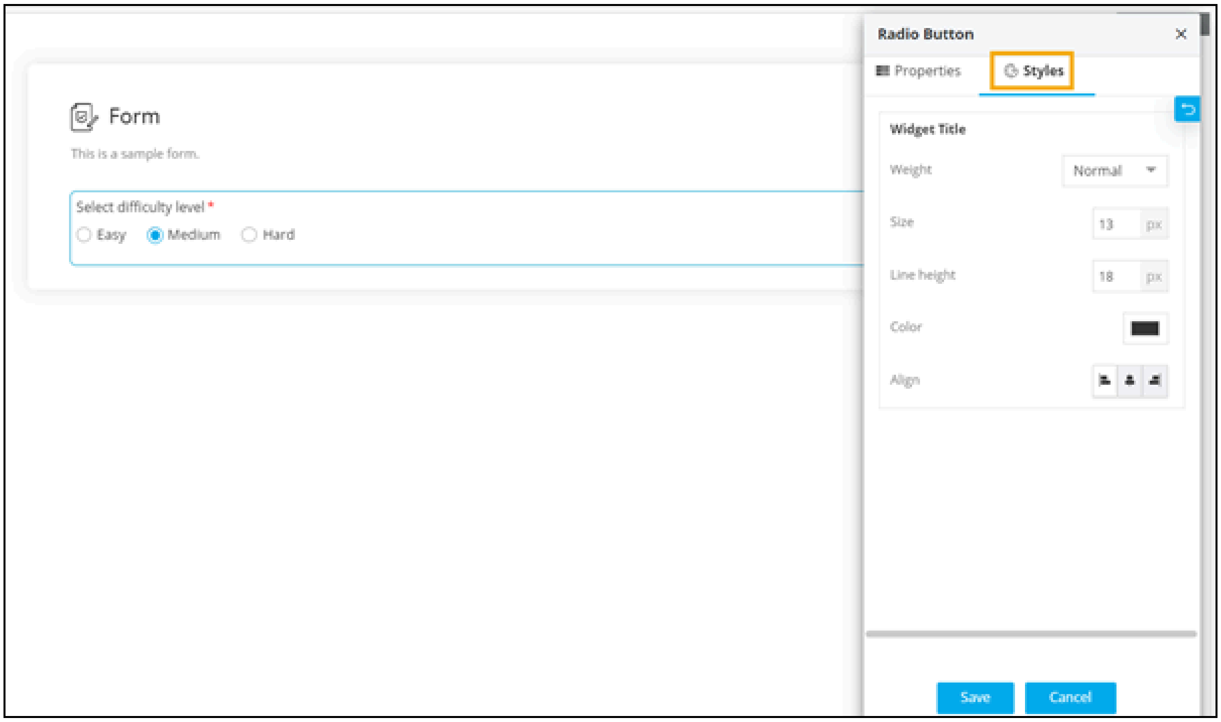
Field	Description
	Note: For more information on Assigning Hooks to Form Fields, click here.
Depends on	Allows you to define dependencies between form fields based on 'Field ID' and 'Values'. • Operators used: '&' and '|' • Max character: 550 • Format: [Field Id1: value1 & Field Id2: value1 & Field Id3: value3], [Field Id1: value1 | Field Id2: value1].
Validation	Allows you to select or define a custom regex pattern against the specific form field. • Select from the predefined regex patterns defined in the regex inventory. • Validation ensures that you capture the right information while creating the request.
Parent ID	Allows you to assign a **Parent ID** for fields in a table. Select the relevant tabular component (Field ID) against which the form field must be mapped to. **Note:** This is used only when defining a tabular element in the form.
Help	Allows you to add **Help** text/tooltip for the field.
Profile Access	Allows you to select the profile from the dropdown to restrict form field display within a template based on the relevant profile.
Mandatory	Allows you to define this field as **Mandatory** or otherwise. If marked as mandatory, an asterisk (*) is displayed next to the field name in the form.
Read only	Allows you to define this field as **Read Only** or otherwise. If marked as Read Only, a red crossed circle is displayed when you hover your mouse over the field and does not allow you to enter any information.
Hidden	Allows you to hide the field in the form.
Global variable	Allows you to define this field as a **Global Variable**.

Field	Description
*: Mandatory fields	

4. Click **Save**.

Configuring the Radio Button field - Styles

1. For more options to customize the Radio Button, in the **Radio Button** window, click **Styles**.



Field descriptions for Styles

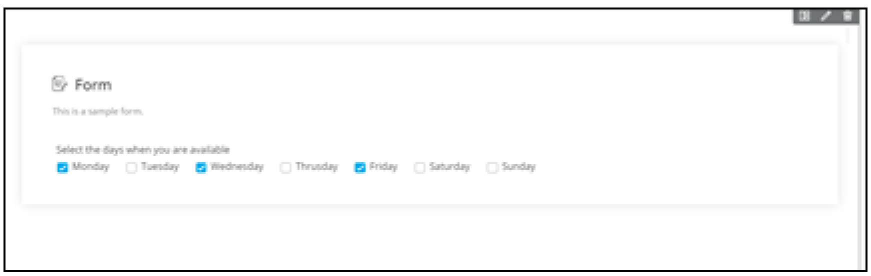
Field	Description
Weight	The **Weight** dropdown list lets you set the font thickness of the field label using one of the following values: • **Default** • **Light** • **Normal** • **Thick**.
Size	Allows you to increase/decrease font size of the field name.

Field	Description
Line Height	Allows you to increase/decrease the line height of the field. This means the space between the field and the field border.
Color	Allows you to define the color of the field name.
Align	Allows you to align the field name to the left, center or right side of the field.

2. To save your settings, click **Save**.
3. To restore the settings under **Styles** to default, click **(Restore)** icon.

Form Field - Checkbox

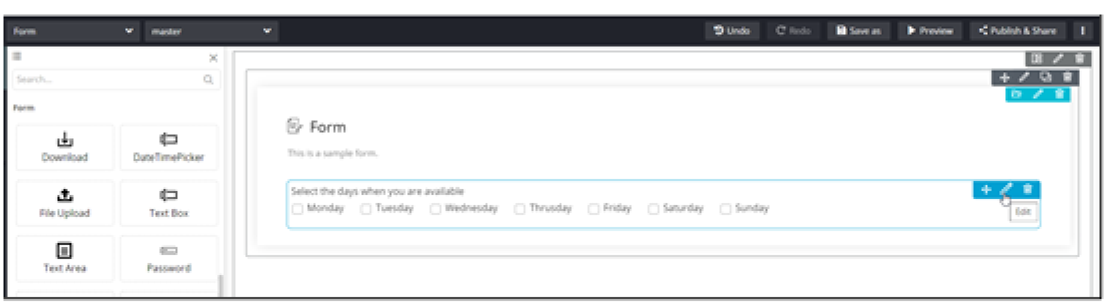
The Checkbox field allows you to make one or more choice(es) from the available options.



- Configuring the Checkbox field - Properties
- Configuring the Checkbox field - Styles

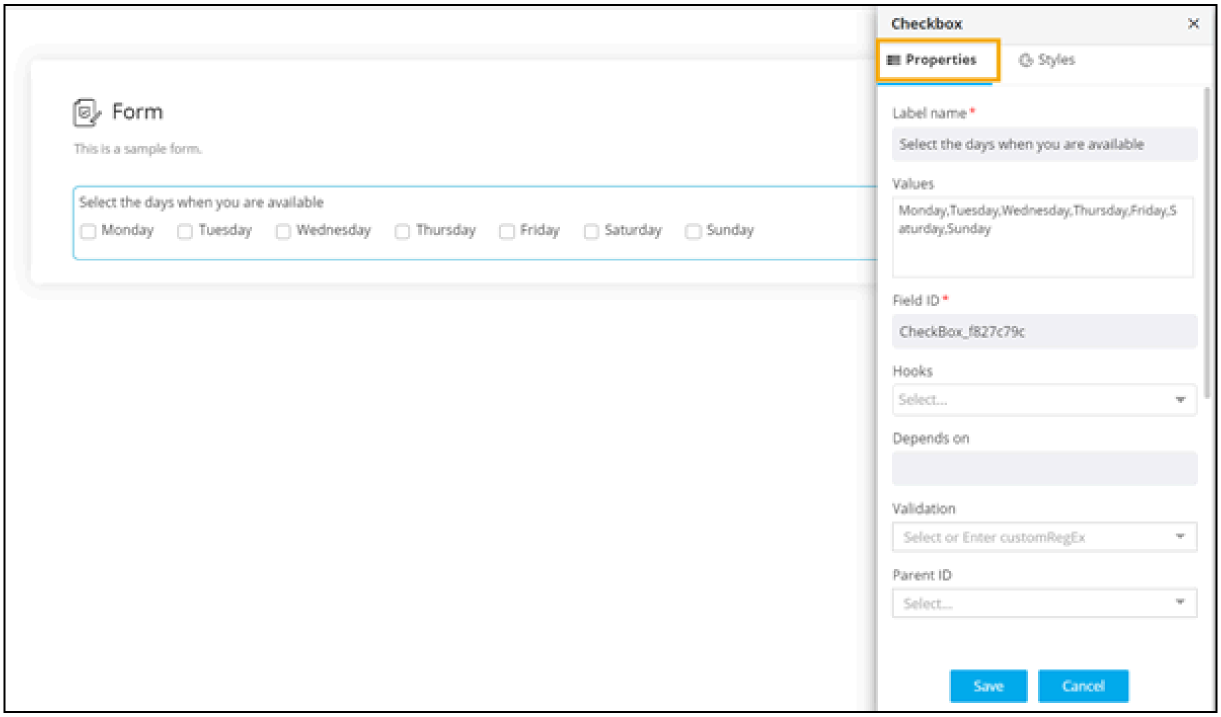
Configuring the Checkbox field - Properties

1. To add the Checkbox field to your form, drag and drop the **Checkbox** component to the design canvas.
2. To configure the Checkbox, hover your mouse over the field and click **(Edit)** icon.



The **Checkbox** pop-up window opens.

3. In the **Checkbox** window, under **Properties**, enter or select the required fields.



Field descriptions for Properties

Field	Description
Label name*	Enter a valid **Label name** for the form field. For example, Select Date and Time.
Values	Allows you to define field **Values** relevant to a form field. • Values can be delimited by a comma. • Characters not allowed: '&' and '|'.
Field ID*	Allows you to assign a unique **Field ID** to refer to the field name. • Allowed characters: '-' and '_' • Character Length: Min (1), Max (50).
Hooks	Allows you to assign an appropriate hook for this field. For example, a hook to fetch the current date. • Select a hook that is relevant to the form field. • Hooks can be either Script or REST API. • Hooks are used to fetch details against a specific form field from the device and/or database and map them to the respective field on the user interface.

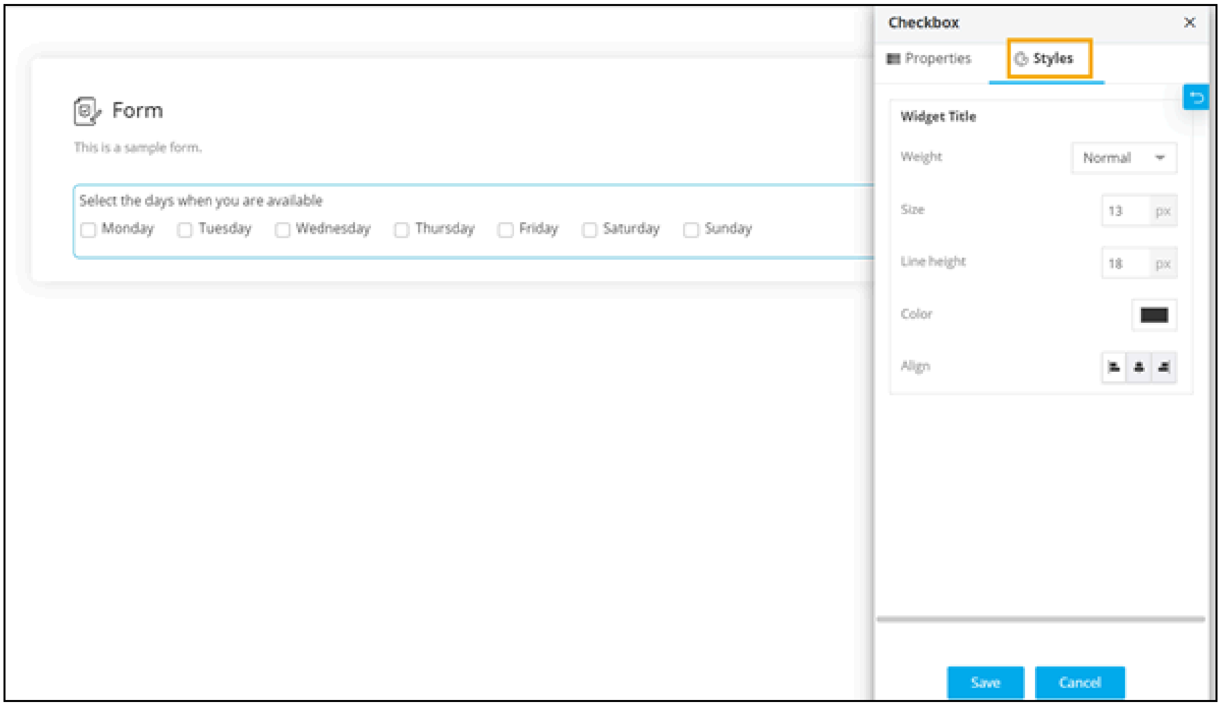
Field	Description
	Note: For more information on Assigning Hooks to Form Fields, click here.
Depends on	Allows you to define dependencies between form fields based on 'Field ID' and 'Values'. • Operators used: '&' and '|' • Max character: 550 • Format: [Field Id1: value1 & Field Id2: value1 & Field Id3: value3], [Field Id1: value1 | Field Id2: value1].
Validation	Allows you to select or define a custom regex pattern against the specific form field. • Select from the predefined regex patterns defined in the regex inventory. • Validation ensures that you capture the right information while creating the request.
Parent ID	Allows you to assign a **Parent ID** for fields in a table. • Select the relevant tabular component (Field ID) against which the form field must be mapped to. **Note:** This is used only when defining a tabular element in the form.
Help	Allows you to add **Help** text/tooltip for the field.
Profile Access	Allows you to select the profile from the dropdown to restrict form field display within a template based on the relevant profile.
Mandatory	Allows you to define this field as **Mandatory** or otherwise. If marked as mandatory, an asterisk (*) is displayed next to the field name in the form.
Read only	Allows you to define this field as **Read Only** or otherwise. If marked as Read Only, a red crossed circle is displayed when you hover your mouse over the read only field and does not allow you to enter any information.
Global variable	Allows you to define this field as a **Global Variable**.

Field	Description
*: Mandatory fields	

4. Click **Save**.

Configuring the Checkbox field - Styles

1. For more options to customize the Checkbox, in the **Checkbox**window, click **Styles**.



Field descriptions for Styles

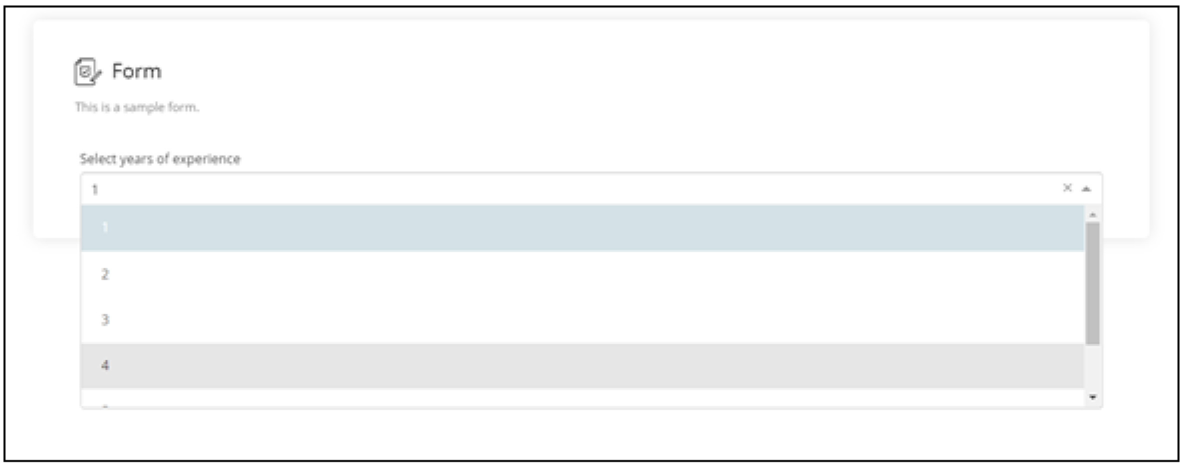
Field	Description
Weight	The **Weight** dropdown list lets you set the font thickness of the field label using one of the following values: • **Default** • **Light** • **Normal** • **Thick**.
Size	Allows you to increase/decrease font size of the field name.
Line Height	Allows you to increase/decrease the line height of the field. This means the space between the field and the field border.

Field	Description
Color	Allows you to define the color of the field name.
Align	Allows you to align the field name to the left, center or right side of the field.

2. To save your settings, click **Save**.
3. To restore the settings under **Styles** to default, click  **(Restore)** icon.

Form Field - Select Box

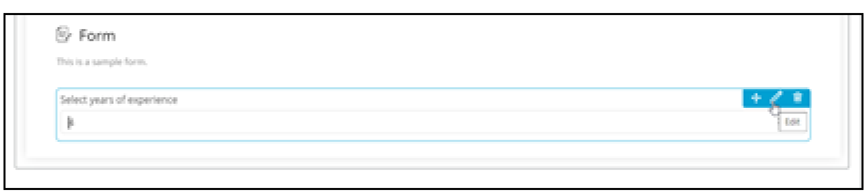
The Select Box field allows you to select one of the predefined options listed in the dropdown.



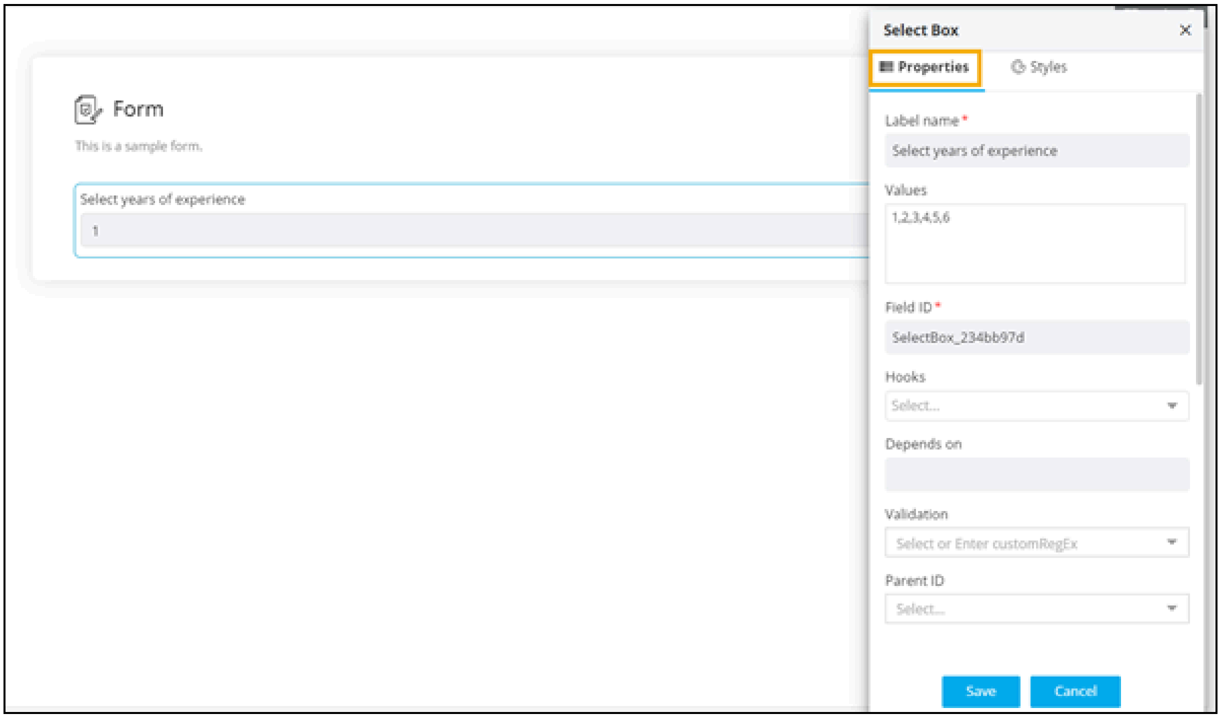
- Configuring the Select Box field - Properties
- Configuring the Select Box field - Styles

Configuring the Select Box field - Properties

1. To add the Select Box field to your form, drag and drop the **Select Box** component to the design area.
2. To configure the Select Box, hover your mouse over the field and click  **(Edit)** icon.



   The**Select Box**pop-up window opens.
3. In the**Select Box** window, under **Properties**, enter or select the required fields.



Field descriptions for Properties

Field	Description
Label name*	Enter a valid **Label name** for the form field. For example, Select Date and Time.
Values	Allows you to define field **Values** relevant to a form field. • Values can be delimited by a comma. • Characters not allowed: ‘&’ and ‘|’.
Field ID*	Allows you to assign a unique **Field ID** to refer to the field name. • Allowed characters: ‘-‘ and ‘_’ • Character Length: Min (1), Max (50).
Hooks	Allows you to assign an appropriate hook for this field. For example, a hook to fetch the current date. • Select a hook that is relevant to the form field. • Hooks can be either Script or REST API. • Hooks are used to fetch details against a specific form field from the device and/or database and map them to the respective field on the user interface.

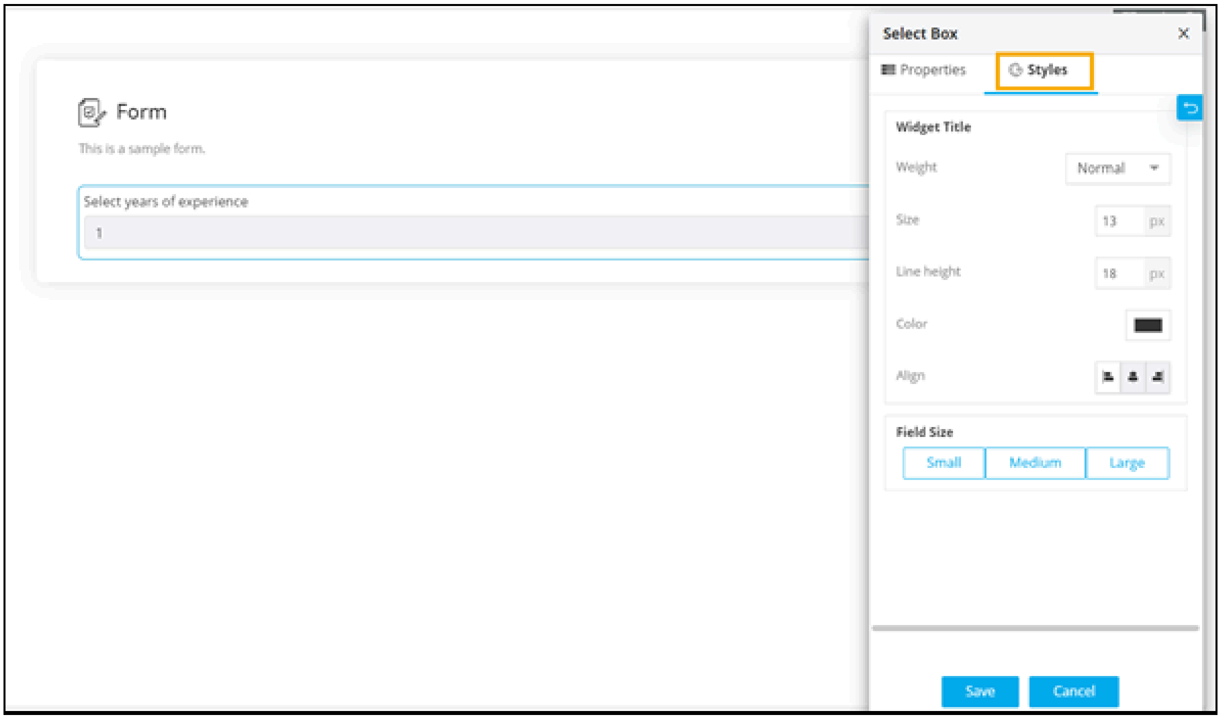
Field	Description
	Note: For more information on Assigning Hooks to Form Fields, click here.
Depends on	Allows you to define dependencies between form fields based on 'Field ID' and 'Values'. • Operators used: '&' and '|' • Max character: 550 • Format: [Field Id1: value1 & Field Id2: value1 & Field Id3: value3], [Field Id1: value1 | Field Id2: value1].
Validation	Allows you to select or define a custom regex pattern against the specific form field. • Select from the predefined regex patterns defined in the regex inventory. • Validation ensures that you capture the right information while creating the request.
Parent ID	Allows you to assign a **Parent ID** for fields in a table. • Select the relevant tabular component (Field ID) against which the form field must be mapped to. **Note:** This is used only when defining a tabular element in the form.
Help	Allows you to add **Help** text/tooltip for the field.
Profile Access	Allows you to select the profile from the dropdown to restrict form field display within a template based on the relevant profile.
Mandatory	Allows you to define this field as **Mandatory** or otherwise. If marked as mandatory, an asterisk (*) is displayed next to the field name in the form.
Read only	Allows you to define this field as **Read Only** or otherwise. If marked as Read Only, a red crossed circle is displayed when you hover your mouse over the field and does not allow you to enter any information.
Global variable	Allows you to define this field as a **Global Variable**.
Create option	Allows you to create option with this field as a **Create option**.

Field	Description
**: Mandatory fields*	

4. Click **Save**.

Configuring the Select Box field - Styles

1. For more options to customize the Select Box, in the **Select Box** window, click **Styles**.



Field descriptions for Styles

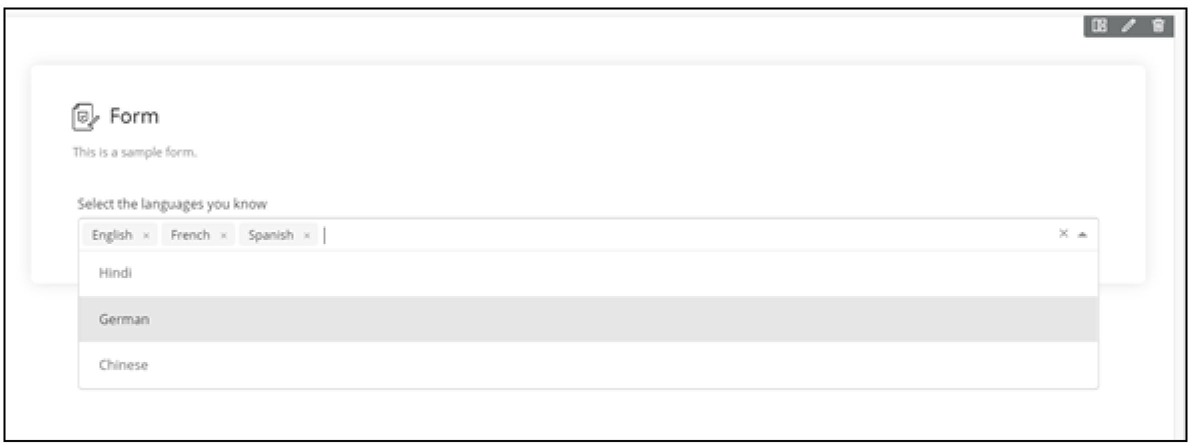
Field	Description
Weight	The **Weight** dropdown list lets you set the font thickness of the field label using one of the following values: • **Default** • **Light** • **Normal** • **Thick**.
Size	Allows you to increase/decrease font size of the field name.

Field	Description
Line Height	Allows you to increase/decrease the line height of the field. This means the space between the field and the field border.
Color	Allows you to define the color of the field name.
Align	Allows you to align the field name to the left, center or right side of the field.
Field Size	Allows you to define the size of the form field as • **Small** • **Medium** • **Large**.

2. To save your settings, click **Save**.
3. To restore the settings under **Styles** to default, click **(Restore)** icon.

Form Field - Multi-select Box

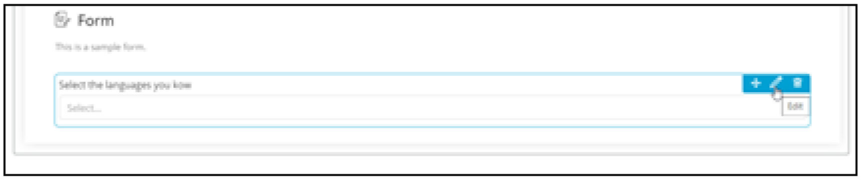
The Multi-select box field allows you to select from a predefined list of multiple options available in a dropdown.



- Configuring the Multi-select Box field - Properties
- Configuring the Multi-select Box field - Styles

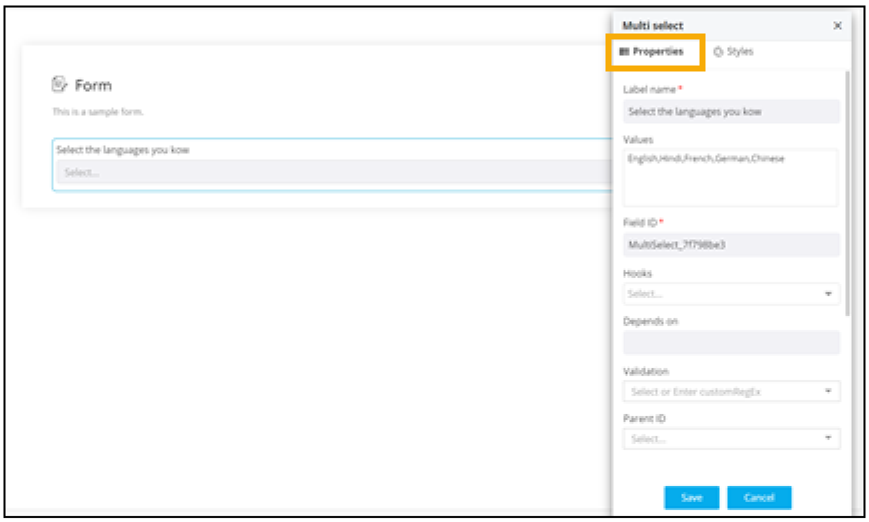
Configuring the Multi-select Box field - Properties

1. To add the Multi-select Box field to your form, drag and drop the **Multi-select Box** component to the design canvas.
2. To configure the Multi-select box, hover your mouse over the field and click **(Edit)** icon.



The **Multi-select** pop-up window opens.

3. In the **Multi-select** window, under **Properties**, enter or select the required fields.



Field descriptions for Properties

Field	Description
Label name	Enter a valid **Label name** for the form field. For example, Select Date and Time.
Values	Allows you to define field **Values** relevant to a form field. • Values can be delimited by a comma. • Characters not allowed: ‘&’ and ‘|’.
Field ID	Allows you to assign a unique **Field ID** to refer to the field name. • Allowed characters: ‘-‘ and ‘_’ • Character Length: Min (1), Max (50).

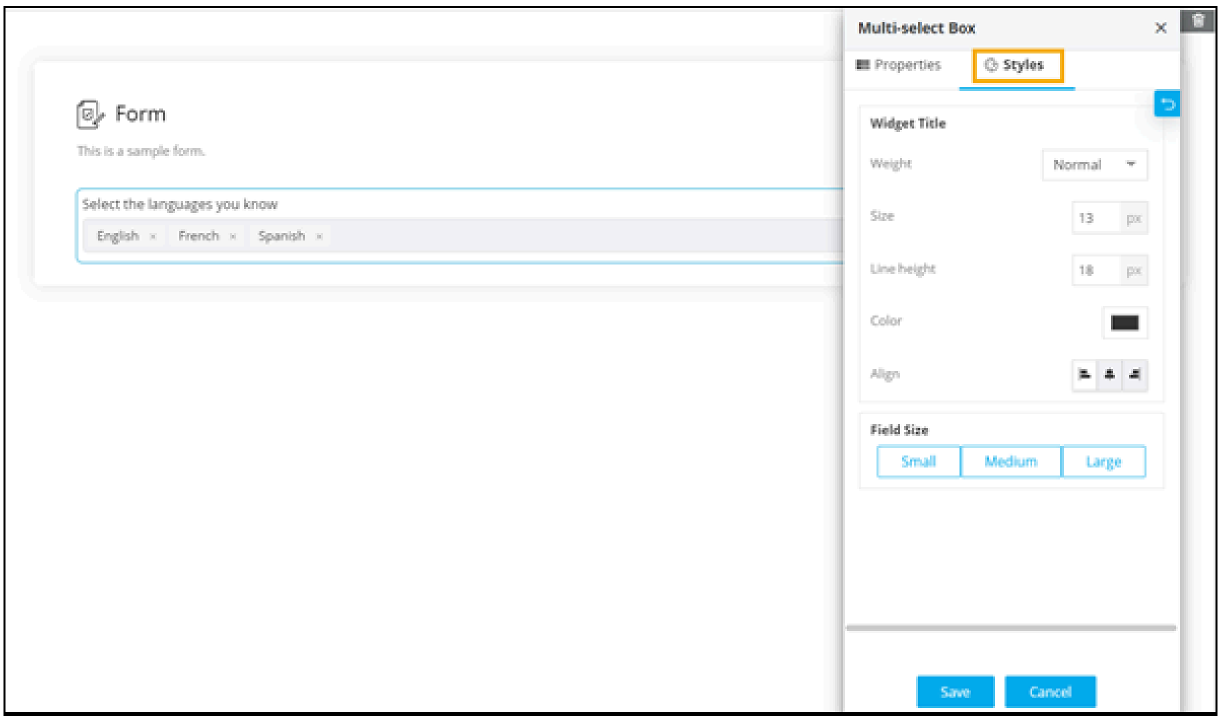
Field	Description
Hooks	Allows you to assign an appropriate hook for this field. For example, a hook to fetch the current date. • Select a hook that is relevant to the form field. • Hooks can be either Script or REST API. • Hooks are used to fetch details against a specific form field from the device and/or database and map them to the respective field on the user interface. **Note:** For more information on Assigning Hooks to Form Fields, click here.
Depends on	Allows you to define dependencies between form fields based on 'Field ID' and 'Values'. • Operators used: '&' and '|' • Max character: 550 • Format: [Field Id1: value1 & Field Id2: value1 & Field Id3: value3], [Field Id1: value1 | Field Id2: value1].
Validation	Allows you to select or define a custom regex pattern against the specific form field. • Select from the predefined regex patterns defined in the regex inventory. • Validation ensures that you capture the right information while creating the request.
Parent ID	Allows you to assign a **Parent ID** for fields in a table. • Select the relevant tabular component (Field ID) against which the form field must be mapped to. **Note:** This is used only when defining a tabular element in the form.
Profile Access	Allows you to select the profile from the dropdown to restrict form field display within a template based on the relevant profile.
Mandatory	Allows you to define this field as **Mandatory** or otherwise. If marked as mandatory, an asterisk (*) is displayed next to the field name in the form.

Field	Description
Read only	Allows you to define this field as **Read Only** or otherwise. If marked as Read Only, a red crossed circle is displayed when you hover your mouse over the read only field and does not allow you to enter any information.
Global variable	Allows you to define this field as a **Global Variable**.
Create option	Allows you to define this field as a **Global variable**.
**: Mandatory fields*	

4. Click **Save**.

Configuring the Multi-select Box field - Styles

1. For more options to customize the Multi-select Box, in the **Multi-select Box**window, click **Styles**.



Field descriptions for Styles

Field	Description
Weight	The **Weight** dropdown list lets you set the font thickness of the field label using one of the following values: • **Default** • **Light** • **Normal** • **Thick**.
Size	Allows you to increase/decrease font size of the field name.
Line Height	Allows you to increase/decrease the line height of the field. This means the space between the field and the field border.
Color	Allows you to define the color of the field name.
Align	Allows you to align the field name to the left, center or right side of the field.
Field Size	Allows you to define the size of the form field as • **Small** • **Medium** • **Large**.

2. To save your settings, click **Save**.
3. To restore the settings under **Styles** to default, click **(Restore)** icon.

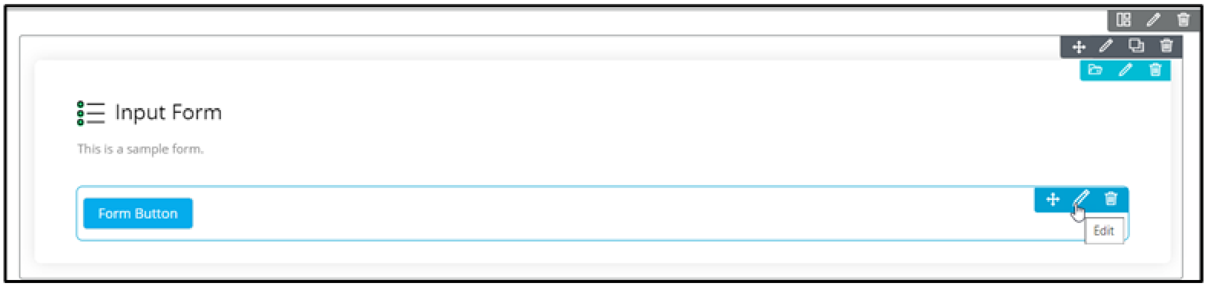
Form Field - Form Button

The Form Button field allows you to add a button to your form to fetch information such as device list, list of certificate authorities and so on.

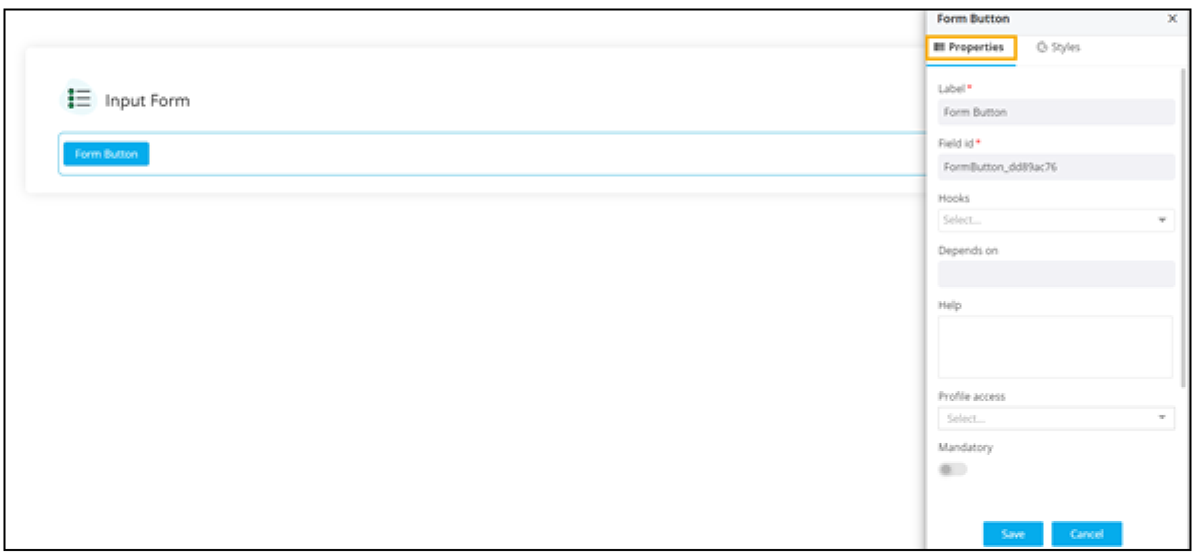
- Configuring the Form Button field - Properties
- Configuring the Form Button field - Styles

Configuring the Form Button field - Properties

1. To add the Form Button field to your form, drag and drop the **Form Button** component to the design area.
2. To configure the Form Button, hover your mouse over the field and click **(Edit)** icon.



   The **Form Button** pop-up window opens.
3. In the **Form Button** window, under **Properties**, enter or select the required fields.



Field descriptions for Properties

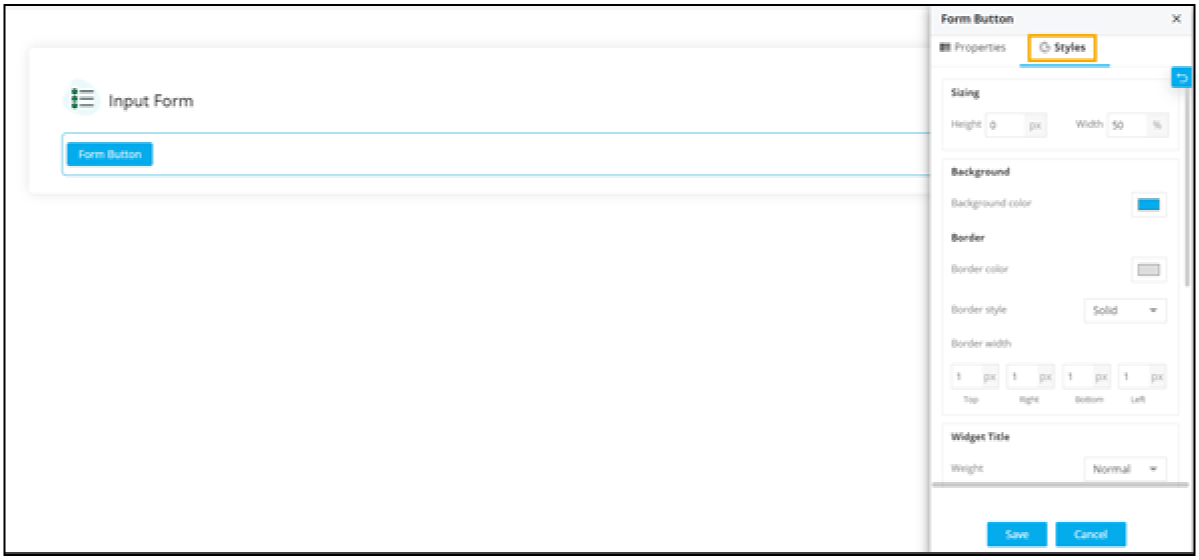
Field	Description
Label	Enter a valid **Label name** for the form field. For example, Select Date and Time.

Field	Description
Field ID	Allows you to assign a unique **Field ID** to refer to the field name. • Allowed characters: '-' and '_' • Character Length: Min (1), Max (50).
Help	Allows you to add **Help** text/tooltip for the field.
Hooks	Allows you to assign an appropriate hook for this field. For example, a hook to fetch the current date. • Select a hook that is relevant to the form field. • Hooks can be either Script or REST API. • Hooks are used to fetch details against a specific form field from the device and/ or database and map them to the respective field on the user interface. **Note:** For more information on Assigning Hooks to Form Fields, click here.
Depends on	Allows you to define dependencies between form fields based on 'Field ID' and 'Values'. • Operators used: '&' and '|' • Max character: 550 • Format: [Field Id1: value1 & Field Id2: value1 & Field Id3: value3], [Field Id1: value1 | Field Id2: value1].
Profile Access	Allows you to select the profile from the dropdown to restrict form field display within a template based on the relevant profile.
Mandatory	Allows you to define this field as **Mandatory** or otherwise. If marked as mandatory, an asterisk (*) is displayed next to the field name in the form.
Read only	Allows you to define this field as **Read Only** or otherwise. If marked as Read Only, a red crossed circle is displayed when you hover your mouse over the read only field and does not allow you to enter any information.
Hidden	Allows you to hide the field in the form.
Select icons	Allows you to select an icon for the Form Button from the available options.

4. Click **Save**.

Configuring the Form Button field - Styles

1. For more options to customize the Form Button, in the **Form Button** window, click **Styles**.



Field descriptions for Styles

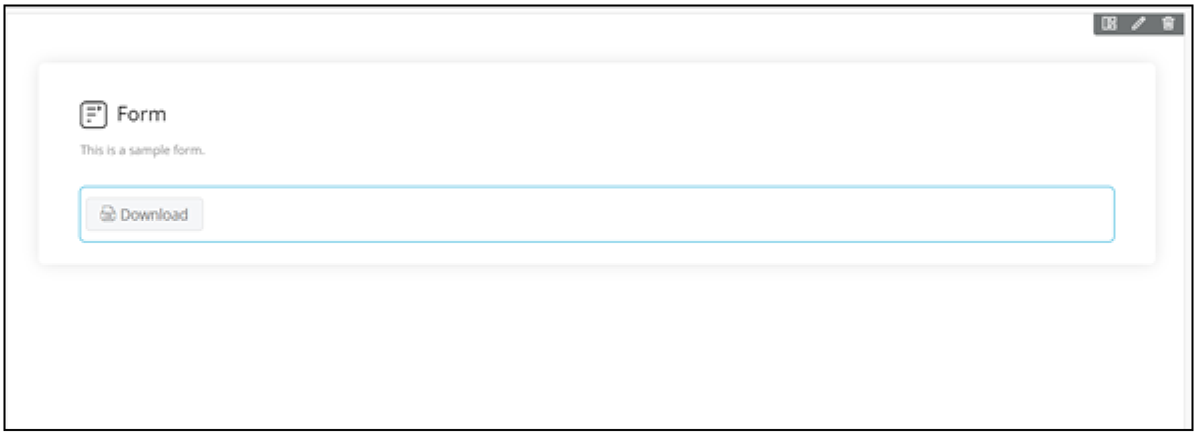
Field	Description
Sizing	Allows you to increase/decrease the height and width of the form field.
Background	Allows you to select the color of the form button.
Border	Allows you to define the color, style, and width of the button's border.
Weight	The **Weight** dropdown list lets you set the font thickness of the field label using one of the following values: • **Default** • **Light** • **Normal** • **Thick**.
Size	Allows you to increase/decrease font size of the field name.
Line Height	Allows you to increase/decrease the line height of the field. This means the space between the field and the field border.
Color	Allows you to define the color of the field name.
Align	Allows you to align the field name to the left, center or right side of the field.

Field	Description
Field Size	Allows you to define the size of the form field as • **Small** • **Medium** • **Large**.

2. To save your settings, click **Save**.
3. To restore the settings under **Styles** to default, click **(Restore)** icon.

Form Field - Download

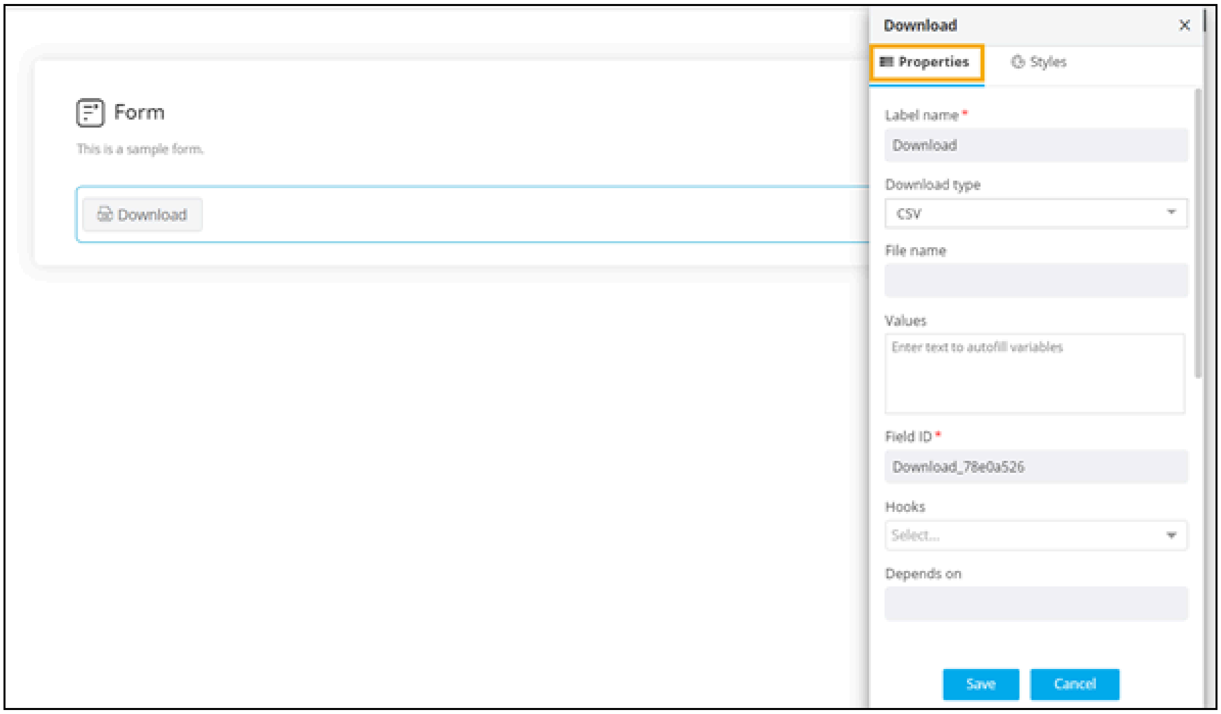
The Download field serves as a URL/button which you can click to download files(s).



- Configuring the Download field - Properties
- Configuring the Download field - Styles

Configuring the Download field - Properties

1. To add the Download field to your form, drag and drop the **Download** component to the design canvas.
2. To configure the **Download** widget, hover your mouse over the widget and click **(Edit)** icon. The **Download** pop-up window opens.
3. In the **Download** window, under **Properties**, enter or select the required fields.



Field descriptions for Properties

Field	Description
Label name	Enter a valid **Label name** for the form field. For example, Download CSV.
Download Type	Select the **Download type** from the available options: • URL • File • CSV.
File Name	Enter the file name with which the file will be downloaded to your machine.
Values	Allows you to define field **Values** relevant to a form field. • Values can be delimited by a comma. • Characters not allowed: '&' and '|'.
Field ID	Allows you to assign a unique **Field ID** to refer to the field name. • Allowed characters: '-' and '_' • Character Length: Min (1), Max (50).

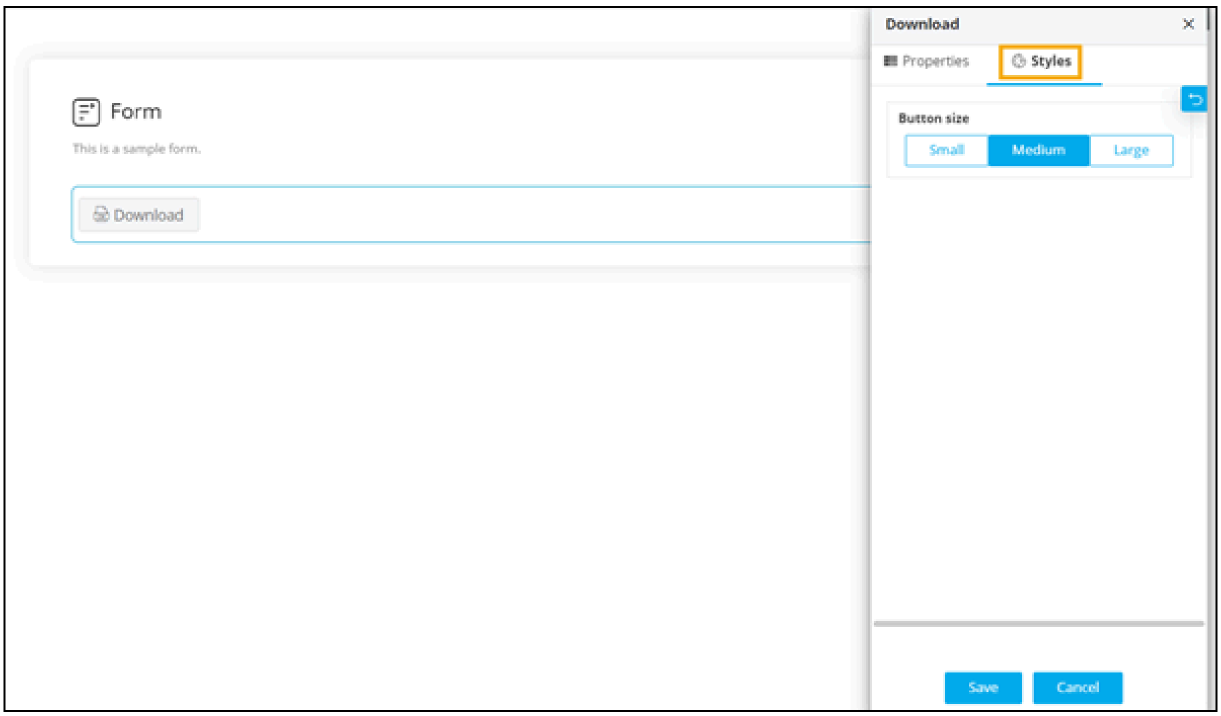
Field	Description
Hooks	Allows you to assign an appropriate hook for this field. For example, a hook to fetch the current date. • Select a hook that is relevant to the form field. • Hooks can be either Script or REST API. • Hooks are used to fetch details against a specific form field from the device and/or database and map them to the respective field on the user interface. **Note:** For more information on Assigning Hooks to Form Fields, click here.
Depends on	Allows you to define dependencies between form fields based on 'Field ID' and 'Values'. • Operators used: '&' and '|' • Max character: 550 • Format: [Field Id1: value1 & Field Id2: value1 & Field Id3: value3], [Field Id1: value1 | Field Id2: value1].
Validation	Allows you to select or define a custom regex pattern against the specific form field. • Select from the predefined regex patterns defined in the regex inventory. • Validation ensures that you capture the right information while creating the request.
Parent ID	Allows you to assign a **Parent ID** for fields in a table. • Select the relevant tabular component (Field ID) against which the form field must be mapped to. **Note:** This is used only when defining a tabular element in the form.
Help	Allows you to add **Help** text/tooltip for the field.
Mandatory	Allows you to define this field as **Mandatory** or otherwise. If marked as mandatory, an asterisk (*) is displayed next to the field name in the form.

Field	Description
Read only	Allows you to define this field as **Read Only** or otherwise. If marked as Read Only, a red crossed circle is displayed when you hover your mouse over the read only field and does not allow you to enter any information.
Global variable	Allows you to define this field as a **Global Variable**.

4. Click **Save**.
5. To download your file, click on the download link or button.

Configuring the Download field - Styles

1. For more options to customize the Download, in the **Download** window, click **Styles**.



2. Under **Button Size**, select the size of the Download button as **Small**, **Medium** or **Large** as required.
3. To save your settings, click **Save**.
4. To restore the settings under **Styles** to default, click  **(Restore)** icon.

Form Field - File Upload

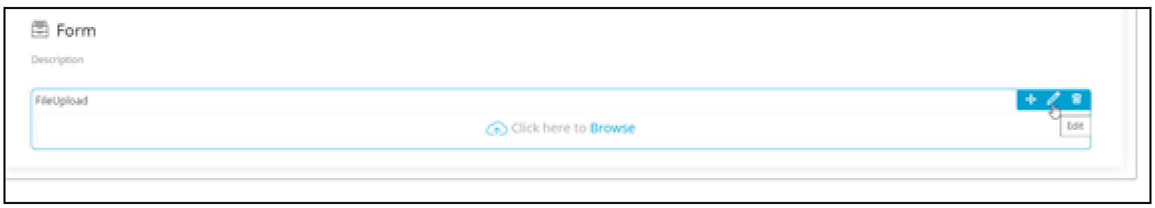
The File Upload field allows you to attach files from your machine to the form.

• Configuring the File Upload field - Properties

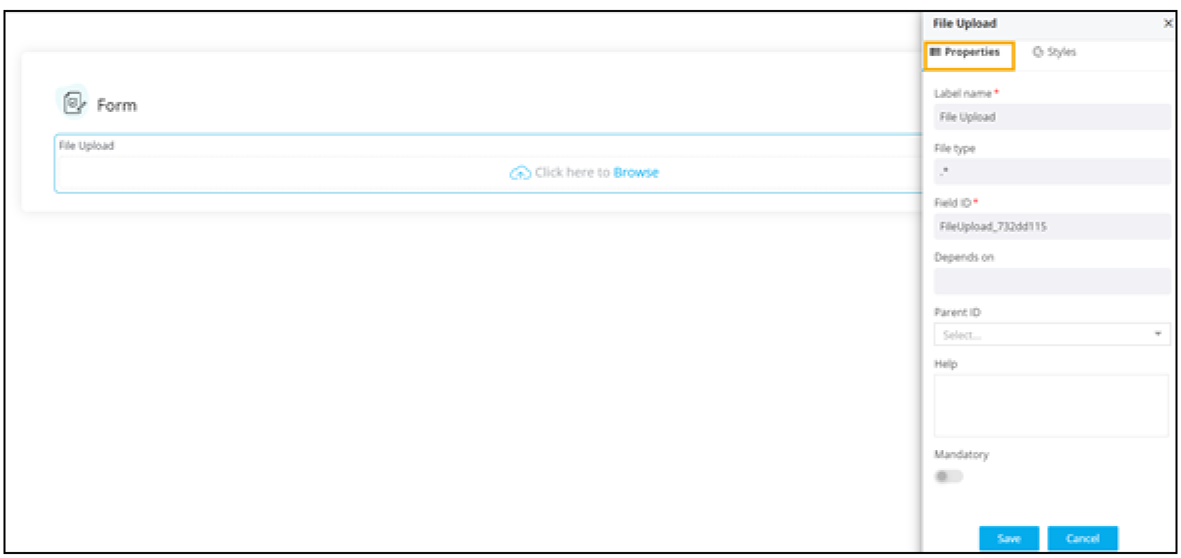
• Configuring the File Upload field - Styles

Configuring the File Upload field - Properties

1. To add the File Upload field to your form, drag and drop the **File Upload** component to the design canvas.
2. To configure the **File Upload** widget, hover your mouse over the widget and click **(Edit)** icon. The **File Upload** pop-up window opens.



3. In the **File Upload** window, under **Properties**, enter or select the required fields.



Field descriptions for Properties

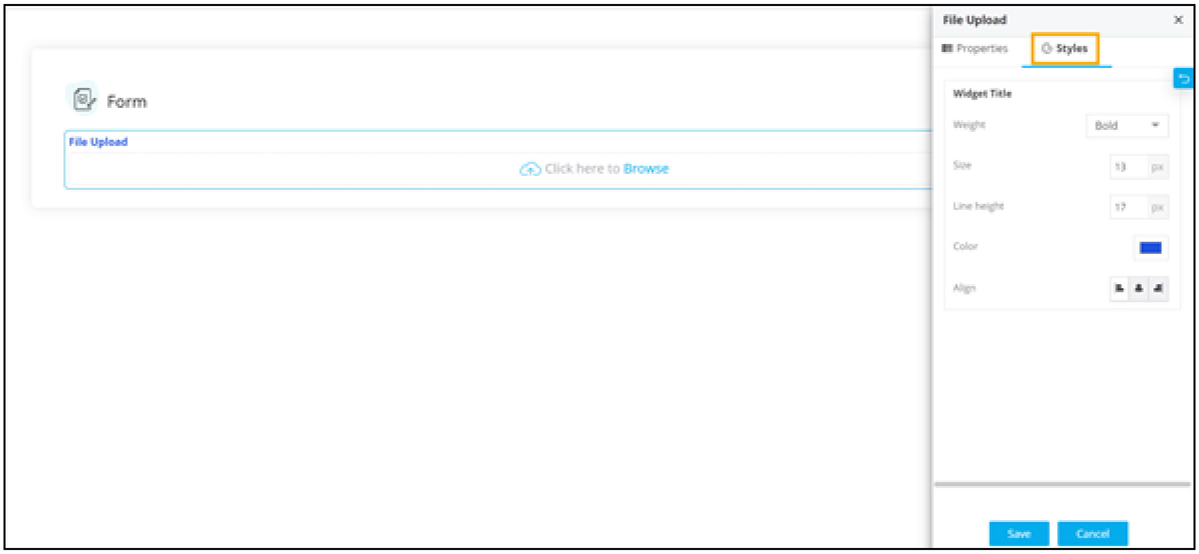
Field	Description
Label name	Enter a valid **Label name** for the form field. For example, Chose File.
File Type	Allows you to define the type of file(s) that can be uploaded. For example, if you enter .* as the file type, it means any file type can be uploaded to this field. Alternatively, you can define the file types as .png, .svg and so on.
***Field ID**	Allows you to assign a unique **Field ID** to refer to the field name. • Allowed characters: '-' and '_' • Character Length: Min (1), Max (50).
Global variable	Allows you to define this field as a **Global Variable**.
Depends on	Allows you to define dependencies between form fields based on 'Field ID' and 'Values'. • Operators used: '&' and '|' • Max character: 550 • Format: [Field Id1: value1 & Field Id2: value1 & Field Id3: value3], [Field Id1: value1 | Field Id2: value1].
Parent ID	Allows you to assign a **Parent ID** for fields in a table. • Select the relevant tabular component (Field ID) against which the form field must be mapped. **Note:** This is used only when defining a tabular element in the form.
Help	Allows you to add **Help** text/tooltip for the field.
Mandatory	Allows you to define this field as **Mandatory** or otherwise. If marked as mandatory, an asterisk (*) is displayed next to the field name in the form.
Read only	Allows you to define this field as **Read Only** or otherwise. If marked as Read Only, a red crossed circle is displayed when you hover your mouse over the read only field and does not allow you to enter any information.
**: Mandatory fields*	

4. Click **Save**.

5. To browse for files to be uploaded from your machine to the page, click **(Upload)** icon.

Configuring the File Upload field - Styles

1. For more options to customize the File Upload, in the **File Upload** window, click **Styles**.



Field descriptions for Styles

Field	Description
Weight	The **Weight** dropdown list lets you set the font thickness of the field label using one of the following values: • **Default** • **Light** • **Normal** • **Thick**.
Size	Allows you to increase/decrease font size of the field name.
Line Height	Allows you to increase/decrease the line height of the field. This means the space between the field and the field border.
Color	Allows you to define the color of the field name.
Align	Allows you to align the field name to the left, center or right side of the field.

2. To save your settings, click **Save**.
3. To restore the settings under **Styles** to default, click **(Restore)** icon.

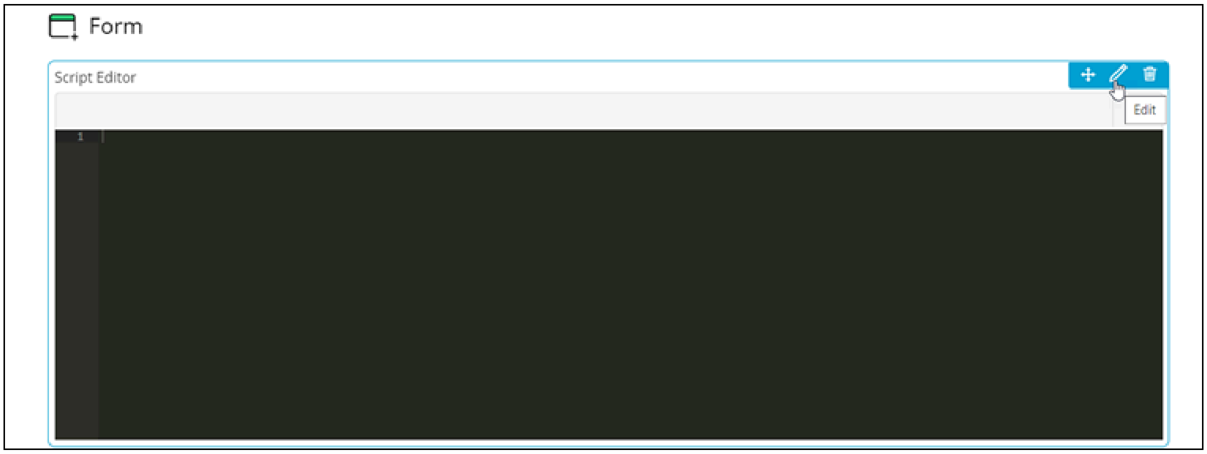
Form Field - Script Editor

The Script Editor field allows you to add a code editor field to the form on your page.

- Configuring the Script Editor- Properties
- Configuring the Script Editor- Styles

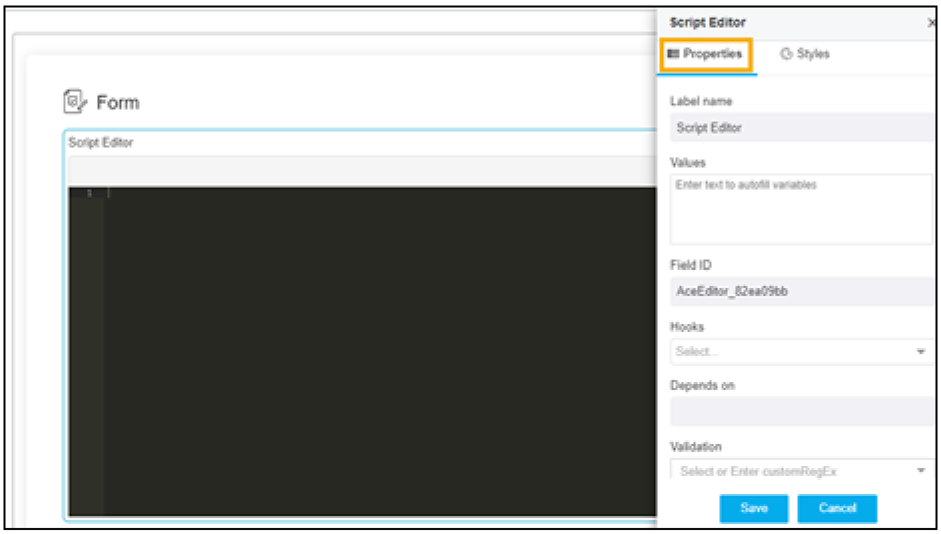
Configuring the Script Editor- Properties

1. To add the Script Editor widget, drag and drop the **Script Editor** component to any blue highlighted area in the design area.
2. To configure the **Script Editor**, hover your mouse over the field and click **(Edit)** icon.



The **Script Editor**pop-up window opens.

3. In the **Script Editor** window, under **Properties**, enter or select the required fields.



Field descriptions for Properties

Field	Description
Label name*	Enter a valid **Label name** for the form field. For example, Select Date and Time.
Values	Allows you to define field **Values** relevant to a form field. • Values can be delimited by a comma. • Characters not allowed: '&' and '|'.
Field ID*	Allows you to assign a unique **Field ID** to refer to the field name. • Allowed characters: '-' and '_' • Character Length: Min (1), Max (50).

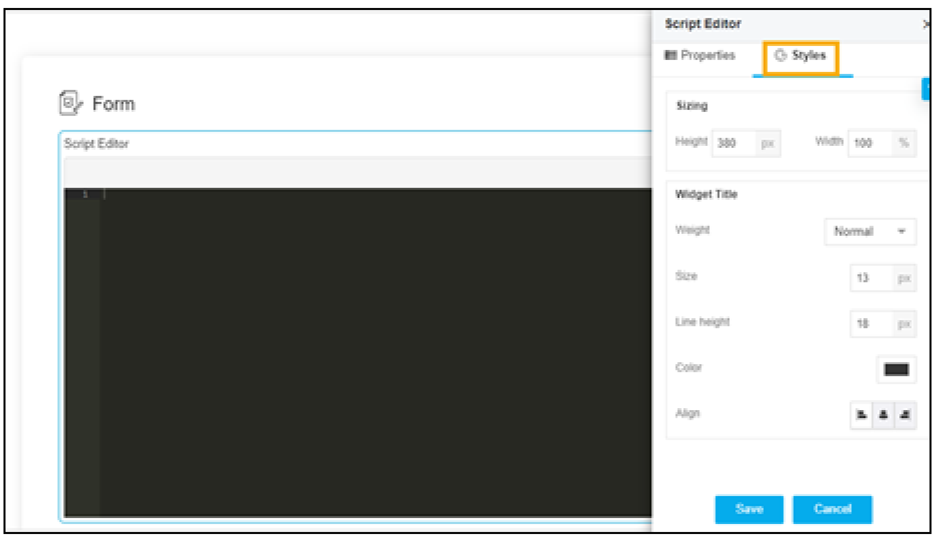
Field	Description
Hooks	Allows you to assign an appropriate hook for this field. For example, a hook to fetch the current date. • Select a hook that is relevant to the form field. • Hooks can be either Script or REST API. • Hooks are used to fetch details against a specific form field from the device and/or database and map them to the respective field on the user interface. **Note:** For more information on Assigning Hooks to Form Fields, click here.
Depends on	Allows you to define dependencies between form fields based on 'Field ID' and 'Values'. • Operators used: '&' and '|' • Max character: 550 • Format: [Field Id1: value1 & Field Id2: value1 & Field Id3: value3], [Field Id1: value1 | Field Id2: value1].
Validation	Allows you to select or define a custom regex pattern against the specific form field. • Select from the predefined regex patterns defined in the regex inventory. • Validation ensures that you capture the right information while creating the request.
Parent ID	Allows you to assign a **Parent ID** for fields in a table. • Select the relevant tabular component (Field ID) against which the form field must be mapped. **Note:** This is used only when defining a tabular element in the form.
Help	Allows you to add **Help** text/tooltip for the field.
Profile Access	Allows you to select the profile from the dropdown to restrict form field display within a template based on the relevant profile.
Mandatory	Allows you to define this field as **Mandatory** or otherwise. If marked as mandatory, an asterisk (*) is displayed next to the field name in the form.
Read only	Allows you to define this field as **Read Only** or otherwise. If marked as Read Only, a red crossed circle is displayed when you hover your mouse over the read only field and does not allow you to enter any information.

Field	Description
Global variable	Allows you to define this field as a **Global Variable**.
Hidden	Allows you to hide the field in the form.
File Upload	Allows you to add an option for uploading files to the ace editor field.
Theme	Allows you to select between a light or dark theme for the ace editor.
Mode	Allows you to select the script from the drop down menu.
**: Mandatory fields*	

4. Click **Save**.

Configuring the Script Editor- Styles

1. For more options to customize the Script Editor, in the **Script Editor**window, click **Styles**.



Field descriptions for Styles

Field	Description
Sizing	Allows you to adjust the height and width of the Ace Editor field in the form.

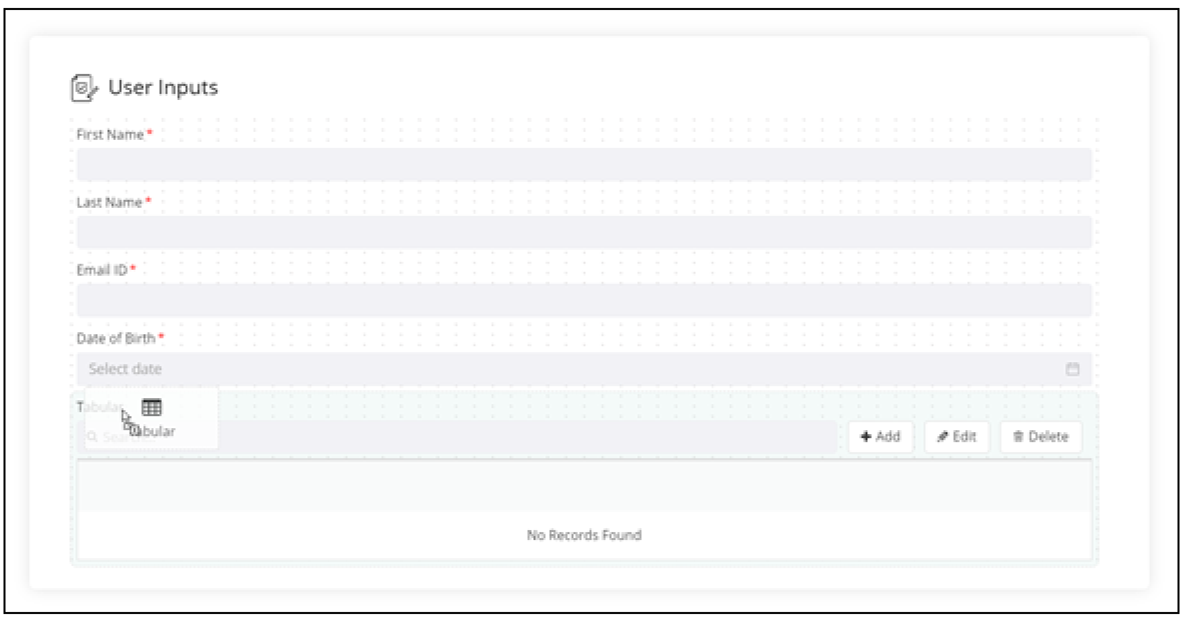
Field	Description
Widget Title	The following options are available here: • **Weight** - Allows you to set the font thickness of the widget title as **Default, Light, Normal, or Thick**. • **Size** - Allows you to increase/decrease the size of the widget title. • **Line height** - Allows you to adjust the line height. • **Color** - Allows you to set a color for the widget title. • **Align** - Allows you to align the widget title to the left, right, or centre.

2. To save your settings, click **Save**.
3. To restore the settings under **Styles** to default, click **(Restore)** icon.

Form Field - Tabular

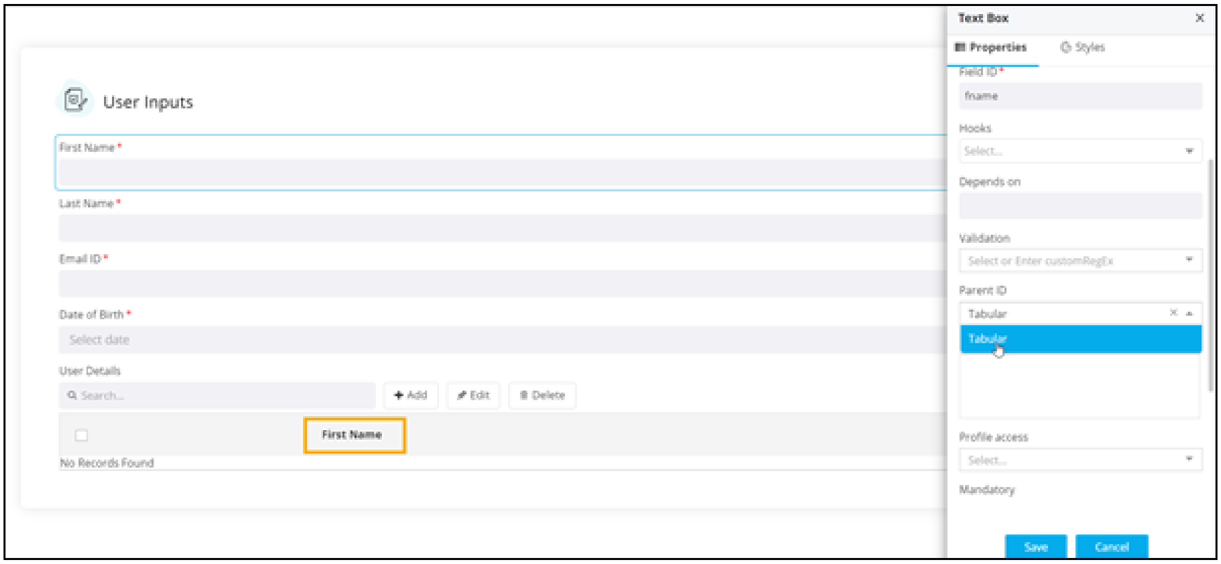
You can add a Tabular field to the form on your page. This field allows you to configure a table within your form by associating the field ID of the tabular as the Parent ID for the form fields required in the table.

1. To add a Tabular element to the form first design a form by adding form fields to the design canvas.
2. Drag and drop the **Tabular** field in the form.



3. Hover your mouse over a field, for example **First Name**, and click **(Edit)** icon.
   The **Text Box** pop-up window opens with the **Properties** tab open by default.
4. In the **Text Box** pop-up window, under **Properties**, select the **Parent ID** as Tabular.

The field, **First Name**, is added to the Tabular.



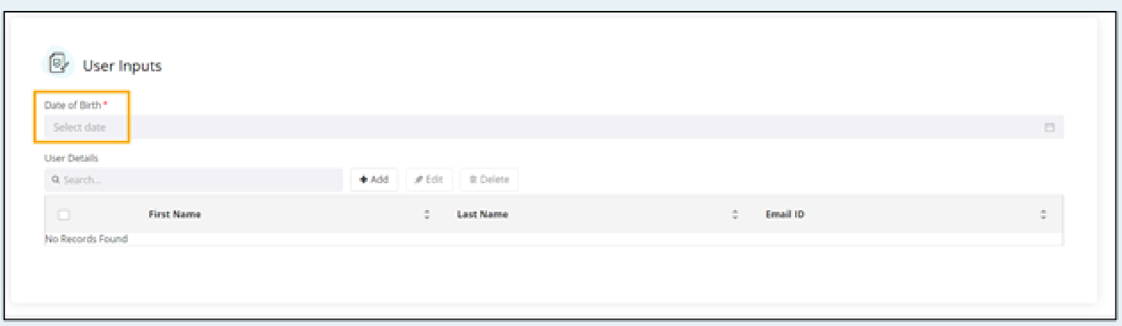
5. Associate the Tabular's field ID as the **Parent ID** for other fields that are to be added to the form.

> **Note:** When the Tabular field within a Form Group only the form fields present within that group can be added to that group's Tabular.

6. Click **Preview**.

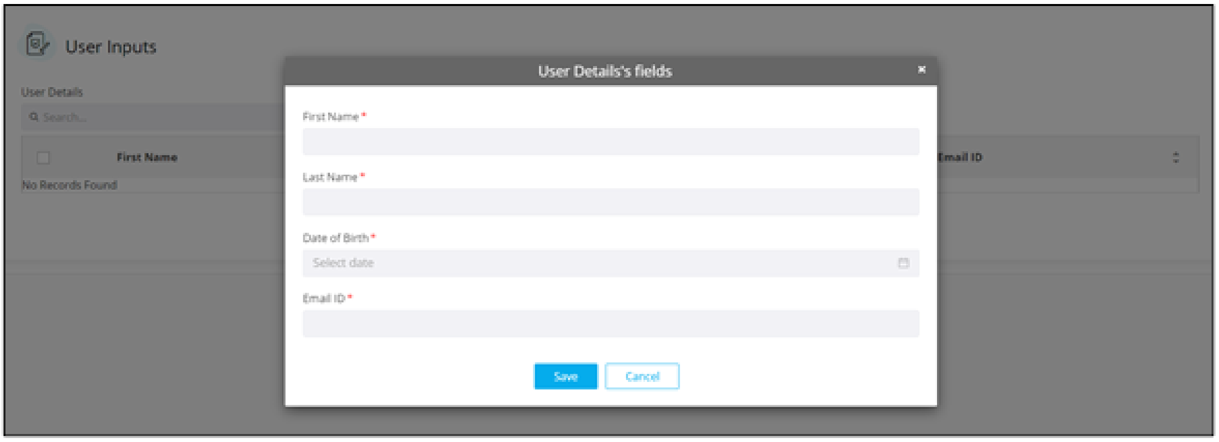
   The Tabular opens in a new tab.

> **Note:** Any form fields not associated with the Tabular field will be displayed outside the Tabular.



7. To add data in the Tabular component, click Add.

   The **User Details's fields** window opens with the fields of the tabular displayed.



8. In the **User Details’s fields** window, enter the values for the form fields.
9. Click **Save**.

   The values are added to the Tabular.

> **Note:** The **User Details’s fields** window remains open for the user to add more fields.

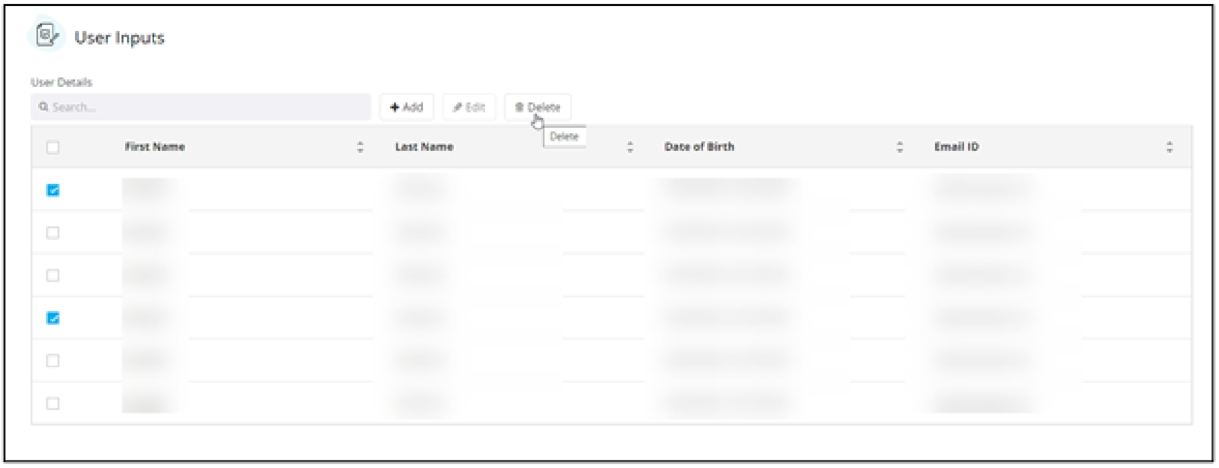
10. To close the **Tabular’s fields** window, click **Cancel**.
11. To edit a row in the Tabular, select the row to be edited.

    The **Edit** button is enabled.
12. Click **Edit**.

> **Note:** The **Edit** button is enabled only when you select exactly one row in the tabular. Selecting zero or more than 1 row will disable the **Edit** button.

13. In the **User Details’s fields** window, modify the form field(s).
14. Click **Update**.

    The Tabular’s fields window closes.
15. To delete a row in the Tabular, select the row to be deleted.
16. Click **Delete**.

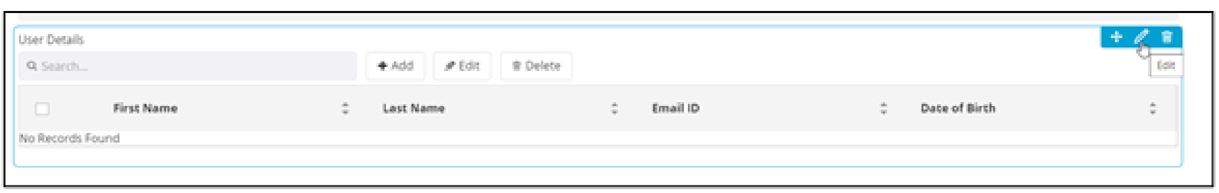


Note: The **Delete** button is enabled only when you select a single or multiple rows.

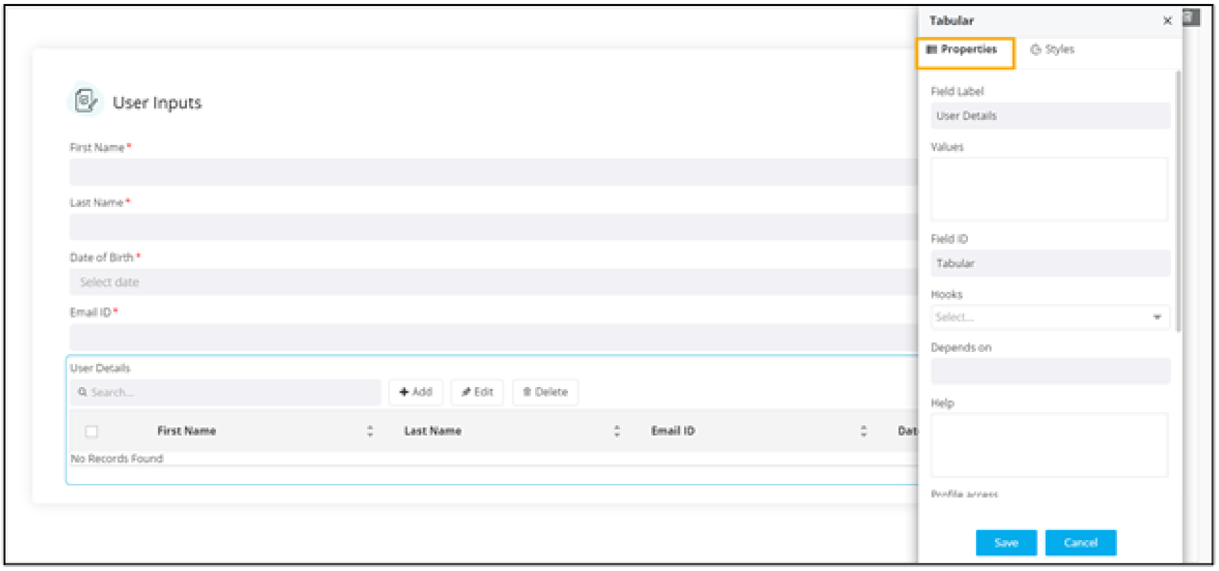
- Configuring the Tabular - Properties
- Tabular - Grid Universal Controls
- Configuring the Tabular with Hooks
- Configuring the Tabular - Styles

Configuring the Tabular - Properties

1. To configure the **Tabular**, hover your mouse over the field and click **(Edit)** icon.



   The **Tabular** pop-up window opens.
2. In the **Tabular** window, under **Properties**, enter or select the required fields.



Field descriptions for Properties

Field	Description
Label name*	Enter a valid **Label name** for the form field. For example, Select Date and Time.
Values	Allows you to define field **Values** relevant to a form field. • Values can be delimited by a comma. • Characters not allowed: '&' and '|'.
Field ID*	Allows you to assign a unique **Field ID** to refer to the field name. • Allowed characters: '-' and '_' • Character Length: Min (1), Max (50).
Hooks	Allows you to assign an appropriate hook for this field. For example, a hook to fetch the current date. • Select a hook that is relevant to the form field. • Hooks can be either Script or REST API. • Hooks are used to fetch details against a specific form field from the device and/or database and map them to the respective field on the user interface. **Note:** For more information on Assigning Hooks to Form Fields, click here.

Field	Description
Depends on	Allows you to define dependencies between form fields based on 'Field ID' and 'Values'. • Operators used: '&' and '|' • Max character: 550 • Format: [Field Id1: value1 & Field Id2: value1 & Field Id3: value3], [Field Id1: value1 | Field Id2: value1].
Help	Allows you to add **Help** text/tooltip for the field.
Profile Access	Allows you to select the profile from the dropdown to restrict form field display within a template based on the relevant profile.
Mandatory	Allows you to define this field as **Mandatory** or otherwise. If marked as mandatory, an asterisk (*) is displayed next to the field name in the form.
Grid Universal Controls	Allows you enable/disable the following features in the Tabular: • **Search** • **Pagination** • **Actions**. **Note:** For more information con **Grid Universal Controls**, click here.
**: Mandatory fields*	

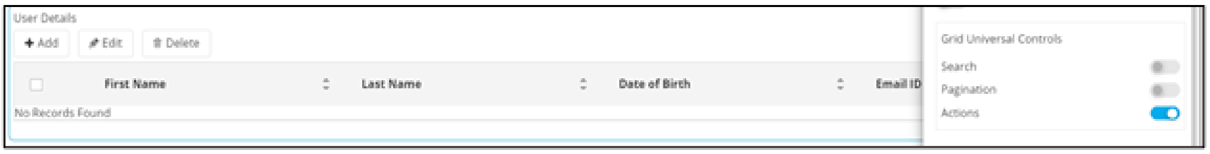
3. Click **Save**.

Tabular - Grid Universal Controls

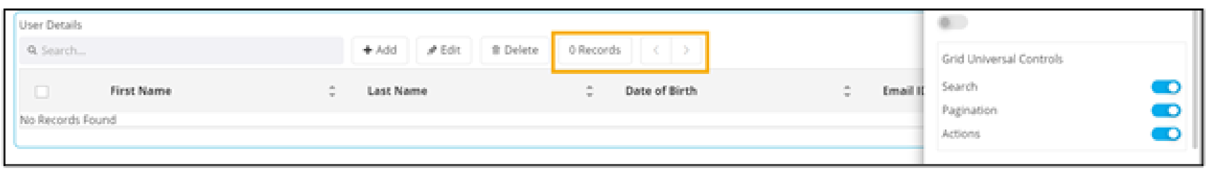
This option in the Tabular allows you to enable/disable the following features:

- **Search** - You can turn off the toggle for Search if you don’t want the Search bar to be displayed in the Tabular. Enabling the Search bar allows you to search for a particular item in the Tabular quickly.
  - Search toggle is on and the search bar is displayed.

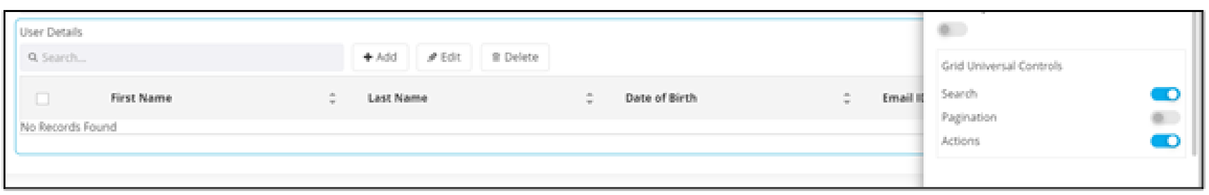
  - Search toggle is off and the search bar is not displayed.



- **Pagination** - You can turn on the toggle for Pagination to display the number of records that can be viewed on a page. Enabling Pagination allows a page with high volumes of data to load faster.
  - Pagination toggle is on and the record view is displayed.

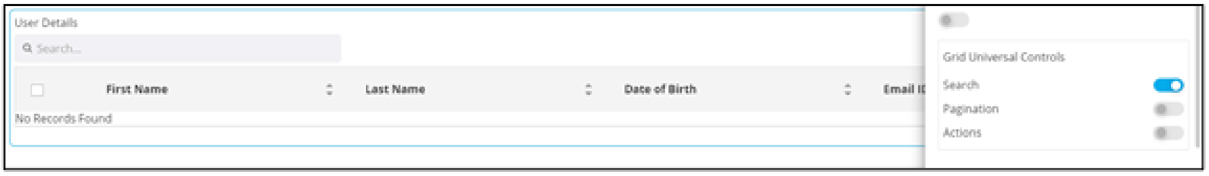


  - Pagination toggle is off and the record view is not displayed.



- **Actions** - You can turn off the Actions toggle if you don’t want the Action buttons to be displayed in the Tabular. Enabling action buttons such as Add, Edit, and Delete allows you to perform these actions and modify the Tabular quickly.
  - Actions toggle is on and the action buttons are displayed.

- Actions toggle is off and the action buttons are not displayed.



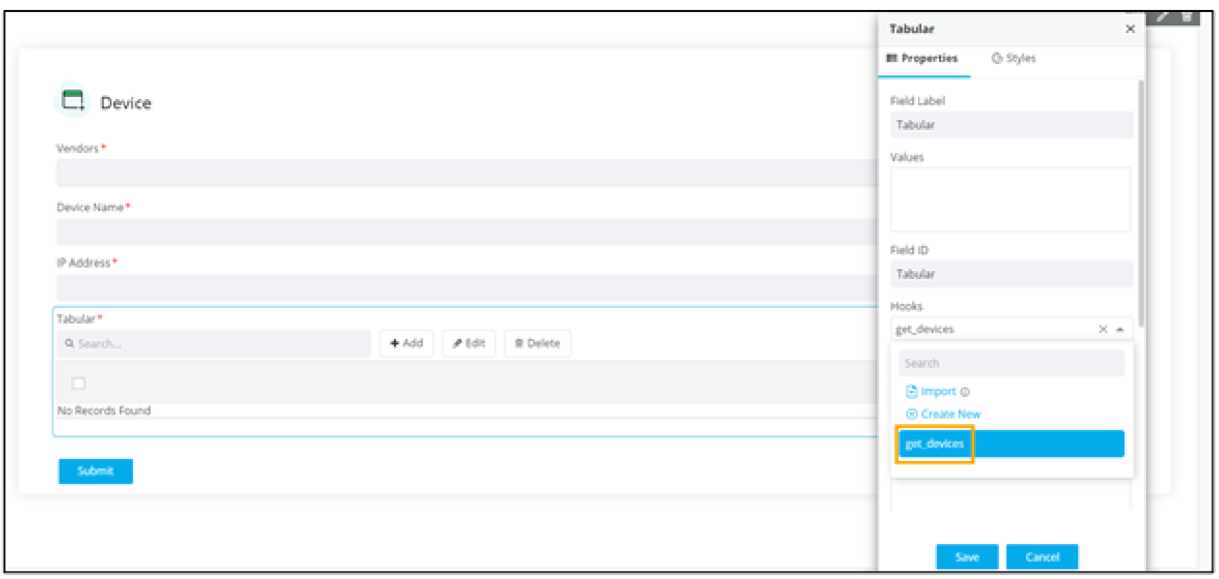
Configuring the Tabular with Hooks

1. Design a page.
2. From the **Build** > **Form** section, drag and drop the following form fields to design a form:

Field Type	Label Name	Field Values	Parent ID
Text Box	Vendors	A10,F5,Citrix	
Text Box	Device Name	-	Tabular
Text Box	IP Address	-	Tabular

3. Drag and drop a Tabular field to the form.
4. To configure the **Tabular**, hover your mouse over the field and click **(Edit)** icon.

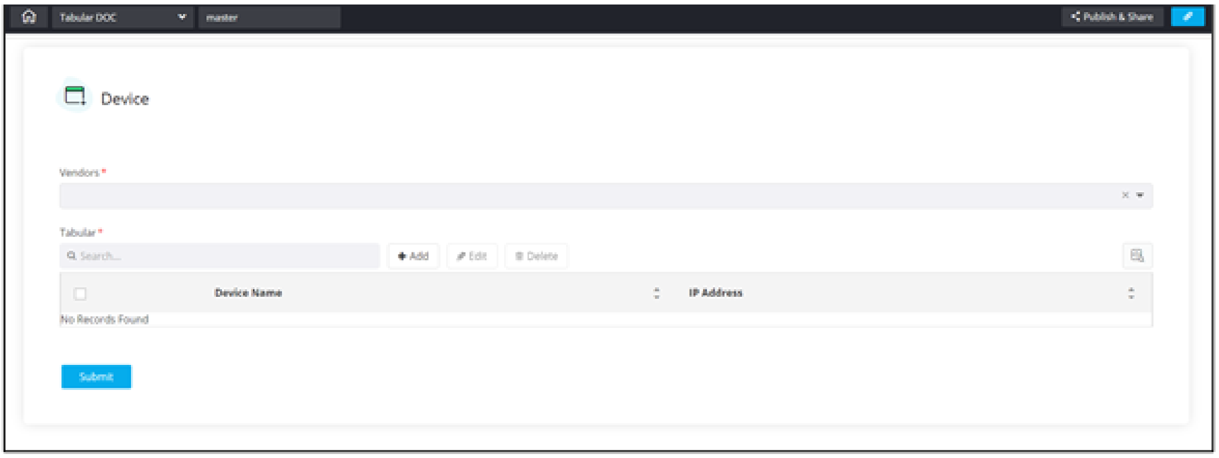
   The **Tabular** pop-up window opens with the **Properties** tab open by default.
5. In the **Tabular** window, under **Properties**, select an appropriate hook for getting the device list.



Note: You can also create a hook that meets your requirements. For more information on assigning hooks to form fields, click here.

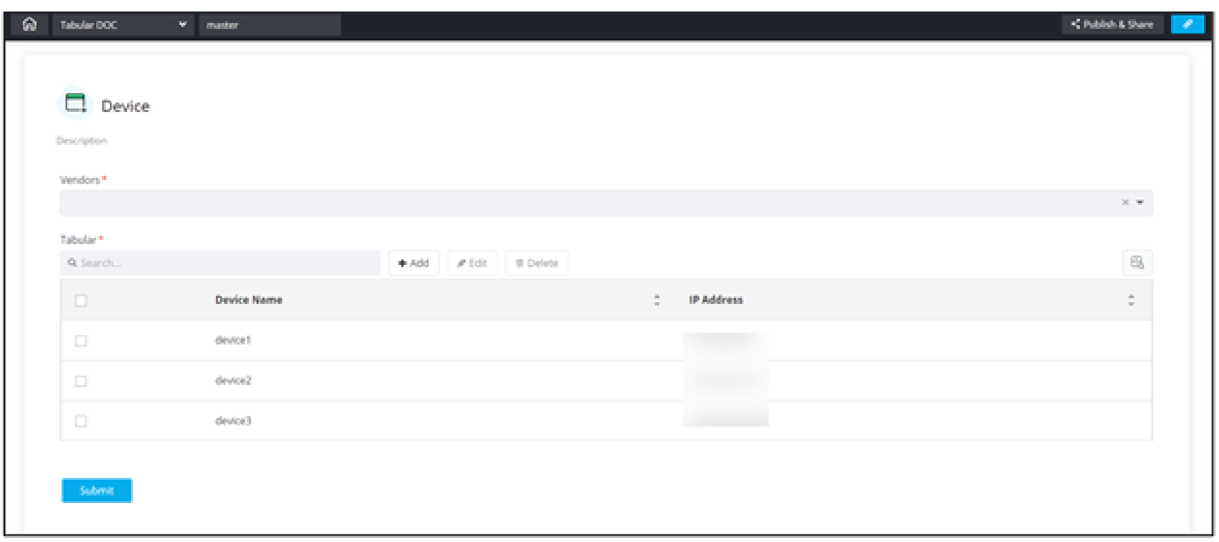
6. Click **Save**.
7. **Preview** the form.

   The form opens in a new tab.



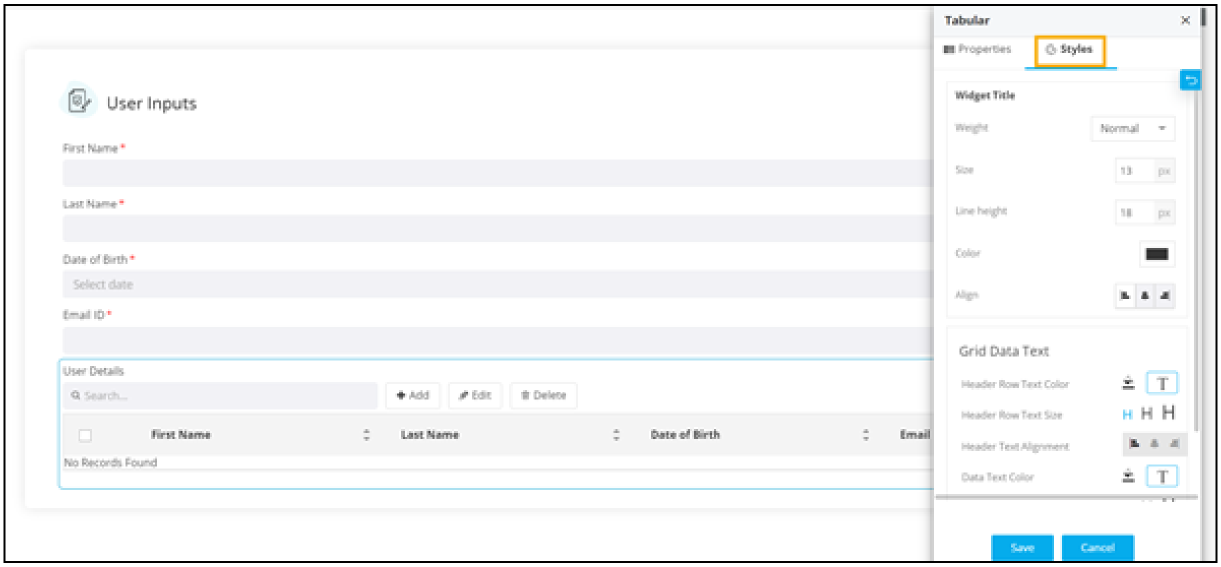
8. In the **Vendors** field, select an option from the dropdown.
9. To trigger the hook and get the device list based on vendor and auto-populate the form, click  above the Tabular field.

   The Tabular is populated with the device list fetched from the hook.



Configuring the Tabular - Styles

1. For more options to customize the Tabular, in the **Tabular** window, click **Styles**.



Field descriptions for Styles

Field	Description
Weight Title	Allows you to define the following features in the Widget Title: • **Weight** - The **Weight** dropdown list lets you set the font thickness of the field label. • **Size** - Allows you to increase/decrease font **Size** of the field name. • **Line height** - Allows you to increase/decrease the **Line height** of the field. This means the space between the field and the field border. • **Color** - Allows you to define the **Color** of the field label. • **Align** - Allows you to **Align** the field label to the left, center or right side of the field.
Grid Data Text	Allows you to define the following features for header text: • **Header Row Text Color** - Allows you to select the color for the text in the Header row from the color picker or the color palette. • **Header Row Text Size** - Allows you to set the size of the text in the Header row. • **Header text Alignment** - Allows you to align the text in the Header row to the left, center or right side. • **Data Text Color** - Allows you to select the color for the data text from the color picker or the color palette. • **Data Text Size** - Allows you to set the size of the data text in the Tabular. • **Data Text Alignment** - Allows you to align the data text to the left, center or right side.

2. To save your settings, click **Save**.
3. To restore the settings under **Styles** to default, click **(Restore)** icon.

Form Field - Rich Text Editor

The Rich Text Editor field allows you to add a text editor to your page with the following features:

- Provision to define the font type and size, add font effects, highlight text in color, and add bulleted and numbered points.
- Provision to enable monospace.
- Provision to insert superscript and subscript.
- Provision to add links to external/internal pages.
- Provision to embed an internal/external page within the text editor.
- Provision to indent and align the text as per requirement.
- Provision to upload files to the page.

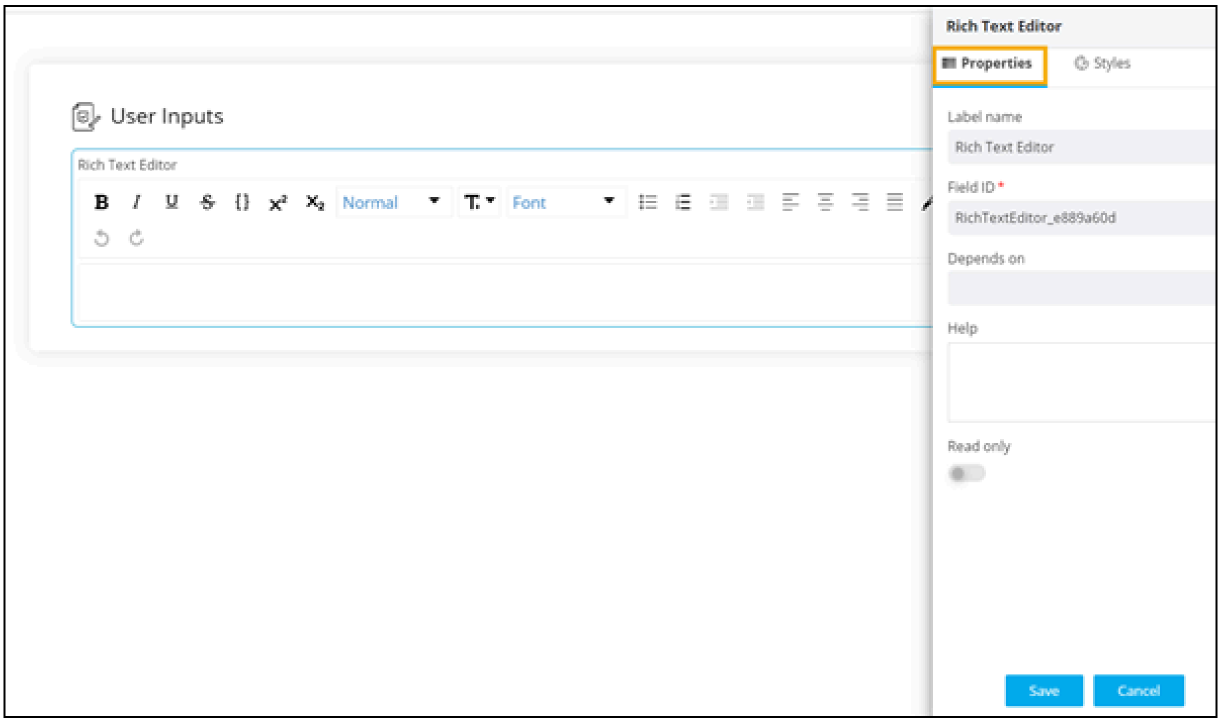
- Configuring the Rich Text Editor - Properties
- Configuring the Rich Text Editor - Styles

Configuring the Rich Text Editor - Properties

1. To add the Rich Text Editor field to your form, drag and drop the **Rich Text Editor** component in the design area.
2. To configure the Rich Text Editor, hover your mouse over the top right corner of the widget and click **(Edit)** icon.

The **Rich Text Editor** pop-up window opens with the **Properties** tab open by default.

3. In the **Rich Text Editor** window, under **Properties**, enter or select the required field information.



Field descriptions for Properties

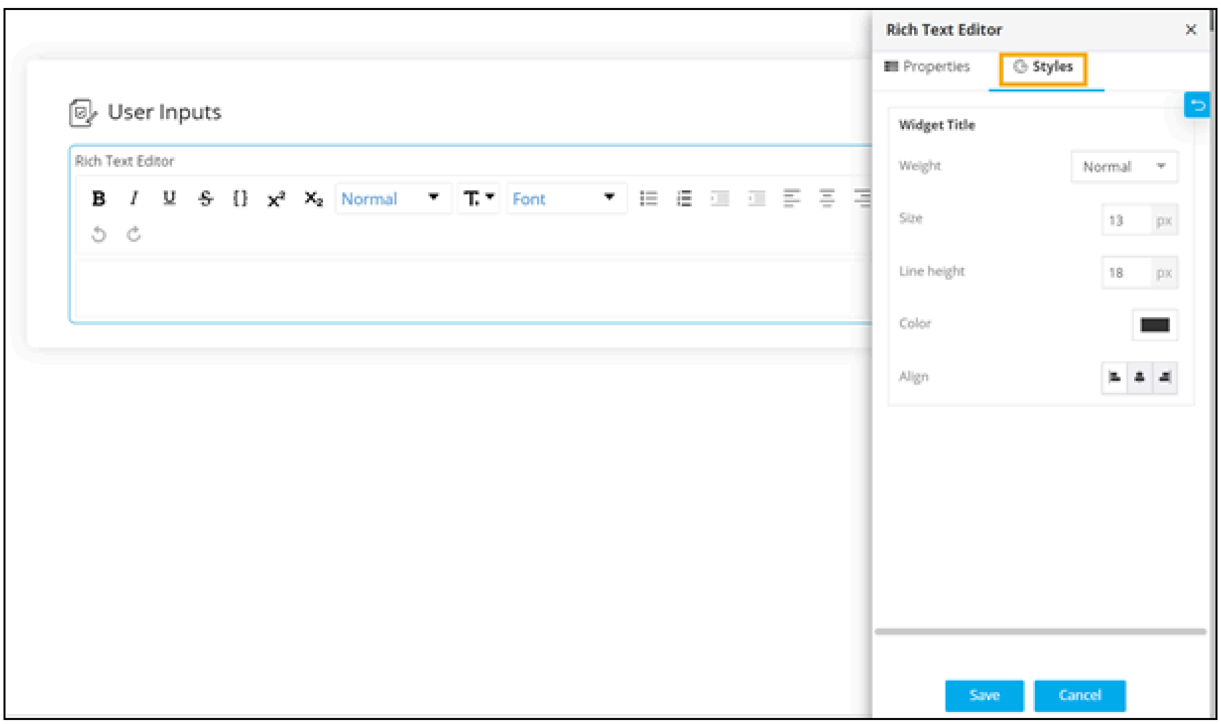
Field	Description
Label name	Enter a valid **Label name** for the form field. For example, Select Date and Time.
Field ID*	Allows you to assign a unique **Field ID** to refer to the field name. • Allowed characters: '-' and '_' • Character Length: Min (1), Max (50).

Field	Description
Depends on	Allows you to define dependencies between form fields based on ‘Field ID’ and ‘Values’. • Operators used: ‘&’ and ‘|’ • Max character: 550 • Format: [Field Id1: value1 & Field Id2: value1 & Field Id3: value3], [Field Id1: value1 | Field Id2: value1].
Help	Allows you to add **Help** text/tooltip for the field.
Read only	Allows you to define this field as **Read Only** or otherwise. If marked as Read Only, a red crossed circle is displayed when you hover your mouse over the read only field and does not allow you to enter any information.
**: Mandatory fields*	

4. Click **Save**.

Configuring the Rich Text Editor - Styles

1. For more options to customize the Rich Text Editor, in the **Rich Text Editor** window, click **Styles**.



Field descriptions for Styles

Option	Description
Weight Title	Allows you to define the following features in the Widget Title: • **Weight** - The **Weight** dropdown list lets you set the font thickness of the field label. • **Size** - Allows you to increase/decrease font **Size** of the field name. • **Line height** - Allows you to increase/decrease the **Line height** of the field. This means the space between the field and the field border. • **Color** - Allows you to define the **Color** of the field label. • **Align** - Allows you to **Align** the field label to the left, center or right side of the field.

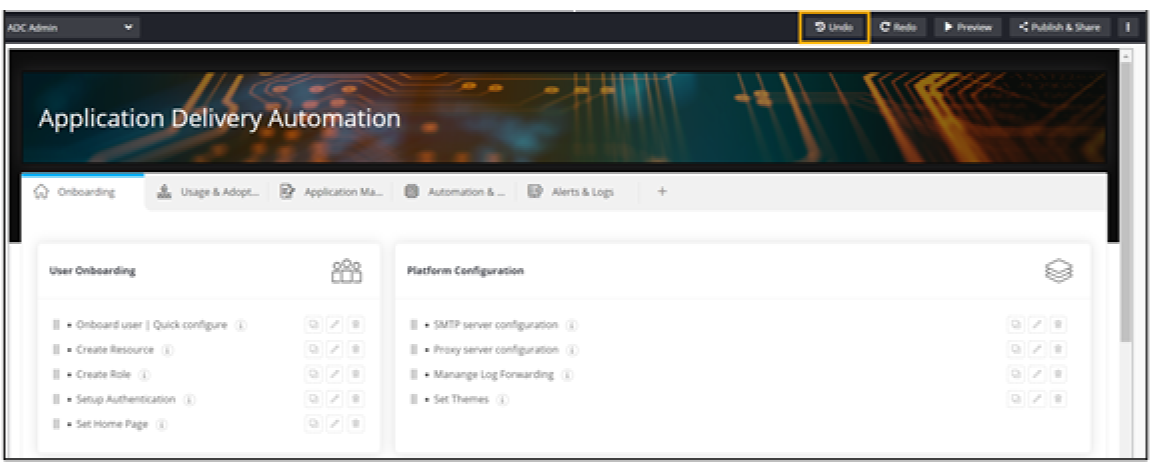
2. To save your settings, click **Save**.
3. To restore the settings under **Styles** to default, click  **(Restore)** icon.

How to Undo/Redo actions

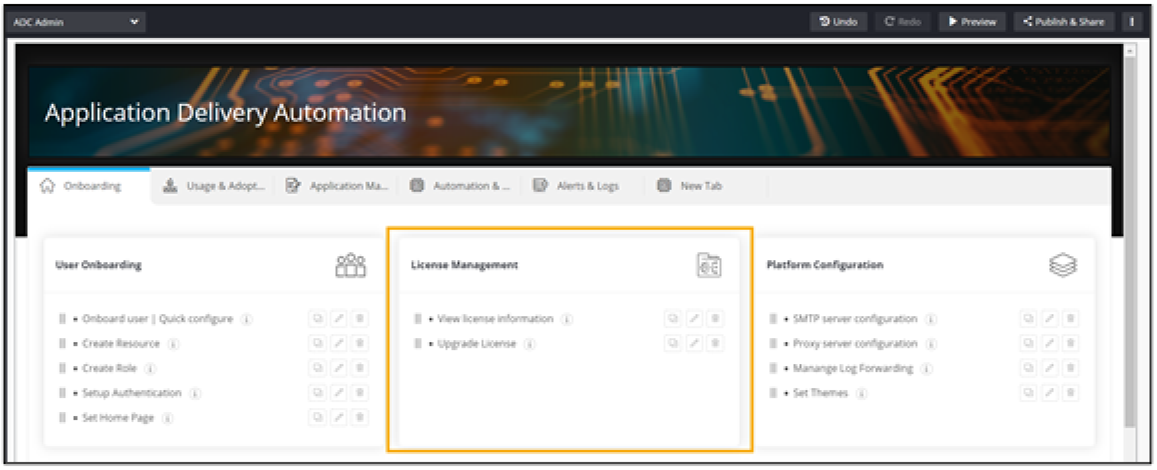
You can undo/redo any actions such as adding or deleting page elements.

To undo an action:

1. Open a page in design mode.
2. From the menu on the top, click **Undo**.



The deleted widget is displayed.



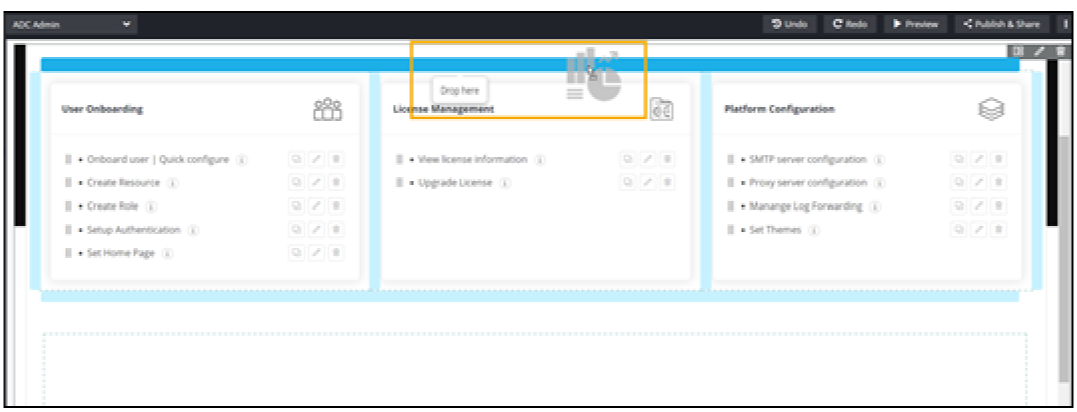
3. To redo the previous action, from the menu on the top, click **Redo**.
   The widget is deleted again.

Moving widgets

You can modify the placement of the widgets on your page and move it around as per your preference.

To move a widget:

1. Hover your mouse over the widget and click  **(Move)** icon.
2. Move the widget to any blue highlighted space in the build area.

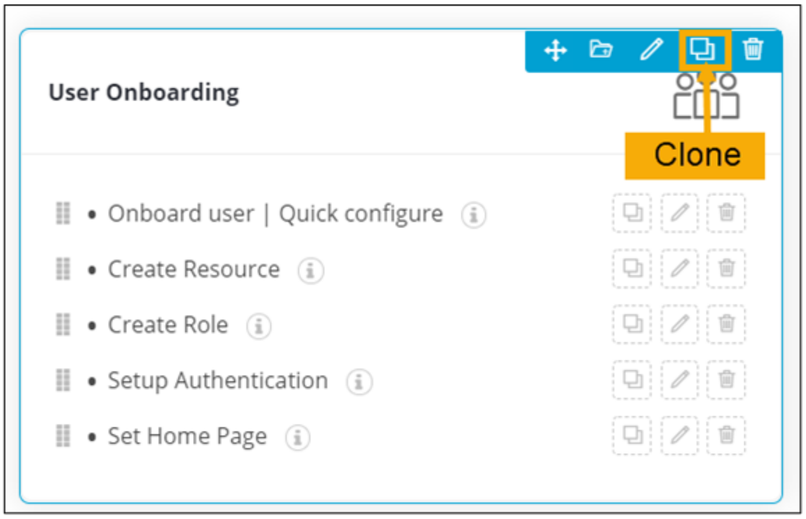


Cloning Widgets

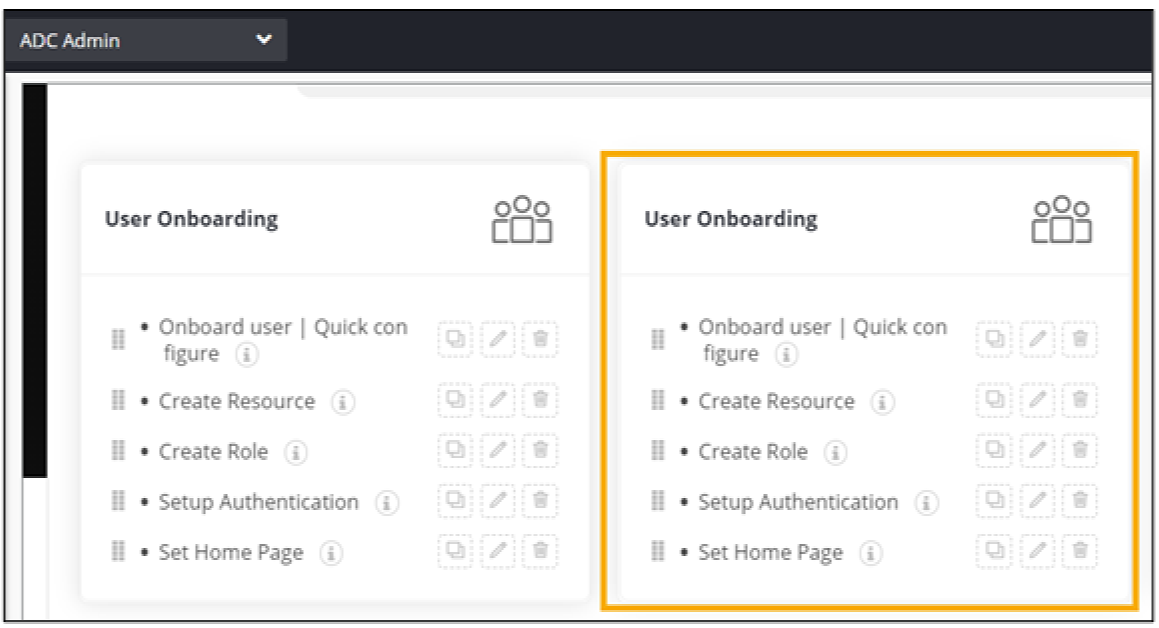
You can clone a widget to have multiple widgets of the same component on your page, without having to drag and drop the components onto the page every time.

To clone a widget:

1. Hover your mouse over the widget.
2. Click  **(Clone)** icon.



A clone of the selected widget is displayed on the page. You can modify this cloned widget as per your requirement.



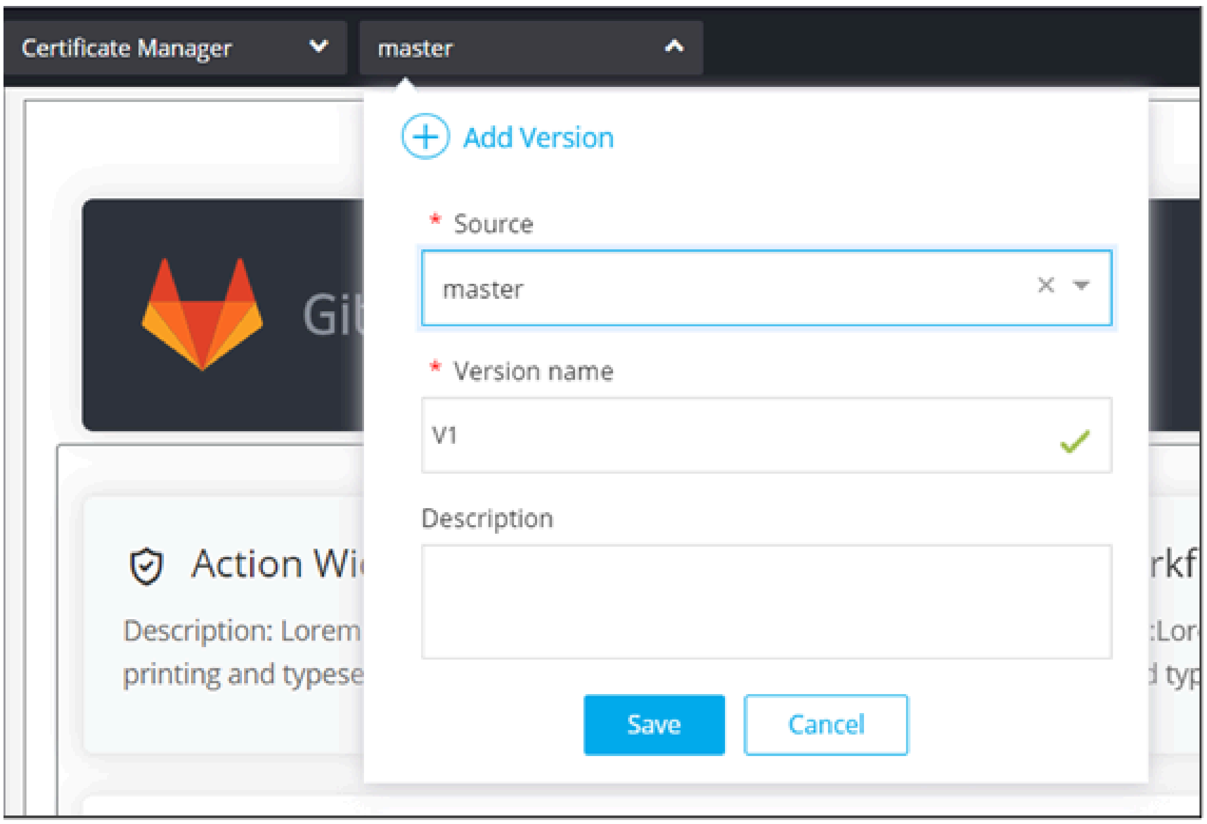
Version Control

This feature allows you to design and have multiple versions of a catalog and enable one master version.

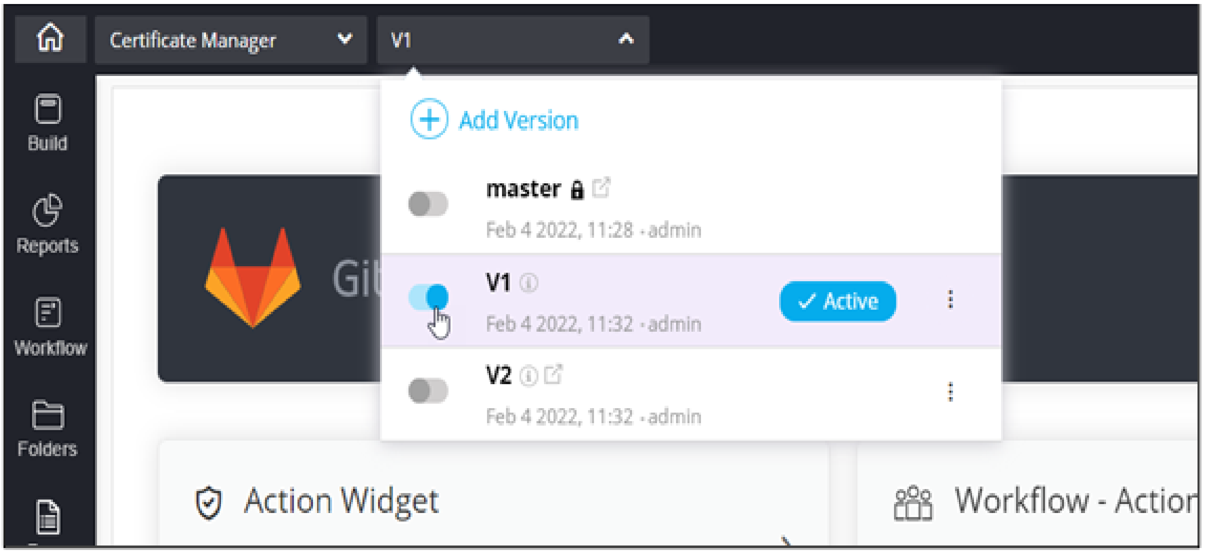
- Provision to version control the catalogs.
- Provision to edit/configure page elements in each version.
- Version control allows for tracking up to ten versions of the catalog.
- Provision to edit and delete versions.
- Provision to preview another version in a new tab/window while working on the current version.

To add a new version:

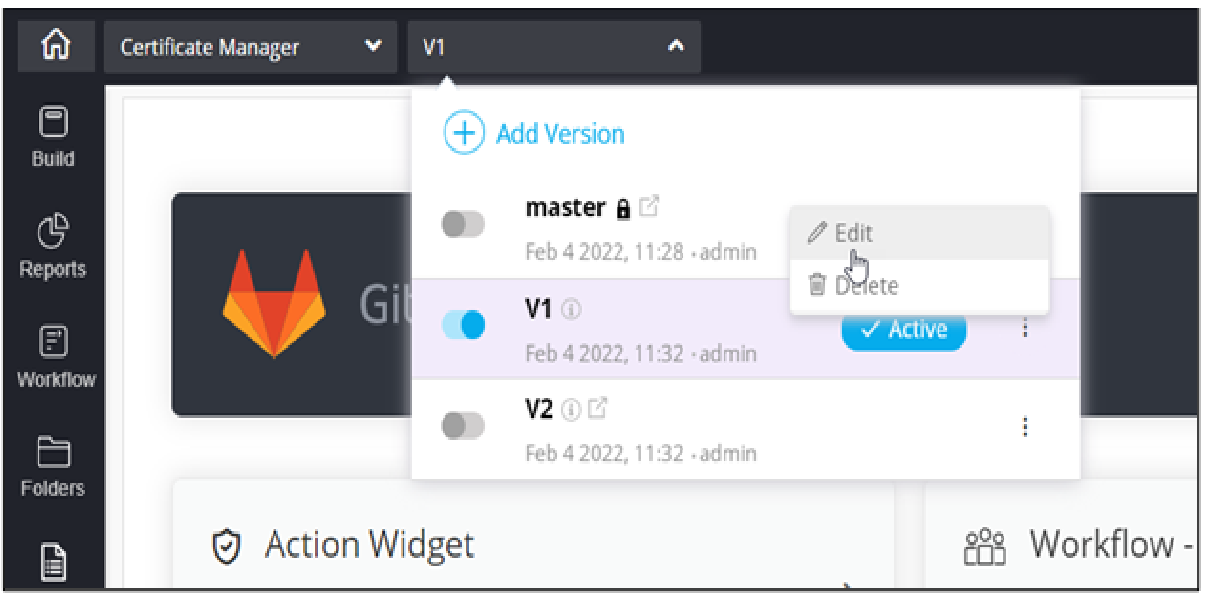
1. From the **master** dropdown menu, select **Add Version**.
2. Select the **Source** and enter a **Version name**.



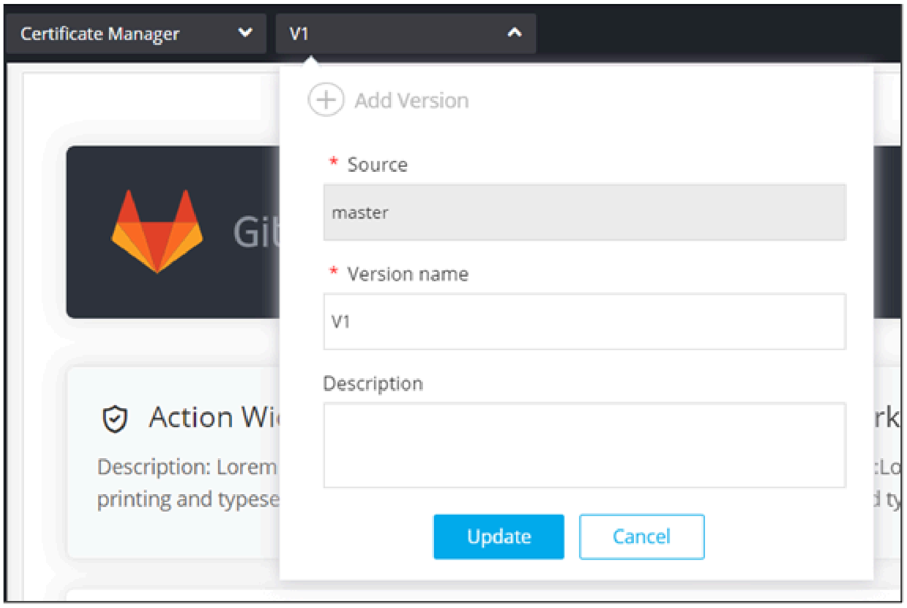
3. Click **Save**.
4. To mark a particular version as the active version for the catalog, enable the toggle next to it.



5. For more options, click .
6. To edit the version details, from the options displayed, click **Edit**.



7. Edit the version details and click **Update**.



> **Important:** The master version cannot be edited.

8. To delete the version, click .
9. From the options displayed, click **Delete**.

> **Important:** The master version or any active version cannot be deleted.

Preview a Service Catalog

You can see a preview of your custom catalog before publishing it. Also, certain functionalities such as modifications in the workflow components are only available in the preview mode.

To preview a service catalog:

1. Open a catalog from the Design inventory.
2. From the top right corner of the screen, click **Preview**.
   A preview of the catalog opens in a new tab on your screen.

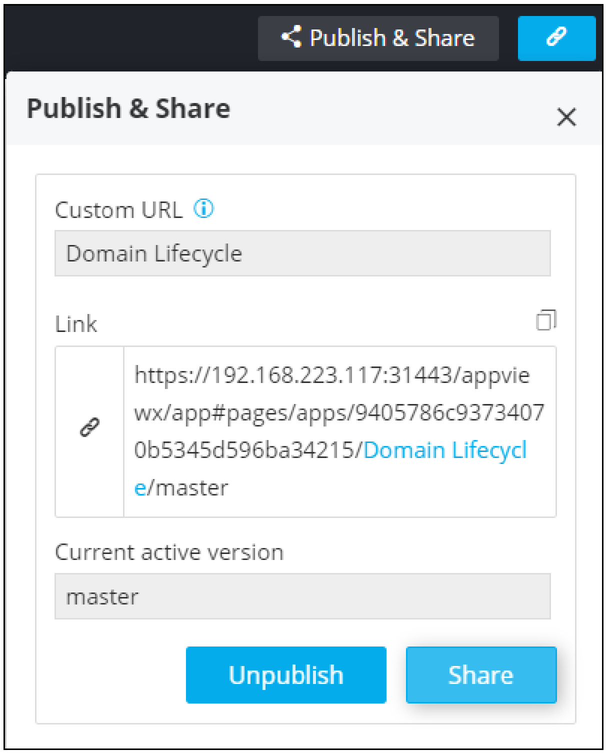
Publishing a Service Catalog

You can publish a service catalog once all the required components have been configured.

To publish a service catalog:

1. From the top right corner of the catalog, click **Publish & Share**.

   The **Publish & Share** pop-up window opens.



Field descriptions for Publish & Share

Field	Description
Custom URL	Allows you to enter a **Custom URL** for the catalog. **Note:** Page name is set as the default URL.
Link	Displays the link for the catalog. Click on the link to open your catalog in a new tab. To copy the link click **(Clone)** icon.
Current active version	Displays the current active version of the catalog. **Note:** This is a read-only field.

2. To publish your catalog, in the **Publish & Share** window, click **Publish**.

The Published catalog is added to the **Published** Inventory.

> **Important:** Only the current active version of the catalog will be published and added to the **Published** inventory.

Sharing a Service Catalog

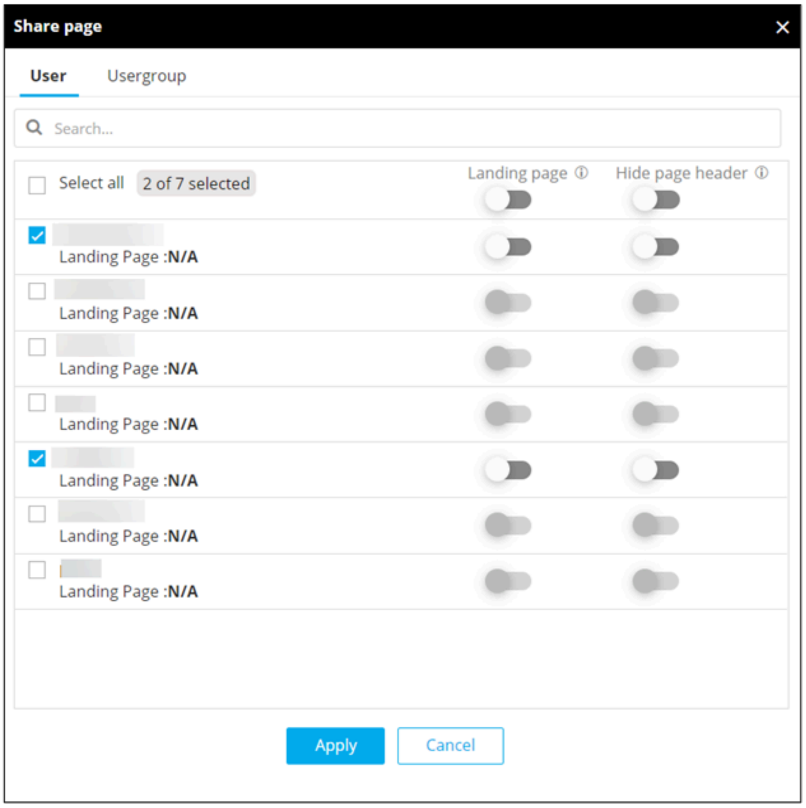
You can share your published service catalogs with other users and/or user groups.

- Sharing a Service Catalog with a User(s)
- Sharing a Service Catalog with a User group(s)
- Scenarios

Sharing a Service Catalog with a User(s)

1. Open a catalog from the Design section.
2. From the top right corner of the screen, click **Publish & Share**.
3. In the **Publish & Share** window, click **Share**.

   The **Share page** pop-up window is displayed.
4. To select the user with whom the catalog will be shared, in the **Share page** pop-up window, under **User**, select the checkbox next to the username.



Note: A catalog can be shared with multiple users.

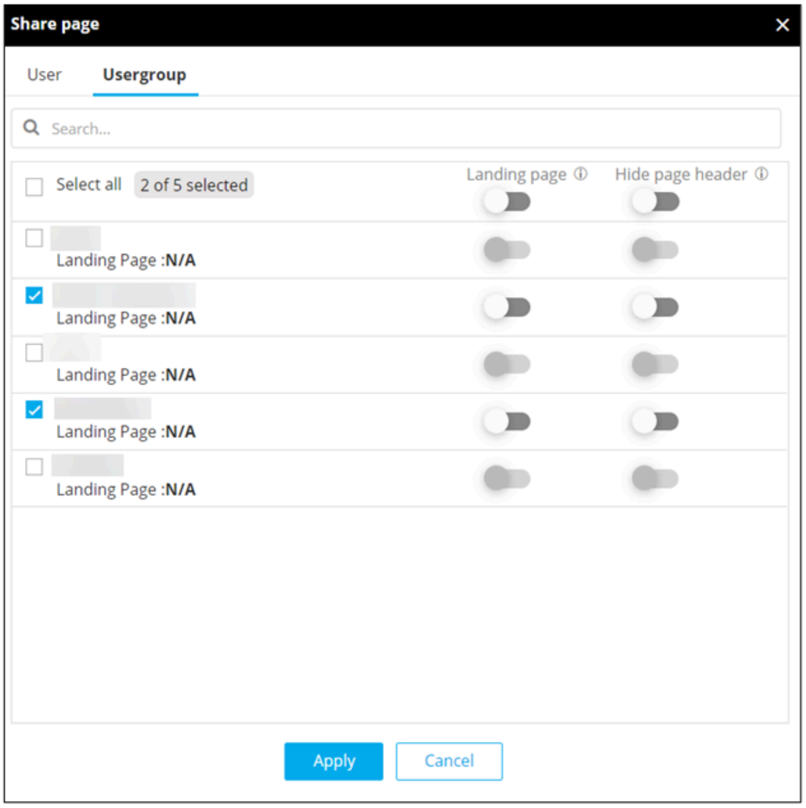
5. To share the catalog with selected user(s), click **Apply**.

Important: To share a specific version of the catalog, ensure it is the current active version.

Sharing a Service Catalog with a User group(s)

1. Open a catalog from the Design section.
2. From the top right corner of the screen, click **Publish & Share**.
3. In the **Publish & Share** window, click **Share**.

   The **Share page** pop-up window is displayed.
4. To select the usergroup with whom the catalog will be shared, in the **Share page** window, under **Usergroup**, select the checkbox next to the usergroup.



Note: A catalog can be shared with multiple user groups.

5. To share the catalog with the selected user group(s), click **Apply**.

Important: To share a specific version of the catalog, ensure it is the current active version.

Scenarios

When a user creates a page and shares it with another user(s) in design mode,

- All actions such as renaming the page, adding/deleting components, changes in widgets and their properties and so on made by the user who created the page will be reflected for the user(s) with which the page has been shared.

Service Catalogs with Birthright Role

Birthright provisioning allows external users to log into AppViewX even when they have not been assigned to any user groups. By enabling birthright to a user group, when an external user logs into AppViewX, they will see the landing page of the user groups for which Birthright has been enabled.

If an external user has been assigned to a user group then,

- If that user group does not have a landing page set up, then the landing page associated with the group for which Birthright is enabled will be the landing page for this user.
- If that user group has a landing page set up, then the landing page that appears when the user logs in will be dynamic. However, any further changes to the landing page settings will impact the landing page of this user. The latest page set as the landing page becomes the landing page of this user when they login.

- Limitation

Limitation

If the birthright provisioning is changed from one user group to another, for example, if the birthright was enabled for the ‘Admin’ user group, but now it has been enabled for the ‘end user’ user group, there is no change to the landing page for the user, i.e. the landing page of the ‘end user’ user group will not be updated.

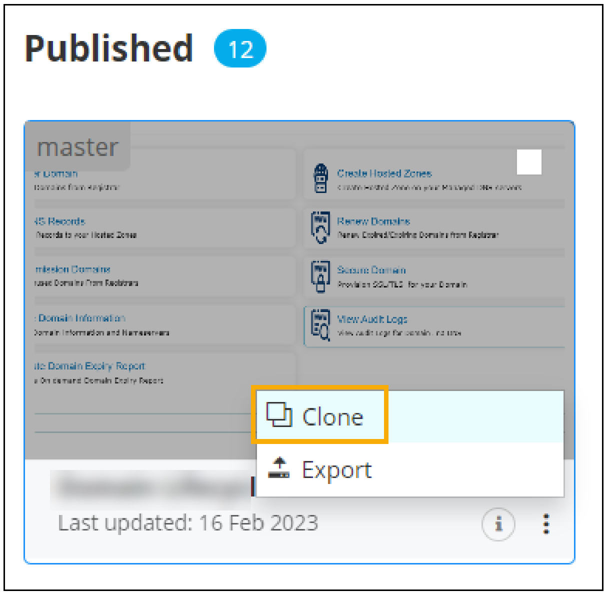
Cloning a Service Catalog

You can clone catalogs that are in the Published inventory as well as in the Design section.

- Cloning a Published Catalog
- Cloning a Catalog in Design mode

Cloning a Published Catalog

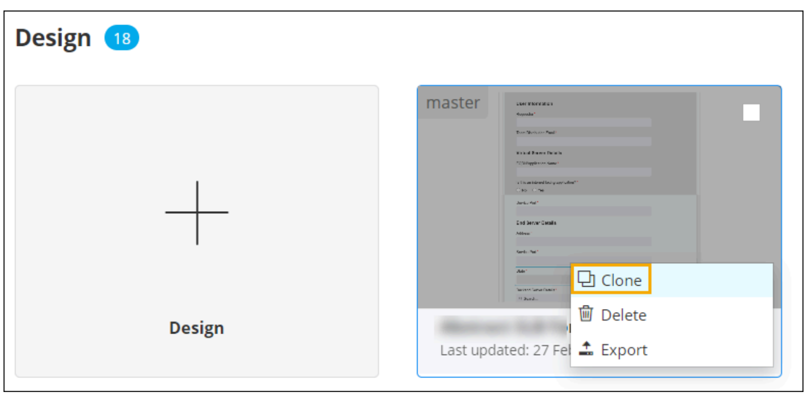
1. On the **Published** page, click ⋮ on the page to be cloned.
2. From the options displayed, click **Clone**.



The cloned catalog opens in the same tab and a copy of the cloned catalog is added to the **Design Inventory**.

Cloning a Catalog in Design mode

1. On the **Design** page, click on the page to be cloned.
2. From the options displayed, click **Clone**.



The selected catalog opens in the same tab and a copy of the cloned catalog is added to the **Design** inventory.

Setting up the Landing Page

The Landing page is the page that appears first on the screen when a user logs into AppViewX. Only a Published service catalog can be shared and set as a landing page. This Landing page can be also accessed by clicking the AppViewX logo on the top left corner of the computer screen.

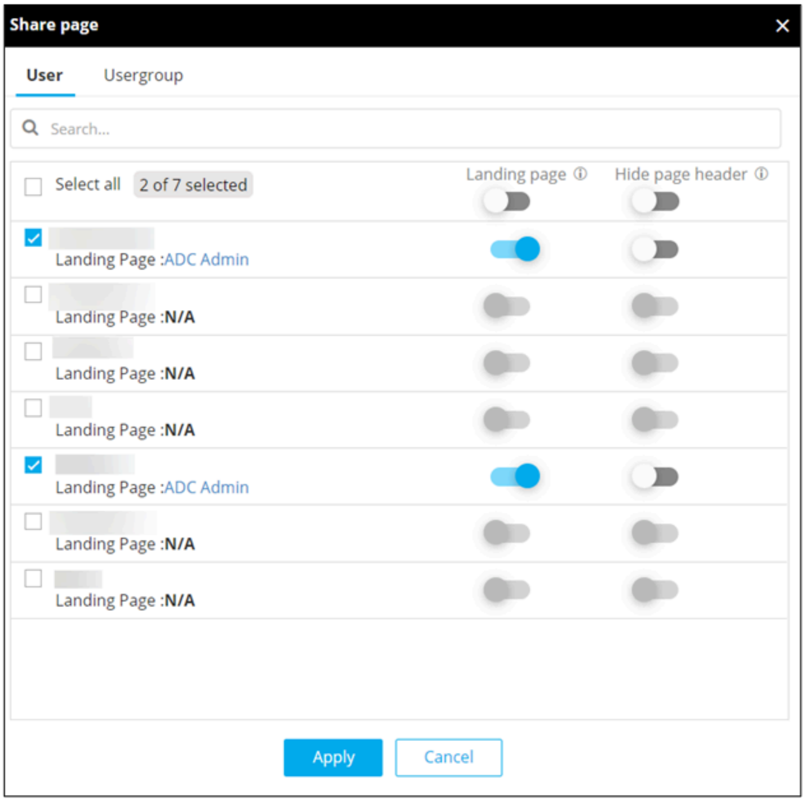
Important: When setting the Landing page, only the current active version of the catalog will be set as the Landing page.

- Setting up the Landing Page when Publishing/Sharing
- Setting up the Landing Page under User Preferences

Setting up the Landing Page when Publishing/Sharing

To set the current page as a landing page for certain user(s) and/or user group(s):

1. Open a Published page.
2. From the top right corner of the screen, click **Publish & Share**.
3. In the **Publish & Share** window, click **Share**.
4. In the **Share page** window, select the user(s) for which the current catalog is to be set as the Landing page.
5. To set the current catalog as the landing page for selected user(s), in the **Share page** window, under **Landing page,** turn on the toggle for the selected user(s).



6. To implement the selected settings, click **Apply**.

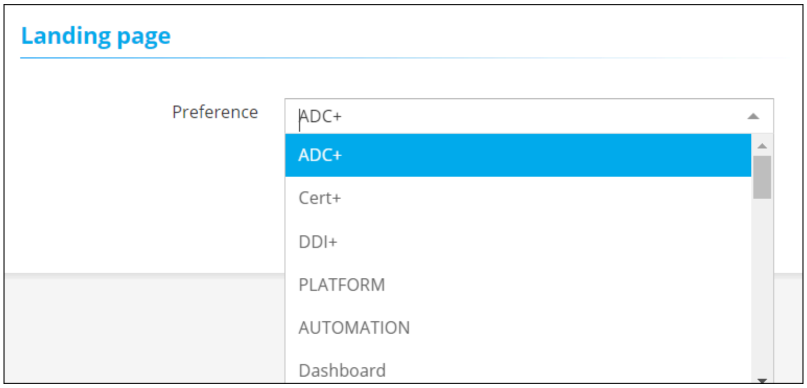
> **Note:** The current active version of the catalog will be set as the Landing page for the selected users.

Setting up the Landing Page under User Preferences

You can also set a specific page as your landing page by modifying user preferences.

To set landing page under user preferences:

1. From the top right corner of the screen, click **(Profile)** icon.
2. From the options displayed, select **User preferences**.
3. On the **User preferences** page, from the drop down in the **Landing page** section, select the catalog that you want to set as the landing page.



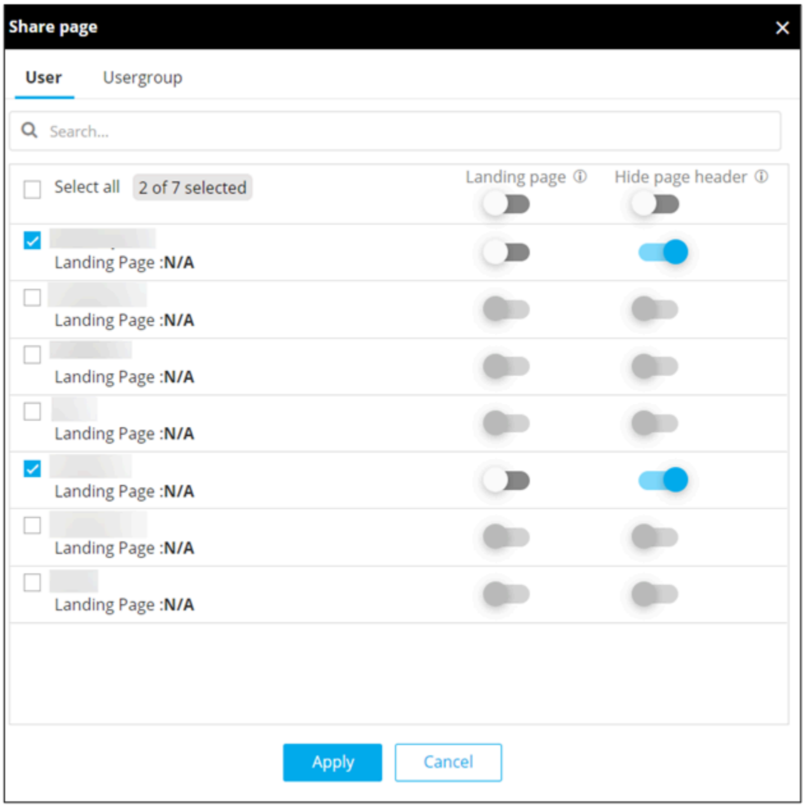
4. To set the selected catalog as your landing page, click **Save**.

Hiding the Page Header

You have the option of hiding the AppViewX header on your published catalogs. This can be done while sharing a published catalog with user(s) and/or usergroup(s).

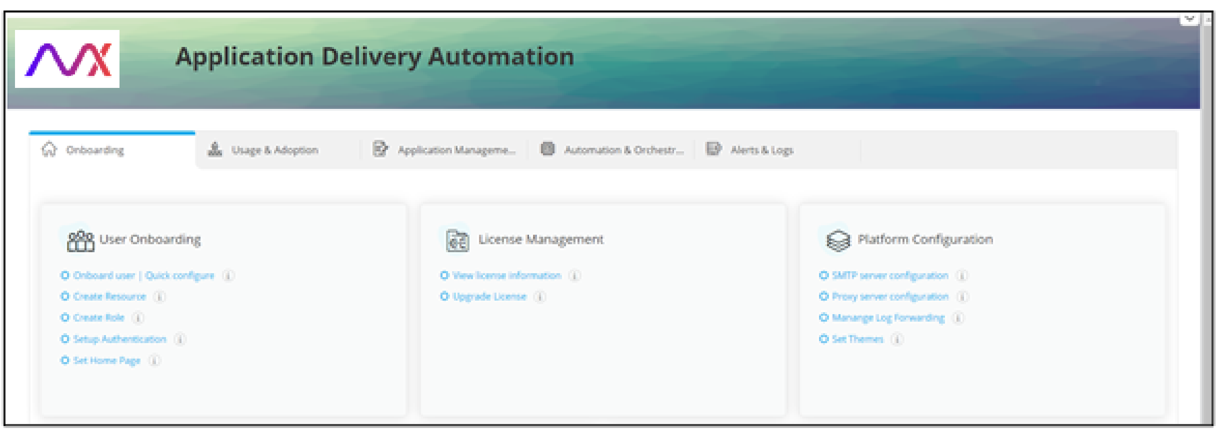
To hide the page header:

1. Open a Published page.
2. From the top right corner of the screen, click **Publish & Share**.
3. In the **Publish & Share** window, click **Share**.
4. In the **Share page** window, select the user(s) from the list.
5. To hide the page header for the selected user(s), under **Hide page header**, turn on the toggle for the selected user(s).



6. To implement the settings, click **Apply**.

The AppViewX header is hidden.



Note: The Hide page header option is independent of whether the catalog is set as the Landing page for a user or not. The page header will be hidden even when the Published catalog is accessed with its URL.

7. To show the application header, from the top right corner of the screen click .

8. To collapse the application header, from the top right of the screen, click .

Troubleshooting

Following are the common issues and their solutions:

Issue	Cause	Solution
Service Catalog module does not load	• avx-web-visual-page-builder pod is down. • avx-subsystems-automation pod is down.	Restart the pods and refresh the catalog.
Design canvas throws error	• avx-web-visual-page-builder pod is down. • visual_page_builder_toolbox is not available as a web plugin/addOn.	• Restart avx-web-visual-page-builder pod. • Deploy the visual_page_builder_toolbox in the required location.
Catalog actions (Create, Publish, Share etc.) not enabled or available	The user might not have permission for these Service catalog actions enabled.	Enable the required Service catalog action permissions for the user's role.
Landing page not set appropriately for shared users	• The catalog might have been shared before it was set as the landing page. • Landing page setting might be overridden in page share tab for user/user group or application user preference.	• The catalog has to be shared after setting the landing page. • Cross verify landing page setting in share tab and application user preference.

Following are the common issues and their solutions: (continued)

Issue	Cause	Solution
Improper data or error in widgets	Discrepancies in data sent/received from server.	Analyze/Share the Service Catalog related logs listed here with dev team: • webApp.log • visual_page_builder_engine.log • avx-subsystems.log