Cisco Crosswork Workflow Manager Solutions 2.1 MOPs User Guide

Cisco Crosswork Workflow Manager 2
Updated January 30, 2026

Topics included

1 Get Started With Crosswork Workflow Manager MOPs....4
CWM Solutions MOPs....5
Application types and MOP structural hierarchy....5
Default MOPs....6
Find MOPs....6
Find MOP actions....7
Inspect MOP and MOP action details....8

2 Customize Crosswork Workflow Manager Solutions MOPs....10
CWM Solutions MOP Customization....11
Export MOPs....11
Import MOPs....11
Clone and edit MOPs....12
Create custom MOPs....13
Custom MOP action types....17
Requirements for MOP action integration....17
MOP action tags....17
MOP action data I/O requirements....19
MOP action data I/O examples....20

3 MOP Actions....26
MOP action categories....27
Runtime app-data for MOP actions....27
Cisco IOS XR MOP actions....29
Cisco IOS XE MOP actions....59
Juniper MX MOP actions....78
Generic MOP workflows....136

1 Get Started With Crosswork Workflow Manager MOPs

Topics:

- CWM Solutions MOPs
- Find MOPs

Identifies the introductory CWM Solutions MOPs topics and the related MOP lookup and inspection tasks in this chapter.

The chapter introduces CWM Solutions MOPs, application types, installation requirements, default MOPs, and tasks for finding and inspecting MOPs and MOP actions.

CWM Solutions MOPs

Explains CWM Solutions MOPs, application type hierarchy, installation requirements, and default MOPs before you find and inspect MOPs and MOP actions.

Cisco Crosswork Workflow Manager Solutions (CWM Solutions) is a collection of pre-built use cases that offers users a convenient and efficient way to manage their network devices. It provides out-of-the-box use cases that are easy to deploy and ready to use.

The CWM Solutions MOPs application lets users create Methods of Procedure (MOPs) and MOP actions that allow the other CWM Solutions applications to automate and routinize a wide variety of network operations, from detecting the current version of a network operating system installed on a particular device, to upgrading the device operating system and applying new device configurations that change the role the device plays in your network.

CWM Solutions MOPs application

CWM Solutions and the MOPs application in particular are extremely powerful network automation products. Their underlying functionalities are enabled by the workflow automation functions provided by Cisco Crosswork Workflow Automation (CWM) and Cisco Network Services Orchestration (NSO) products.

A full explanation of the capabilities of these products is outside the scope of this guide, which concentrates exclusively on how to use the CWM Solutions MOPs application.

For a deeper dive, and examples of how to create workflows that can serve as MOPs and MOP actions, refer to the *Cisco Crosswork Workflow Manager 2.1 Get Started Guide*.

Application types and MOP structural hierarchy

Explains how application types, MOPs, and MOP actions relate so that you can manage valid network automation strategies in CWM Solutions.

When creating or modifying a MOP, the **Application Type** you select acts as a primary filter. It automatically adds workflows to the **Available Actions** list for that specific automation strategy.

To effectively manage network automation within CWM Solutions, you must understand the hierarchical relationship between:

- application types,
- MOPs,
- and MOP actions.

Application type hierarchy

The application type serves as the root object that governs the execution strategy and product associations.

If you select the Fleet Upgrade application type, only upgrade-related actions are available. This approach helps you prevent configuration errors and ensures valid execution across the network.

The hierarchy contains these objects:

- **Application Type (root)**: This is the highest level of categorization, such as Fleet Upgrade or Golden Configuration. It defines the overarching operational logic and ensures that only compatible MOP actions and device series are associated. The application type defines the stages for the procedure: pre, distribute, activate, commit, and post.
- **MOP (procedure)**: A MOP is a specific instance of an application type. It organizes selected MOP actions into a plan for a specific vendor and product series.

- **MOP Action (step)**: Each MOP action is an individual workflow and serves as a building block of a MOP. These actions perform specific tasks, such as checking disk space or capturing configuration summaries. The system logs actions to specific application types to ensure they are used only in the correct operational context.

Default MOPs

Describes the default MOPs that are installed with the MOPs application for supported CWM Solutions applications, vendors, and device families.

The MOPs application comes installed with several default, or pre-built, MOPs. These MOPs are provided by Cisco, cannot be modified directly, although they can be copied and the copies modified, and are intended for use with specific CWM Solutions applications, specific supported vendors, and designated device families.

These default MOPs are installed with the MOPs application.

- **Default Juniper Upgrade**: For performing fleet upgrades on Juniper Networks Inc. MX series devices running JunOS.
- **Default XE Upgrade**: For performing fleet upgrades on Cisco Systems ASR 900 series devices running Cisco IOS XE. This also includes the Cisco ASR 900 Series Aggregation Services Routers, Cisco ASR 920 Series Aggregation Services Routers, Cisco ASR 1000 Series Aggregation Services Routers, Cisco Catalyst 8000V Edge Chassis Platforms, Cisco Catalyst 8300 Series Edge Platforms, and the Cisco Catalyst 9500 Series Switch.
- **Default XR Upgrade**: For performing fleet upgrades on Cisco Systems 8000 series devices running Cisco IOS XR. This also includes the Cisco 8000 Series Routers, Cisco ASR 9000 Series Aggregation Services Routers, Cisco Network Convergence System 5500 Series, Cisco Network Convergence System 540 Series Routers, Cisco Network Convergence System 540L Series Routers, and the Cisco Network Convergence System 5700 Series Routers.
- **Default Golden Config – Juniper**: For applying Golden Configuration templates on Juniper devices.
- **Default Golden Config – Cisco**: For applying Golden Configuration on Cisco Systems devices.

> **Note**
>
> In most cases, you will want to select one of these default MOPs when performing a fleet upgrade or applying a golden configuration. They are usually your safest choice. Each of the default MOPs is customized for the supported vendor and product series, and the customizations are extensive. Each MOP varies significantly in the number of action types it makes available, the selected actions it performs during an upgrade, and the stages it goes through as it performs these actions.

Find MOPs

Teaches you how to find MOPs and MOP actions, and inspect MOP details by using menus, search, sorting, filters, and pagination controls.

Find MOPs in the Crosswork Workflow Manager Solutions database.

Procedure

1. From the main menu, choose **CWM Solutions** > **MOPs** > **MOP list**.

   Crosswork displays the list of MOPs.

2. Use any of these quick methods to find the MOP you want.

- Scroll the **MOP list** until the name of the MOP you want appears.
- Begin entering characters in the **Search** field to narrow the table listing to only those MOPs with values that match your search string. For example, enter `Juniper` to search for MOPs where the Vendor is "Juniper Networks".
- Click any of the table column headings to sort the list by the values in that column. You can sort on any of these column headings: **Status**, **MOP Name**, **Application type**, **Vendor**, **Product series**, **Actions**, **Tags**, or **Created by**.

3. Enter or select values in one or more of the filter fields at the top of the **MOP list**.

   The filters you set narrow the table listing to just those MOPs that match the values you set.

4. If you want to use all of the filter fields at once, click the **Filters** icon to display all the filter fields in a **Filter** popup list at the right side of the table.

   When you have chosen all the filter values you want, click **Apply** to narrow the table listing to only those MOPs with matching values.

   Click **Cancel** or the **X** icon to exit the popup but retain the filters you set.

   Click **Reset all** to clear the filters you set.

5. If there is more than one page of listings, use the **Rows per page** and page number controls at the bottom of the table to scroll to the page containing the MOP you want.

   You can also click the **Table settings** icon to change the table display density and the number of table columns shown.

Find MOP actions

Teaches you how to find MOP actions in the Crosswork Workflow Manager Solutions database by using menus, search, sorting, filters, and pagination controls.

Find MOP actions in the Crosswork Workflow Manager Solutions database.

Procedure

1. From the main menu, choose **CWM Solutions** > **MOPs** > **MOP actions**.

   Crosswork displays the list of MOP actions.

2. Use any of these quick methods to find the MOP actions you want.
   - Scroll the **MOP actions** list until the name of the MOP action you want appears.
   - Begin entering characters in the **Search** field to narrow the table listing to only those MOP actions with values that match your search string. For example, enter `Fleet Upgrade` to search for MOP actions where the Application type is "Fleet Upgrade".
   - Click any of the table column headings to sort the MOP actions list by the values in that column. You can sort on any of these column headings: **Name**, **Version**, **Description**, **Application type**, **Vendor**, **Product series**, or **Created by**.

3. Enter or select values in one or more of the filter fields at the top of the **MOP actions** list.

   The filters you set narrow the table listing to just those MOP actions that match the values you set.

4. If you want to use all of the filter fields at once, click the **Filters** icon to display all the filter fields in a **Filter** popup list at the right side of the table.

   When you have chosen all the filter values you want, click **Apply** to narrow the table listing to only those MOP actions with matching values.

   Click **Cancel** or the **X** icon to exit the popup but retain the filters you set.

   Click **Reset all** to clear the filters you set.

5. If there is more than one page of listings, use the **Rows per page** and page number controls at the bottom of the table to scroll to the page containing the MOP action you want.

   You can also click the **Table settings** icon to change the table display density and the number of table columns shown.

Inspect MOP and MOP action details

Teaches you how to inspect MOP details, selected MOP actions, and available MOP actions from the MOP list for editable MOPs.

Inspect MOP and MOP action details.

In addition to the information available to you in the **MOP list** table view, you can click the **MOP Name** to view the list of available MOP actions and the MOP actions included in the MOP.

You can also inspect details for any MOP action that has been selected for inclusion in a MOP and, if the MOP is editable, any of the actions available for selection.

Procedure

1. From the main menu, choose **CWM Solutions** > **MOPs**.
2. Click the **MOP List** tab to display the list of MOPs.
3. Under the **MOP Name** column, click the name of the MOP whose actions you want to view.

   Crosswork displays details for the selected MOP, including the lists of **Available actions** and **Selected actions** for the MOP.

4. To view MOP action details, including **Input**, **Version**, **Stage**, Description, and the **workflow definition** for a **Selected action**, click the MOP action **Name**.

   Crosswork displays a popup window with these details to the right of the actions lists.

5. If the MOP is editable, to view details for available MOP actions not included in the MOP, click the **+** icon next to the **Action type** name in the **Available Actions** list at left.

   Crosswork displays a popup window with the details to the right of the two actions lists.

2 Customize Crosswork Workflow Manager Solutions MOPs

Topics:

- CWM Solutions MOP Customization
- Custom MOP action types

Identifies the MOP customization tasks and custom MOP action type information in this chapter.

The chapter includes tasks for exporting and importing MOPs, cloning and editing MOPs, and creating custom MOPs. It also includes information about custom MOP action types, MOP action integration requirements, tags, data I/O requirements, and data I/O examples.

CWM Solutions MOP Customization

Explains how to customize CWM Solutions MOPs by exporting, importing, cloning, editing, creating MOPs, and creating custom MOP action types.

CWM Solutions MOP customization lets you keep backups of MOPs, share custom MOPs with other locations in your organization, clone default MOPs, create MOPs from scratch, and create custom MOP actions.

Customization methods

You can export MOPs as JSON files. You can also use the Import function to share custom MOPs with other locations in your organization that are using CWM Solutions.

The quickest way to create a MOP is to clone one of the default MOPs supplied with Crosswork Workflow Manager Solutions and then modify it as needed.

You can use the MOPs graphic user interface to create MOPs from scratch, in free-form fashion, using the MOP action types that you choose.

You can create custom MOP actions that you can then use whenever you create a custom MOP from scratch or clone and edit an existing MOP.

Export MOPs

Describes how to export MOPs as JSON files so that you can keep backups or share custom MOPs with other locations in your organization.

Export a MOP as a JSON file.

You can export MOPs as JSON files. This is a handy way to keep backups of MOPs you create. You can also use this with the Import function to share custom MOPs with other locations in your organization that are using CWM Solutions.

Procedure

1. From the main menu, choose **CWM Solutions** > **MOPs**. Crosswork displays the **MOP list**, listing all existing MOPs.
2. Follow the steps in Find MOPs to find the MOP you want to export.
3. At the far right, in the same row as the MOP you want to export, click the **More** icon (**...**). Crosswork displays a drop-down menu.
4. From the drop-down menu, choose **Export (JSON)** . MOPs displays a message indicating that the MOP has been exported to a JSON file. Check your local default download folder for a file with a filename that matches the name of the exported MOP, with a `JSON` filename extension.

   Store or transmit the exported MOP file as needed.

Import MOPs

Describes how to import a MOP JSON file into the MOP list.

Import a MOP JSON file.

Note that, although the exported MOP file is editable, you cannot edit and import it again. The edited MOP will be non-functional.

Procedure

1. From the main menu, choose **CWM Solutions** > **MOPs**.
2. Click **Import MOP (JSON)**.
3. Browse to and select the exported file, then click **Import MOP**. When the import is finished, the file appears in the **MOP list**.

Clone and edit MOPs

Describes how to clone a default MOP, give the cloned MOP a unique name, and edit it as needed.

Clone and edit an existing MOP.

The quickest way to create a MOP is to clone one of the default MOPs supplied with Crosswork Workflow Manager Solutions and then modify it as needed. You will be required to give the cloned MOP a unique name, but you can then work with the cloned MOP as though it were entirely new.

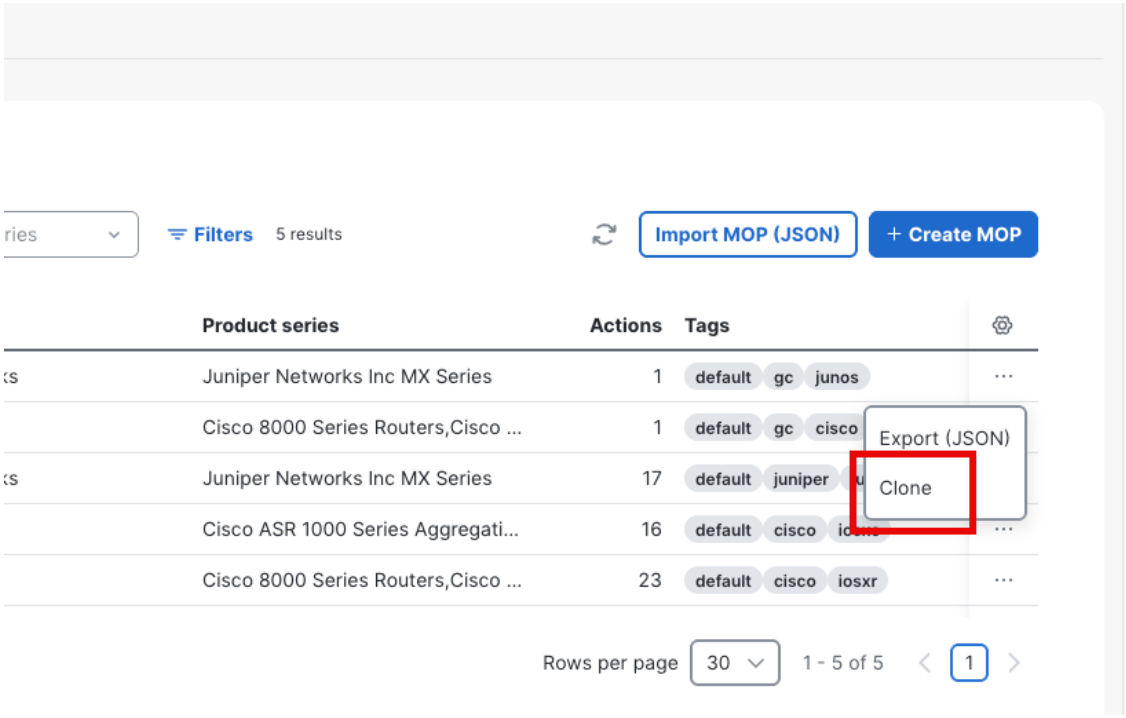
You can create MOPs using other means, such as:

- Creating a MOP from scratch using the GUI, as explained in Create custom MOPs.
- Adding custom MOP action types and adding them to a modified MOP as explained in Custom MOP action types.

Procedure

1. From the main menu, choose **CWM Solutions** > **MOPs** to display the **MOP list**, listing all existing MOPs.
2. At the far right, in the same row as the MOP you want to clone, click the **More** icon (**...**). CWM Solutions displays a drop-down menu like the one at right in the following figure.

Figure 1: Clone a MOP



3. From the drop-down menu, choose **Clone**. CWM Solutions displays an edit window for the MOP you are cloning. All the fields except the **MOP name** will be filled with exact copies of the values they had in the MOP you cloned. This includes the cloned MOP's **Selected actions** and input configurations.
4. Enter a new, unique name for the cloned MOP and click **Save changes**.

   You can now change the values in the editable fields, add or remove MOP actions, and change action inputs in the cloned MOP as you would for a newly created MOP, as explained Create custom MOPs.

   When you are finished, click **Save changes**.

Create custom MOPs

Describes how to create a custom MOP from scratch by using the MOPs graphic user interface.

Create a custom MOP from scratch.

You can use the MOPs graphic user interface to create MOPs from scratch, in free-form fashion, using the MOP action types that you choose. The steps below show how to do this.

You can also create MOPs using other means, such as:

- Cloning and editing an existing MOP, as explained in Clone and edit MOPs.
- Adding custom MOP action types and adding them to a modified MOP as explained in Custom MOP action types.

Procedure

1. From the main menu, choose **CWM Solutions** > **MOPs** to display the **MOP list**, listing all existing MOPs.
2. Click **+ Create MOP**. CWM Solutions displays the **Create new MOP** window, with most of the first five required identification fields blank.
3. Edit the first three of the five identification fields at the top of the window as shown in the table below:

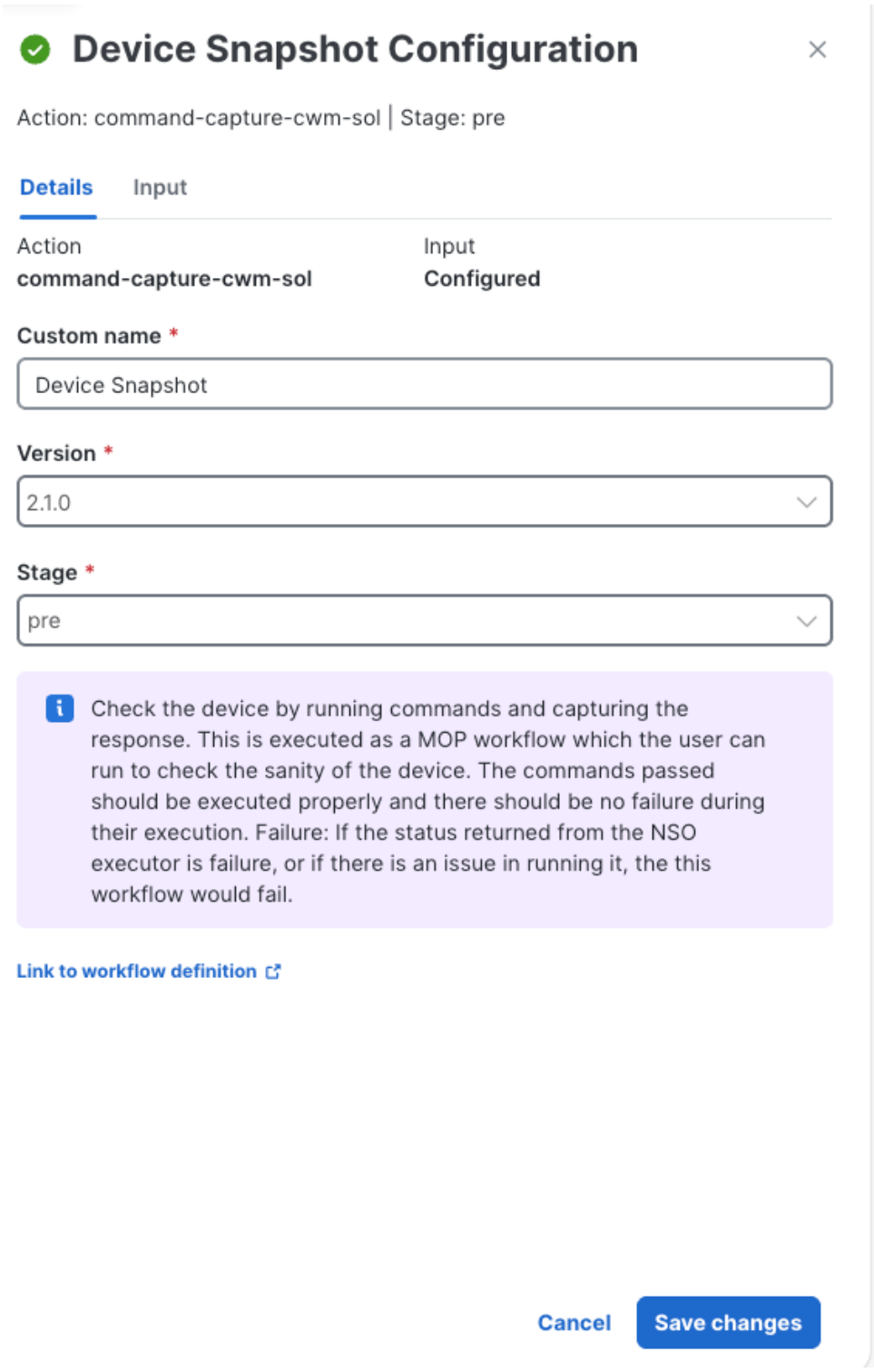
In this field...	Enter or select...
MOP name *	A new name for the new MOP. You must enter a unique name.
Application Type*	The name of the CWM Solutions application for which the MOP is intended (for example, `Device Migration`).
Vendor*	The name of the vendor for whose devices the MOP is intended (for example, `Cisco Systems`)
Product series*	Select one or more product series from the chosen vendor to include in the MOP. Multiple selections allow the MOP to apply to all selected product series.
Tags*	A comma-separated list of tags to be applied to the job whenever a user runs this MOP.

4. As soon as you enter the **tags**, the **Create new MOP** window populates the **Available actions** list with Action Types appropriate for the **Vendor** and **Product series** you selected.
5. Add one of the **Available actions** on the left to the **Selected actions** list on the right:
   a) In the **Action type** list on the left, click the **+** icon next to the **Action type** you want to add. An **Add Action** window for the new Action Type appears, like the one shown below.
   b) Enter a **Custom name** for the newly selected Action, and select the **Stage** at which you want it to be performed and the **Version** of the action if multiple versions of the same action are available.

c) Click **Add**. The newly added Action appears in the **Selected actions** list, at the end of the Actions in the **Stage** you selected for it.

6. To remove one of the **Selected actions** on the right:
   a) Click the more icon (…) shown to the right on the same row as the name of the selected action you want to delete. Crosswork displays a drop-down menu.
   b) Select **Delete**. You will be prompted to verify that you want to delete the Action. Click **Delete** again to remove it.
7. To edit the **Custom name**, **Stage**, **Position**, or **Configuration** details for one of the **Selected actions** on the right:
   a) Click the more icon (…) shown to the right on the same row as the name of the selected action you want to edit. Crosswork displays a drop-down menu.
   b) Select **Edit**. A popup action configuration window for the action appears, with **Details** and **Input** tabs, like the one shown below.

Figure 2: Device snapshot configuration - edit details



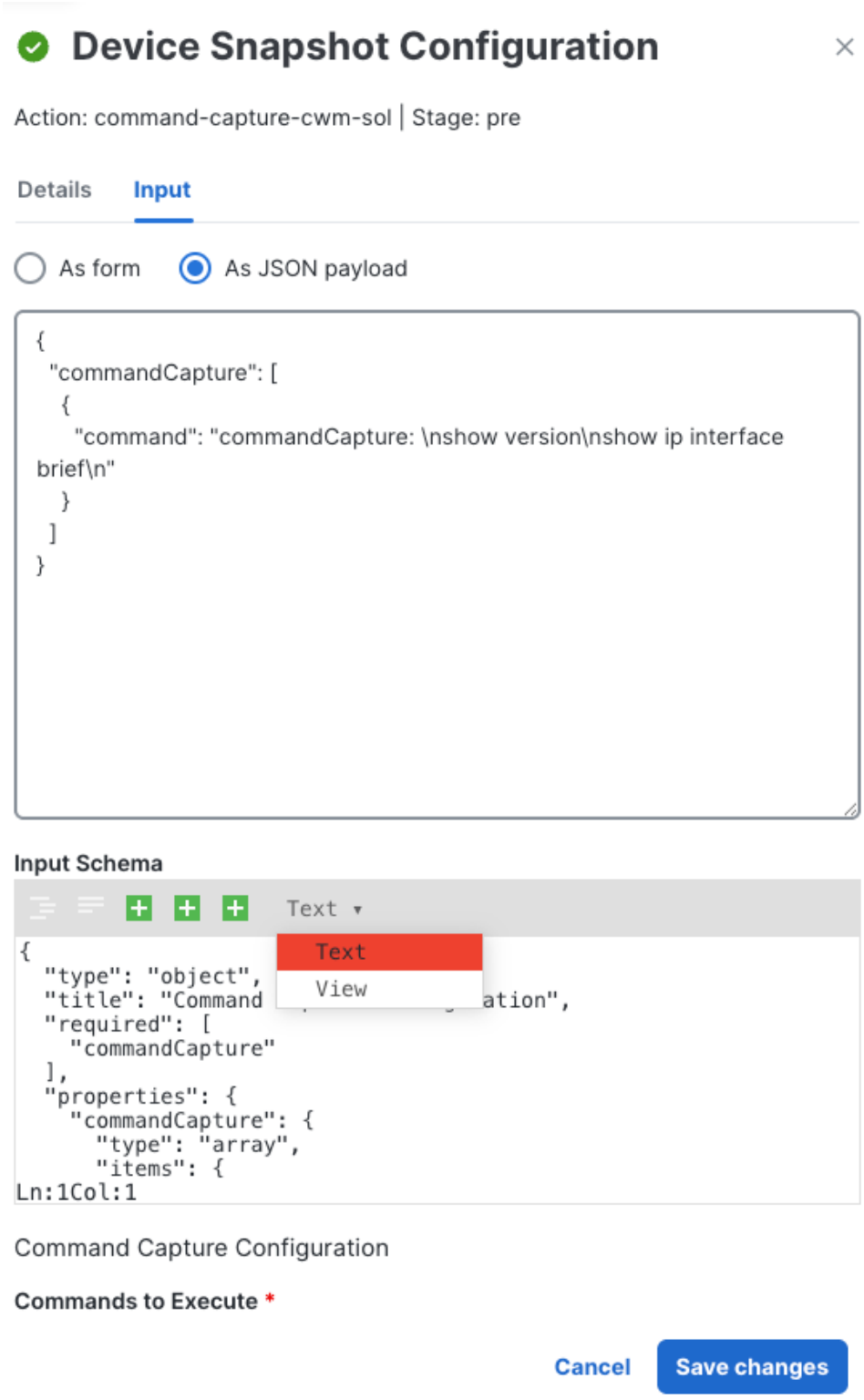
c) To change the selected Action's **Custom name**, **Version**, or **Stage**, edit the respective fields on the **Details** tab.

d) For actions that have configurable inputs: To change the action's input configuration, click the **Input** tab to display the **Input Schema** tools as shown below.

Toggle the **As form** and **As JSON payload** buttons to toggle the configuration view.

Use the **Text/View** drop-down to toggle between JSON and JavaScript views, and format JSON text. You can also use the + icons to format JMEScript queries and sorts.

Figure 3: Device snapshot configuration - toggle



8. When you are finished, click **Save changes**.

Custom MOP action types

Explains how custom MOP actions work as workflows that you can use in custom or cloned MOPs.

You can create custom MOP actions that you can then use whenever you create a custom MOP from scratch or clone and edit an existing MOP.

Your custom MOP action is a workflow, with outputs, inputs, logic, name and tags. Making the most of this MOP action customization capability, you can extend MOP pre- and post-check functionality for supported devices in any way you wish: providing device health snapshots, making configuration changes to divert traffic while you perform upgrades, and so on.

Workflow integration

Creating a custom workflow from scratch is covered in depth in the *Cisco Crosswork Workflow Manager 2.1 Workflow Creator Guide* . But it is important to remember that MOP actions, default or custom, are not *standalone* workflows. They are intended to integrate smoothly with other MOP actions into a MOP, and therefore must follow the constraints and guidelines set by the workflow platform.

You can develop workflows independently or generate a MOP action skeleton in the MOP Actions UI. This approach creates a foundational structure that meets the mandatory constraints for versioning and tagging. After you create the skeleton, go to the Workflow Automation interface to refine the logic and execution steps. To complete the integration, go to the MOP Actions UI and select Sync Actions. This step ensures that the updated workflow is recognized as a functional MOP action.

Those constraints are explained in the next section, Guidelines for CWM Solutions MOP actions, followed by additional sections with examples showing how you can follow these guidelines. The examples come from the MOP actions supplied with CWM Solutions default MOPs. You can view many similar examples by selecting **Workflow Automation** > **Workflows** from the main menu. Click on any of the CWM Solutions workflows in the **Workflow name** list and then click **Graph** or **Code** to explore the MOP action code.

Requirements for MOP action integration

Lists the versioning, tag, and data I/O requirements that MOP actions must satisfy to integrate with CWM Solutions applications.

In addition to the normal features that all workflows must have, such as a unique name, MOP actions must also incorporate features that allow them to integrate smoothly with other CWM Solutions applications and complete successfully:

- **Version**: Each MOP action must be assigned a version number consisting of three digits separated by periods. This standard follows the Semantic Versioning v1.0.0 Specification for major, minor and patch versioning; for details, refer to the link.
- **Tags**: Each MOP action must have tags assigned that will identify it as a type of CWM Solutions MOP action. The tags also identify other MOP action characteristics, such as the vendor and device types with which you want the MOP action to work. Tags allow CWM Solutions to control how it processes the MOP action and to display it properly in the user interface's list of available MOP action types. For a list of the tags you can use and examples of ways to combine them, refer to Tags for CWM Solutions MOP Actions.
- **Data I/O**: The MOP action must be designed to accept data input from multiple sources. The MOP action should also generate a response in a pre-defined format to parse and process the workflow outcome. For more information and examples, refer to Data I/O for CWM Solutions MOP actions.

MOP action tags

Lists MOP action tags and shows how tags identify and control workflows used as MOP actions.

You can assign any of the tags shown in the table below to a MOP action by entering them in the **Tags** field when creating or editing a MOP action workflow. Only the **mopActivity** tag is required. How you use the optional tags will depend on your purpose and the level of generalization you want to achieve using it. For example: The `vendor` tag is optional, so you do not need to assign a `vendor` tag value if the MOP action can be used any of the devices of any vendor. However, you may want to assign `vendor:Cisco` and `vendor:Juniper` tags if you want to restrict use of the MOP action to just those two vendors.

Table 1: MOP Action Tags

This Tag...	Indicates	Required?	Case Sensitive?
`mopActivity`	This workflow is a MOP activity (aka a MOP action). CWM Solutions uses the `mopActivity` tag to recognize any workflow that needs to be identified as a MOP action and to distinguish it from general workflows.	Yes	Yes

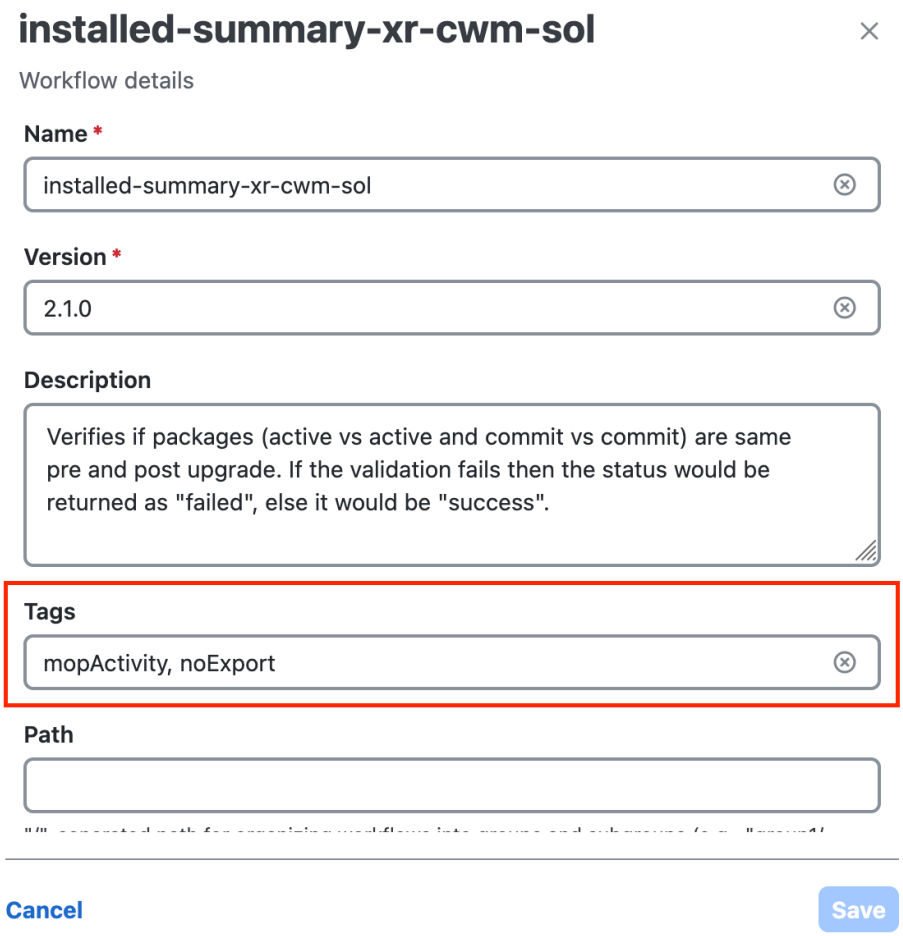
`noExport`	Prevents a MOP action or sub-workflow from being included when a MOP is exported. You can use this tag to exclude specific workflow definitions, such as generic sub-workflows, from the exported JSON file. For example, if a MOP action calls a chain of sub-workflows, only those without the `noExport` tag are included:
	`parent-cwm-sol (tags: mopActivity)`
	`└──child-cwm-sol (tags: )`
	`└──grandchild-cwm-sol (tags: noExport)`
	In this case, only the definitions for `parent-cwm-sol` and `child-cwm-sol` are exported because the grandchild, `grandchild-cwm-sol`, is tagged with `noExport`.
		No	No

Figure 1, below, shows the internal `wftags` entry from the `installed-summary-xr-cwm-sol` Fleet Upgrade MOP action, supplied with the **Default XR Upgrade** MOP and executed under the name "Capture Package Summary Configuration" during the **Default XR Upgrade** MOP's `post` stage. Figure 2 shows the same MOP action's tags when edited in the **Tags** field in the user interface.

wfTags entry from installed-summary-xr-cwm-sol :

```
"wfTags": [
  "mopActivity",
  "noExport",
]
```

Figure 4: Tag Entries in Workflow UI



MOP action data I/O requirements

Lists input sections, execution modes, mandatory input fields, and mandatory output fields for MOP actions.

Prepare MOP actions to accept input from the system and return structured outputs. This enables you to track execution success.

Organize the input data into three primary sections. Each section provides information for a network operation.

1. **CWM Solutions Job Data (app-data)**: The product framework provides data, including device details, software image selections, and required system metadata, such as `jobId`, `runId`, and `actionTimeout`.
2. **MOP Action-Specific Data (input-key)**: Enter commands or parameters using the **Job Intent** form when prompted during MOP job creation.
3. **Stash and Dependencies**: When you execute a MOP action in multiple stages (such as `pre` and `post` stages), use data from previous stages. The workflow stores output from the `pre` stage as a **stash** and automatically includes it in the input for the `post` stage so you can use results from the `pre` stage.
   - **Stash**: Results from a previous stage of the same action. For example, the system can use data captured in a `pre` stage during a `post` stage.
   - **Passthrough**: Data modified by a previous action in the MOP sequence and passed forward by the `app-data` object.

Execution scope and mode

Before processing data, identify the execution scope of the MOP action using the `executionMode` field. This determines how the device data is structured in the input.

- **Per-Device (mode 1)**: The workflow runs for each device. The input includes a single device object.
- **Fleet-Wide (mode 2)**: The workflow runs once for a group of devices. The input includes a devices object with a list of all targeted resources.

Mandatory input schema

Include these fields in the `app-data` object for every MOP action:

- **`jobId`** or **`runId`** : Unique identifiers for execution tracking within CWM.
- **`actionTimeout`** : The workflow's system-calculated timeout value.
- **`executionMode`** : Integer (1 for Per-Device, 2 for Fleet-Wide).
- **`stage`** : The current MOP stage (`pre`, `distribute`, `activate`, `commit`, or `post`).

Mandatory output schema

Return a JSON object with these required fields to ensure correct processing and display of execution results in the Job Details UI:

- **`status`** : The UI uses this value for status reporting. Supported values are success, failed, warning, or terminated.
- **`message`** : Enter a user-readable description of the action outcome.
- **`processedResource`** : List the device UUIDs or resources handled by the action.
- **`app-data`** : Return the modified or original app-data object to continue the workflow for further MOP actions.

MOP action data I/O examples

Provides JSON examples for MOP action input, output, stash, and framework-provided data.

These examples show MOP action input and output structures.

Example 1: output structure

```
{
  "status": <status>,
  "message": <description_of_outcome>,
  "app-data": { ... },
  "processedResource": [<device_uuid>],
  "stash": { ... }
}
```

In the example 2, the MOP action is a disk-space `pre` stage check. There is no workflow-specific input data required beyond data injected directly by the product framework itself: the image and device information, the name of the resource, and the MOP stage. All of this is either specified by the user when setting up the MOP job, or specified in the MOP action itself.

Example 2: MOP Action with framework-only data

```
{
  "app-data": {
    "data": {
      "imageSize": 1095680,
      "imageVersion": "7.8.2",
      "softwareImages": [
```

```
        "ncs540l-7.8.2.CSCwe80628.tar"
      ]
    },
    "device": {
      "host": "NCS540-RON-TSDN.cisco.com",
      "ip": "172.22.141.249",
      "name": "NCS540-RON-TSDN",
     "productSeries": "Cisco Network Convergence System 540L Series Routers",

      "productType": "Cisco NCS 540-24Q8L2DD-SYS-A Router",
      "softwareType": "IOS XR",
      "softwareVersion": "7.8.2",
      "uuid": "fde8fd5d-98cf-4326-bf83-ef319c82223b",
      "vendor": "Cisco Systems"
    },
    "resource": "nso-217"
  },
  "phase": "",
  "stage": "pre"
}
```

In example 3, the MOP action is a `pre` stage check on the operational status of the router, which requires as input a couple of commands specified by the user: `show version` and `show ip interface brief`. All other the details about the device on which these two commands are to be executed came from the product framework, as in Figure 2.

Example 3: MOP Action with user commands

```
{
  "app-data": {
    "data": {
      "imageSize": 1095680,
      "imageVersion": "7.8.2",
      "softwareImages": [
        "ncs540l-7.8.2.CSCwe80628.tar"
      ]
    },
    "device": {
      "host": "NCS540-RON-TSDN.cisco.com",
      "ip": "172.22.141.249",
      "name": "NCS540-RON-TSDN",
     "productSeries": "Cisco Network Convergence System 540L Series Routers",

      "productType": "Cisco NCS 540-24Q8L2DD-SYS-A Router",
      "softwareType": "IOS XR",
      "softwareVersion": "7.8.2",
      "uuid": "fde8fd5d-98cf-4326-bf83-ef319c82223b",
      "vendor": "Cisco Systems"
    },
    "resource": "nso-217"
},
"commandCapture": [
    "show version",
    "show ip interface brief"
 ],
  "phase": "",
  "stage": "pre"
}
```

Example 4 shows a MOP action executed in the `pre` stage, with the idea that the stash data saved during the `pre` stage will be provided to the workflow job of the `post` stage. In this example, `command-capture` is run in both the `pre` and `post` stages, and the details stored during the `pre` stage for the device are injected as stash data by the product framework.

Example 4: MOP Action with stashed `pre` phase output


```
{
  "app-data": {
    "data": {
      "imageSize": 1504059392,
      "imageVersion": "7.9.2",
      "softwareImages": [
        "8000-x64-7.9.2.iso"
      ]
    },
    "device": {
      "host": "PE1-UI",
      "ip": "172.20.221.183",
      "name": "PE1-UI",
      "productSeries": "Cisco 8000 Series Routers",
      "productType": "Cisco 8201 Router",
      "softwareType": "IOS XR",
      "softwareVersion": "7.9.2",
      "uuid": "a068190e-31cb-4b21-95fe-306768921a62",
      "vendor": "Cisco Systems"
    },
    "resource": "nso_resource"
  },
  "commandCapture": [
    {
      "command": "show version"
    },
    {
      "command": "show ip interface brief"
    }
  ],
  "phase": "",
  "stage": "post",
  "stash" : [
      "\r\n\rMon Aug  4 10:39:38.489 UTC\r\nCisco IOS XR Software, Version
7.9.2 LNT
\r\nCopyright (c) 2013-2023 by Cisco Systems, Inc.
\r\n\r\nBuild Information:\r\n Built By     : xxxxxxxxx
\r\n Built On     : Thu Jun 29 03:07:05 UTC 2023
\r\n Build Host   : bdb7eb8f4e82\r\n Workspace    :
/auto/srcarchive16/prod/7.9.2/8000/ws
\r\n Version      : 7.9.2
\r\n Label        : 7.9.2
\r\n\r\ncisco 8000 (VXR)\r\ncisco 8201-SYS (VXR) processor with 32GB of memory
\r\nPE1-UI uptime is 16 weeks, 4 days, 35 minutes
\r\nCisco 8201 1RU Chassis\r\n\r\nRP/0/RP0/CPU0:PE1-UI#;
command: show version",
      "\r\n\rMon Aug  4 10:39:38.473 UTC\r\n
\r\nInterface                      IP-Address      Status          Protocol
Vrf-Name
\r\nLoopback0                      150.1.1.1       Up              Up
default
\r\nFourHundredGigE0/0/0/0         10.1.13.1       Up              Up
paris
\r\nFourHundredGigE0/0/0/1         150.1.12.1      Up              Up
default
\r\nFourHundredGigE0/0/0/2         unassigned      Shutdown        Down
default
\r\nFourHundredGigE0/0/0/3         unassigned      Shutdown        Down
default
...
\r\nFourHundredGigE0/0/0/23        unassigned      Shutdown        Down
```

```
default
\r\nHundredGigE0/0/0/24            unassigned      Shutdown        Down
default
\r\nHundredGigE0/0/0/25            unassigned      Shutdown        Down
default
\r\nHundredGigE0/0/0/26            unassigned      Shutdown        Down
default
...
\r\nHundredGigE0/0/0/35            unassigned      Shutdown        Down
default
\r\nMgmtEth0/RP0/CPU0/0            192.168.122.58  Up              Up
MGMT
\r\nRP/0/RP0/CPU0:PE1-UI#; command: show ip interface brief"
    ],
}
```

In example 5, when the MOP action is executed in the `pre` and `post` stages, the stash data saved during the `pre` stage will be provided to the workflow job of the `post` stage. In this example, a package summary check is run in both the `pre` and `post` stages, and the details stored during the `pre` stage for the device are injected as stash data by the product framework.

Example 5: MOP Action with package summary

```
{
  "app-data": {
    "data": {
      "imageSize": 1504059392,
      "imageVersion": "7.9.2",
      "softwareImages": [
        "8000-x64-7.9.2.iso"
      ]
    },
    "device": {
      "host": "PE1-UI",
      "ip": "172.20.221.183",
      "name": "PE1-UI",
      "productSeries": "Cisco 8000 Series Routers",
      "productType": "Cisco 8201 Router",
      "softwareType": "IOS XR",
      "softwareVersion": "7.9.2",
      "uuid": "a068190e-31cb-4b21-95fe-306768921a62",
      "vendor": "Cisco Systems"
    },
    "resource": "nso_resource"
  },
  "phase": "",
  "stage": "post",
  "stash": {
    "activated": [
      {
        "Package": "xr-8000-l2mcast",
        "Version": "7.9.2v1.0.0-1"
      },
      {
        "Package": "xr-8000-mcast",
        "Version": "7.9.2v1.0.0-1"
      },
...
      {
        "Package": "xr-track",
        "Version": "7.9.2v1.0.0-1"
      },
```

```
      {
        "Package": "xr-8000-fpd",
        "Version": "7.9.2v1.0.1-1"
      }
    ],
    "committed": [
      {
        "Package": "xr-8000-l2mcast",
        "Version": "7.9.2v1.0.0-1"
      },
      {
        "Package": "xr-8000-mcast",
        "Version": "7.9.2v1.0.0-1"
      },
...
      {
        "Package": "xr-track",
        "Version": "7.9.2v1.0.0-1"
      },
      {
        "Package": "xr-8000-fpd",
        "Version": "7.9.2v1.0.1-1"
      }
    ]
  },
}
```

For examples 6 and 7, consider a simple MOP action that performs only a validation check on a device and appears to have no "stash" data output, only a status message:

Example 6: output without stash data

```
{
  "Data": {
    "message": "Device disk check passed.",
    "status": "success"
  }
}
```

The stash data is useful in two ways. First, it lets users view detailed information about each MOP action that ran during the CWM Solutions job. In the job execution view, when someone clicks on a specific MOP action, the stash shows the actual device configuration or any relevant data used at that step. This gives more context than just a message or status, so users can view exactly what data was involved in the decision.

Second, if that same MOP action is run again in a later stage (such as the `post` stage), the workflow automatically passes the `pre` stash data as input for the `post` stage. This means the workflow can re-use earlier results for post-processing, keeping things connected and efficient.

Example 7: output with stash data

```
{
  "Data": {
    "message": "Interface pre-upgrade check is successful.",
    "stash": [
      {
        "AdminDown": "0",
        "Down": "0",
        "InterfaceType": "IFT_ETHERNET",
        "Total": "1",
        "Up": "1"
      },
      {
```

```
        "AdminDown": "0",
        "Down": "0",
        "InterfaceType": "IFT_LOOPBACK",
        "Total": "1",
        "Up": "1"
      },
      {
        "AdminDown": "0",
        "Down": "0",
        "InterfaceType": "IFT_NULL",
        "Total": "1",
        "Up": "1"
      },
      {
        "AdminDown": "12",
        "Down": "0",
        "InterfaceType": "IFT_HUNDREDGE",
        "Total": "12",
        "Up": "0"
      },
      {
        "AdminDown": "22",
        "Down": "0",
        "InterfaceType": "IFT_FOURHUNDREDGE",
        "Total": "24",
        "Up": "2"
      },
      {
        "AdminDown": "34",
        "Down": "0",
        "InterfaceType": "ALL TYPES",
        "Total": "39",
        "Up": "5"
      }
    ],
    "status": "success"
  }
}
```

A MOP action should return output using the following JSON schema (the `stash` element is optional):

```
{
"status": "string",
"message": "string",
"stash": {output stash}
}
```

3 MOP Actions

Topics:

- MOP action categories

This chapter provides reference information for MOP actions, workflow schemas, input details, and output examples.

Use this chapter to review MOP action schemas, input details, output examples, runtime app-data, Cisco IOS XR and IOS XE MOP actions, Juniper MX MOP actions, and generic MOP workflows.

MOP action categories

This chapter provides reference information for MOP actions, workflow schemas, input details, and output examples.

Refer to this topic to review MOP actions, workflow schemas, input details, and output examples.

Reference overview

Table 2: MOP action reference areas

Reference area	Description
Runtime app-data	Provides app-data examples that the orchestrator sends to MOP action workflows at runtime.
Cisco IOS XR and IOS XE MOP actions	Provides action details, commands, stages, schemas, input details, and output examples for Cisco device workflows.
Juniper MX MOP actions	Provides action details, RPC calls, schemas, input examples, and output examples for Juniper MX workflows.
Generic MOP workflows	Provides reusable workflow details, schemas, and input/output examples for common MOP workflow patterns.

Related topics

Use these topics to review related MOP action reference information.

- Runtime app-data for MOP actions
- Cisco IOS XR MOP actions
- Cisco IOS XE MOP actions
- Juniper MX MOP actions
- Generic MOP workflows

Runtime app-data for MOP actions

The orchestrator provides app-data as input to each MOP action workflow at runtime.

Refer to this topic to review runtime app-data examples for MOP action workflows.

Cisco IOS XR app-data

The orchestrator provides the following app-data as input to each IOS XR workflow.

```
{
"app-data": {
"actionTimeout": "PT15M",
"data": {
"imageSize": 163840,
"imageVersion": "24.4.2",
"softwareImages": [
"8000-24.4.2.CSCwq11442.tar"
]
},
```

```
"device": {
"customData": null,
"host": "spitfire248.cisco.com",
"ip": "172.22.141.248",
"label": "spitfire248.cisco.com",
"name": "spit-248",
"productSeries": "Cisco 8000 Series Routers",
"productType": "Cisco 8201 Router",
"resource": "cwm.sol.system.NSO-243",
"softwareType": "IOS XR",
"softwareVersion": "24.4.2",
"uuid": "95dcdf88-8097-4b75-a5c1-d4d4f8599827",
"vendor": "Cisco Systems"
},
"executionId": 10,
"executionMode": 1,
"jobName": "XR-spit248",
"phase": "",
"resource": "cwm.sol.system.NSO-243",
"stage": "pre"
}
}
```


Cisco IOS XE app-data

The orchestrator provides the following app-data as input to each IOS XE workflow.


```
{
"app-data": {
"actionTimeout": "PT15M",
"data": {
"imageSize": 175093760,
"imageVersion": "17.16.01",
"softwareImages": [
"cat9k_iosxe.17.16.01.SPA.bin"
]
},
"device": {
"customData": null,
"host": "cat9300-153",
"label": "catalyst9300",
"ip": "192.168.66.153",
"name": "catalyst9300",
"productSeries": "Cisco Catalyst 9300 Series Switches",
"resource": "cwm.sol.system.NSO-243",
"softwareType": "IOS XE",
"softwareVersion": "17.11.2",
"uuid": "6cb6ff1e-f266-4d34-8db8-ff12d34ee1c7",
"vendor": "Cisco Systems"
},
"executionId": 11,
"executionMode": 1,
"jobName": "Catalyst 9300 Upgrade - 17.16.01",
"phase": "",
"resource": "cwm.sol.system.NSO-243",
"stage": "pre"
}
}
```

Juniper MX app-data

The orchestrator provides the following app-data as input to each Juniper MX workflow.


```
{
"app-data": {
"actionTimeout": "PT15M",
"data": {
"imageSize": 1487,
"imageVersion": "21.1R3.11",
"repository":
"M664W5Uy+CRcIlJesLBdokdzDjn4jEbgMIdb5BSx2WI17EhGIvz40QAMQLVOSucAsARyQpD9HFgf/lwqNqm32HqpcrQ3ZwuXAIUMa0w+4zWh8fSyZ4LtIQ==",
"softwareImages": [
"junos-install-mx-x86-64-21.1R3.11.tgz"
]
},
"device": {
"customData": null,
"host": "MX960-2",
"ip": "172.20.160.24",
"label": "MX960-2",
"name": "junos-24",
"productSeries": "Juniper Networks Inc MX Series",
"productType": "juniperMX960",
"resource": "cwm.sol.system.NSO-243",
"softwareType": "Junos",
"softwareVersion": "21.1R3.11",
"uuid": "098c2715-aba3-48cd-8d93-f1233ced2e0a",
"vendor": "Juniper Networks"
},
"executionId": 13,
"executionMode": 1,
"jobName": "test-ver-check2",
"phase": "",
"resource": "cwm.sol.system.NSO-243",
"stage": "activate"
}
}
```


Parent topic

MOP action reference overview

Cisco IOS XR MOP actions

Cisco IOS XR MOP actions include system health checks, interface monitoring, upgrade validation, schemas, input details, and output examples.

Refer to this topic to review Cisco IOS XR MOP action commands, usage, stages, schemas, input details, and output examples.

Overview

The table below shows complete command reference for Cisco IOS XR and XE platforms including system health checks, interface monitoring, and upgrade validation with specific success/failure thresholds for network reliability.

Table 3: Cisco IOS XR MOP actions overview

Sr. No.	Command/Check Name	Command	Usage	Stages Applicable
1	CFS Check	cfs check	Validates CFS integrity and configuration database consistency; warns if any CFS status line is not OK.	Pre Post
2	Disk Space Check	show media	Validates storage capacity across harddisk (>2x imgSize GB free), rootfs (<5% used), and disk0 (<5% used); fails if any threshold exceeded.	Distribute
3	Interface Summary	show interfaces summary	Validates interface availability and operational status by type (Total/Up/Down/AdminDown); fails if no interfaces found.	Pre Post
4	No Redundancy Check	show platform	Verifies device is configured for single-node operation without standby components; fails if multiple Active/Standby nodes detected.	Pre
5	Software Version Check	show version	Validates current software version matches target release; fails on version mismatch; warns if version found only in image filename.	Activate
6	No Install Operation Check	show install request	Verifies no active install operations are running so device is ready for new install requests; fails if operations detected.	Pre

Sr. No.	Command/Check Name	Command	Usage	Stages Applicable
7	Installed Summary Active	show install active summary	Verifies active package consistency against expected state; fails if package verification fails; warns on package differences.	Pre Post
8	Installed Summary Committed	show install committed summary	Verifies committed package consistency against expected state for Post-restart stability; fails if verification fails; warns on differences.	Pre Post
9	Route Summary	show route summary	Captures routing table summary for Pre/Post comparison; fails if no routes found or retrieval fails; warns on route differences.	Pre Post
11	Package Summary	show install active summary + show install committed summary	Compares active vs committed packages using dual command execution; fails if comparison fails; warns on package differences.	Pre
12	Log Analyzer	show logging last [n]	Analyzes recent system logs with configurable depth; fails if severity 0/1 logs detected; warns on severity 2 logs.	Pre Post
13	Startup Config Check	show configuration failed startup	Validates startup configuration integrity; fails if startup config errors detected; warns on config inconsistencies.	Pre Activate
14	Platform Check/ Check Nodes	show platform	Validates platform component health across all nodes (line cards/RPs/fabric); fails if components are offline/failed; warns on platform warnings.	Pre Post

Sr. No.	Command/Check Name	Command	Usage	Stages Applicable
15	Upgrade Matrix Check	show install upgrade-matrix iso [image]	Validates image compatibility with current hardware/software; fails on incorrect file extension or multiple ISOs; warns if upgrade restrictions exist.	Distribute
16	Install Commit	EMS commit operations	Executes software commit via EMS with SSH validation and 90-minute timeout; fails if device offline/host keys fail or commit fails.	Commit
17	Ping Check	ping [address] [options]	Tests end-to-end network reachability with VRF/repeat/timeout options; fails if success rate is less than 100%.	Pre Post
18	Command Capture	User-defined commands	Executes user-defined custom commands with output capture and error handling; fails on command execution failures; warns on command warnings.	Pre Post
19	Expression Evaluation	N/A	Evaluates custom expressions and validation logic against command output; fails if expression evaluation fails; warns on expression warnings.	Pre Post
20	Install Distribute	EMS/NSO distribute operations	Distributes software images via EMS/NSO to the device; fails if distribution is unsuccessful.	Distribute

Sr. No.	Command/Check Name	Command	Usage	Stages Applicable
21	Install Activate	EMS/NSO activate operations	Activates distributed software images via EMS/NSO with ISSU support; fails if activation is unsuccessful.	Activate
22	OLBIT Toggle	ISIS configuration commands	Manages ISIS overload bit for traffic diversion during maintenance; sets OL-bit Pre-upgrade and resets Post-upgrade; fails if operation unsuccessful.	Pre Post

Parent topic

MOP action reference overview

Related topics

Use these topics to review related MOP action reference information.

Table 4: Related topics

Topic	Description
CFS Check	Review the CFS Check MOP action reference.
Disk Space Check	Review the Disk Space Check MOP action reference.
Interface Summary	Review the Interface Summary MOP action reference.
No Redundancy Check	Review the No Redundancy Check MOP action reference.
Software Version Check	Review the Software Version Check MOP action reference.
No Install Operation Check	Review the No Install Operation Check MOP action reference.
Installed Summary Active	Review the Installed Summary Active MOP action reference.
Installed Summary Committed	Review the Installed Summary Committed MOP action reference.
Route Summary	Review the Route Summary MOP action reference.
Package Summary	Review the Package Summary MOP action reference.
Log Analyzer	Review the Log Analyzer MOP action reference.
Startup Config Check	Review the Startup Config Check MOP action reference.
Platform Check/ Check Nodes	Review the Platform Check/ Check Nodes MOP action reference.
Upgrade Matrix Check	Review the Upgrade Matrix Check MOP action reference.

Topic	Description
Install Commit	Review the Install Commit MOP action reference.
Ping Check	Review the Ping Check MOP action reference.
Command Capture	Review the Command Capture MOP action reference.
Expression Evaluation	Review the Expression Evaluation MOP action reference.
Install Distribute	Review the Install Distribute MOP action reference.
Install Activate	Review the Install Activate MOP action reference.
OLBIT Toggle	Review the OLBIT Toggle MOP action reference.

Cisco IOS XR CFS Check MOP Action

Validates CFS integrity and configuration database consistency; warns if any CFS status line is not OK.

Refer to this topic to review the schema, input details, and output examples for CFS Check.

Action details
Table 5: CFS Check details

Field	Value
Sr. No.	1
Device type	IOS XR
Command/Check Name	CFS Check
Command	cfs check
Usage	Validates CFS integrity and configuration database consistency; warns if any CFS status line is not OK.
Stages Applicable	Pre Post

Schema

Configurable schema not applicable.

Input

The orchestrator sends the app-data as mentioned above to each workflow at runtime.

Output

```
{
"Data": {
"message": "device cfs-check passed",
"status": "success"
```

```
}
}
```

Parent topic

Cisco IOS XR MOP actions

Cisco IOS XE Disk Space Check MOP Action

Validates bootflash/flash storage capacity to ensure adequate space (Less than 2x imgSize GB free) for software upgrade; fails if free space greater than is equal to 2x imgSize GB.

Refer to this topic to review the schema, input details, and output examples for Disk Space Check.

Action details
Table 6: Disk Space Check details

Field	Value
Sr. No.	1
Device Family	IOS XE
Command/Check Name	Disk Space Check
Command	show file systems
Usage	Validates bootflash/flash storage capacity to ensure adequate space (>2x imgSize GB free) for software upgrade; fails if free space <= 2x imgSize GB.
Stages Applicable	pre post

Schema

Configurable schema not applicable.

Input

The orchestrator sends the app-data as mentioned above to each workflow at runtime.

Output

```
{
"Data": {
"message": "Device disk check passed.",
"status": "success"
}
}
```

Parent topic

Cisco IOS XE MOP actions

Cisco IOS XR Interface Summary MOP Action

Validates interface availability and operational status by type (Total/Up/Down/AdminDown); fails if no interfaces found.

Refer to this topic to review the schema, input details, and output examples for Interface Summary.

Action details
Table 7: Interface Summary details

Field	Value
Sr. No.	3
Device type	IOS XR
Command/Check Name	Interface Summary
Command	show interfaces summary
Usage	Validates interface availability and operational status by type (Total/Up/Down/AdminDown); fails if no interfaces found.
Stages Applicable	Pre Post

Schema

Configurable schema not applicable.

Input

The orchestrator sends the app-data as mentioned above to each workflow at runtime.

Output


```
{
"Data": {
"message": "Interface pre-upgrade check is successful.",
"stash": [
{
"AdminDown": "0",
"Down": "0",
"InterfaceType": "IFT_ETHERNET",
"Total": "1",
"Up": "1"
},
{
"AdminDown": "0",
"Down": "0",
"InterfaceType": "IFT_LOOPBACK",
"Total": "1",
"Up": "1"
},
{
"AdminDown": "0",
"Down": "0",
"InterfaceType": "IFT_NULL",
```

```
"Total": "1",
"Up": "1"
},
{
"AdminDown": "12",
"Down": "0",
"InterfaceType": "IFT_HUNDREDGE",
"Total": "12",
"Up": "0"
},
{
"AdminDown": "22",
"Down": "0",
"InterfaceType": "IFT_FOURHUNDREDGE",
"Total": "24",
"Up": "2"
},
{
"AdminDown": "34",
"Down": "0",
"InterfaceType": "ALL TYPES",
"Total": "39",
"Up": "5"
}
],
"status": "success"
}
}
```


Parent topic

Cisco IOS XR MOP actions

Cisco IOS XR No Redundancy Check MOP Action

Verifies device is configured for single-node operation without standby components; fails if multiple Active/Standby nodes detected.

Refer to this topic to review the schema, input details, and output examples for No Redundancy Check.

Action details
Table 8: No Redundancy Check details

Field	Value
Sr. No.	4
Device type	IOS XR
Command/Check Name	No Redundancy Check
Command	show platform
Usage	Verifies device is configured for single-node operation without standby components; fails if multiple Active/Standby nodes detected.
Stages Applicable	Pre

Schema

Configurable schema not applicable.

Input

The orchestrator sends the app-data as mentioned above to each workflow at runtime.

Output

```
{
"Data": {
"message": "Device redundancy check passed.",
"status": "success"
}
}
```

Parent topic

Cisco IOS XR MOP actions

Cisco IOS XR Software Version Check MOP Action

Validates current software version matches target release; fails on version mismatch; warns if version found only in image filename.

Refer to this topic to review the schema, input details, and output examples for Software Version Check.

Action details

Table 9: Software Version Check details

Field	Value
Sr. No.	5
Device type	IOS XR
Command/Check Name	Software Version Check
Command	show version
Usage	Validates current software version matches target release; fails on version mismatch; warns if version found only in image filename.
Stages Applicable	Activate

Schema

Configurable schema not applicable.

Input

The orchestrator sends the app-data as mentioned above to each workflow at runtime.

Output

```
{
"Data": {
"message": "Version check completed successfully",
"status": "success"
}
}
```

Parent topic

Cisco IOS XR MOP actions

Cisco IOS XR No Install Operation Check MOP Action

Verifies no active install operations are running so device is ready for new install requests; fails if operations detected.

Refer to this topic to review the schema, input details, and output examples for No Install Operation Check.

Action details
Table 10: No Install Operation Check details

Field	Value
Sr. No.	6
Device type	IOS XR
Command/Check Name	No Install Operation Check
Command	show install request
Usage	Verifies no active install operations are running so device is ready for new install requests; fails if operations detected.
Stages Applicable	Pre

Schema

Configurable schema not applicable.

Input

The orchestrator sends the app-data as mentioned above to each workflow at runtime.

Output

```
{
"Data": {
"message": "device is ready for install request ",
"status": "success"
}
}
```

Parent topic

Cisco IOS XR MOP actions

Cisco IOS XR Installed Summary Active MOP Action

Verifies active package consistency against expected state; fails if package verification fails; warns on package differences.

Refer to this topic to review the schema, input details, and output examples for Installed Summary Active.

Action details
Table 11: Installed Summary Active details

Field	Value
Sr. No.	7
Device type	IOS XR
Command/Check Name	Installed Summary Active
Command	show install active summary
Usage	Verifies active package consistency against expected state; fails if package verification fails; warns on package differences.
Stages Applicable	Pre Post

Schema

Configurable schema not applicable.

Input

The orchestrator sends the app-data as mentioned above to each workflow at runtime.

Output


```
{
"Data": {
"message": "Installed summary check completed successfully. Stash would be
generated.",
"stash": {
"activated": [
{
"Package": "xr-8000-l2mcast",
"Version": "7.8.2v1.0.0-1"
},
{
"Package": "xr-8000-mcast",
"Version": "7.8.2v1.0.0-1"
},
{
"Package": "xr-8000-netflow",
"Version": "7.8.2v1.0.0-1"
},
{
"Package": "xr-bgp",
"Version": "7.8.2v1.0.0-1"
},
```

```
{
"Package": "xr-ipsla",
"Version": "7.8.2v1.0.0-1"
},
{
"Package": "xr-is-is",
"Version": "7.8.2v1.0.0-1"
},
{
"Package": "xr-lldp",
"Version": "7.8.2v1.0.0-1"
},
{
"Package": "xr-mcast",
"Version": "7.8.2v1.0.0-1"
},
{
"Package": "xr-mpls-oam",
"Version": "7.8.2v1.0.0-1"
},
{
"Package": "xr-netflow",
"Version": "7.8.2v1.0.0-1"
},
{
"Package": "xr-ospf",
"Version": "7.8.2v1.0.0-1"
},
{
"Package": "xr-perf-meas",
"Version": "7.8.2v1.0.0-1"
},
{
"Package": "xr-perfmgmt",
"Version": "7.8.2v1.0.0-1"
},
{
"Package": "xr-track",
"Version": "7.8.2v1.0.0-1"
}
],
"committed": [
{
"Package": "xr-8000-l2mcast",
"Version": "7.8.2v1.0.0-1"
},
{
"Package": "xr-8000-mcast",
"Version": "7.8.2v1.0.0-1"
},
{
"Package": "xr-8000-netflow",
"Version": "7.8.2v1.0.0-1"
},
{
"Package": "xr-bgp",
"Version": "7.8.2v1.0.0-1"
},
{
"Package": "xr-ipsla",
"Version": "7.8.2v1.0.0-1"
},
```

```
{
"Package": "xr-is-is",
"Version": "7.8.2v1.0.0-1"
},
{
"Package": "xr-lldp",
"Version": "7.8.2v1.0.0-1"
},
{
"Package": "xr-mcast",
"Version": "7.8.2v1.0.0-1"
},
{
"Package": "xr-mpls-oam",
"Version": "7.8.2v1.0.0-1"
},
{
"Package": "xr-netflow",
"Version": "7.8.2v1.0.0-1"
},
{
"Package": "xr-ospf",
"Version": "7.8.2v1.0.0-1"
},
{
"Package": "xr-perf-meas",
"Version": "7.8.2v1.0.0-1"
},
{
"Package": "xr-perfmgmt",
"Version": "7.8.2v1.0.0-1"
},
{
"Package": "xr-track",
"Version": "7.8.2v1.0.0-1"
}
]
},
"status": "success"
}
}
```


Parent topic

Cisco IOS XR MOP actions

Cisco IOS XR Installed Summary Committed MOP Action

Verifies committed package consistency against expected state for Post-restart stability; fails if verification fails; warns on differences.

Refer to this topic to review the schema and input details for Installed Summary Committed.

Action details
Table 12: Installed Summary Committed details

Field	Value
Sr. No.	8
Device type	IOS XR

Field	Value
Command/Check Name	Installed Summary Committed
Command	show install committed summary
Usage	Verifies committed package consistency against expected state for Post-restart stability; fails if verification fails; warns on differences.
Stages Applicable	Pre Post

Schema

Configurable schema not applicable.

Input

The orchestrator sends the app-data as mentioned above to each workflow at runtime.

Parent topic

Cisco IOS XR MOP actions

Cisco IOS XR Route Summary MOP Action

Captures routing table summary for Pre/Post comparison; fails if no routes found or retrieval fails; warns on route differences.

Refer to this topic to review the schema, input details, and output examples for Route Summary.

Action details
Table 13: Route Summary details

Field	Value
Sr. No.	9
Device type	IOS XR
Command/Check Name	Route Summary
Command	show route summary
Usage	Captures routing table summary for Pre/Post comparison; fails if no routes found or retrieval fails; warns on route differences.
Stages Applicable	Pre Post

Schema

Configurable schema not applicable.

Input

The orchestrator sends the app-data as mentioned above to each workflow at runtime.

Output


```
{
"Data": {
"message": "Route summary is captured.",
"stash": {
"GRT": [
{
"Backup": "0",
"Deleted": "0",
"Memory": "0",
"RouteSource": "application fib_mgr",
"Routes": "0"
},
{
"Backup": "1",
"Deleted": "0",
"Memory": "432",
"RouteSource": "connected",
"Routes": "1"
},
{
"Backup": "0",
"Deleted": "0",
"Memory": "432",
"RouteSource": "local",
"Routes": "2"
},
{
"Backup": "0",
"Deleted": "0",
"Memory": "0",
"RouteSource": "ospf paris",
"Routes": "0"
},
{
"Backup": "0",
"Deleted": "0",
"Memory": "0",
"RouteSource": "vxlan",
"Routes": "0"
},
{
"Backup": "0",
"Deleted": "0",
"Memory": "0",
"RouteSource": "dagr",
"Routes": "0"
},
{
"Backup": "0",
"Deleted": "0",
"Memory": "0",
"RouteSource": "isis core",
"Routes": "0"
}
],
"VRF": [
```

```
{
"Backup": "0",
"Deleted": "0",
"Memory": "216",
"RouteSource": "local",
"Routes": "1",
"VRF": "MGMT"
},
{
"Backup": "0",
"Deleted": "0",
"Memory": "216",
"RouteSource": "static",
"Routes": "1",
"VRF": "MGMT"
},
{
"Backup": "0",
"Deleted": "0",
"Memory": "216",
"RouteSource": "connected",
"Routes": "1",
"VRF": "MGMT"
},
{
"Backup": "0",
"Deleted": "0",
"Memory": "0",
"RouteSource": "dagr",
"Routes": "0",
"VRF": "MGMT"
},
{
"Backup": "0",
"Deleted": "0",
"Memory": "0",
"RouteSource": "ospf paris",
"Routes": "0",
"VRF": "paris"
},
{
"Backup": "0",
"Deleted": "0",
"Memory": "216",
"RouteSource": "local",
"Routes": "1",
"VRF": "paris"
},
{
"Backup": "0",
"Deleted": "0",
"Memory": "216",
"RouteSource": "connected",
"Routes": "1",
"VRF": "paris"
},
{
"Backup": "0",
"Deleted": "0",
"Memory": "0",
"RouteSource": "bgp 1",
"Routes": "0",
```

```
"VRF": "paris"
},
{
"Backup": "0",
"Deleted": "0",
"Memory": "0",
"RouteSource": "dagr",
"Routes": "0",
"VRF": "paris"
},
{
"Backup": "0",
"Deleted": "0",
"Memory": "0",
"RouteSource": "connected",
"Routes": "0",
"VRF": "**iid"
},
{
"Backup": "0",
"Deleted": "0",
"Memory": "0",
"RouteSource": "local",
"Routes": "0",
"VRF": "**iid"
}
]
},
"status": "success"
}
}
```


Parent topic

Cisco IOS XR MOP actions

Cisco IOS XR Package Summary MOP Action

Compares active vs committed packages using dual command execution; fails if comparison fails; warns on package differences.

Refer to this topic to review the schema, input details, and output examples for Package Summary.

Action details
Table 14: Package Summary details

Field	Value
Sr. No.	11
Device type	IOS XR
Command/Check Name	Package Summary
Command	show install active summary + show install committed summary
Usage	Compares active vs committed packages using dual command execution; fails if comparison fails; warns on package differences.

Field	Value
Stages Applicable	Pre

Schema

Configurable schema not applicable.

Input

The orchestrator sends the app-data as mentioned above to each workflow at runtime.

Output

```
{
"Data": {
"message": "Package summary check passed",
"status": "success"
}
}
```

Parent topic

Cisco IOS XR MOP actions

Cisco IOS XR Log Analyzer MOP Action

Analyzes recent system logs with configurable depth; fails if severity 0/1 logs detected; warns on severity 2 logs.

Refer to this topic to review the schema, input details, and output examples for Log Analyzer.

Action details
Table 15: Log Analyzer details

Field	Value
Sr. No.	12
Device type	IOS XR
Command/Check Name	Log Analyzer
Command	show logging last [n]
Usage	Analyzes recent system logs with configurable depth; fails if severity 0/1 logs detected; warns on severity 2 logs.
Stages Applicable	Pre Post

Schema

```
{
"logAnalyzer": {
```

```
"lastLines": "integer (optional, default: 30) - number of log lines to analyze"
}
}
```

Input

```
{
"logAnalyzer": {
"lastLines": 30
}
}
The orchestrator sends the app-data  as mentioned above to each workflow at
runtime.
```

Output

```
{
"Data": {
"message": "Log analyzer check passed",
"status": "success"
}
}
```

Parent topic

Cisco IOS XR MOP actions

Cisco IOS XR Startup Config Check MOP Action

Validates startup configuration integrity; fails if startup config errors detected; warns on config inconsistencies.

Refer to this topic to review the schema, input details, and output examples for Startup Config Check.

Action details
Table 16: Startup Config Check details

Field	Value
Sr. No.	13
Device type	IOS XR
Command/Check Name	Startup Config Check
Command	show configuration failed startup
Usage	Validates startup configuration integrity; fails if startup config errors detected; warns on config inconsistencies.
Stages Applicable	Pre Activate

Schema

Configurable schema not applicable.

Input

The orchestrator sends the app-data as mentioned above to each workflow at runtime.

Output

```
{
"Data": {
"message": "device startup-config-check passed",
"status": "success"
}
}
```

Parent topic

Cisco IOS XR MOP actions

Cisco IOS XR Platform Check/ Check Nodes MOP Action

Validates platform component health across all nodes (line cards/RPs/fabric); fails if components are offline/failed; warns on platform warnings.

Refer to this topic to review the schema, input details, and output examples for Platform Check/ Check Nodes.

Action details

Table 17: Platform Check/ Check Nodes details

Field	Value
Sr. No.	14
Device type	IOS XR
Command/Check Name	Platform Check/ Check Nodes
Command	show platform
Usage	Validates platform component health across all nodes (line cards/RPs/fabric); fails if components are offline/failed; warns on platform warnings.
Stages Applicable	Pre Post

Schema

Configurable schema not applicable.

Input

The orchestrator sends the app-data as mentioned above to each workflow at runtime.

Output

```
{
"Data": {
```

```
"message": "Device nodes check passed.",
"status": "success"
}
}
```

Parent topic

Cisco IOS XR MOP actions

Cisco IOS XR Upgrade Matrix Check MOP Action

Validates image compatibility with current hardware/software; fails on incorrect file extension or multiple ISOs; warns if upgrade restrictions exist.

Refer to this topic to review the schema, input details, and output examples for Upgrade Matrix Check.

Action details
Table 18: Upgrade Matrix Check details

Field	Value
Sr. No.	15
Device type	IOS XR
Command/Check Name	Upgrade Matrix Check
Command	show install upgrade-matrix iso [image]
Usage	Validates image compatibility with current hardware/software; fails on incorrect file extension or multiple ISOs; warns if upgrade restrictions exist.
Stages Applicable	Distribute

Schema

Configurable schema not applicable.

Input

The orchestrator sends the app-data as mentioned above to each workflow at runtime.

Output

```
{
"Data": {
"message": "Image upgrade is supported.",
"status": "success"
}
}
```

Parent topic

Cisco IOS XR MOP actions

Cisco IOS XR Install Commit MOP Action

Executes software commit via EMS with SSH validation and 90-minute timeout; fails if device offline/host keys fail or commit fails.

Refer to this topic to review the schema, input details, and output examples for Install Commit.

Action details
Table 19: Install Commit details

Field	Value
Sr. No.	16
Device type	IOS XR
Command/Check Name	Install Commit
Command	EMS commit operations
Usage	Executes software commit via EMS with SSH validation and 90-minute timeout; fails if device offline/host keys fail or commit fails.
Stages Applicable	Commit

Schema

Configurable schema not applicable.

Input

The orchestrator sends the app-data as mentioned above to each workflow at runtime.

Output

```
{
"Data": {
"message": "Commit is successful.",
"status": "success"
}
}
```

Parent topic

Cisco IOS XR MOP actions

Cisco IOS XR Ping Check MOP Action

Tests end-to-end network reachability with VRF/repeat/timeout options; fails if success rate is less than 100%.

Refer to this topic to review the schema, input details, and output examples for Ping Check.

Action details
Table 20: Ping Check details

Field	Value
Sr. No.	17
Device type	IOS XR
Command/Check Name	Ping Check
Command	ping [address] [options]
Usage	Tests end-to-end network reachability with VRF/repeat/timeout options; fails if success rate is less than 100%.
Stages Applicable	Pre Post

Schema

```
{
"ping": {
"address": "string - target IP address (required)",
"vrf": "string (optional) - VRF name",
"repeat": "integer (optional) - repeat count (1-1000)",
"timeout": "integer (optional) - timeout in seconds",
"verbose": "boolean (optional) - verbose output (XR/Cat8kv only)"
}
}
```

Input

```
{
"ping": {
"address": "10.195.73.40",
"repeat": 2,
"timeout": 5,
"verbose": true,
"vrf": "MGMT"
}
}
```

Output

```
{
"message": "Ping check passed.",
"status": "success"
}
```

Parent topic

Cisco IOS XR MOP actions

Cisco IOS XR Command Capture MOP Action

Executes user-defined custom commands with output capture and error handling; fails on command execution failures; warns on command warnings.

Refer to this topic to review the schema, input details, and output examples for Command Capture.

Action details
Table 21: Command Capture details

Field	Value
Sr. No.	18
Device type	IOS XR
Command/Check Name	Command Capture
Command	User-defined commands
Usage	Executes user-defined custom commands with output capture and error handling; fails on command execution failures; warns on command warnings.
Stages Applicable	Pre Post

Schema

```
{
"commandCapture": (required)
[
{
"command": string
}
],
"deviceSelection": {
"jsonPath": "string (optional) - JSON path expression for device filtering"
}
}
```

Input

```
{
"commandCapture": [
{
"command": "show version"
},
{
"command": "show install log"
}
]
}
```

Output


```
{
"Data": {
"message": "captured output for 2 command(s)",
"stash": [
"\r\n\rFri Jan 23 17:59:45.361 UTC\r\nCisco IOS XR Software, Version 7.8.2
LNT\r\nCopyright (c) 2013-2023 by Cisco Systems, Inc.\r\n\r\nBuild
Information:\r\n Built By     : ingunawa\r\n Built On     : Wed Mar 15 16:45:19
 UTC 2023\r\n Build Host   : iox-ucs-060\r\n Workspace    :
/auto/srcarchive13/prod/7.8.2/8000/ws\r\n Version      : 7.8.2\r\n Label
   : 7.8.2\r\n\r\ncisco 8000 (VXR)\r\ncisco 8201-SYS (VXR) processor with 32GB
 of memory\r\nPE1 uptime is 15 weeks, 3 days, 13 hours, 2 minutes\r\nCisco
8201 1RU Chassis\r\n\r\nRP/0/RP0/CPU0:PE1#; command: show version",
"\r\n\rFri Jan 23 17:59:44.618 UTC\r\n2026-01-21 14:52:02 UTC    Transaction
195 started\r\n2026-01-21 14:53:38 UTC      Replace (Success)\r\n2026-01-21
14:53:38 UTC        /harddisk:/8000-x64-7.8.2.iso\r\n2026-01-21 14:54:18 UTC
     Atomic change 195.1 successfully applied by process restart\r\n2026-01-21
 14:58:38 UTC    Transaction 195 completed without error\r\nRP/0/RP0/CPU0:PE1#;
 command: show install log"
],
"status": "success"
}
}
```


Parent topic

Cisco IOS XR MOP actions

Cisco IOS XR Expression Evaluation MOP Action

Evaluates custom expressions and validation logic against command output; fails if expression evaluation fails; warns on expression warnings.

Refer to this topic to review the schema, input details, and output examples for Expression Evaluation.

Action details
Table 22: Expression Evaluation details

Field	Value
Sr. No.	19
Device type	IOS XR
Command/Check Name	Expression Evaluation
Command	N/A
Usage	Evaluates custom expressions and validation logic against command output; fails if expression evaluation fails; warns on expression warnings.
Stages Applicable	Pre Post

Schema

```
{
"expressions": (required)[
{
"condition": "string",
"message": "string",
"status": "string"
}
]
}
```

Input

```
{
"expressions": [
{
"condition": "(.\"app-data\".device.vendor | contains(\"Cisco\"))",
"message": "Device check passed successfully",
"status": "success"
}
]
}
```

Output

```
{
"Data": {
"message": "Device check passed successfullytrue",
"status": "success"
}
}
```

Parent topic

Cisco IOS XR MOP actions

Cisco IOS XR Install Distribute MOP Action

Distributes software images via EMS/NSO to the device; fails if distribution is unsuccessful.

Refer to this topic to review the schema, input details, and output examples for Install Distribute.

Action details
Table 23: Install Distribute details

Field	Value
Sr. No.	20
Device type	IOS XR
Command/Check Name	Install Distribute
Command	EMS/NSO distribute operations
Usage	Distributes software images via EMS/NSO to the device; fails if distribution is unsuccessful.

Field	Value
Stages Applicable	Distribute

Schema

```
{
"distribute": {
"fpd": "boolean (optional, default: false) - FPD upgrade",
"vrfName": "string (optional) - VRF name",
"location": "string (optional) - storage location (harddisk:)",
"isISSUUpgrade": "boolean (optional, default: false) - ISSU upgrade mode",
"swimPreference": {
"copyByServer": "boolean (optional) - server-side copy",
"imageTransferProtocolOrder": "array (optional) - list of { protocol: string
}"
}
}
}
```

Input

```
"distribute": {
"fpd": false,
"isISSUUpgrade": false,
"swimPreference": {
"copyByServer": false,
"imageTransferProtocolOrder": [
{
"protocol": "ftp"
},
{
"protocol": "scp"
},
{
"protocol": "sftp"
}
]
}
}
```

Output

```
{
"Data": {
"message": "Distribution of 8000-x64-7.8.2.iso is successful.",
"status": "success"
}
}
```

Parent topic

Cisco IOS XR MOP actions

Cisco IOS XR Install Activate MOP Action

Activates distributed software images via EMS/NSO with ISSU support; fails if activation is unsuccessful.

Refer to this topic to review the schema, input details, and output examples for Install Activate.

Action details
Table 24: Install Activate details

Field	Value
Sr. No.	21
Device type	IOS XR
Command/Check Name	Install Activate
Command	EMS/NSO activate operations
Usage	Activates distributed software images via EMS/NSO with ISSU support; fails if activation is unsuccessful.
Stages Applicable	Activate

Schema

```
{
"activate": {
"fpd": "boolean (optional, default: false) - FPD upgrade",
"location": "string (optional) - storage location",
"isISSUUpgrade": "boolean (optional, default: false) - ISSU upgrade mode",
"configuredRepo": "string (optional) - configured repository override"
}
}
```

Input

```
{
"activate": {
"fpd": false,
"isISSUUpgrade": false,
"location": "harddisk"
}
}
```

Output

```
{
"Data": {
"message": "Activation of 8000-x64-7.8.2.iso is successful. SSH key has been
fetched successfully. Device has been synced successfully.",
"status": "success"
}
}
```

Parent topic

Cisco IOS XR MOP actions

Cisco IOS XR OLBIT Toggle MOP Action

Manages ISIS overload bit for traffic diversion during maintenance; sets OL-bit Pre-upgrade and resets Post-upgrade; fails if operation unsuccessful.

Refer to this topic to review the schema, input details, and output examples for OLBIT Toggle.

Action details
Table 25: OLBIT Toggle details

Field	Value
Sr. No.	22
Device type	IOS XR
Command/Check Name	OLBIT Toggle
Command	ISIS configuration commands
Usage	Manages ISIS overload bit for traffic diversion during maintenance; sets OL-bit Pre-upgrade and resets Post-upgrade; fails if operation unsuccessful.
Stages Applicable	Pre Post

Schema

```
{
"olbitToggle": {
"instance": "string (required) - ISIS instance name"
}
}
```

Input

```
{
"olbitToggle": {
"instance": "core"
}
}
```

Output

```
{
"Data": {
"message": "isis ol bit has been set by /golden-config/application
isis-set-VXR1",
"stash": [
"core"
],
"status": "success"
}
}
```

Parent topic

Cisco IOS XR MOP actions

Cisco IOS XE MOP actions

Cisco IOS XE MOP actions include system health checks, interface monitoring, upgrade validation, schemas, input details, and output examples.

Refer to this topic to review Cisco IOS XE MOP action commands, usage, stages, schemas, input details, and output examples.

Overview

The table below shows complete command reference for Cisco IOS XR and XE platforms including system health checks, interface monitoring, and upgrade validation with specific success/failure thresholds for network reliability.

Table 26: Cisco IOS XE MOP actions overview

Sr. No.	Command/Check Name	Command	Usage	Stages Applicable
1	Disk Space Check	show file systems	Validates bootflash/flash storage capacity to ensure adequate space (>2x imgSize GB free) for software upgrade; fails if free space <= 2x imgSize GB.	pre post
2	Interface Summary	show ip interface brief	Validates interface availability and operational status across all IP-enabled interfaces; fails if no interface data is retrieved.	pre post
3	No Redundancy Check	show platform	Verifies device is configured for single-node operation without standby components; fails if multiple Active/Standby nodes detected.	pre post
4	Software Version Check	show version	Validates current software version matches target release; fails on version mismatch; warns if version found only in image filename.	pre post

Sr. No.	Command/Check Name	Command	Usage	Stages Applicable
5	Installed Summary	show install summary	Verifies all installed packages are activated and committed (C status); fails on package inconsistency.	pre post
6	Route Summary	show ip route summary	Captures IP routing table summary for pre/post comparison; fails if no routes found or retrieval fails; warns on route differences.	pre post
7	Log Analyzer	show logging	Analyzes system logs for critical errors; fails if severity 0/1/2 messages detected; warns on warning-level issues.	pre post
8	Ping Check	ping [address] [options]	Tests end-to-end network reachability with VRF/repeat/timeout options; fails if success rate is less than 100%.	pre post
9	Command Capture	User-defined commands	Executes user-defined custom commands with output capture; fails if command execution fails; warns on command warnings.	pre post
10	Expression Evaluation	N/A	Evaluates custom expressions and validation logic against command output; fails if expression evaluation fails; warns on expression warnings.	pre post
11	Install Distribute	EMS/NSO distribute operations	Distributes software images via EMS/NSO to the device; fails if distribution is unsuccessful.	Distribute

Sr. No.	Command/Check Name	Command	Usage	Stages Applicable
12	Install Activate	EMS/NSO activate operations	Activates distributed software images via EMS/NSO with ISSU support; fails if activation is unsuccessful.	Activate
13	OLBIT Toggle	ISIS configuration commands	Manages ISIS overload bit for traffic diversion during maintenance; sets OL-bit pre-upgrade and resets post-upgrade; fails if operation unsuccessful.	pre post

Parent topic

MOP action reference overview

Related topics

Use these topics to review related MOP action reference information.

Table 27: Related topics

Topic	Description
Disk Space Check	Review the Disk Space Check MOP action reference.
Interface Summary	Review the Interface Summary MOP action reference.
No Redundancy Check	Review the No Redundancy Check MOP action reference.
Software Version Check	Review the Software Version Check MOP action reference.
Installed Summary	Review the Installed Summary MOP action reference.
Route Summary	Review the Route Summary MOP action reference.
Log Analyzer	Review the Log Analyzer MOP action reference.
Ping Check	Review the Ping Check MOP action reference.
Command Capture	Review the Command Capture MOP action reference.
Expression Evaluation	Review the Expression Evaluation MOP action reference.
Install Distribute	Review the Install Distribute MOP action reference.
Install Activate	Review the Install Activate MOP action reference.
OLBIT Toggle	Review the OLBIT Toggle MOP action reference.

Cisco IOS XE Disk Space Check MOP Action

Validates bootflash/flash storage capacity to ensure adequate space (Less than 2x imgSize GB free) for software upgrade; fails if free space greater than is equal to 2x imgSize GB.

Refer to this topic to review the schema, input details, and output examples for Disk Space Check.

Action details
Table 28: Disk Space Check details

Field	Value
Sr. No.	1
Device Family	IOS XE
Command/Check Name	Disk Space Check
Command	show file systems
Usage	Validates bootflash/flash storage capacity to ensure adequate space (>2x imgSize GB free) for software upgrade; fails if free space <= 2x imgSize GB.
Stages Applicable	pre post

Schema

Configurable schema not applicable.

Input

The orchestrator sends the app-data as mentioned above to each workflow at runtime.

Output

```
{
"Data": {
"message": "Device disk check passed.",
"status": "success"
}
}
```

Parent topic

Cisco IOS XE MOP actions

Cisco IOS XE Interface Summary MOP Action

Validates interface availability and operational status across all IP-enabled interfaces; fails if no interface data is retrieved.

Refer to this topic to review the schema, input details, and output examples for Interface Summary.

Action details
Table 29: Interface Summary details

Field	Value
Sr. No.	2
Device Family	IOS XE
Command/Check Name	Interface Summary
Command	show ip interface brief
Usage	Validates interface availability and operational status across all IP-enabled interfaces; fails if no interface data is retrieved.
Stages Applicable	pre post

Schema

Configurable schema not applicable.

Input

The orchestrator sends the app-data as mentioned above to each workflow at runtime.

Output


```
{
"Data": {
"message": "Interface pre-upgrade check is successful.",
"stash": [
{
"AdminDown": "0",
"Down": "0",
"InterfaceType": "IFT_ETHERNET",
"Total": "1",
"Up": "1"
},
{
"AdminDown": "0",
"Down": "0",
"InterfaceType": "IFT_NULL",
"Total": "1",
"Up": "1"
},
{
"AdminDown": "0",
"Down": "1",
"InterfaceType": "IFT_TUNNEL_TE",
"Total": "1",
"Up": "0"
},
{
"AdminDown": "48",
"Down": "0",
```

```
"InterfaceType": "IFT_TENGETHERNET",
"Total": "48",
"Up": "0"
},
{
"AdminDown": "54",
"Down": "1",
"InterfaceType": "ALL TYPES",
"Total": "57",
"Up": "2"
},
{
"AdminDown": "6",
"Down": "0",
"InterfaceType": "IFT_HUNDREDGE",
"Total": "6",
"Up": "0"
}
],
"status": "success"
}
}
```


Parent topic

Cisco IOS XE MOP actions

Cisco IOS XE No Redundancy Check MOP Action

Verifies device is configured for single-node operation without standby components; fails if multiple Active/Standby nodes detected.

Refer to this topic to review the schema, input details, and output examples for No Redundancy Check.

Action details
Table 30: No Redundancy Check details

Field	Value
Sr. No.	3
Device Family	IOS XE
Command/Check Name	No Redundancy Check
Command	show platform
Usage	Verifies device is configured for single-node operation without standby components; fails if multiple Active/Standby nodes detected.
Stages Applicable	pre post

Schema

Configurable schema not applicable.

Input

The orchestrator sends the app-data as mentioned above to each workflow at runtime.

Output

```
{
"Data": {
"message": "Device redundancy check passed.",
"status": "success"
}
}
```

Parent topic

Cisco IOS XE MOP actions

Cisco IOS XE Software Version Check MOP Action

Validates current software version matches target release; fails on version mismatch; warns if version found only in image filename.

Refer to this topic to review the schema, input details, and output examples for Software Version Check.

Action details

Table 31: Software Version Check details

Field	Value
Sr. No.	4
Device Family	IOS XE
Command/Check Name	Software Version Check
Command	show version
Usage	Validates current software version matches target release; fails on version mismatch; warns if version found only in image filename.
Stages Applicable	pre post

Schema

Configurable schema not applicable.

Input

The orchestrator sends the app-data as mentioned above to each workflow at runtime.

Output

```
{
"Data": {
```

```
"message": "Version check completed successfully",
"status": "success"
}
}
```

Parent topic

Cisco IOS XE MOP actions

Cisco IOS XE Installed Summary MOP Action

Verifies all installed packages are activated and committed (C status); fails on package inconsistency.

Refer to this topic to review the schema, input details, and output examples for Installed Summary.

Action details
Table 32: Installed Summary details

Field	Value
Sr. No.	5
Device Family	IOS XE
Command/Check Name	Installed Summary
Command	show install summary
Usage	Verifies all installed packages are activated and committed (C status); fails on package inconsistency.
Stages Applicable	pre post

Schema

Configurable schema not applicable.

Input

The orchestrator sends the app-data as mentioned above to each workflow at runtime.

Output

```
{
"Data": {
"message": "Installed summary check completed successfully.",
"status": "success"
}
}
```

Parent topic

Cisco IOS XE MOP actions

Cisco IOS XE Route Summary MOP Action

Captures IP routing table summary for pre/post comparison; fails if no routes found or retrieval fails; warns on route differences.

Refer to this topic to review the schema, input details, and output examples for Route Summary.

Action details
Table 33: Route Summary details

Field	Value
Sr. No.	6
Device Family	IOS XE
Command/Check Name	Route Summary
Command	show ip route summary
Usage	Captures IP routing table summary for pre/post comparison; fails if no routes found or retrieval fails; warns on route differences.
Stages Applicable	pre post

Schema

Configurable schema not applicable.

Input

The orchestrator sends the app-data as mentioned above to each workflow at runtime.

Output


```
{
"Data": {
"message": "Route summary is captured.",
"stash": {
"RouteSummary": [
{
"Memory": "0",
"Networks": "0",
"Overhead": "0",
"Replicates": "0",
"RouteSource": "static",
"Subnets": "0"
},
{
"Memory": "312",
"Networks": "0",
"Overhead": "112",
"Replicates": "0",
"RouteSource": "connected",
"Subnets": "1"
```

```
},
{
"Memory": "704",
"Networks": "1",
"Overhead": "112",
"Replicates": "0",
"RouteSource": "Total",
"Subnets": "1"
}
]
},
"status": "success"
}
}
```


Parent topic

Cisco IOS XE MOP actions

Cisco IOS XE Log Analyzer MOP Action

Analyzes system logs for critical errors; fails if severity 0/1/2 messages detected; warns on warning-level issues.

Refer to this topic to review the schema, input details, and output examples for Log Analyzer.

Action details
Table 34: Log Analyzer details

Field	Value
Sr. No.	7
Device Family	IOS XE
Command/Check Name	Log Analyzer
Command	show logging
Usage	Analyzes system logs for critical errors; fails if severity 0/1/2 messages detected; warns on warning-level issues.
Stages Applicable	pre post

Schema


```
{
"logAnalyzer": {
"lastLines": "integer (optional, default: 30) - number of log lines to analyze"
}
}
```


Input


```
{
"logAnalyzer": {
```

```
"lastLines": 30
}
}
The orchestrator sends the app-data  as mentioned above to each workflow at
runtime.
```

Output

```
{
"Data": {
"message": "Log analyzer check passed",
"status": "success"
}
}
```

Parent topic

Cisco IOS XE MOP actions

Cisco IOS XE Ping Check MOP Action

Tests end-to-end network reachability with VRF/repeat/timeout options; fails if success rate is less than 100%.

Refer to this topic to review the schema, input details, and output examples for Ping Check.

Action details
Table 35: Ping Check details

Field	Value
Sr. No.	8
Device Family	IOS XE
Command/Check Name	Ping Check
Command	ping [address] [options]
Usage	Tests end-to-end network reachability with VRF/repeat/timeout options; fails if success rate is less than 100%.
Stages Applicable	pre post

Schema

```
{
"ping": {
"address": "string - target IP address (required)",
"vrf": "string (optional) - VRF name",
"repeat": "integer (optional) - repeat count (1-1000)",
"timeout": "integer (optional) - timeout in seconds",
"verbose": "boolean (optional) - verbose output (XR/Cat8kv only)"
}
}
```

Input

```
{
"ping": {
"address": "10.195.73.40",
"repeat": 2,
"timeout": 5,
"verbose": true,
"vrf": "MGMT"
}
}
```

Output

```
{
"message": "Ping check passed.",
"status": "success"
}
```

Parent topic

Cisco IOS XE MOP actions

Cisco IOS XE Command Capture MOP Action

Executes user-defined custom commands with output capture; fails if command execution fails; warns on command warnings.

Refer to this topic to review the schema, input details, and output examples for Command Capture.

Action details
Table 36: Command Capture details

Field	Value
Sr. No.	9
Device Family	IOS XE
Command/Check Name	Command Capture
Command	User-defined commands
Usage	Executes user-defined custom commands with output capture; fails if command execution fails; warns on command warnings.
Stages Applicable	pre post

Schema

```
{
"commandCapture": "array (required) - list of { command: string } objects",
"deviceSelection": {
"jsonPath": "string (optional) - JSON path expression for device filtering"
```

```
}
}
```

Input

```
{
"commandCapture": [
{
"command": "show version"
},
{
"command": "show ip interface brief"
}
]
}
```

Output

```
{
"Data": {
"message": "captured output for 2 command(s)",
"stash": [
"\r\nCisco IOS XE Software, Version 17.12.04\r\nCisco IOS Software [Dublin],
Catalyst L3 Switch Software (CAT9K_IOSXE), Version 17.12.4, RELEASE SOFTWARE
(fc3)\r\nTechnical Support: http://www.cisco.com/techsupport\r\nCopyright (c)
 1986-2024 by Cisco Systems, Inc.\r\nCompiled Tue 23-Jul-24 09:40 by
mcpre\r\n\r\n\r\nCisco IOS-XE software, Copyright (c) 2005-2024 by cisco
Systems, Inc.\r\nAll rights reserved.  Certain components of Cisco IOS-XE
software are\r\nlicensed under the GNU General Public License (\"GPL\") Version
 2.0.  The\r\nsoftware code licensed under GPL Version 2.0 is free software
that comes\r\nwith ABSOLUTELY NO WARRANTY.  You can redistribute and/or modify
 such\r\nGPL code under the terms of GPL Version 2.0.  For more details, see
the\r\ndocumentation or \"License Notice\" file accompanying the IOS-XE
software,\r\nor the applicable URL provided on the flyer accompanying the
IOS-XE\r\nsoftware.\r\n\r\n\r\nROM: IOS-XE ROMMON\r\nBOOTLDR: System Bootstrap,
 Version 17.12.1r, RELEASE SOFTWARE (P)\r\n\r\ncat9300-153 uptime is 15 weeks,
 5 days, 6 hours, 42 minutes\r\nUptime for this control processor is 15 weeks,
 5 days, 6 hours, 47 minutes\r\nSystem returned to ROM by Reload
Command\r\nSystem image file is \"flash:packages.conf\"\r\nLast reload reason:
 Reload Command\r\n\r\n\r\n\r\n\r\nThis product contains cryptographic features
and is subject to United\r\nStates and local country laws governing import,
export, transfer and\r\nuse. Delivery of Cisco cryptographic products does not
 imply\r\nthird-party authority to import, export, distribute or use
encryption.\r\nImporters, exporters, distributors and users are responsible
for\r\ncompliance with U.S. and local country laws. By using this product
you\r\nagree to comply with applicable laws and regulations. If you are
unable\r\nto comply with U.S. and local laws, return this product
immediately.\r\n\r\nA summary of U.S. laws governing Cisco cryptographic
products may be found
at:\r\nhttp://www.cisco.com/wwl/export/crypto/tool/stqrg.html\r\n\r\nIf you
require further assistance please contact us by sending email
to\r\nexport@cisco.com.\r\n\r\n\r\nTechnology Package License Information:
\r\n\r\n------------------------------------------------------------------\r\nTechnology-package
                                    Technology-package\r\nCurrent
            Type                       Next reboot
\r\n------------------------------------------------------------------\r\nnetwork-advantage
   \tSmart License                \t network-advantage   \r\ndna-advantage
     \tSubscription Smart License    \t dna-advantage                 \r\nAIR
 License Level: AIR DNA Advantage\r\nNext reload AIR license Level: AIR DNA
Advantage\r\n\r\n\r\nSmart Licensing Status: Smart Licensing Using
```

```
Policy\r\n\r\ncisco C9300-24UX (X86) processor with 1301943K/6147K bytes of
memory.\r\nProcessor board ID FCW2138G01H\r\n1 Virtual Ethernet interface\r\n4
 Gigabit Ethernet interfaces\r\n32 Ten Gigabit Ethernet interfaces\r\n2
TwentyFive Gigabit Ethernet interfaces\r\n2 Forty Gigabit Ethernet
interfaces\r\n2048K bytes of non-volatile configuration memory.\r\n8388608K
bytes of physical memory.\r\n1638400K bytes of Crash Files at
crashinfo:.\r\n1638400K bytes of Crash Files at crashinfo-2:.\r\n11264000K
bytes of Flash at flash:.\r\n11264000K bytes of Flash at flash-2:.\r\n7810048K
 bytes of USB Flash at usbflash0-2:.\r\n\r\nBase Ethernet MAC Address
  : f8:7b:20:53:6e:80\r\nMotherboard Assembly Number        :
73-17958-06\r\nMotherboard Serial Number          : FOC21371QAU\r\nModel
Revision Number             : A0\r\nMotherboard Revision Number        :
A0\r\nModel Number                       : C9300-24UX\r\nSystem Serial Number
             : FCW2138G01H\r\nCLEI Code Number                 :
INM3Y10BRA\r\n\r\n\r\nSwitch Ports Model              SW Version        SW
Image              Mode   \r\n------ ----- -----                ----------
   ----------            ----   \r\n*    1 41    C9300-24UX         17.12.04
        CAT9K_IOSXE           INSTALL\r\n\r\n\r\ncat9300-153#; command: show
 version",
"\r\nInterface              IP-Address      OK? Method Status
Protocol\r\nVlan1                  unassigned      YES NVRAM  administratively
 down down    \r\nGigabitEthernet0/0     192.168.66.153  YES NVRAM  up
             up      \r\nTe1/0/1               unassigned      YES unset  up
                   down    \r\nTe1/0/2               unassigned      YES
unset  down                    down    \r\nTe1/0/3               unassigned
   YES unset  down                   down    \r\nTe1/0/4
unassigned      YES unset  up                    down    \r\nTe1/0/5
       unassigned      YES unset  down                  down    \r\nTe1/0/6
              unassigned      YES unset  down                  down
\r\nTe1/0/7               unassigned      YES unset  down
down    \r\nTe1/0/8               unassigned      YES unset  down
       down    \r\nTe1/0/9               unassigned      YES unset  down
              down    \r\nTe1/0/10              unassigned      YES unset
down                   down    \r\nTe1/0/11              unassigned      YES
unset  down                    down    \r\nTe1/0/12              unassigned
   YES unset  down                   down    \r\nTe1/0/13
unassigned      YES unset  down                  down    \r\nTe1/0/14
       unassigned      YES unset  down                  down    \r\nTe1/0/15
              unassigned      YES unset  down                  down
\r\nTe1/0/16              unassigned      YES unset  down
down    \r\nTe1/0/17              unassigned      YES unset  down
       down    \r\nTe1/0/18              unassigned      YES unset  down
              down    \r\nTe1/0/19              unassigned      YES unset
down                   down    \r\nTe1/0/20              unassigned      YES
unset  down                    down    \r\nTe1/0/21              unassigned
   YES unset  down                   down    \r\nTe1/0/22
unassigned      YES unset  down                  down    \r\nTe1/0/23
       unassigned      YES unset  down                  down    \r\nTe1/0/24
              unassigned      YES unset  down                  down
\r\nGigabitEthernet1/1/1   unassigned      YES unset  down
down    \r\nGigabitEthernet1/1/2   unassigned      YES unset  down
       down    \r\nGigabitEthernet1/1/3   unassigned      YES unset  down
              down    \r\nGigabitEthernet1/1/4   unassigned      YES unset
down                   down    \r\nTe1/1/1               unassigned      YES
unset  down                    down    \r\nTe1/1/2               unassigned
   YES unset  down                   down    \r\nTe1/1/3
unassigned      YES unset  down                  down    \r\nTe1/1/4
       unassigned      YES unset  down                  down    \r\nTe1/1/5
              unassigned      YES unset  down                  down
\r\nTe1/1/6               unassigned      YES unset  down
down    \r\nTe1/1/7               unassigned      YES unset  down
       down    \r\nTe1/1/8               unassigned      YES unset  down
```

```
              down    \r\nFo1/1/1                unassigned      YES unset
down                  down    \r\nFo1/1/2                unassigned      YES
unset  down                   down    \r\nTwentyFiveGigE1/1/1    unassigned
   YES unset  down                  down    \r\nTwentyFiveGigE1/1/2
unassigned      YES unset  down                  down    \r\nAp1/0/1
       unassigned      YES unset  up                    up
\r\nPort-channel120        unassigned      YES unset  down
down    \r\nPort-channel121        unassigned      YES unset  down
       down    \r\nLoopback0              20.20.20.1      YES NVRAM  up
              up      \r\ncat9300-153#; command: show ip interface brief"
],
"status": "success"
}
}
```


Parent topic

Cisco IOS XE MOP actions

Cisco IOS XE Expression Evaluation MOP Action

Evaluates custom expressions and validation logic against command output; fails if expression evaluation fails; warns on expression warnings.

Refer to this topic to review the schema, input details, and output examples for Expression Evaluation.

Action details
Table 37: Expression Evaluation details

Field	Value
Sr. No.	10
Device Family	IOS XE
Command/Check Name	Expression Evaluation
Command	N/A
Usage	Evaluates custom expressions and validation logic against command output; fails if expression evaluation fails; warns on expression warnings.
Stages Applicable	pre post

Schema

```
{
"expressions": (required)[
{
"condition": "string",
"message": "string",
"status": "string"
}
]
}
```

Input

```
{
"expressions": [
{
"condition": "(.\"app-data\".device.vendor | contains(\"Cisco\"))",
"message": "Device check passed successfully",
"status": "success"
}
]
}
```

Output

```
{
"Data": {
"message": "Device check passed successfullytrue",
"status": "success"
}
}
```

Parent topic

Cisco IOS XE MOP actions

Cisco IOS XE Install Distribute MOP Action

Distributes software images via EMS/NSO to the device; fails if distribution is unsuccessful.

Refer to this topic to review the schema, input details, and output examples for Install Distribute.

Action details
Table 38: Install Distribute details

Field	Value
Sr. No.	11
Device Family	IOS XE
Command/Check Name	Install Distribute
Command	EMS/NSO distribute operations
Usage	Distributes software images via EMS/NSO to the device; fails if distribution is unsuccessful.
Stages Applicable	Distribute

Schema

```
{
"distribute": {
"fpd": "boolean (optional, default: false) - FPD upgrade",
"vrfName": "string (optional) - VRF name",
"location": "string (optional) - storage location (harddisk:)",
"isISSUUpgrade": "boolean (optional, default: false) - ISSU upgrade mode",
```

```
"swimPreference": {
"copyByServer": "boolean (optional) - server-side copy",
"imageTransferProtocolOrder": "array (optional) - list of { protocol: string
}"
}
}
}
```

Input

```
"distribute": {
"fpd": false,
"isISSUUpgrade": false,
"swimPreference": {
"copyByServer": false,
"imageTransferProtocolOrder": [
{
"protocol": "ftp"
},
{
"protocol": "scp"
},
{
"protocol": "sftp"
}
]
}
}
```

Output

```
{
"Data": {
"message": "Distribution of asr1000-universalk9.17.09.06.SPA.bin is
successful.",
"status": "success"
}
}
```

Parent topic

Cisco IOS XE MOP actions

Cisco IOS XE Install Activate MOP Action

Activates distributed software images via EMS/NSO with ISSU support; fails if activation is unsuccessful.

Refer to this topic to review the schema, input details, and output examples for Install Activate.

Action details
Table 39: Install Activate details

Field	Value
Sr. No.	12
Device Family	IOS XE

Field	Value
Command/Check Name	Install Activate
Command	EMS/NSO activate operations
Usage	Activates distributed software images via EMS/NSO with ISSU support; fails if activation is unsuccessful.
Stages Applicable	Activate

Schema

```
{
"activate": {
"fpd": "boolean (optional, default: false) - FPD upgrade",
"location": "string (optional) - storage location",
"isISSUUpgrade": "boolean (optional, default: false) - ISSU upgrade mode",
"configuredRepo": "string (optional) - configured repository override"
}
}
```

Input

```
{
"activate": {
"fpd": false,
"isISSUUpgrade": false
}
}
```

Output

```
{
"Data": {
"message": "Activation of asr1000-universalk9.17.09.06.SPA.bin is successful.
 SSH key has been fetched successfully. Device has been synced successfully.",
"status": "success"
}
}
```

Parent topic

Cisco IOS XE MOP actions

Cisco IOS XE OLBIT Toggle MOP Action

Manages ISIS overload bit for traffic diversion during maintenance; sets OL-bit pre-upgrade and resets post-upgrade; fails if operation unsuccessful.

Refer to this topic to review the schema, input details, and output examples for OLBIT Toggle.

Action details
Table 40: OLBIT Toggle details

Field	Value
Sr. No.	13
Device Family	IOS XE
Command/Check Name	OLBIT Toggle
Command	ISIS configuration commands
Usage	Manages ISIS overload bit for traffic diversion during maintenance; sets OL-bit pre-upgrade and resets post-upgrade; fails if operation unsuccessful.
Stages Applicable	pre post

Schema

```
{
"olbitToggle": {
"instance": "string (required) - ISIS instance name"
}
}
```

Input

```
{
"olbitToggle": {
"instance": "core"
}
}
```

Output

```
{
"Data": {
"message": "isis ol bit has been set by /golden-config/application
isis-set-VXR1",
"stash": [
"core"
],
"status": "success"
}
}
```

Parent topic

Cisco IOS XE MOP actions

Juniper MX MOP actions

Juniper MX MOP actions include pre-upgrade and post-upgrade validation commands, schemas, input details, and output examples.

Refer to this topic to review Juniper MX MOP action RPC calls, usage, schemas, input examples, and output examples.

Overview

This table shows the comprehensive pre/post upgrade validation commands for Juniper MX routers covering hardware health, routing protocols, and system performance with detailed pass/fail criteria for operational assurance.

Table 41: Juniper MX MOP actions overview

Sr. No.	MOP action Name	RPC Call	Usage
1	check device memory	rpc-get-system-memory-information	Validates system memory utilization across all routing engines. Free memory should be greater than 15%; the check fails if it drops to 15% or below.
2	check disk space	rpc-get-system-storage	Validates available disk storage for software images and upgrade files. The available blocks multiplied by 512 must exceed the required space; otherwise the check fails.
3	check alarms	rpc-get-system-alarm-information	Monitors active system alarms for critical issues. No critical alarms should be present. The check fails when critical alarms are detected and raises a warning if minor alarms exist.
4	check route engines	rpc-get-route-engine-information	Validates route engine health and control plane status. The route engine status must be 'OK'; any other status causes the check to fail.
5	check fpc status	rpc-get-fpc-information	Validates FPC/line card operational status. FPC details should match the pre-upgrade state. The check fails if execution errors occur, and raises a warning when pre/post differences are detected.

Sr. No.	MOP action Name	RPC Call	Usage
6	check interfaces	rpc-get-interface-information	Validates physical interface status and connectivity. Physical interfaces should be present and their states must match the pre-upgrade baseline. The check fails if no interface data is returned, and raises a warning when state differences are found.
7	arp summary	rpc-get-arp-table-information	Validates ARP table entries and neighbor relationships. ARP entries should be present and match the pre-upgrade state. The check fails if no entries are found, and raises a warning when entries are missing or newly added.
8	Route Summary Check	rpc-get-route-summary-information	Validates routing table summary and convergence. Route tables should match the pre-upgrade state. The check fails if no route tables are found, and raises a warning when routes have been added or removed.
9	Software Version Check	rpc-get-software-information	Validates the device is running the expected software version. The version should match the target image or be found within the image filename. The check fails if a version mismatch occurs and the version is not found in the image file.
10	Package Summary Check	rpc-get-software-information	Verifies package consistency between committed and active states. Committed and active packages should be identical. The check fails if package verification encounters errors, and raises a warning when packages have been added or removed.

Sr. No.	MOP action Name	RPC Call	Usage
11	Ping Check	rpc-ping	Tests end-to-end network reachability. The target should be reachable with zero packet loss. Any packet loss causes the check to fail.
12	distribute image	rpc-file-copy	Copies the software image to the device. The file should be transferred successfully; the check fails if the copy command returns an error.
13	activate image	New image: rpc-request-package-add	Installs a new image or activates from the rollback/installed version list. The device should reboot successfully with the new image; the check fails if the reboot does not complete.

Parent topic

MOP action reference overview

Related topics

Use these topics to review related MOP action reference information.

Table 42: Related topics

Topic	Description
check device memory	Review the check device memory MOP action reference.
check disk space	Review the check disk space MOP action reference.
check alarms	Review the check alarms MOP action reference.
check route engines	Review the check route engines MOP action reference.
check fpc status	Review the check fpc status MOP action reference.
check interfaces	Review the check interfaces MOP action reference.
arp summary	Review the arp summary MOP action reference.
Route Summary Check	Review the Route Summary Check MOP action reference.
Software Version Check	Review the Software Version Check MOP action reference.
Package Summary Check	Review the Package Summary Check MOP action reference.
Ping Check	Review the Ping Check MOP action reference.
distribute image	Review the distribute image MOP action reference.

Topic	Description
activate image	Review the activate image MOP action reference.

Juniper MX check device memory MOP Action

Validates system memory utilization across all routing engines. Free memory should be greater than 15%; the check fails if it drops to 15% or below.

Refer to this topic to review the schema, input details, and output examples for check device memory.

Action details

Table 43: check device memory details

Field	Value
Sr. No.	1
Device Family	MX Series
MOP action Name	check device memory
RPC Call	rpc-get-system-memory-information
Usage	Validates system memory utilization across all routing engines. Free memory should be greater than 15%; the check fails if it drops to 15% or below.

Input Schema

Configurable schema not applicable.

Input Example

The orchestrator sends the app-data as mentioned above to each workflow at runtime.

Output

```
{
"Data": {
"message": "Memory check passed.",
"status": "success"
}
}
```

Parent topic

Juniper MX MOP actions

Juniper MX check disk space MOP Action

Validates available disk storage for software images and upgrade files. The available blocks multiplied by 512 must exceed the required space; otherwise the check fails.

Refer to this topic to review the schema, input details, and output examples for check disk space.

Action details
Table 44: check disk space details

Field	Value
Sr. No.	2
Device Family	MX Series
MOP action Name	check disk space
RPC Call	rpc-get-system-storage
Usage	Validates available disk storage for software images and upgrade files. The available blocks multiplied by 512 must exceed the required space; otherwise the check fails.

Input Schema

Configurable schema not applicable.

Input Example

The orchestrator sends the app-data as mentioned above to each workflow at runtime.

Output

```
{
"Data": {
"message": "Device disk check passed.",
"status": "success"
}
}
```

Parent topic

Juniper MX MOP actions

Juniper MX check alarms MOP Action

Monitors active system alarms for critical issues. No critical alarms should be present. The check fails when critical alarms are detected and raises a warning if minor alarms exist.

Refer to this topic to review the schema, input details, and output examples for check alarms.

Action details
Table 45: check alarms details

Field	Value
Sr. No.	3
Device Family	MX Series
MOP action Name	check alarms
RPC Call	rpc-get-system-alarm-information

Field	Value
Usage	Monitors active system alarms for critical issues. No critical alarms should be present. The check fails when critical alarms are detected and raises a warning if minor alarms exist.

Input Schema

Configurable schema not applicable.

Input Example

The orchestrator sends the app-data as mentioned above to each workflow at runtime.

Output

```
{
"Data": {
"message": "Alarm check completed with warnings.",
"stash": [
"CB 0 Failure",
"PEM 3 Unrecognized",
"Top Fan Tray Fail"
],
"status": "warning"
}
}
```

Parent topic

Juniper MX MOP actions

Juniper MX check route engines MOP Action

Validates route engine health and control plane status. The route engine status must be 'OK'; any other status causes the check to fail.

Refer to this topic to review the schema, input details, and output examples for check route engines.

Action details
Table 46: check route engines details

Field	Value
Sr. No.	4
Device Family	MX Series
MOP action Name	check route engines
RPC Call	rpc-get-route-engine-information
Usage	Validates route engine health and control plane status. The route engine status must be 'OK'; any other status causes the check to fail.

Input Schema

Configurable schema not applicable.

Input Example

The orchestrator sends the app-data as mentioned above to each workflow at runtime.

Output

```
{
"Data": {
"message": "Route engine check passed.",
"status": "success"
}
}
```

Parent topic

Juniper MX MOP actions

Juniper MX check fpc status MOP Action

Validates FPC/line card operational status. FPC details should match the pre-upgrade state. The check fails if execution errors occur, and raises a warning when pre/post differences are detected.

Refer to this topic to review the schema, input details, and output examples for check fpc status.

Action details

Table 47: check fpc status details

Field	Value
Sr. No.	5
Device Family	MX Series
MOP action Name	check fpc status
RPC Call	rpc-get-fpc-information
Usage	Validates FPC/line card operational status. FPC details should match the pre-upgrade state. The check fails if execution errors occur, and raises a warning when pre/post differences are detected.

Input Schema

Configurable schema not applicable.

Input Example

The orchestrator sends the app-data as mentioned above to each workflow at runtime.

Output


```
{
"Data": {
"message": "fpc pre-upgrade check is successful.",
"stash": [
{
"cpu-interrupt": 0,
"cpu-total": 1,
"memory-buffer-utilization": 29,
"memory-dram-size": 1024,
"memory-heap-utilization": 17,
"slot": 10,
"state": "Online",
"temperature": "26"
},
{
"cpu-interrupt": 0,
"cpu-total": 1,
"memory-buffer-utilization": 29,
"memory-dram-size": 1024,
"memory-heap-utilization": 17,
"slot": 3,
"state": "Online",
"temperature": "26"
},
{
"slot": 0,
"state": "Empty"
},
{
"slot": 1,
"state": "Empty"
},
{
"slot": 11,
"state": "Empty"
},
{
"slot": 2,
"state": "Empty"
},
{
"slot": 4,
"state": "Empty"
},
{
"slot": 5,
"state": "Empty"
},
{
"slot": 6,
"state": "Empty"
},
{
"slot": 7,
"state": "Empty"
},
{
"slot": 8,
"state": "Empty"
},
{
```

```
"slot": 9,
"state": "Empty"
}
],
"status": "success"
}
}
```


Parent topic

Juniper MX MOP actions

Juniper MX check interfaces MOP Action

Validates physical interface status and connectivity. Physical interfaces should be present and their states must match the pre-upgrade baseline. The check fails if no interface data is returned, and raises a warning when state differences are found.

Refer to this topic to review the schema, input details, and output examples for check interfaces.

Action details
Table 48: check interfaces details

Field	Value
Sr. No.	6
Device Family	MX Series
MOP action Name	check interfaces
RPC Call	rpc-get-interface-information
Usage	Validates physical interface status and connectivity. Physical interfaces should be present and their states must match the pre-upgrade baseline. The check fails if no interface data is returned, and raises a warning when state differences are found.

Input Schema

Configurable schema not applicable.

Input Example

The orchestrator sends the app-data as mentioned above to each workflow at runtime.

Output


```
{
"Data": {
"message": "Interface pre-upgrade check is successful.",
"stash": [
{
"admin-status": "up",
"clocking": "1",
"if-config-flags": {
```

```
"iff-point-to-point": [
null
],
"iff-snmp-traps": [
null
]
},
"if-device-flags": {
"ifdf-present": [
null
],
"ifdf-running": [
null
]
},
"if-media-flags": {
"ifmf-none": [
null
]
},
"if-type": "Software-Pseudo",
"interface-flapped": "Never",
"link-type": "Full-Duplex",
"local-index": 128,
"mtu": "9192",
"name": "demux0",
"oper-status": "up",
"snmp-index": 502,
"traffic-statistics": {
"input-packets": 0,
"output-packets": 0
}
},
{
"admin-status": "up",
"current-physical-address": "00:1d:b5:2a:48:11",
"hardware-physical-address": "00:1d:b5:2a:48:11",
"if-config-flags": {
"iff-snmp-traps": [
null
]
},
"if-device-flags": {
"ifdf-present": [
null
],
"ifdf-running": [
null
]
},
"if-media-flags": {
"ifmf-none": [
null
]
},
"if-type": "Ethernet",
"interface-flapped": "Never",
"link-level-type": "Ethernet",
"link-type": "Full-Duplex",
"local-index": 129,
"mtu": "9192",
"name": "cbp0",
```

```
"oper-status": "up",
"snmp-index": 501,
"traffic-statistics": {
"input-packets": 0,
"output-packets": 0
}
},
{
"admin-status": "up",
"current-physical-address": "00:1d:b5:2a:4f:b0",
"hardware-physical-address": "00:1d:b5:2a:4f:b0",
"if-config-flags": {
"iff-snmp-traps": [
null
]
},
"if-device-flags": {
"ifdf-present": [
null
],
"ifdf-running": [
null
]
},
"if-media-flags": {
"ifmf-none": [
null
]
},
"if-type": "Ethernet",
"interface-flapped": "Never",
"link-level-type": "Ethernet",
"link-type": "Full-Duplex",
"local-index": 130,
"mtu": "9192",
"name": "pip0",
"oper-status": "up",
"snmp-index": 504,
"traffic-statistics": {
"input-packets": 0,
"output-packets": 0
}
},
{
"admin-status": "up",
"current-physical-address": "00:1d:b5:2a:4f:c0",
"hardware-physical-address": "00:1d:b5:2a:4f:c0",
"if-config-flags": {},
"if-device-flags": {
"ifdf-present": [
null
],
"ifdf-running": [
null
]
},
"if-media-flags": {
"ifmf-none": [
null
]
},
"if-type": "Ethernet",
```

```
"interface-flapped": "Never",
"link-level-type": "Ethernet",
"link-type": "Full-Duplex",
"local-index": 145,
"logical-interface": [
{
"address-family": [
{
"address-family-flags": {
"ifff-is-primary": [
null
]
},
"address-family-name": "inet",
"interface-address": [
{
"ifa-broadcast": "191.255.255.255",
"ifa-destination": "128/2",
"ifa-flags": {
"ifaf-current-default": [
null
],
"ifaf-current-preferred": [
null
],
"ifaf-current-primary": [
null
],
"ifaf-primary": [
null
]
},
"ifa-local": "128.0.0.127"
}
],
"mtu": "1514"
}
],
"encapsulation": "unknown",
"filter-information": [
null
],
"if-config-flags": {
"iff-up": [
null
],
"internal-flags": "0x24004000"
},
"irb-domain": {
"irb-bridge": "None",
"irb-routing-instance": "None"
},
"local-index": 324,
"logical-interface-bandwidth": "1Gbps",
"name": "jsrv.1",
"policer-overhead": {},
"snmp-index": 537,
"traffic-statistics": {
"input-packets": 0,
"output-packets": 0
}
}
```

```
],
"mtu": "1514",
"name": "jsrv",
"oper-status": "up",
"snmp-index": 536,
"traffic-statistics": {
"input-packets": 0,
"output-packets": 0
}
},
{
"admin-status": "up",
"current-physical-address": "00:1d:b5:2a:4f:f0",
"hardware-physical-address": "00:1d:b5:2a:4f:f0",
"if-config-flags": {
"iff-snmp-traps": [
null
]
},
"if-device-flags": {
"ifdf-present": [
null
],
"ifdf-running": [
null
]
},
"if-media-flags": {
"ifmf-none": [
null
]
},
"if-type": "Ethernet",
"interface-flapped": "Never",
"link-level-type": "Ethernet",
"link-type": "Full-Duplex",
"local-index": 132,
"mtu": "1514",
"name": "irb",
"oper-status": "up",
"snmp-index": 503,
"traffic-statistics": {
"input-packets": 0,
"output-packets": 0
}
},
{
"admin-status": "up",
"current-physical-address": "00:a0:a5:71:d4:cb",
"hardware-physical-address": "00:a0:a5:71:d4:cb",
"if-config-flags": {
"iff-snmp-traps": [
null
]
},
"if-device-flags": {
"ifdf-present": [
null
],
"ifdf-running": [
null
]
```

```
},
"if-media-flags": {},
"if-type": "Ethernet",
"interface-flapped": "2026-03-16 07:10:36 PDT (1w2d 03:23 ago)",
"link-level-type": "Ethernet",
"link-type": "Full-Duplex",
"local-index": 64,
"logical-interface": [
{
"address-family": [
{
"address-family-flags": {
"ifff-is-primary": [
null
],
"ifff-sendbcast-pkt-to-re": [
null
]
},
"address-family-name": "inet",
"interface-address": [
{
"ifa-broadcast": "172.20.160.127",
"ifa-destination": "172.20.160.0/25",
"ifa-flags": {
"ifaf-current-default": [
null
],
"ifaf-current-preferred": [
null
],
"ifaf-current-primary": [
null
]
},
"ifa-local": "172.20.160.24"
}
],
"mtu": "1500"
}
],
"encapsulation": "ENET2",
"filter-information": [
null
],
"if-config-flags": {
"iff-snmp-traps": [
null
],
"iff-up": [
null
],
"internal-flags": "0x4000000"
},
"local-index": 6,
"name": "fxp0.0",
"policer-overhead": {},
"snmp-index": 13,
"traffic-statistics": {
"input-packets": 897961,
"output-packets": 347854
}
```

```
}
],
"mtu": "1514",
"name": "fxp0",
"oper-status": "up",
"snmp-index": 1,
"speed": "1000mbps",
"traffic-statistics": {
"input-packets": 897983,
"output-packets": 347846
}
},
{
"admin-status": "up",
"current-physical-address": "02:00:00:00:00:04",
"hardware-physical-address": "02:00:00:00:00:04",
"if-config-flags": {
"iff-snmp-traps": [
null
]
},
"if-device-flags": {
"ifdf-present": [
null
],
"ifdf-running": [
null
]
},
"if-media-flags": {},
"if-type": "Ethernet",
"interface-flapped": "2026-03-16 07:10:36 PDT (1w2d 03:23 ago)",
"link-level-type": "Ethernet",
"link-type": "Full-Duplex",
"local-index": 8,
"logical-interface": [
{
"address-family": [
{
"address-family-flags": {
"ifff-is-primary": [
null
]
},
"address-family-name": "inet",
"interface-address": [
{
"ifa-broadcast": "10.255.255.255",
"ifa-destination": "10/8",
"ifa-flags": {
"ifaf-current-preferred": [
null
]
},
"ifa-local": "10.0.0.4"
},
{
"ifa-broadcast": "191.255.255.255",
"ifa-destination": "128/2",
"ifa-flags": {
"ifaf-current-preferred": [
null
```

```
],
"ifaf-kernel": [
null
],
"ifaf-preferred": [
null
]
},
"ifa-local": "128.0.0.1"
},
{
"ifa-broadcast": "191.255.255.255",
"ifa-destination": "128/2",
"ifa-flags": {
"ifaf-current-default": [
null
],
"ifaf-current-primary": [
null
],
"ifaf-primary": [
null
]
},
"ifa-local": "128.0.0.4"
}
],
"mtu": "1500"
},
{
"address-family-flags": {
"ifff-is-primary": [
null
]
},
"address-family-name": "inet6",
"interface-address": [
{
"ifa-destination": "fe80::/64",
"ifa-flags": {
"ifaf-current-preferred": [
null
]
},
"ifa-local": "fe80::200:ff:fe00:4"
},
{
"ifa-destination": "fec0::/64",
"ifa-flags": {
"ifaf-current-default": [
null
],
"ifaf-current-preferred": [
null
],
"ifaf-current-primary": [
null
]
},
"ifa-local": "fec0::a:0:0:4"
}
],
```

```
"mtu": "1500"
},
{
"address-family-flags": {
"ifff-is-primary": [
null
],
"ifff-primary": [
null
]
},
"address-family-name": "tnp",
"interface-address": [
{
"ifa-local": "0x4"
}
],
"mtu": "1500"
}
],
"encapsulation": "ENET2",
"filter-information": [
null
],
"if-config-flags": {
"iff-snmp-traps": [
null
],
"iff-up": [
null
],
"internal-flags": "0x4000000"
},
"local-index": 4,
"name": "em0.0",
"policer-overhead": {},
"snmp-index": 18,
"traffic-statistics": {
"input-packets": 15618569,
"output-packets": 17349670
}
}
],
"mtu": "1514",
"name": "em0",
"oper-status": "up",
"snmp-index": 17,
"speed": "1000mbps",
"traffic-statistics": {
"input-packets": 17249696,
"output-packets": 17349814
}
},
{
"admin-status": "up",
"current-physical-address": "02:00:01:00:00:04",
"hardware-physical-address": "02:00:01:00:00:04",
"if-config-flags": {
"iff-snmp-traps": [
null
]
},
```

```
"if-device-flags": {
"ifdf-present": [
null
],
"ifdf-running": [
null
]
},
"if-media-flags": {},
"if-type": "Ethernet",
"interface-flapped": "2026-03-16 07:10:36 PDT (1w2d 03:23 ago)",
"link-level-type": "Ethernet",
"link-type": "Full-Duplex",
"local-index": 9,
"logical-interface": [
{
"address-family": [
{
"address-family-flags": {
"ifff-none": [
null
]
},
"address-family-name": "inet",
"interface-address": [
{
"ifa-broadcast": "10.255.255.255",
"ifa-destination": "10/8",
"ifa-flags": {
"ifaf-current-preferred": [
null
]
},
"ifa-local": "10.0.0.4"
},
{
"ifa-broadcast": "191.255.255.255",
"ifa-destination": "128/2",
"ifa-flags": {
"ifaf-current-preferred": [
null
],
"ifaf-kernel": [
null
],
"ifaf-preferred": [
null
]
},
"ifa-local": "128.0.0.1"
},
{
"ifa-broadcast": "191.255.255.255",
"ifa-destination": "128/2",
"ifa-flags": {
"ifaf-current-primary": [
null
],
"ifaf-primary": [
null
]
},
```

```
"ifa-local": "128.0.0.4"
}
],
"mtu": "1500"
},
{
"address-family-flags": {
"ifff-none": [
null
]
},
"address-family-name": "inet6",
"interface-address": [
{
"ifa-destination": "fe80::/64",
"ifa-flags": {
"ifaf-current-preferred": [
null
]
},
"ifa-local": "fe80::200:1ff:fe00:4"
},
{
"ifa-destination": "fec0::/64",
"ifa-flags": {
"ifaf-current-preferred": [
null
],
"ifaf-current-primary": [
null
]
},
"ifa-local": "fec0::a:0:0:4"
}
],
"mtu": "1500"
},
{
"address-family-flags": {
"ifff-none": [
null
]
},
"address-family-name": "tnp",
"interface-address": [
{
"ifa-local": "0x4"
}
],
"mtu": "1500"
}
],
"encapsulation": "ENET2",
"filter-information": [
null
],
"if-config-flags": {
"iff-snmp-traps": [
null
],
"iff-up": [
null
```

```
],
"internal-flags": "0x4000000"
},
"local-index": 5,
"name": "em1.0",
"policer-overhead": {},
"snmp-index": 24,
"traffic-statistics": {
"input-packets": 0,
"output-packets": 789751
}
}
],
"mtu": "1514",
"name": "em1",
"oper-status": "up",
"snmp-index": 23,
"speed": "1000mbps",
"traffic-statistics": {
"input-packets": 0,
"output-packets": 789751
}
},
{
"admin-status": "up",
"if-config-flags": {
"iff-point-to-point": [
null
],
"iff-snmp-traps": [
null
]
},
"if-device-flags": {
"ifdf-present": [
null
],
"ifdf-running": [
null
]
},
"if-media-flags": {
"ifmf-none": [
null
]
},
"if-type": "PPPoE",
"link-level-type": "PPPoE",
"link-type": "Full-Duplex",
"local-index": 131,
"mtu": "1532",
"name": "pp0",
"oper-status": "up",
"snmp-index": 505
},
{
"admin-status": "up",
"if-config-flags": {
"iff-point-to-point": [
null
],
"iff-snmp-traps": [
```

```
null
]
},
"if-device-flags": {
"ifdf-present": [
null
],
"ifdf-running": [
null
]
},
"if-media-flags": {
"ifmf-none": [
null
]
},
"if-type": "Software-Pseudo",
"interface-flapped": "Never",
"local-index": 5,
"mtu": "Unlimited",
"name": "dsc",
"oper-status": "up",
"snmp-index": 5,
"traffic-statistics": {
"input-packets": 0,
"output-packets": 0
}
},
{
"admin-status": "up",
"if-config-flags": {
"iff-point-to-point": [
null
],
"iff-snmp-traps": [
null
]
},
"if-device-flags": {
"ifdf-present": [
null
],
"ifdf-running": [
null
]
},
"if-type": "GRE",
"link-level-type": "GRE",
"local-index": 10,
"mtu": "Unlimited",
"name": "gre",
"oper-status": "up",
"snmp-index": 8,
"speed": "Unlimited",
"traffic-statistics": {
"input-packets": 0,
"output-packets": 0
}
},
{
"admin-status": "up",
"if-config-flags": {
```

```
"iff-snmp-traps": [
null
]
},
"if-device-flags": {
"ifdf-loopback": [
null
],
"ifdf-present": [
null
],
"ifdf-running": [
null
]
},
"if-media-flags": {
"ifmf-none": [
null
]
},
"if-type": "Loopback",
"interface-flapped": "Never",
"local-index": 6,
"logical-interface": [
{
"encapsulation": "Unspecified",
"filter-information": [
null
],
"if-config-flags": {
"iff-snmp-traps": [
null
]
},
"local-index": 322,
"name": "lo0.0",
"policer-overhead": {},
"snmp-index": 16,
"traffic-statistics": {
"input-packets": 0,
"output-packets": 0
}
},
{
"address-family": [
{
"address-family-flags": {},
"address-family-name": "inet",
"interface-address": [
{
"ifa-local": "127.0.0.1"
}
],
"mtu": "Unlimited"
}
],
"encapsulation": "Unspecified",
"filter-information": [
null
],
"if-config-flags": {
"iff-snmp-traps": [
```

```
null
]
},
"local-index": 321,
"name": "lo0.16384",
"policer-overhead": {},
"snmp-index": 21,
"traffic-statistics": {
"input-packets": 0,
"output-packets": 0
}
},
{
"address-family": [
{
"address-family-flags": {},
"address-family-name": "inet",
"mtu": "Unlimited"
}
],
"encapsulation": "Unspecified",
"filter-information": [
null
],
"if-config-flags": {
"iff-snmp-traps": [
null
]
},
"local-index": 320,
"name": "lo0.16385",
"policer-overhead": {},
"snmp-index": 22,
"traffic-statistics": {
"input-packets": 690013,
"output-packets": 690013
}
}
],
"mtu": "Unlimited",
"name": "lo0",
"oper-status": "up",
"snmp-index": 6,
"traffic-statistics": {
"input-packets": 690013,
"output-packets": 690013
}
},
{
"admin-status": "up",
"if-config-flags": {
"iff-snmp-traps": [
null
]
},
"if-device-flags": {
"ifdf-present": [
null
],
"ifdf-running": [
null
]
```

```
},
"if-media-flags": {
"ifmf-none": [
null
]
},
"if-type": "FTI",
"interface-flapped": "Never",
"link-level-type": "Flexible-tunnel-Interface",
"link-type": "Full-Duplex",
"local-index": 136,
"mtu": "Unlimited",
"name": "fti0",
"oper-status": "up",
"snmp-index": 1191,
"speed": "Unlimited",
"traffic-statistics": {
"input-packets": 0,
"output-packets": 0
}
},
{
"admin-status": "up",
"if-config-flags": {
"iff-snmp-traps": [
null
]
},
"if-device-flags": {
"ifdf-present": [
null
],
"ifdf-running": [
null
]
},
"if-media-flags": {
"ifmf-none": [
null
]
},
"if-type": "FTI",
"interface-flapped": "Never",
"link-level-type": "Flexible-tunnel-Interface",
"link-type": "Full-Duplex",
"local-index": 137,
"mtu": "Unlimited",
"name": "fti1",
"oper-status": "up",
"snmp-index": 1217,
"speed": "Unlimited",
"traffic-statistics": {
"input-packets": 0,
"output-packets": 0
}
},
{
"admin-status": "up",
"if-config-flags": {
"iff-snmp-traps": [
null
]
```

```
},
"if-device-flags": {
"ifdf-present": [
null
],
"ifdf-running": [
null
]
},
"if-media-flags": {
"ifmf-none": [
null
]
},
"if-type": "FTI",
"interface-flapped": "Never",
"link-level-type": "Flexible-tunnel-Interface",
"link-type": "Full-Duplex",
"local-index": 138,
"mtu": "Unlimited",
"name": "fti2",
"oper-status": "up",
"snmp-index": 1368,
"speed": "Unlimited",
"traffic-statistics": {
"input-packets": 0,
"output-packets": 0
}
},
{
"admin-status": "up",
"if-config-flags": {
"iff-snmp-traps": [
null
]
},
"if-device-flags": {
"ifdf-present": [
null
],
"ifdf-running": [
null
]
},
"if-media-flags": {
"ifmf-none": [
null
]
},
"if-type": "FTI",
"interface-flapped": "Never",
"link-level-type": "Flexible-tunnel-Interface",
"link-type": "Full-Duplex",
"local-index": 139,
"mtu": "Unlimited",
"name": "fti3",
"oper-status": "up",
"snmp-index": 1369,
"speed": "Unlimited",
"traffic-statistics": {
"input-packets": 0,
"output-packets": 0
```

```
}
},
{
"admin-status": "up",
"if-config-flags": {
"iff-snmp-traps": [
null
]
},
"if-device-flags": {
"ifdf-present": [
null
],
"ifdf-running": [
null
]
},
"if-media-flags": {
"ifmf-none": [
null
]
},
"if-type": "FTI",
"interface-flapped": "Never",
"link-level-type": "Flexible-tunnel-Interface",
"link-type": "Full-Duplex",
"local-index": 140,
"mtu": "Unlimited",
"name": "fti4",
"oper-status": "up",
"snmp-index": 1370,
"speed": "Unlimited",
"traffic-statistics": {
"input-packets": 0,
"output-packets": 0
}
},
{
"admin-status": "up",
"if-config-flags": {
"iff-snmp-traps": [
null
]
},
"if-device-flags": {
"ifdf-present": [
null
],
"ifdf-running": [
null
]
},
"if-media-flags": {
"ifmf-none": [
null
]
},
"if-type": "FTI",
"interface-flapped": "Never",
"link-level-type": "Flexible-tunnel-Interface",
"link-type": "Full-Duplex",
"local-index": 141,
```

```
"mtu": "Unlimited",
"name": "fti5",
"oper-status": "up",
"snmp-index": 1371,
"speed": "Unlimited",
"traffic-statistics": {
"input-packets": 0,
"output-packets": 0
}
},
{
"admin-status": "up",
"if-config-flags": {
"iff-snmp-traps": [
null
]
},
"if-device-flags": {
"ifdf-present": [
null
],
"ifdf-running": [
null
]
},
"if-media-flags": {
"ifmf-none": [
null
]
},
"if-type": "FTI",
"interface-flapped": "Never",
"link-level-type": "Flexible-tunnel-Interface",
"link-type": "Full-Duplex",
"local-index": 142,
"mtu": "Unlimited",
"name": "fti6",
"oper-status": "up",
"snmp-index": 1372,
"speed": "Unlimited",
"traffic-statistics": {
"input-packets": 0,
"output-packets": 0
}
},
{
"admin-status": "up",
"if-config-flags": {
"iff-snmp-traps": [
null
]
},
"if-device-flags": {
"ifdf-present": [
null
],
"ifdf-running": [
null
]
},
"if-media-flags": {
"ifmf-none": [
```

```
null
]
},
"if-type": "FTI",
"interface-flapped": "Never",
"link-level-type": "Flexible-tunnel-Interface",
"link-type": "Full-Duplex",
"local-index": 143,
"mtu": "Unlimited",
"name": "fti7",
"oper-status": "up",
"snmp-index": 1373,
"speed": "Unlimited",
"traffic-statistics": {
"input-packets": 0,
"output-packets": 0
}
},
{
"admin-status": "up",
"if-config-flags": {
"iff-snmp-traps": [
null
]
},
"if-device-flags": {
"ifdf-present": [
null
],
"ifdf-running": [
null
]
},
"if-media-flags": {
"ifmf-none": [
null
]
},
"if-type": "Software-Pseudo",
"interface-flapped": "Never",
"link-level-type": "Interface-Specific",
"link-type": "Full-Duplex",
"local-index": 144,
"mtu": "1440",
"name": "mif",
"oper-status": "up",
"snmp-index": 1377,
"traffic-statistics": {
"input-packets": 0,
"output-packets": 0
}
},
{
"admin-status": "up",
"if-config-flags": {
"iff-snmp-traps": [
null
]
},
"if-device-flags": {
"ifdf-present": [
null
```

```
],
"ifdf-running": [
null
]
},
"if-media-flags": {
"ifmf-none": [
null
]
},
"if-type": "Software-Pseudo",
"interface-flapped": "Never",
"link-level-type": "Interface-Specific",
"local-index": 12,
"mtu": "Unlimited",
"name": "tap",
"oper-status": "up",
"snmp-index": 7,
"speed": "Unlimited",
"traffic-statistics": {
"input-packets": 0,
"output-packets": 0
}
},
{
"admin-status": "up",
"if-config-flags": {
"iff-snmp-traps": [
null
]
},
"if-device-flags": {
"ifdf-present": [
null
],
"ifdf-running": [
null
]
},
"if-media-flags": {
"ifmf-none": [
null
]
},
"if-type": "Software-Pseudo",
"interface-flapped": "Never",
"link-level-type": "Remote-BEB",
"link-type": "Full-Duplex",
"local-index": 135,
"mtu": "Unlimited",
"name": "rbeb",
"oper-status": "up",
"snmp-index": 678,
"speed": "Unlimited",
"traffic-statistics": {
"input-packets": 0,
"output-packets": 0
}
},
{
"admin-status": "up",
"if-config-flags": {
```

```
"iff-snmp-traps": [
null
]
},
"if-device-flags": {
"ifdf-present": [
null
],
"ifdf-running": [
null
]
},
"if-media-flags": {
"ifmf-none": [
null
]
},
"if-type": "Software-Pseudo",
"interface-flapped": "Never",
"link-level-type": "VxLAN-Tunnel-Endpoint",
"link-type": "Full-Duplex",
"local-index": 133,
"mtu": "Unlimited",
"name": "vtep",
"oper-status": "up",
"snmp-index": 685,
"speed": "Unlimited",
"traffic-statistics": {
"input-packets": 0,
"output-packets": 0
}
},
{
"admin-status": "up",
"if-config-flags": {
"iff-snmp-traps": [
null
]
},
"if-device-flags": {
"ifdf-present": [
null
],
"ifdf-running": [
null
]
},
"if-media-flags": {
"ifmf-none": [
null
]
},
"if-type": "Software-Pseudo",
"interface-flapped": "Never",
"link-level-type": "VxLAN-Tunnel-Endpoint",
"link-type": "Full-Duplex",
"local-index": 134,
"mtu": "Unlimited",
"name": "esi",
"oper-status": "up",
"snmp-index": 535,
"speed": "Unlimited",
```

```
"traffic-statistics": {
"input-packets": 0,
"output-packets": 0
}
},
{
"admin-status": "up",
"if-config-flags": {
"iff-snmp-traps": [
null
]
},
"if-device-flags": {
"ifdf-present": [
null
],
"ifdf-running": [
null
]
},
"if-type": "IPIP",
"link-level-type": "IP-over-IP",
"local-index": 11,
"mtu": "Unlimited",
"name": "ipip",
"oper-status": "up",
"snmp-index": 9,
"speed": "Unlimited",
"traffic-statistics": {
"input-packets": 0,
"output-packets": 0
}
},
{
"admin-status": "up",
"if-config-flags": {
"iff-snmp-traps": [
null
]
},
"if-device-flags": {
"ifdf-present": [
null
],
"ifdf-running": [
null
]
},
"if-type": "Multicast-GRE",
"link-level-type": "GRE",
"local-index": 65,
"mtu": "Unlimited",
"name": "mtun",
"oper-status": "up",
"snmp-index": 12,
"speed": "Unlimited",
"traffic-statistics": {
"input-packets": 0,
"output-packets": 0
}
},
{
```

```
"admin-status": "up",
"if-config-flags": {},
"if-device-flags": {
"ifdf-present": [
null
],
"ifdf-running": [
null
]
},
"if-media-flags": {
"ifmf-none": [
null
]
},
"if-type": "Software-Pseudo",
"interface-flapped": "Never",
"link-level-type": "LSI",
"local-index": 4,
"mtu": "Unlimited",
"name": "lsi",
"oper-status": "up",
"snmp-index": 4,
"speed": "Unlimited",
"traffic-statistics": {
"input-packets": 0,
"output-packets": 0
}
},
{
"admin-status": "up",
"if-config-flags": {},
"if-device-flags": {
"ifdf-present": [
null
],
"ifdf-running": [
null
]
},
"if-media-flags": {
"ifmf-none": [
null
]
},
"interface-flapped": "Never",
"local-index": 146,
"logical-interface": [
{
"address-family": [
{
"address-family-flags": {
"ifff-none": [
null
]
},
"address-family-name": "inet",
"mtu": "Unlimited"
}
],
"encapsulation": "ENET2",
"filter-information": [
```

```
null
],
"if-config-flags": {
"iff-snmp-traps": [
null
],
"iff-up": [
null
]
},
"local-index": 329,
"logical-interface-bandwidth": "0",
"name": "pfh-3/0/0.16383",
"policer-overhead": {},
"snmp-index": 523,
"traffic-statistics": {
"input-packets": 0,
"output-packets": 0
}
},
{
"address-family": [
{
"address-family-flags": {
"ifff-is-primary": [
null
]
},
"address-family-name": "inet",
"mtu": "Unlimited"
}
],
"encapsulation": "ENET2",
"filter-information": [
null
],
"if-config-flags": {
"iff-snmp-traps": [
null
],
"iff-up": [
null
]
},
"local-index": 330,
"logical-interface-bandwidth": "0",
"name": "pfh-3/0/0.16384",
"policer-overhead": {},
"snmp-index": 1232,
"traffic-statistics": {
"input-packets": 0,
"output-packets": 0
}
}
],
"name": "pfh-3/0/0",
"oper-status": "up",
"snmp-index": 522,
"speed": "800mbps",
"traffic-statistics": {
"input-packets": 0,
"output-packets": 0
```

```
}
},
{
"admin-status": "up",
"if-config-flags": {},
"if-device-flags": {
"ifdf-present": [
null
],
"ifdf-running": [
null
]
},
"if-media-flags": {
"ifmf-none": [
null
]
},
"interface-flapped": "Never",
"local-index": 147,
"logical-interface": [
{
"address-family": [
{
"address-family-flags": {
"ifff-none": [
null
]
},
"address-family-name": "inet",
"mtu": "Unlimited"
},
{
"address-family-flags": {
"ifff-none": [
null
]
},
"address-family-name": "inet6",
"mtu": "Unlimited"
}
],
"encapsulation": "ENET2",
"filter-information": [
null
],
"if-config-flags": {
"iff-snmp-traps": [
null
],
"iff-up": [
null
]
},
"local-index": 331,
"logical-interface-bandwidth": "0",
"name": "pfe-3/0/0.16383",
"policer-overhead": {},
"snmp-index": 526,
"traffic-statistics": {
"input-packets": 0,
"output-packets": 0
```

```
}
}
],
"name": "pfe-3/0/0",
"oper-status": "up",
"snmp-index": 524,
"speed": "800mbps",
"traffic-statistics": {
"input-packets": 0,
"output-packets": 0
}
},
{
"admin-status": "up",
"if-config-flags": {},
"if-device-flags": {
"ifdf-present": [
null
],
"ifdf-running": [
null
]
},
"if-media-flags": {
"ifmf-none": [
null
]
},
"interface-flapped": "Never",
"local-index": 148,
"logical-interface": [
{
"address-family": [
{
"address-family-flags": {
"ifff-none": [
null
]
},
"address-family-name": "inet",
"mtu": "Unlimited"
},
{
"address-family-flags": {
"ifff-none": [
null
]
},
"address-family-name": "inet6",
"mtu": "Unlimited"
}
],
"encapsulation": "ENET2",
"filter-information": [
null
],
"if-config-flags": {
"iff-snmp-traps": [
null
],
"iff-up": [
null
```

```
]
},
"local-index": 332,
"logical-interface-bandwidth": "0",
"name": "pfe-3/1/0.16383",
"policer-overhead": {},
"snmp-index": 1234,
"traffic-statistics": {
"input-packets": 0,
"output-packets": 0
}
}
],
"name": "pfe-3/1/0",
"oper-status": "up",
"snmp-index": 1219,
"speed": "800mbps",
"traffic-statistics": {
"input-packets": 0,
"output-packets": 0
}
},
{
"admin-status": "up",
"if-config-flags": {},
"if-device-flags": {
"ifdf-present": [
null
],
"ifdf-running": [
null
]
},
"if-media-flags": {
"ifmf-none": [
null
]
},
"interface-flapped": "Never",
"local-index": 149,
"logical-interface": [
{
"address-family": [
{
"address-family-flags": {
"ifff-none": [
null
]
},
"address-family-name": "inet",
"mtu": "Unlimited"
},
{
"address-family-flags": {
"ifff-none": [
null
]
},
"address-family-name": "inet6",
"mtu": "Unlimited"
}
],
```

```
"encapsulation": "ENET2",
"filter-information": [
null
],
"if-config-flags": {
"iff-snmp-traps": [
null
],
"iff-up": [
null
]
},
"local-index": 333,
"logical-interface-bandwidth": "0",
"name": "pfe-3/2/0.16383",
"policer-overhead": {},
"snmp-index": 527,
"traffic-statistics": {
"input-packets": 0,
"output-packets": 0
}
}
],
"name": "pfe-3/2/0",
"oper-status": "up",
"snmp-index": 525,
"speed": "800mbps",
"traffic-statistics": {
"input-packets": 0,
"output-packets": 0
}
},
{
"admin-status": "up",
"if-config-flags": {},
"if-device-flags": {
"ifdf-present": [
null
],
"ifdf-running": [
null
]
},
"if-media-flags": {
"ifmf-none": [
null
]
},
"interface-flapped": "Never",
"local-index": 150,
"logical-interface": [
{
"address-family": [
{
"address-family-flags": {
"ifff-none": [
null
]
},
"address-family-name": "inet",
"mtu": "Unlimited"
},
```

```
{
"address-family-flags": {
"ifff-none": [
null
]
},
"address-family-name": "inet6",
"mtu": "Unlimited"
}
],
"encapsulation": "ENET2",
"filter-information": [
null
],
"if-config-flags": {
"iff-snmp-traps": [
null
],
"iff-up": [
null
]
},
"local-index": 334,
"logical-interface-bandwidth": "0",
"name": "pfe-3/3/0.16383",
"policer-overhead": {},
"snmp-index": 1366,
"traffic-statistics": {
"input-packets": 0,
"output-packets": 0
}
}
],
"name": "pfe-3/3/0",
"oper-status": "up",
"snmp-index": 1228,
"speed": "800mbps",
"traffic-statistics": {
"input-packets": 0,
"output-packets": 0
}
},
{
"admin-status": "up",
"if-config-flags": {},
"if-device-flags": {
"ifdf-present": [
null
],
"ifdf-running": [
null
]
},
"if-media-flags": {
"ifmf-none": [
null
]
},
"interface-flapped": "Never",
"local-index": 151,
"logical-interface": [
{
```

```
"address-family": [
{
"address-family-flags": {
"ifff-none": [
null
]
},
"address-family-name": "inet",
"mtu": "Unlimited"
}
],
"encapsulation": "ENET2",
"filter-information": [
null
],
"if-config-flags": {
"iff-snmp-traps": [
null
],
"iff-up": [
null
]
},
"local-index": 335,
"logical-interface-bandwidth": "0",
"name": "pfh-10/0/0.16383",
"policer-overhead": {},
"snmp-index": 651,
"traffic-statistics": {
"input-packets": 0,
"output-packets": 0
}
},
{
"address-family": [
{
"address-family-flags": {
"ifff-none": [
null
]
},
"address-family-name": "inet",
"mtu": "Unlimited"
}
],
"encapsulation": "ENET2",
"filter-information": [
null
],
"if-config-flags": {
"iff-snmp-traps": [
null
],
"iff-up": [
null
]
},
"local-index": 336,
"logical-interface-bandwidth": "0",
"name": "pfh-10/0/0.16384",
"policer-overhead": {},
"snmp-index": 1367,
```

```
"traffic-statistics": {
"input-packets": 0,
"output-packets": 0
}
}
],
"name": "pfh-10/0/0",
"oper-status": "up",
"snmp-index": 646,
"speed": "800mbps",
"traffic-statistics": {
"input-packets": 0,
"output-packets": 0
}
},
{
"admin-status": "up",
"if-config-flags": {},
"if-device-flags": {
"ifdf-present": [
null
],
"ifdf-running": [
null
]
},
"if-media-flags": {
"ifmf-none": [
null
]
},
"interface-flapped": "Never",
"local-index": 152,
"logical-interface": [
{
"address-family": [
{
"address-family-flags": {
"ifff-none": [
null
]
},
"address-family-name": "inet",
"mtu": "Unlimited"
},
{
"address-family-flags": {
"ifff-none": [
null
]
},
"address-family-name": "inet6",
"mtu": "Unlimited"
}
],
"encapsulation": "ENET2",
"filter-information": [
null
],
"if-config-flags": {
"iff-snmp-traps": [
null
```

```
],
"iff-up": [
null
]
},
"local-index": 337,
"logical-interface-bandwidth": "0",
"name": "pfe-10/0/0.16383",
"policer-overhead": {},
"snmp-index": 652,
"traffic-statistics": {
"input-packets": 0,
"output-packets": 0
}
}
],
"name": "pfe-10/0/0",
"oper-status": "up",
"snmp-index": 647,
"speed": "800mbps",
"traffic-statistics": {
"input-packets": 0,
"output-packets": 0
}
},
{
"admin-status": "up",
"if-config-flags": {},
"if-device-flags": {
"ifdf-present": [
null
],
"ifdf-running": [
null
]
},
"if-media-flags": {
"ifmf-none": [
null
]
},
"interface-flapped": "Never",
"local-index": 153,
"logical-interface": [
{
"address-family": [
{
"address-family-flags": {
"ifff-none": [
null
]
},
"address-family-name": "inet",
"mtu": "Unlimited"
},
{
"address-family-flags": {
"ifff-none": [
null
]
},
"address-family-name": "inet6",
```

```
"mtu": "Unlimited"
}
],
"encapsulation": "ENET2",
"filter-information": [
null
],
"if-config-flags": {
"iff-snmp-traps": [
null
],
"iff-up": [
null
]
},
"local-index": 338,
"logical-interface-bandwidth": "0",
"name": "pfe-10/1/0.16383",
"policer-overhead": {},
"snmp-index": 653,
"traffic-statistics": {
"input-packets": 0,
"output-packets": 0
}
}
],
"name": "pfe-10/1/0",
"oper-status": "up",
"snmp-index": 648,
"speed": "800mbps",
"traffic-statistics": {
"input-packets": 0,
"output-packets": 0
}
},
{
"admin-status": "up",
"if-config-flags": {},
"if-device-flags": {
"ifdf-present": [
null
],
"ifdf-running": [
null
]
},
"if-media-flags": {
"ifmf-none": [
null
]
},
"interface-flapped": "Never",
"local-index": 154,
"logical-interface": [
{
"address-family": [
{
"address-family-flags": {
"ifff-none": [
null
]
},
```

```
"address-family-name": "inet",
"mtu": "Unlimited"
},
{
"address-family-flags": {
"ifff-none": [
null
]
},
"address-family-name": "inet6",
"mtu": "Unlimited"
}
],
"encapsulation": "ENET2",
"filter-information": [
null
],
"if-config-flags": {
"iff-snmp-traps": [
null
],
"iff-up": [
null
]
},
"local-index": 339,
"logical-interface-bandwidth": "0",
"name": "pfe-10/2/0.16383",
"policer-overhead": {},
"snmp-index": 654,
"traffic-statistics": {
"input-packets": 0,
"output-packets": 0
}
}
],
"name": "pfe-10/2/0",
"oper-status": "up",
"snmp-index": 649,
"speed": "800mbps",
"traffic-statistics": {
"input-packets": 0,
"output-packets": 0
}
},
{
"admin-status": "up",
"if-config-flags": {},
"if-device-flags": {
"ifdf-present": [
null
],
"ifdf-running": [
null
]
},
"if-media-flags": {
"ifmf-none": [
null
]
},
"interface-flapped": "Never",
```

```
"local-index": 155,
"logical-interface": [
{
"address-family": [
{
"address-family-flags": {
"ifff-none": [
null
]
},
"address-family-name": "inet",
"mtu": "Unlimited"
},
{
"address-family-flags": {
"ifff-none": [
null
]
},
"address-family-name": "inet6",
"mtu": "Unlimited"
}
],
"encapsulation": "ENET2",
"filter-information": [
null
],
"if-config-flags": {
"iff-snmp-traps": [
null
],
"iff-up": [
null
]
},
"local-index": 340,
"logical-interface-bandwidth": "0",
"name": "pfe-10/3/0.16383",
"policer-overhead": {},
"snmp-index": 655,
"traffic-statistics": {
"input-packets": 0,
"output-packets": 0
}
}
],
"name": "pfe-10/3/0",
"oper-status": "up",
"snmp-index": 650,
"speed": "800mbps",
"traffic-statistics": {
"input-packets": 0,
"output-packets": 0
}
},
{
"admin-status": "up",
"if-config-flags": {},
"if-device-flags": {
"ifdf-present": [
null
],
```

```
"ifdf-running": [
null
]
},
"if-type": "PIMD",
"link-level-type": "PIM-Decapsulator",
"local-index": 26,
"mtu": "Unlimited",
"name": "pimd",
"oper-status": "up",
"snmp-index": 11,
"speed": "Unlimited",
"traffic-statistics": {
"input-packets": 0,
"output-packets": 0
}
},
{
"admin-status": "up",
"if-config-flags": {},
"if-device-flags": {
"ifdf-present": [
null
],
"ifdf-running": [
null
]
},
"if-type": "PIME",
"link-level-type": "PIM-Encapsulator",
"local-index": 25,
"mtu": "Unlimited",
"name": "pime",
"oper-status": "up",
"snmp-index": 10,
"speed": "Unlimited",
"traffic-statistics": {
"input-packets": 0,
"output-packets": 0
}
}
],
"status": "success"
}
}
```


Parent topic

Juniper MX MOP actions

Juniper MX arp summary MOP Action

Validates ARP table entries and neighbor relationships. ARP entries should be present and match the pre-upgrade state. The check fails if no entries are found, and raises a warning when entries are missing or newly added.

Refer to this topic to review the schema, input details, and output examples for arp summary.

Action details
Table 49: arp summary details

Field	Value
Sr. No.	7
Device Family	MX Series
MOP action Name	arp summary
RPC Call	rpc-get-arp-table-information
Usage	Validates ARP table entries and neighbor relationships. ARP entries should be present and match the pre-upgrade state. The check fails if no entries are found, and raises a warning when entries are missing or newly added.

Input Schema

Configurable schema not applicable.

Input Example

The orchestrator sends the app-data as mentioned above to each workflow at runtime.

Output


```
{
"Data": {
"message": "ARP pre-upgrade check is successful.",
"stash": [
{
"hostname": "172.20.160.1",
"interface-name": "fxp0.0",
"ip-address": "172.20.160.1",
"mac-address": "00:00:0c:07:ac:0a"
},
{
"hostname": "172.20.160.2",
"interface-name": "fxp0.0",
"ip-address": "172.20.160.2",
"mac-address": "00:1b:0d:ef:79:40"
},
{
"hostname": "172.20.160.3",
"interface-name": "fxp0.0",
"ip-address": "172.20.160.3",
"mac-address": "00:1b:0d:ef:7f:00"
},
{
"hostname": "fpc10",
"interface-name": "em0.0",
"ip-address": "128.0.0.26",
"mac-address": "02:00:00:00:00:1a"
},
{
"hostname": "fpc3",
"interface-name": "em0.0",
```

```
"ip-address": "128.0.0.19",
"mac-address": "02:00:00:00:00:13"
}
],
"status": "success"
}
}
```


Parent topic

Juniper MX MOP actions

Juniper MX Route Summary Check MOP Action

Validates routing table summary and convergence. Route tables should match the pre-upgrade state. The check fails if no route tables are found, and raises a warning when routes have been added or removed.

Refer to this topic to review the schema, input details, and output examples for Route Summary Check.

Action details
Table 50: Route Summary Check details

Field	Value
Sr. No.	8
Device Family	MX Series
MOP action Name	Route Summary Check
RPC Call	rpc-get-route-summary-information
Usage	Validates routing table summary and convergence. Route tables should match the pre-upgrade state. The check fails if no route tables are found, and raises a warning when routes have been added or removed.

Input Schema

Configurable schema not applicable.

Input Example

The orchestrator sends the app-data as mentioned above to each workflow at runtime.

Output


```
{
"Data": {
"message": "Route summary pre-upgrade check is successful.",
"stash": [
{
"active-route-count": 1,
"destination-count": 1,
"hidden-route-count": 0,
"holddown-route-count": 0,
"protocols": [
```

```
{
"active-route-count": 1,
"protocol-name": "INET6",
"protocol-route-count": 1
}
],
"table-name": "inet6.0",
"total-route-count": 1
},
{
"active-route-count": 8,
"destination-count": 8,
"hidden-route-count": 0,
"holddown-route-count": 0,
"protocols": [
{
"active-route-count": 1,
"protocol-name": "Direct",
"protocol-route-count": 1
},
{
"active-route-count": 1,
"protocol-name": "Local",
"protocol-route-count": 1
},
{
"active-route-count": 1,
"protocol-name": "OSPF",
"protocol-route-count": 1
},
{
"active-route-count": 5,
"protocol-name": "Static",
"protocol-route-count": 5
}
],
"table-name": "inet.0",
"total-route-count": 8
}
],
"status": "success"
}
}
```


Parent topic

Juniper MX MOP actions

Juniper MX Software Version Check MOP Action

Validates the device is running the expected software version. The version should match the target image or be found within the image filename. The check fails if a version mismatch occurs and the version is not found in the image file.

Refer to this topic to review the schema, input details, and output examples for Software Version Check.

Action details
Table 51: Software Version Check details

Field	Value
Sr. No.	9

Field	Value
Device Family	MX Series
MOP action Name	Software Version Check
RPC Call	rpc-get-software-information
Usage	Validates the device is running the expected software version. The version should match the target image or be found within the image filename. The check fails if a version mismatch occurs and the version is not found in the image file.

Input Schema

Configurable schema not applicable.

Input Example

The orchestrator sends the app-data as mentioned above to each workflow at runtime.

Output

```
{
"Data": {
"message": "Version check failed.",
"stash": [
"21.1R3.11"
],
"status": "failed"
}
}
```

Parent topic

Juniper MX MOP actions

Juniper MX Package Summary Check MOP Action

Verifies package consistency between committed and active states. Committed and active packages should be identical. The check fails if package verification encounters errors, and raises a warning when packages have been added or removed.

Refer to this topic to review the schema, input details, and output examples for Package Summary Check.

Action details
Table 52: Package Summary Check details

Field	Value
Sr. No.	10
Device Family	MX Series
MOP action Name	Package Summary Check

Field	Value
RPC Call	rpc-get-software-information
Usage	Verifies package consistency between committed and active states. Committed and active packages should be identical. The check fails if package verification encounters errors, and raises a warning when packages have been added or removed.

Input Schema

Configurable schema not applicable.

Input Example

The orchestrator sends the app-data as mentioned above to each workflow at runtime.

Output


```
{
"Data": {
"message": "Package summary pre-upgrade check is successful.",
"stash": [
{
"comment": "JUNOS AppId Services [20211215.004903_builder_junos_211_r3]",
"name": "jservices-appid"
},
{
"comment": "JUNOS Broadband Egde User Plane Apps [21.1R3.11]",
"name": "junos-bbe-up"
},
{
"comment": "JUNOS Extension Toolkit [20211215.004903_builder_junos_211_r3]",
"name": "jsd-jet-1"
},
{
"comment": "JUNOS IDP Services [20211215.004903_builder_junos_211_r3]",
"name": "jservices-idp"
},
{
"comment": "JUNOS J-Insight [20211215.004903_builder_junos_211_r3]",
"name": "jinsight"
},
{
"comment": "JUNOS Juniper Malware Removal Tool (JMRT)
[1.0.0+20211215.004903_builder_junos_211_r3]",
"name": "jmrt-base"
},
{
"comment": "JUNOS OS 32-bit compatibility [20211011.a9cfe96_builder_stable_11]",
"name": "os-compat32"
},
{
"comment": "JUNOS OS Kernel 64-bit  [20211011.a9cfe96_builder_stable_11]",
"name": "os-kernel"
},
{
"comment": "JUNOS OS crypto [20211011.a9cfe96_builder_stable_11]",
```

```
"name": "os-crypto"
},
{
"comment": "JUNOS OS libs [20211011.a9cfe96_builder_stable_11]",
"name": "os-libs"
},
{
"comment": "JUNOS OS libs compat32 [20211011.a9cfe96_builder_stable_11]",
"name": "os-libs-compat32"
},
{
"comment": "JUNOS OS runtime [20211011.a9cfe96_builder_stable_11]",
"name": "os-runtime"
},
{
"comment": "JUNOS OS time zone information
[20211011.a9cfe96_builder_stable_11]",
"name": "zoneinfo"
},
{
"comment": "JUNOS Online Documentation [20211215.004903_builder_junos_211_r3]",
"name": "jdocs"
},
{
"comment": "JUNOS Openconfig [21.1R3.11]",
"name": "junos-openconfig"
},
{
"comment": "JUNOS Packet Forwarding Engine Support (M/T Common)
[20211215.004903_builder_junos_211_r3]",
"name": "jpfe-common"
},
{
"comment": "JUNOS Packet Forwarding Engine Support (MX Common)
[20211215.004903_builder_junos_211_r3]",
"name": "jpfe-X"
},
{
"comment": "JUNOS Packet Forwarding Engine Support (MX/EX92XX Common)
[20211215.004903_builder_junos_211_r3]",
"name": "jpfe-platform"
},
{
"comment": "JUNOS Packet Forwarding Engine Support (MXSPC3) [21.1R3.11]",
"name": "jpfe-spc3-mx"
},
{
"comment": "JUNOS Packet Forwarding Engine Support (aft)
[20211215.004903_builder_junos_211_r3]",
"name": "jpfe-aft"
},
{
"comment": "JUNOS Packet Forwarding Engine Support (wrlinux9)
[20211215.004903_builder_junos_211_r3]",
"name": "jpfe-wrlinux9"
},
{
"comment": "JUNOS RPD Telemetry Application [21.1R3.11]",
"name": "junos-rpd-telemetry-application"
},
{
"comment": "JUNOS Routing 32-bit Compatible Version
```

```
[20211215.004903_builder_junos_211_r3]",
"name": "junos-routing-compat32"
},
{
"comment": "JUNOS Routing aggregated [20211215.004903_builder_junos_211_r3]",
"name": "junos-routing-aggregated"
},
{
"comment": "JUNOS Routing controller-external
[20211215.004903_builder_junos_211_r3]",
"name": "junos-routing-controller-external"
},
{
"comment": "JUNOS Routing controller-internal
[20211215.004903_builder_junos_211_r3]",
"name": "junos-routing-controller-internal"
},
{
"comment": "JUNOS Routing lsys [20211215.004903_builder_junos_211_r3]",
"name": "junos-routing-lsys"
},
{
"comment": "JUNOS Routing mpls-oam-advanced
[20211215.004903_builder_junos_211_r3]",
"name": "junos-routing-mpls-oam-advanced"
},
{
"comment": "JUNOS Routing mpls-oam-basic
[20211215.004903_builder_junos_211_r3]",
"name": "junos-routing-mpls-oam-basic"
},
{
"comment": "JUNOS SDN Software Suite [20211215.004903_builder_junos_211_r3]",
"name": "jsdn"
},
{
"comment": "JUNOS SQL Sync Daemon [20211215.004903_builder_junos_211_r3]",
"name": "junos-jsqlsync"
},
{
"comment": "JUNOS Security Intelligence [20211215.004903_builder_junos_211_r3]",
"name": "junos-secintel"
},
{
"comment": "JUNOS Services AACL Container package
[20211215.004903_builder_junos_211_r3]",
"name": "jservices-aacl"
},
{
"comment": "JUNOS Services Application Level Gateways
[20211215.004903_builder_junos_211_r3]",
"name": "jservices-alg"
},
{
"comment": "JUNOS Services COS [20211215.004903_builder_junos_211_r3]",
"name": "jservices-cos"
},
{
"comment": "JUNOS Services Captive Portal and Content Delivery Container package
 [20211215.004903_builder_junos_211_r3]",
"name": "jservices-cpcd"
},
```

```
{
"comment": "JUNOS Services Crypto [20211215.004903_builder_junos_211_r3]",
"name": "jservices-crypto-base"
},
{
"comment": "JUNOS Services DNS Filter package (i386)
[20211215.004903_builder_junos_211_r3]",
"name": "jservices-dnsf"
},
{
"comment": "JUNOS Services Deep Packet Inspection package
[20211215.004903_builder_junos_211_r3]",
"name": "jservices-jdpi"
},
{
"comment": "JUNOS Services HTTP Content Management package
[20211215.004903_builder_junos_211_r3]",
"name": "jservices-hcm"
},
{
"comment": "JUNOS Services IDS [20211215.004903_builder_junos_211_r3]",
"name": "jservices-ids"
},
{
"comment": "JUNOS Services IPSec [20211215.004903_builder_junos_211_r3]",
"name": "jservices-ipsec"
},
{
"comment": "JUNOS Services Jflow Container package
[20211215.004903_builder_junos_211_r3]",
"name": "jservices-jflow"
},
{
"comment": "JUNOS Services LL-PDF Container package
[20211215.004903_builder_junos_211_r3]",
"name": "jservices-llpdf"
},
{
"comment": "JUNOS Services Logging Report Framework package
[20211215.004903_builder_junos_211_r3]",
"name": "jservices-lrf"
},
{
"comment": "JUNOS Services Mobile Subscriber Service Container package
[20211215.004903_builder_junos_211_r3]",
"name": "jservices-mss"
},
{
"comment": "JUNOS Services MobileNext Software package
[20211215.004903_builder_junos_211_r3]",
"name": "jservices-mobile"
},
{
"comment": "JUNOS Services NAT [20211215.004903_builder_junos_211_r3]",
"name": "jservices-nat"
},
{
"comment": "JUNOS Services PCEF package [20211215.004903_builder_junos_211_r3]",
"name": "jservices-pcef"
},
{
"comment": "JUNOS Services RPM [20211215.004903_builder_junos_211_r3]",
```

```
"name": "jservices-rpm"
},
{
"comment": "JUNOS Services RTCOM [20211215.004903_builder_junos_211_r3]",
"name": "jservices-rtcom"
},
{
"comment": "JUNOS Services SOFTWIRE [20211215.004903_builder_junos_211_r3]",
"name": "jservices-softwire"
},
{
"comment": "JUNOS Services SSL [20211215.004903_builder_junos_211_r3]",
"name": "jservices-ssl"
},
{
"comment": "JUNOS Services Stateful Firewall
[20211215.004903_builder_junos_211_r3]",
"name": "jservices-sfw"
},
{
"comment": "JUNOS Services TCP-LOG [20211215.004903_builder_junos_211_r3]",
"name": "jservices-tcp-log"
},
{
"comment": "JUNOS Services TLB Service PIC package
[20211215.004903_builder_junos_211_r3]",
"name": "jservices-traffic-dird"
},
{
"comment": "JUNOS Services Telemetry [20211215.004903_builder_junos_211_r3]",
"name": "jservices-telemetry"
},
{
"comment": "JUNOS Services URL Filter package
[20211215.004903_builder_junos_211_r3]",
"name": "jservices-urlf"
},
{
"comment": "JUNOS appidd-mx application-identification daemon
[20211215.004903_builder_junos_211_r3]",
"name": "junos-appidd"
},
{
"comment": "JUNOS common platform support
[20211215.004903_builder_junos_211_r3]",
"name": "junos-platform"
},
{
"comment": "JUNOS daemons [20211215.004903_builder_junos_211_r3]",
"name": "junos-daemons"
},
{
"comment": "JUNOS jail runtime [20211011.a9cfe96_builder_stable_11]",
"name": "jail-runtime"
},
{
"comment": "JUNOS jfirmware [20211215.004903_builder_junos_211_r3]",
"name": "jfirmware"
},
{
"comment": "JUNOS libs [20211215.004903_builder_junos_211_r3]",
"name": "junos-libs"
```

```
},
{
"comment": "JUNOS libs compat32 [20211215.004903_builder_junos_211_r3]",
"name": "junos-libs-compat32"
},
{
"comment": "JUNOS modules [20211215.004903_builder_junos_211_r3]",
"name": "junos-modules"
},
{
"comment": "JUNOS mtx Data Plane Crypto Support
[20211215.004903_builder_junos_211_r3]",
"name": "junos-dp-crypto-support-platform"
},
{
"comment": "JUNOS mtx network modules [20211215.004903_builder_junos_211_r3]",
"name": "junos-net-platform"
},
{
"comment": "JUNOS mx daemons [20211215.004903_builder_junos_211_r3]",
"name": "junos-daemons-platform"
},
{
"comment": "JUNOS mx libs [20211215.004903_builder_junos_211_r3]",
"name": "junos-libs-platform"
},
{
"comment": "JUNOS mx libs compat32 [20211215.004903_builder_junos_211_r3]",
"name": "junos-libs-compat32-platform"
},
{
"comment": "JUNOS mx modules [20211215.004903_builder_junos_211_r3]",
"name": "junos-modules-platform"
},
{
"comment": "JUNOS mx runtime [20211215.004903_builder_junos_211_r3]",
"name": "junos-runtime-platform"
},
{
"comment": "JUNOS na telemetry [21.1R3.11]",
"name": "na-telemetry"
},
{
"comment": "JUNOS network stack and utilities
[20211215.004903_builder_junos_211_r3]",
"name": "netstack"
},
{
"comment": "JUNOS probe utility [20211215.004903_builder_junos_211_r3]",
"name": "junos-probe"
},
{
"comment": "JUNOS py base [20211215.004903_builder_junos_211_r3]",
"name": "py-base"
},
{
"comment": "JUNOS py extensions [20211215.004903_builder_junos_211_r3]",
"name": "py-extensions"
},
{
"comment": "JUNOS runtime [20211215.004903_builder_junos_211_r3]",
"name": "junos-runtime"
```

```
},
{
"comment": "JUNOS sflow mx [20211215.004903_builder_junos_211_r3]",
"name": "sflow-platform"
},
{
"comment": "Redis [20211215.004903_builder_junos_211_r3]",
"name": "junos-redis"
}
],
"status": "success"
}
}
```


Parent topic

Juniper MX MOP actions

Juniper MX Ping Check MOP Action

Tests end-to-end network reachability. The target should be reachable with zero packet loss. Any packet loss causes the check to fail.

Refer to this topic to review the schema, input details, and output examples for Ping Check.

Action details
Table 53: Ping Check details

Field	Value
Sr. No.	11
Device Family	MX Series
MOP action Name	Ping Check
RPC Call	rpc-ping
Usage	Tests end-to-end network reachability. The target should be reachable with zero packet loss. Any packet loss causes the check to fail.

Input Schema

```
{
"ping": {
"address": "string, required",
"count": "number, optional, default: 1",
"verbose": "string, optional, default: ''"
}
}
```

Input Example

```
{
"ping": {
```

```
"address": "10.0.0.1",
"count": 5,
"verbose": ""
}
}
```

Output

```
{
"Data": {
"message": "Ping check passed. Success rate is 100 percent.",
"status": "success"
}
}
```

Parent topic

Juniper MX MOP actions

Juniper MX distribute image MOP Action

Copies the software image to the device. The file should be transferred successfully; the check fails if the copy command returns an error.

Refer to this topic to review the schema, input details, and output examples for distribute image.

Action details
Table 54: distribute image details

Field	Value
Sr. No.	12
Device Family	MX Series
MOP action Name	distribute image
RPC Call	rpc-file-copy
Usage	Copies the software image to the device. The file should be transferred successfully; the check fails if the copy command returns an error.

Input Schema

```
{
"distribute": {
"fpd": "boolean, optional, default: false",
"vrfName": "string, optional",
"location": "string, optional",
"isISSUUpgrade": "boolean, optional, default: false",
"swimPreference": {
"copyByServer": "boolean, optional, default: false",
"imageTransferProtocolOrder": [
{
"protocol": "string, required, enum: scp|sftp|ftp|tftp|http|telnet|ssh"
```

```
}
]
}
}
}
```

Input Example

```
{
"distribute": {
"fpd": false,
"isISSUUpgrade": false,
"swimPreference": {
"copyByServer": false
}
}
}
```

Output

```
{
"Data": {
"message": "Distribution of junos-install-mx-x86-64-21.1R3.11.tgz is
successful.",
"status": "success"
}
}
```

Parent topic

Juniper MX MOP actions

Juniper MX activate image MOP Action

Installs a new image or activates from the rollback/installed version list. The device should reboot successfully with the new image; the check fails if the reboot does not complete.

Refer to this topic to review the schema, input details, and output examples for activate image.

Action details
Table 55: activate image details

Field	Value
Sr. No.	13
Device Family	MX Series
MOP action Name	activate image
RPC Call	New image: rpc-request-package-add
Usage	Installs a new image or activates from the rollback/installed version list. The device should reboot successfully with the new image; the check fails if the reboot does not complete.

Input Schema

```
{
"activate": {
"fpd": "boolean, optional, default: false",
"location": "string, optional",
"isISSUUpgrade": "boolean, optional, default: false",
"configuredRepo": "string, optional"
}
}
```

Input Example

```
{
"activate": {
"fpd": false,
"isISSUUpgrade": false
}
}
```

Output

```
{
"Data": {
"message": "Activation of junos-install-mx-x86-64-21.1R3.11.tgz is successful.
 SSH key has been fetched successfully. Device has been synced successfully.",
"status": "success"
}
}
```

Parent topic

Juniper MX MOP actions

Generic MOP workflows

Generic MOP workflows provide reusable workflow patterns, schemas, and input/output examples for device and service operations.

Refer to this topic to review generic MOP workflow patterns, schemas, and input/output examples.

Overview

The Generic MOP framework provides a comprehensive set of reusable workflows that enable users to automate device operations and service management tasks.

These workflows capture common patterns and logic frequently used in device / network operations, allowing operators to execute complex procedures through standardized, configurable workflow templates.

The Generic MOP approach abstracts common operational patterns into parameterized workflows, enabling users to focus on defining their specific input configurations rather than implementing the underlying execution logic.

This feature ensures standardization by using consistent patterns across different operations and promotes reusability through a template-based approach, significantly reducing development time. It offers flexibility through configurable parameters for diverse use cases.

Architecture benefits

- Standardization: Consistent patterns across all network operations

- Reusability: Template-based approach reduces development time
- Reliability: Built-in error handling, timeouts, and validation
- Flexibility: Configurable parameters adapt to various use cases
- Maintainability: Centralized logic updates benefit all implementations
- Observability: Comprehensive logging and status reporting

Integration patterns

These Generic MOP workflows are designed to work together, enabling complex automation scenarios through workflow composition:

- Sequential Composition: Chain multiple workflows for complex procedures
- Conditional Branching: Use pause workflows for decision points
- Data Flow Integration: Pass results between workflows using standard data formats
- Error Recovery: Implement fallback procedures using multiple workflow combinations

This framework provides the foundation for building sophisticated network automation solutions while maintaining simplicity and reliability in execution.

Workflow overview

Table 56: Generic MOP workflow overview

Sr. No.	MOP action name	Description
1	run-and-review-cwm-sol	This workflow automates the execution of device commands, validates output using pre-defined templates and filters, handles errors and retries, and ensures reliable results and output evaluation. We can execute any operation on the device then validate it using the user-defined expressions.
2	wait-until-cwm-sol	This workflow is designed to execute a device command repeatedly in a loop until a specified evaluation condition is met or the operation times out. The purpose of this device is to keep trying to execute an operation on the device till the provided conditions are not met or the timeout is not reached. Within the timeout, as soon as the valid condition is satisfied, the execution ends.

Sr. No.	MOP action name	Description
3	command-chaining-cwm-sol	The workflow executes the an operation and applies filter to the command results to build inputs for the following operation. There are two ways in which this can be executed. If loop-enabled in operation 2 is provided as true, then the output of operation 1 give an array of length n. This array would be used as inputs, for n operations, with all using the operation defined in op2ration 2. If loop-enabled in operation 2 is false or is not provided, then starting from operation 1, the output of the preceding operation would be used as input to the following one.
4	nso-service-deployment-cwm-sol	This workflow automates the deployment of Network Service Orchestrator (NSO) services with built-in safety mechanisms, dry-run capabilities, and error handling for both new service creation and existing service updates.
5	pause-mop-cwm-sol	This workflow provides a manual checkpoint mechanism for Method of Procedure (MOP) execution, allowing users to pause automation workflows for review, verification, and decision-making before continuing with subsequent operations.
6	sleep-cwm-sol.	This is a workflow to sleep for a specified amount of time. The user needs to provide the number of seconds the workflow needs to sleep as input and the workflow would remain idle for that period of time.

Parent topic

MOP action reference overview

Related topics

Use these topics to review related MOP action reference information.

- run-and-review-cwm-sol
- wait-until-cwm-sol
- command-chaining-cwm-sol

- nso-service-deployment-cwm-sol
- pause-mop-cwm-sol
- sleep-cwm-sol.

run-and-review-cwm-sol Workflow

This workflow automates the execution of device commands, validates output using pre-defined templates and filters, handles errors and retries, and ensures reliable results and output evaluation. We can execute any operation on the device then validate it using the user-defined expressions.

Refer to this topic to review the schema and input/output examples for run-and-review-cwm-sol.

Workflow details
Table 57: run-and-review-cwm-sol details

Field	Value
Sr. No.	1
MOP action name	run-and-review-cwm-sol
Description	This workflow automates the execution of device commands, validates output using pre-defined templates and filters, handles errors and retries, and ensures reliable results and output evaluation. We can execute any operation on the device then validate it using the user-defined expressions.

Schema

```
{
"operation": {
"method": "string (required) - execCli|get|post|delete",
"command": "string (required for execCli) - CLI command",
"path": "string (required for get/post/delete) - RESTCONF path",
"payload": "string (required for post) - request body",
"template": "string (required for execCli) - TextFSM template",
"filter": "string (optional) - jq filter on parsed output",
"delay": "integer (optional) - delay in seconds",
"compare": "boolean (optional) - enable stash comparison",
"expressions": [
{
"condition": "string (required) - jq condition",
"message": "string (required) - result message",
"status": "string (required) - success|warning|failed"
}
]
},
"deviceSelection": {
"jsonPath": "string (optional) - JSON path for device filtering"
}
}
```

Input and Output Example

```
Input passed without stash. Method passed as post.
Input
{
"app-data": {
"device": {
"name": "junos-24",
"vendor": "Juniper Networks"
},
"resource": "NSO-243"
},
"operation": {
"expressions": [
{
"condition": "(.\"junos-rpc:output\".\"chassis-inventory\".chassis | length
== 0) or (.hasDifference == true)",
"message": "Chassis hardware details check failed.",
"status": "warning"
}
],
"method": "post",
"path":
"restconf/operations/devices/device=${.\"app-data\".device.name}/rpc/rpc-get-chassis-inventory/get-chassis-inventory"
}
}
Output
{
"compareOutput": "There are no issues with the data",
"conditionMatch": "true",
"message": "Chassis hardware details check passed.",
"stash": {
"junos-rpc:output": {
"chassis-inventory": {
"chassis": [
{
"chassis-module": [
{
"description": "MX960 Backplane",
"model-number": "CHAS-BP-MX960-S",
"name": "Midplane",
"part-number": "710-013698",
"serial-number": "TR1019",
"version": "REV 03"
},
{
"description": "Front Panel Display",
"model-number": "CRAFT-MX960-S",
"name": "FPM Board",
"part-number": "710-014974",
"serial-number": "KD5830",
"version": "REV 03"
},
{
"description": "Power Distribution Module",
"name": "PDM",
"part-number": "740-013110",
"serial-number": "QCS11405041",
"version": "Rev 03"
},
{
"description": "PS 1.7kW; 200-240VAC in",
"model-number": "PWR-MX960-AC-S",
```

```
"name": "PEM 0",
"part-number": "740-013682",
"serial-number": "QCS1127401K",
"version": "Rev 03"
},
{
"model-number": "",
"name": "PEM 3"
},
{
"clei-code": "COUCALBBAA",
"description": "RE-S-1800x2",
"model-number": "RE-S-1800X2-16G-S",
"name": "Routing Engine 0",
"part-number": "740-031114",
"serial-number": "9009065630",
"version": "REV 05"
},
{
"clei-code": "COUCARCBAA",
"description": "Enhanced MX SCB",
"model-number": "SCBE-MX-S",
"name": "CB 0",
"part-number": "750-031391",
"serial-number": "ZN5334",
"version": "REV 14"
},
{
"chassis-sub-module": [
{
"description": "DPC PMB",
"name": "CPU",
"part-number": "710-013713",
"serial-number": "KA8096",
"version": "REV 06"
}
],
"description": "DPCE 4x 10GE R",
"model-number": "DPCE-R-4XGE-XFP",
"name": "FPC 3",
"part-number": "750-018124",
"serial-number": "KA8605",
"version": "REV 04"
},
{
"chassis-sub-module": [
{
"description": "DPC PMB",
"name": "CPU",
"part-number": "710-013713",
"serial-number": "KC3439",
"version": "REV 06"
}
],
"description": "DPC 40x 1GE R",
"model-number": "DPC-R-40GE-SFP",
"name": "FPC 10",
"part-number": "710-013699",
"serial-number": "KB6305",
"version": "REV 07"
},
{
```

```
"description": "Fan Tray",
"model-number": "FFANTRAY-MX960-S",
"name": "Fan Tray 0",
"part-number": "740-014971",
"serial-number": "TP3538",
"version": "REV 04"
},
{
"description": "Fan Tray",
"model-number": "FFANTRAY-MX960-S",
"name": "Fan Tray 1",
"part-number": "740-014971",
"serial-number": "TP3472",
"version": "REV 04"
}
],
"description": "MX960",
"name": "Chassis",
"serial-number": "JN10C82F8AFA"
}
]
}
}
},
"status": "success"
}
```


2. Input passed with unchanged stash. Method passed as post.

Input


```
{
"app-data": {
"device": {
"name": "junos-24",
"vendor": "Juniper Networks"
},
"resource": "NSO-243"
},
"compare": true,
"operation": {
"expressions": [
{
"condition": "(.\"junos-rpc:output\".\"chassis-inventory\".chassis | length
== 0) or (.hasDifference == true)",
"message": "Chassis hardware details check failed.",
"status": "warning"
}
],
"method": "post",
"path":
"restconf/operations/devices/device=${.\"app-data\".device.name}/rpc/rpc-get-chassis-inventory/get-chassis-inventory"
},
"stash": {
"junos-rpc:output": {
"chassis-inventory": {
"chassis": [
{
"chassis-module": [
{
"description": "MX960 Backplane",
"model-number": "CHAS-BP-MX960-S",
"name": "Midplane",
"part-number": "710-013698",
"serial-number": "TR1019",
```

```
"version": "REV 03"
},
{
"description": "Front Panel Display",
"model-number": "CRAFT-MX960-S",
"name": "FPM Board",
"part-number": "710-014974",
"serial-number": "KD5830",
"version": "REV 03"
},
{
"description": "Power Distribution Module",
"name": "PDM",
"part-number": "740-013110",
"serial-number": "QCS11405041",
"version": "Rev 03"
},
{
"description": "PS 1.7kW; 200-240VAC in",
"model-number": "PWR-MX960-AC-S",
"name": "PEM 0",
"part-number": "740-013682",
"serial-number": "QCS1127401K",
"version": "Rev 03"
},
{
"model-number": "",
"name": "PEM 3"
},
{
"clei-code": "COUCALBBAA",
"description": "RE-S-1800x2",
"model-number": "RE-S-1800X2-16G-S",
"name": "Routing Engine 0",
"part-number": "740-031114",
"serial-number": "9009065630",
"version": "REV 05"
},
{
"clei-code": "COUCARCBAA",
"description": "Enhanced MX SCB",
"model-number": "SCBE-MX-S",
"name": "CB 0",
"part-number": "750-031391",
"serial-number": "ZN5334",
"version": "REV 14"
},
{
"chassis-sub-module": [
{
"description": "DPC PMB",
"name": "CPU",
"part-number": "710-013713",
"serial-number": "KA8096",
"version": "REV 06"
}
],
"description": "DPCE 4x 10GE R",
"model-number": "DPCE-R-4XGE-XFP",
"name": "FPC 3",
"part-number": "750-018124",
"serial-number": "KA8605",
```

```
"version": "REV 04"
},
{
"chassis-sub-module": [
{
"description": "DPC PMB",
"name": "CPU",
"part-number": "710-013713",
"serial-number": "KC3439",
"version": "REV 06"
}
],
"description": "DPC 40x 1GE R",
"model-number": "DPC-R-40GE-SFP",
"name": "FPC 10",
"part-number": "710-013699",
"serial-number": "KB6305",
"version": "REV 07"
},
{
"description": "Fan Tray",
"model-number": "FFANTRAY-MX960-S",
"name": "Fan Tray 0",
"part-number": "740-014971",
"serial-number": "TP3538",
"version": "REV 04"
},
{
"description": "Fan Tray",
"model-number": "FFANTRAY-MX960-S",
"name": "Fan Tray 1",
"part-number": "740-014971",
"serial-number": "TP3472",
"version": "REV 04"
}
],
"description": "MX960",
"name": "Chassis",
"serial-number": "JN10C82F8AFA"
}
]
}
}
}
}
Output
{
"compareOutput": {
"differenceSummary": "No differences",
"hasDifference": false
},
"conditionMatch": "true",
"message": "Chassis hardware details check failed.",
"stash": {
"junos-rpc:output": {
"chassis-inventory": {
"chassis": [
{
"chassis-module": [
{
"description": "MX960 Backplane",
"model-number": "CHAS-BP-MX960-S",
```

```
"name": "Midplane",
"part-number": "710-013698",
"serial-number": "TR1019",
"version": "REV 03"
},
{
"description": "Front Panel Display",
"model-number": "CRAFT-MX960-S",
"name": "FPM Board",
"part-number": "710-014974",
"serial-number": "KD5830",
"version": "REV 03"
},
{
"description": "Power Distribution Module",
"name": "PDM",
"part-number": "740-013110",
"serial-number": "QCS11405041",
"version": "Rev 03"
},
{
"description": "PS 1.7kW; 200-240VAC in",
"model-number": "PWR-MX960-AC-S",
"name": "PEM 0",
"part-number": "740-013682",
"serial-number": "QCS1127401K",
"version": "Rev 03"
},
{
"model-number": "",
"name": "PEM 3"
},
{
"clei-code": "COUCALBBAA",
"description": "RE-S-1800x2",
"model-number": "RE-S-1800X2-16G-S",
"name": "Routing Engine 0",
"part-number": "740-031114",
"serial-number": "9009065630",
"version": "REV 05"
},
{
"clei-code": "COUCARCBAA",
"description": "Enhanced MX SCB",
"model-number": "SCBE-MX-S",
"name": "CB 0",
"part-number": "750-031391",
"serial-number": "ZN5334",
"version": "REV 14"
},
{
"chassis-sub-module": [
{
"description": "DPC PMB",
"name": "CPU",
"part-number": "710-013713",
"serial-number": "KA8096",
"version": "REV 06"
}
],
"description": "DPCE 4x 10GE R",
"model-number": "DPCE-R-4XGE-XFP",
```

```
"name": "FPC 3",
"part-number": "750-018124",
"serial-number": "KA8605",
"version": "REV 04"
},
{
"chassis-sub-module": [
{
"description": "DPC PMB",
"name": "CPU",
"part-number": "710-013713",
"serial-number": "KC3439",
"version": "REV 06"
}
],
"description": "DPC 40x 1GE R",
"model-number": "DPC-R-40GE-SFP",
"name": "FPC 10",
"part-number": "710-013699",
"serial-number": "KB6305",
"version": "REV 07"
},
{
"description": "Fan Tray",
"model-number": "FFANTRAY-MX960-S",
"name": "Fan Tray 0",
"part-number": "740-014971",
"serial-number": "TP3538",
"version": "REV 04"
},
{
"description": "Fan Tray",
"model-number": "FFANTRAY-MX960-S",
"name": "Fan Tray 1",
"part-number": "740-014971",
"serial-number": "TP3472",
"version": "REV 04"
}
],
"description": "MX960",
"name": "Chassis",
"serial-number": "JN10C82F8AFA"
}
]
}
}
},
"status": "success"
}
```


3. Input passed with changed stash. Method passed as post.

Input


```
{
"app-data": {
"device": {
"name": "junos-24",
"vendor": "Juniper Networks"
},
"resource": "NSO-243"
},
"compare": true,
"operation": {
"expressions": [
```

```
{
"condition": "(.\"junos-rpc:output\".\"chassis-inventory\".chassis | length
== 0) or (.hasDifference == true)",
"message": "Chassis hardware details check failed.",
"status": "warning"
}
],
"method": "post",
"path":
"restconf/operations/devices/device=${.\"app-data\".device.name}/rpc/rpc-get-chassis-inventory/get-chassis-inventory"
},
"stash": {
"junos-rpc:output": {
"chassis-inventory": {
"chassis": [
{
"chassis-module": [
{
"description": "MX960 Backplane",
"model-number": "CHAS-BP-MX960-S",
"name": "Midplane",
"part-number": "710-0136981",
"serial-number": "TR1019",
"version": "REV 03"
},
{
"description": "Front Panel Display",
"model-number": "CRAFT-MX960-S",
"name": "FPM Board",
"part-number": "710-014974",
"serial-number": "KD5830",
"version": "REV 031"
},
{
"description": "Power Distribution Module",
"name": "PDM",
"part-number": "740-013110",
"serial-number": "QCS11405041",
"version": "Rev 03"
},
{
"description": "PS 1.7kW; 200-240VAC in",
"model-number": "PWR-MX960-AC-S",
"name": "PEM 0",
"part-number": "740-013682",
"serial-number": "QCS1127401K",
"version": "Rev 03"
},
{
"model-number": "",
"name": "PEM 3"
},
{
"clei-code": "COUCALBBAA",
"description": "RE-S-1800x2",
"model-number": "RE-S-1800X2-16G-S",
"name": "Routing Engine 0",
"part-number": "740-031114",
"serial-number": "9009065630",
"version": "REV 05"
},
{
```

```
"clei-code": "COUCARCBAA",
"description": "Enhanced MX SCB",
"model-number": "SCBE-MX-S",
"name": "CB 0",
"part-number": "750-031391",
"serial-number": "ZN5334",
"version": "REV 14"
},
{
"chassis-sub-module": [
{
"description": "DPC PMB",
"name": "CPU",
"part-number": "710-013713",
"serial-number": "KA8096",
"version": "REV 06"
}
],
"description": "DPCE 4x 10GE R",
"model-number": "DPCE-R-4XGE-XFP",
"name": "FPC 3",
"part-number": "750-018124",
"serial-number": "KA8605",
"version": "REV 04"
},
{
"chassis-sub-module": [
{
"description": "DPC PMB",
"name": "CPU",
"part-number": "710-013713",
"serial-number": "KC3439",
"version": "REV 06"
}
],
"description": "DPC 40x 1GE R",
"model-number": "DPC-R-40GE-SFP",
"name": "FPC 10",
"part-number": "710-013699",
"serial-number": "KB6305",
"version": "REV 07"
},
{
"description": "Fan Tray",
"model-number": "FFANTRAY-MX960-S",
"name": "Fan Tray 0",
"part-number": "740-014971",
"serial-number": "TP3538",
"version": "REV 04"
},
{
"description": "Fan Tray",
"model-number": "FFANTRAY-MX960-S",
"name": "Fan Tray 1",
"part-number": "740-014971",
"serial-number": "TP3472",
"version": "REV 04"
}
],
"description": "MX960",
"name": "Chassis",
"serial-number": "JN10C82F8AFA"
```

```
}
]
}
}
}
}
Output
{
"compareOutput": {
"differenceSummary": "Post-check has 2 differences compared to pre-check.",
"hasDifference": true
},
"conditionMatch": "true",
"message": "Chassis hardware details check failed.",
"stash": {
"junos-rpc:output": {
"chassis-inventory": {
"chassis": [
{
"chassis-module": [
{
"description": "MX960 Backplane",
"model-number": "CHAS-BP-MX960-S",
"name": "Midplane",
"part-number": "710-013698",
"serial-number": "TR1019",
"version": "REV 03"
},
{
"description": "Front Panel Display",
"model-number": "CRAFT-MX960-S",
"name": "FPM Board",
"part-number": "710-014974",
"serial-number": "KD5830",
"version": "REV 03"
},
{
"description": "Power Distribution Module",
"name": "PDM",
"part-number": "740-013110",
"serial-number": "QCS11405041",
"version": "Rev 03"
},
{
"description": "PS 1.7kW; 200-240VAC in",
"model-number": "PWR-MX960-AC-S",
"name": "PEM 0",
"part-number": "740-013682",
"serial-number": "QCS1127401K",
"version": "Rev 03"
},
{
"model-number": "",
"name": "PEM 3"
},
{
"clei-code": "COUCALBBAA",
"description": "RE-S-1800x2",
"model-number": "RE-S-1800X2-16G-S",
"name": "Routing Engine 0",
"part-number": "740-031114",
"serial-number": "9009065630",
```

```
"version": "REV 05"
},
{
"clei-code": "COUCARCBAA",
"description": "Enhanced MX SCB",
"model-number": "SCBE-MX-S",
"name": "CB 0",
"part-number": "750-031391",
"serial-number": "ZN5334",
"version": "REV 14"
},
{
"chassis-sub-module": [
{
"description": "DPC PMB",
"name": "CPU",
"part-number": "710-013713",
"serial-number": "KA8096",
"version": "REV 06"
}
],
"description": "DPCE 4x 10GE R",
"model-number": "DPCE-R-4XGE-XFP",
"name": "FPC 3",
"part-number": "750-018124",
"serial-number": "KA8605",
"version": "REV 04"
},
{
"chassis-sub-module": [
{
"description": "DPC PMB",
"name": "CPU",
"part-number": "710-013713",
"serial-number": "KC3439",
"version": "REV 06"
}
],
"description": "DPC 40x 1GE R",
"model-number": "DPC-R-40GE-SFP",
"name": "FPC 10",
"part-number": "710-013699",
"serial-number": "KB6305",
"version": "REV 07"
},
{
"description": "Fan Tray",
"model-number": "FFANTRAY-MX960-S",
"name": "Fan Tray 0",
"part-number": "740-014971",
"serial-number": "TP3538",
"version": "REV 04"
},
{
"description": "Fan Tray",
"model-number": "FFANTRAY-MX960-S",
"name": "Fan Tray 1",
"part-number": "740-014971",
"serial-number": "TP3472",
"version": "REV 04"
}
],
```

```
"description": "MX960",
"name": "Chassis",
"serial-number": "JN10C82F8AFA"
}
]
}
}
},
"status": "warning"
}
4. Device passed in app-data and the method is execCli.
Input
{
"app-data": {
"data": {
"imageSize": 14412823,
"imageVersion": "",
"softwareImages": [
"cisco8k.tar"
]
},
"device": {
"host": "spitfire248.cisco.com",
"ip": "172.22.141.248",
"label": "spitfire248.cisco.com",
"name": "spit-248",
"productSeries": "Cisco 8000 Series Routers",
"productType": "Cisco 8201 Router",
"resource": "nso-173",
"softwareType": "IOS XR",
"softwareVersion": "7.8.2",
"uuid": "b8bf6dbb-7e01-4982-b45f-48ec5f11a585",
"vendor": "Cisco Systems"
},
"executionMode": 1,
"phase": "",
"resource": "nso-173",
"stage": "post"
},
"operation": {
"command": "cfs check",
"compare": true,
"expressions": [
{
"condition": "[.[] | select(. | keys != [\"app-data\"]) | select(.Status !=
\"OK\")] | length > 0",
"message": "configuration file system failed the sanity check.",
"status": "warning"
},
{
"condition": "[.[] | select(. | keys != [\"app-data\"]) | select(.Status !=
\"OK\")] | length == 0",
"message": "configuration file system passed successfully.",
"status": "success"
}
],
"method": "execCli",
"template": "Value Line (.+)\nValue Status (\\S+)\n\nStart\n
^\\s*${Line}\\.\\.\\.${Status} -> Record\n  ^.-> End"
}
}
Output
```

```
{
"compareOutput": "There is no issues with the data",
"conditionMatch": "true",
"message": "configuration file system passed successfully.",
"processedResource": [
"b8bf6dbb-7e01-4982-b45f-48ec5f11a585"
],
"stash": [
{
"Line": "Creating any missing directories in Configuration File system",
"Status": "OK"
},
{
"Line": "Initializing Configuration Version Manager",
"Status": "OK"
},
{
"Line": "Syncing commit database with running configuration",
"Status": "OK"
}
],
"status": "success"
}
```


5. Device passed in deviceSelection as a JSON and the method is execCli.

Input


```
{
"app-data": {
"data": {},
"devices": {
"source": [
{
"host": "spitfire248.cisco.com",
"ip": "172.22.141.248",
"label": "source",
"name": "spit-248",
"productSeries": "Cisco 8000 Series Routers",
"productType": "Cisco 8201 Router",
"resource": "nso-173",
"softwareType": "IOS XR",
"softwareVersion": "7.8.2",
"uuid": "5951a4de-0a6e-4599-9c38-dcb963b82c66",
"vendor": "Cisco Systems"
}
]
},
"executionMode": 2,
"phase": "",
"resource": "nso-173",
"stage": "pre-provision"
},
"deviceSelection": {
"jsonPath": ".app-data.devices.source[0]"
},
"operation": {
"command": "cfs check",
"expressions": [
{
"condition": "[.[] | select(. | keys != [\"app-data\"]) | select(.Status !=
\"OK\")] | length == 0",
"message": "configuration file system passed successfully.",
"status": "success"
},
```

```
{
"condition": "[.[] | select(. | keys != [\"app-data\"]) | select(.Status !=
\"OK\")] | length > 0",
"message": "configuration file system failed the sanity check.",
"status": "warning"
}
],
"method": "execCli",
"template": "Value Line (.+)\nValue Status (\\S+)\n\nStart\n
^\\s*${Line}\\.\\.\\.${Status} -> Record\n  ^.-> End"
}
}
Output
{
"compareOutput": "There is no issues with the data",
"conditionMatch": "true",
"message": "configuration file system passed successfully.",
"processedResource": [
"5951a4de-0a6e-4599-9c38-dcb963b82c66"
],
"stash": [
{
"Line": "Creating any missing directories in Configuration File system",
"Status": "OK"
},
{
"Line": "Initializing Configuration Version Manager",
"Status": "OK"
},
{
"Line": "Syncing commit database with running configuration",
"Status": "OK"
}
],
"status": "success"
}
6. Device passed in deviceSelection as an array and the method is execCli.
Input
{
"app-data": {
"resource": "NSO-243"
},
"deviceSelection": {
"jsonPath": ".devices[0]"
},
"devices": [
{
"name": "spit-248",
"vendor": "Cisco Systems"
}
],
"operation": {
"command": "show interfaces summary",
"expressions": [
{
"condition": "[.[]] | length > 0",
"message": "Interface summary check successful.",
"status": "success"
}
],
"method": "execCli",
"template": "Value InterfaceType (.*?)\nValue Total (\\d+)\nValue Up
```

```
(\\d+)\nValue Down (\\d+)\nValue AdminDown (\\d+)\n\nStart\n
^${InterfaceType}\\s+${Total}\\s+${Up}\\s+${Down}\\s+${AdminDown} -> Record"
}
}
Output
{
"compareOutput": "There are no issues with the data",
"conditionMatch": "true",
"message": "Interface summary check successful.",
"stash": [
{
"AdminDown": "0",
"Down": "0",
"InterfaceType": "IFT_ETHERNET",
"Total": "1",
"Up": "1"
},
{
"AdminDown": "0",
"Down": "0",
"InterfaceType": "IFT_LOOPBACK",
"Total": "1001",
"Up": "1001"
},
{
"AdminDown": "0",
"Down": "0",
"InterfaceType": "IFT_NULL",
"Total": "1",
"Up": "1"
},
{
"AdminDown": "12",
"Down": "0",
"InterfaceType": "IFT_HUNDREDGE",
"Total": "12",
"Up": "0"
},
{
"AdminDown": "24",
"Down": "0",
"InterfaceType": "IFT_FOURHUNDREDGE",
"Total": "24",
"Up": "0"
},
{
"AdminDown": "36",
"Down": "0",
"InterfaceType": "ALL TYPES",
"Total": "1039",
"Up": "1003"
}
],
"status": "success"
}
7. Pre-check with get method:
Input
{
"app-data": {
"data": {
"imageSize": 14412823,
"imageVersion": "",
```

```
"softwareImages": [
"Invalid_ciscoXE.tar"
]
},
"device": {
"host": "cat9300-153",
"ip": "192.168.66.153",
"label": "cat9300-153",
"name": "catalyst9300",
"productSeries": "Cisco Catalyst 9300 Series Switches",
"productType": "Cisco Catalyst 9300 Switch",
"resource": "nso-173",
"softwareType": "IOS XE",
"softwareVersion": "17.12.4",
"uuid": "deecbb44-e9b9-42f3-99bb-0083957291f0",
"vendor": "Cisco Systems"
},
"executionMode": 1,
"phase": "",
"resource": "nso-173",
"stage": "pre"
},
"operation": {
"expressions": [
{
"condition": ".\"tailf-ned-cisco-ios-stats:version\"== \"17.12.04\"",
"message": "Version check passed successfully.",
"status": "success"
}
],
"method": "get",
"path":
"restconf/data/tailf-ncs:devices/device=${.\"app-data\".device.name}/live-status/tailf-ned-cisco-ios-stats:version/version"
}
}
Output
{
"compareOutput": "There is no issues with the data",
"conditionMatch": "true",
"message": "Version check passed successfully.",
"processedResource": [
"deecbb44-e9b9-42f3-99bb-0083957291f0"
],
"stash": {
"tailf-ned-cisco-ios-stats:version": "17.12.04"
},
"status": "success"
}
8. Post-check with get method:
Input
{
"app-data": {
"data": {
"imageSize": 14412823,
"imageVersion": "",
"softwareImages": [
"Invalid_ciscoXE.tar"
]
},
"device": {
"host": "cat9300-153",
"ip": "192.168.66.153",
```

```
"label": "cat9300-153",
"name": "catalyst9300",
"productSeries": "Cisco Catalyst 9300 Series Switches",
"productType": "Cisco Catalyst 9300 Switch",
"resource": "nso-173",
"softwareType": "IOS XE",
"softwareVersion": "17.12.4",
"uuid": "deecbb44-e9b9-42f3-99bb-0083957291f0",
"vendor": "Cisco Systems"
},
"executionMode": 1,
"phase": "",
"resource": "nso-173",
"stage": "post"
},
"operation": {
"compare": true,
"expressions": [
{
"condition": ".\"tailf-ned-cisco-ios-stats:version\"== \"17.12.04\" and
.hasDifference== false",
"message": "Version check passed successfully.",
"status": "success"
}
],
"method": "get",
"path":
"restconf/data/tailf-ncs:devices/device=${.\"app-data\".device.name}/live-status/tailf-ned-cisco-ios-stats:version/version"
},
"stash": {
"tailf-ned-cisco-ios-stats:version": "17.12.04"
}
}
Output
{
"compareOutput": {
"differenceSummary": "No differences",
"hasDifference": false
},
"conditionMatch": "true",
"message": "Version check passed successfully.",
"processedResource": [
"deecbb44-e9b9-42f3-99bb-0083957291f0"
],
"stash": {
"tailf-ned-cisco-ios-stats:version": "17.12.04"
},
"status": "success"
}
9. Pre-check with execCli method:
Input
{
"app-data": {
"data": {
"imageSize": 14412823,
"imageVersion": "",
"softwareImages": [
"cisco8k.tar"
]
},
"device": {
"host": "spitfire248.cisco.com",
```

```
"ip": "172.22.141.248",
"label": "spitfire248.cisco.com",
"name": "spit-248",
"productSeries": "Cisco 8000 Series Routers",
"productType": "Cisco 8201 Router",
"resource": "nso-173",
"softwareType": "IOS XR",
"softwareVersion": "7.8.2",
"uuid": "b8bf6dbb-7e01-4982-b45f-48ec5f11a585",
"vendor": "Cisco Systems"
},
"executionMode": 1,
"phase": "",
"resource": "nso-173",
"stage": "pre"
},
"operation": {
"command": "cfs check",
"expressions": [
{
"condition": "[.[] | select(. | keys != [\"app-data\"]) | select(.Status !=
\"OK\")] | length == 0",
"message": "configuration file system passed successfully.",
"status": "success"
},
{
"condition": "[.[] | select(. | keys != [\"app-data\"]) | select(.Status !=
\"OK\")] | length > 0",
"message": "configuration file system failed the sanity check.",
"status": "warning"
}
],
"method": "execCli",
"template": "Value Line (.+)\nValue Status (\\S+)\n\nStart\n
^\\s*${Line}\\.\\.\\.${Status} -> Record\n  ^.-> End"
}
}
Output
{
"compareOutput": "There is no issues with the data",
"conditionMatch": "true",
"message": "configuration file system passed successfully.",
"processedResource": [
"b8bf6dbb-7e01-4982-b45f-48ec5f11a585"
],
"stash": [
{
"Line": "Creating any missing directories in Configuration File system",
"Status": "OK"
},
{
"Line": "Initializing Configuration Version Manager",
"Status": "OK"
},
{
"Line": "Syncing commit database with running configuration",
"Status": "OK"
}
],
"status": "success"
}
10. Post-check with execCli method:
```

```
Input
{
"app-data": {
"data": {
"imageSize": 14412823,
"imageVersion": "",
"softwareImages": [
"cisco8k.tar"
]
},
"device": {
"host": "spitfire248.cisco.com",
"ip": "172.22.141.248",
"label": "spitfire248.cisco.com",
"name": "spit-248",
"productSeries": "Cisco 8000 Series Routers",
"productType": "Cisco 8201 Router",
"resource": "nso-173",
"softwareType": "IOS XR",
"softwareVersion": "7.8.2",
"uuid": "b8bf6dbb-7e01-4982-b45f-48ec5f11a585",
"vendor": "Cisco Systems"
},
"executionMode": 1,
"phase": "",
"resource": "nso-173",
"stage": "post"
},
"operation": {
"command": "cfs check",
"compare": true,
"expressions": [
{
"condition": "([.[] | select(. | keys != [\"app-data\", \"hasDifference\"]) |
 select(.Status != \"OK\")] | length == 0) and (.[] | .hasDifference == false)",
"message": "configuration file system passed successfully.",
"status": "success"
},
{
"condition": "([.[] | select(. | keys != [\"app-data\", \"hasDifference\"]) |
 select(.Status != \"OK\")] | length > 0) or (.[] | .hasDifference == true)",
"message": "configuration file system failed the sanity check.",
"status": "warning"
}
],
"method": "execCli",
"template": "Value Line (.+)\nValue Status (\\S+)\n\nStart\n
^\\s*${Line}\\.\\.\\.${Status} -> Record\n  ^.-> End"
},
"stash": [
{
"Line": "Creating any missing directories in Configuration File system",
"Status": "OK"
},
{
"Line": "Initializing Configuration Version Manager",
"Status": "OK"
},
{
"Line": "Syncing commit database with running configuration",
"Status": "OK"
}
```

```
]
}
Output
{
"compareOutput": {
"differenceSummary": "No differences",
"hasDifference": false
},
"conditionMatch": "true",
"message": "configuration file system passed successfully.",
"processedResource": [
"b8bf6dbb-7e01-4982-b45f-48ec5f11a585"
],
"stash": [
{
"Line": "Creating any missing directories in Configuration File system",
"Status": "OK"
},
{
"Line": "Initializing Configuration Version Manager",
"Status": "OK"
},
{
"Line": "Syncing commit database with running configuration",
"Status": "OK"
}
],
"status": "success"
}
11. Post with payload:
Input
{
"app-data": {
"command": "cfs check",
"data": {
"imageSize": 14412823,
"imageVersion": "",
"softwareImages": [
"cisco8k.tar"
]
},
"device": {
"host": "spitfire250.cisco.com",
"ip": "172.22.141.250",
"label": "spitfire250.cisco.com",
"name": "spitfire250",
"productSeries": "Cisco 8000 Series Routers",
"productType": "Cisco 8201 Router",
"resource": "nso-173",
"softwareType": "IOS XR",
"softwareVersion": "24.4.2",
"uuid": "9c10abb5-1396-40ff-ba5c-9f7250f791d1",
"vendor": "Cisco Systems"
},
"phase": "",
"resource": "nso-173",
"stage": "activate"
},
"operation": {
"expressions": [
{
"condition": ". != null and . != \"\"",
```

```
"message": "The command outputs have been captured properly",
"status": "success"
}
],
"method": "post",
"path":
"restconf/operations/devices/device==${.\"app-data\".device.name}/live-status/exec/any",
"payload": "{\n\"args\": \"${.\\\"app-data\\\".command}\"\n}"
}
}
Output
{
"compareOutput": "There is no issues with the data",
"conditionMatch": "true",
"message": "The command outputs have been captured properly",
"processedResource": [
"9c10abb5-1396-40ff-ba5c-9f7250f791d1"
],
"stash": {
"tailf-ned-cisco-ios-xr-stats:output": {
"result": "\r\n\rFri Oct 31 02:54:57.532 PDT\r\n\r\nCreating any missing
directories in Configuration File system...OK\r\nInitializing Configuration
Version Manager...OK\r\nSyncing commit database with running
configuration...OK\r\nRP/0/RP0/CPU0:spitfire250#"
}
},
"status": "success"
}
```


Parent topic

Generic MOP workflows

wait-until-cwm-sol Workflow

This workflow is designed to execute a device command repeatedly in a loop until a specified evaluation condition is met or the operation times out. The purpose of this device is to keep trying to execute an operation on the device till the provided conditions are not met or the timeout is not reached. Within the timeout, as soon as the valid condition is satisfied, the execution ends.

Refer to this topic to review the schema and input/output examples for wait-until-cwm-sol.

Workflow details
Table 58: wait-until-cwm-sol details

Field	Value
Sr. No.	2
MOP action name	wait-until-cwm-sol
Description	This workflow is designed to execute a device command repeatedly in a loop until a specified evaluation condition is met or the operation times out. The purpose of this device is to keep trying to execute an operation on the device till the provided conditions are not met or the timeout is not reached. Within the timeout, as soon as the valid condition is satisfied, the execution ends.

Schema

```
{
"operation": {
"method": "string (required) - execCli|get|post|delete",
"command": "string (required for execCli) - CLI command",
"path": "string (required for get/post/delete) - RESTCONF path",
"payload": "string (required for post) - request body",
"template": "string (required for execCli) - TextFSM template",
"filter": "string (optional) - jq filter on parsed output",
"delay": "integer (required) - polling interval in seconds",
"timeout": "integer (required) - max wait time in seconds",
"compare": "boolean (optional) - enable stash comparison",
"expressions": [
{
"condition": "string (required) - jq condition",
"message": "string (required) - result message",
"status": "string (required) - success|warning|failed"
}
]
},
"deviceSelection": {
"jsonPath": "string (optional) - JSON path for device filtering"
}
}
```

Input and Output Example

```
Device passed in app-data, timeout occurred due to success condition not being
 met.
Input
{
"app-data": {
"data": {
"imageSize": 14412823,
"imageVersion": "",
"softwareImages": [
"cisco8k.tar"
]
},
"device": {
"host": "spitfire248.cisco.com",
"ip": "172.22.141.248",
"label": "spitfire248.cisco.com",
"name": "spit-248",
"productSeries": "Cisco 8000 Series Routers",
"productType": "Cisco 8201 Router",
"resource": "nso-173",
"softwareType": "IOS XR",
"softwareVersion": "7.8.2",
"uuid": "b8bf6dbb-7e01-4982-b45f-48ec5f11a585",
"vendor": "Cisco Systems"
},
"executionMode": 1,
"phase": "",
"resource": "nso-173",
"stage": "pre"
},
"operation": {
"command": "cfs check",
"delay": 10,
```

```
"expressions": [
{
"condition": "[.[] | select(. | keys != [\"app-data\"]) | select(.Status !=
\"OK\")] | length > 0",
"message": "configuration file system failed the sanity check.",
"status": "warning"
}
],
"method": "execCli",
"template": "Value Line (.+)\nValue Status (\\S+)\n\nStart\n
^\\s*${Line}\\.\\.\\.${Status} -> Record\n  ^.-> End",
"timeout": 30
}
}
Output
{
"message": "Loop interrupted due to timeout",
"processedResource": [
"b8bf6dbb-7e01-4982-b45f-48ec5f11a585"
],
"stash": [
{
"Line": "Creating any missing directories in Configuration File system",
"Status": "OK"
},
{
"Line": "Initializing Configuration Version Manager",
"Status": "OK"
},
{
"Line": "Syncing commit database with running configuration",
"Status": "OK"
}
],
"status": "failed"
}
2. Device passed in app-data, execution successful.
Input
{
"app-data": {
"data": {
"imageSize": 14412823,
"imageVersion": "",
"softwareImages": [
"cisco8k.tar"
]
},
"device": {
"host": "spitfire248.cisco.com",
"ip": "172.22.141.248",
"label": "spitfire248.cisco.com",
"name": "spit-248",
"productSeries": "Cisco 8000 Series Routers",
"productType": "Cisco 8201 Router",
"resource": "nso-173",
"softwareType": "IOS XR",
"softwareVersion": "7.8.2",
"uuid": "b8bf6dbb-7e01-4982-b45f-48ec5f11a585",
"vendor": "Cisco Systems"
},
"executionMode": 1,
"phase": "",
```

```
"resource": "nso-173",
"stage": "pre"
},
"operation": {
"command": "cfs check",
"delay": 10,
"expressions": [
{
"condition": "[.[] | select(. | keys != [\"app-data\"]) | select(.Status !=
\"OK\")] | length == 0",
"message": "configuration file system passed successfully.",
"status": "success"
},
{
"condition": "[.[] | select(. | keys != [\"app-data\"]) | select(.Status !=
\"OK\")] | length > 0",
"message": "configuration file system failed the sanity check.",
"status": "warning"
}
],
"method": "execCli",
"template": "Value Line (.+)\nValue Status (\\S+)\n\nStart\n
^\\s*${Line}\\.\\.\\.${Status} -> Record\n  ^.-> End",
"timeout": 30
}
}
Output
{
"message": "configuration file system passed successfully.",
"processedResource": [
"b8bf6dbb-7e01-4982-b45f-48ec5f11a585"
],
"stash": [
{
"Line": "Creating any missing directories in Configuration File system",
"Status": "OK"
},
{
"Line": "Initializing Configuration Version Manager",
"Status": "OK"
},
{
"Line": "Syncing commit database with running configuration",
"Status": "OK"
}
],
"status": "success"
}
3. deviceSelection is not null and success condition is met.
Input
{
"app-data": {
"data": {},
"devices": {
"source": [
{
"host": "spitfire250.cisco.com",
"ip": "172.22.141.250",
"label": "source",
"name": "spitfire250",
"productSeries": "Cisco 8000 Series Routers",
"productType": "Cisco 8201 Router",
```

```
"resource": "nso-173",
"softwareType": "IOS XR",
"softwareVersion": "24.4.2",
"uuid": "2c611a0a-31fc-4a11-80bf-170c9417014c",
"vendor": "Cisco Systems"
}
]
},
"executionMode": 2,
"phase": "",
"resource": "nso-173",
"stage": "pre-provision"
},
"deviceSelection": {
"jsonPath": ".app-data.devices.source[0]"
},
"operation": {
"command": "cfs check",
"delay": 1,
"expressions": [
{
"condition": "[.[] | select(. | keys != [\"app-data\"]) | select(.Status !=
\"OK\")] | length == 0",
"message": "configuration file system passed successfully.",
"status": "success"
},
{
"condition": "[.[] | select(. | keys != [\"app-data\"]) | select(.Status !=
\"OK\")] | length > 0",
"message": "configuration file system failed the sanity check.",
"status": "warning"
}
],
"method": "execCli",
"template": "Value Line (.+)\nValue Status (\\S+)\n\nStart\n
^\\s*${Line}\\.\\.\\.${Status} -> Record\n  ^.-> End",
"timeout": 30
}
}
Output
{
"message": "configuration file system passed successfully.",
"processedResource": [
"2c611a0a-31fc-4a11-80bf-170c9417014c"
],
"stash": [
{
"Line": "Creating any missing directories in Configuration File system",
"Status": "OK"
},
{
"Line": "Initializing Configuration Version Manager",
"Status": "OK"
},
{
"Line": "Syncing commit database with running configuration",
"Status": "OK"
}
],
"status": "success"
}
4. Device passed in app-data, execution successful and method is a post with
```

```
payload:
Input
{
"app-data": {
"data": {
"imageSize": 14412823,
"imageVersion": "",
"softwareImages": [
"cisco8k.tar"
]
},
"device": {
"host": "spitfire248.cisco.com",
"ip": "172.22.141.248",
"label": "spitfire248.cisco.com",
"name": "spit-248",
"productSeries": "Cisco 8000 Series Routers",
"productType": "Cisco 8201 Router",
"resource": "nso-173",
"softwareType": "IOS XR",
"softwareVersion": "7.8.2",
"uuid": "b8bf6dbb-7e01-4982-b45f-48ec5f11a585",
"vendor": "Cisco Systems"
},
"executionMode": 1,
"phase": "",
"resource": "nso-173",
"stage": "pre"
},
"operation": {
"delay": 10,
"expressions": [
{
"condition": ". != null and . != \"\"",
"message": "The command outputs have been captured properly",
"status": "success"
}
],
"method": "post",
"path":
"restconf/operations/devices/device==${.\"app-data\".device.name}/live-status/exec/any",
"payload": "{\n\"args\": \"cfs check\"\n}\n",
"timeout": 30
}
}
Output
{
"message": "The command outputs have been captured properly",
"processedResource": [
"b8bf6dbb-7e01-4982-b45f-48ec5f11a585"
],
"stash": {
"tailf-ned-cisco-ios-xr-stats:output": {
"result": "\r\n\rFri Oct 31 09:38:35.317 PDT\r\n\r\nCreating any missing
directories in Configuration File system...OK\r\nInitializing Configuration
Version Manager...OK\r\nSyncing commit database with running
configuration...OK\r\nRP/0/RP0/CPU0:spitfire248#"
}
},
"status": "success"
}
```

Parent topic

Generic MOP workflows

command-chaining-cwm-sol Workflow

The workflow executes the an operation and applies filter to the command results to build inputs for the following operation. There are two ways in which this can be executed. If loop-enabled in operation 2 is provided as true, then the output of operation 1 give an array of length n. This array would be used as inputs, for n operations, with all using the operation defined in op2ration 2. If loop-enabled in operation 2 is false or is not provided, then starting from operation 1, the output of the preceding operation would be used as input to the following one.

Refer to this topic to review the schema and input/output examples for command-chaining-cwm-sol.

Workflow details
Table 59: command-chaining-cwm-sol details

Field	Value
Sr. No.	3
MOP action name	command-chaining-cwm-sol
Description	The workflow executes the an operation and applies filter to the command results to build inputs for the following operation. There are two ways in which this can be executed. If loop-enabled in operation 2 is provided as true, then the output of operation 1 give an array of length n. This array would be used as inputs, for n operations, with all using the operation defined in op2ration 2. If loop-enabled in operation 2 is false or is not provided, then starting from operation 1, the output of the preceding operation would be used as input to the following one.

Schema

```
{
"operations": [
{
"id": "string (required) - unique operation identifier",
"loop-enabled": "boolean (optional) - enable looping over filtered results",
"method": "string (required) - execCli|get|post|delete",
"command": "string (required for execCli) - CLI command",
"path": "string (required for get/post/delete) - RESTCONF path",
"payload": "string (required for post) - request body",
"template": "string (required for execCli) - TextFSM template",
"filter": "string (optional) - jq filter on parsed output",
"expressions": [
{
"condition": "string (required) - jq condition",
"message": "string (required) - result message",
"status": "string (required) - success|warning|failed"
}
]
}
],
```

```
"deviceSelection": {
"jsonPath": "string (optional) - JSON path for device filtering"
}
}
```

Input and Output Example

```
1. loop_enabled is true and child method is "get".
Input
{
"app-data": {
"data": {
"imageSize": 14412823,
"imageVersion": "",
"softwareImages": [
"cisco8k.tar"
]
},
"device": {
"host": "spitfire250.cisco.com",
"ip": "172.22.141.250",
"label": "spitfire250.cisco.com",
"name": "spitfire250",
"productSeries": "Cisco 8000 Series Routers",
"productType": "Cisco 8201 Router",
"resource": "nso-173",
"softwareType": "IOS XR",
"softwareVersion": "24.4.2",
"uuid": "9c10abb5-1396-40ff-ba5c-9f7250f791d1",
"vendor": "Cisco Systems"
},
"executionMode": 1,
"phase": "",
"resource": "nso-173",
"stage": "pre"
},
"operations": [
{
"expressions": [
{
"condition": ".\"tailf-ned-cisco-ios-xr:interface\" | length > 0",
"message": "The interfaces are provided in the output",
"status": "success"
}
],
"id": 1,
"method": "get",
"path":
"restconf/data/tailf-ncs:devices/device=${.\"app-data\".device.name}/config/tailf-ned-cisco-ios-xr:interface"
},
{
"expressions": [
{
"condition": ". != null",
"message": "The interfaces details have been captured properly",
"status": "success"
}
],
"id": 2,
"loop-enabled": true,
"method": "get",
```

```
"path": "${. as $data | (\"restconf/data/tailf-ncs:devices/device=\" +
.\"app-data\".device.name + \"/config/tailf-ned-cisco-ios-xr:interface/\") as
 $path | $path | $data.stash[].stash[].\"tailf-ned-cisco-ios-xr:interface\" |
 keys | map($path + .)}"
}
]
}
Output
{
"message": "All commands executed and evaluated successfully",
"processedResource": [
"9c10abb5-1396-40ff-ba5c-9f7250f791d1"
],
"stash": [
{
"id": 1,
"stash": [
{
"tailf-ned-cisco-ios-xr:interface": {
"FourHundredGigE": [
{
"id": "0/0/0/0",
"shutdown": [
null
]
},
{
"id": "0/0/0/1",
"shutdown": [
null
]
},
{
"id": "0/0/0/10",
"shutdown": [
null
]
},
{
"id": "0/0/0/11",
"shutdown": [
null
]
},
{
"id": "0/0/0/12",
"shutdown": [
null
]
},
{
"id": "0/0/0/13",
"shutdown": [
null
]
},
{
"id": "0/0/0/14",
"shutdown": [
null
]
},
```

```
{
"id": "0/0/0/15",
"shutdown": [
null
]
},
{
"id": "0/0/0/16",
"shutdown": [
null
]
},
{
"id": "0/0/0/17",
"shutdown": [
null
]
},
{
"id": "0/0/0/18",
"shutdown": [
null
]
},
{
"id": "0/0/0/19",
"shutdown": [
null
]
},
{
"id": "0/0/0/2",
"shutdown": [
null
]
},
{
"id": "0/0/0/20",
"shutdown": [
null
]
},
{
"id": "0/0/0/21",
"shutdown": [
null
]
},
{
"id": "0/0/0/22",
"shutdown": [
null
]
},
{
"id": "0/0/0/23",
"shutdown": [
null
]
},
{
"id": "0/0/0/3",
```

```
"shutdown": [
null
]
},
{
"id": "0/0/0/4",
"shutdown": [
null
]
},
{
"id": "0/0/0/5",
"shutdown": [
null
]
},
{
"id": "0/0/0/6",
"shutdown": [
null
]
},
{
"id": "0/0/0/7",
"shutdown": [
null
]
},
{
"id": "0/0/0/8",
"shutdown": [
null
]
},
{
"id": "0/0/0/9",
"shutdown": [
null
]
}
],
"HundredGigE": [
{
"id": "0/0/0/24",
"shutdown": [
null
]
},
{
"id": "0/0/0/25",
"shutdown": [
null
]
},
{
"id": "0/0/0/26",
"shutdown": [
null
]
},
{
"id": "0/0/0/27",
```

```
"shutdown": [
null
]
},
{
"id": "0/0/0/28",
"shutdown": [
null
]
},
{
"id": "0/0/0/29",
"shutdown": [
null
]
},
{
"id": "0/0/0/30",
"shutdown": [
null
]
},
{
"id": "0/0/0/31",
"shutdown": [
null
]
},
{
"id": "0/0/0/32",
"shutdown": [
null
]
},
{
"id": "0/0/0/33",
"shutdown": [
null
]
},
{
"id": "0/0/0/34",
"shutdown": [
null
]
},
{
"id": "0/0/0/35",
"shutdown": [
null
]
}
],
"MgmtEth": [
{
"id": "0/RP0/CPU0/0",
"ipv4": {
"address": {
"ip": "172.22.141.250",
"mask": "255.255.255.0"
}
},
```

```
"vrf": "MGMT"
}
],
"preconfigure": {
"HundredGigE": [
{
"id": "0/0/0/3/0"
}
]
}
}
}
]
},
{
"id": "2-1",
"stash": {
"tailf-ned-cisco-ios-xr:FourHundredGigE": [
{
"id": "0/0/0/0",
"shutdown": [
null
]
},
{
"id": "0/0/0/1",
"shutdown": [
null
]
},
{
"id": "0/0/0/10",
"shutdown": [
null
]
},
{
"id": "0/0/0/11",
"shutdown": [
null
]
},
{
"id": "0/0/0/12",
"shutdown": [
null
]
},
{
"id": "0/0/0/13",
"shutdown": [
null
]
},
{
"id": "0/0/0/14",
"shutdown": [
null
]
},
{
"id": "0/0/0/15",
```

```
"shutdown": [
null
]
},
{
"id": "0/0/0/16",
"shutdown": [
null
]
},
{
"id": "0/0/0/17",
"shutdown": [
null
]
},
{
"id": "0/0/0/18",
"shutdown": [
null
]
},
{
"id": "0/0/0/19",
"shutdown": [
null
]
},
{
"id": "0/0/0/2",
"shutdown": [
null
]
},
{
"id": "0/0/0/20",
"shutdown": [
null
]
},
{
"id": "0/0/0/21",
"shutdown": [
null
]
},
{
"id": "0/0/0/22",
"shutdown": [
null
]
},
{
"id": "0/0/0/23",
"shutdown": [
null
]
},
{
"id": "0/0/0/3",
"shutdown": [
null
```

```
]
},
{
"id": "0/0/0/4",
"shutdown": [
null
]
},
{
"id": "0/0/0/5",
"shutdown": [
null
]
},
{
"id": "0/0/0/6",
"shutdown": [
null
]
},
{
"id": "0/0/0/7",
"shutdown": [
null
]
},
{
"id": "0/0/0/8",
"shutdown": [
null
]
},
{
"id": "0/0/0/9",
"shutdown": [
null
]
}
]
}
},
{
"id": "2-2",
"stash": {
"tailf-ned-cisco-ios-xr:HundredGigE": [
{
"id": "0/0/0/24",
"shutdown": [
null
]
},
{
"id": "0/0/0/25",
"shutdown": [
null
]
},
{
"id": "0/0/0/26",
"shutdown": [
null
]
```

```
},
{
"id": "0/0/0/27",
"shutdown": [
null
]
},
{
"id": "0/0/0/28",
"shutdown": [
null
]
},
{
"id": "0/0/0/29",
"shutdown": [
null
]
},
{
"id": "0/0/0/30",
"shutdown": [
null
]
},
{
"id": "0/0/0/31",
"shutdown": [
null
]
},
{
"id": "0/0/0/32",
"shutdown": [
null
]
},
{
"id": "0/0/0/33",
"shutdown": [
null
]
},
{
"id": "0/0/0/34",
"shutdown": [
null
]
},
{
"id": "0/0/0/35",
"shutdown": [
null
]
}
]
}
},
{
"id": "2-3",
"stash": {
"tailf-ned-cisco-ios-xr:MgmtEth": [
```

```
{
"id": "0/RP0/CPU0/0",
"ipv4": {
"address": {
"ip": "172.22.141.250",
"mask": "255.255.255.0"
}
},
"vrf": "MGMT"
}
]
}
},
{
"id": "2-4",
"stash": {
"tailf-ned-cisco-ios-xr:preconfigure": {
"HundredGigE": [
{
"id": "0/0/0/3/0"
}
]
}
}
}
],
"status": "success"
}
```


2. loop_enabled is true and child method is "execCli".

Input


```
{
"app-data": {
"data": {
"imageSize": 14412823,
"imageVersion": "",
"softwareImages": [
"cisco8k.tar"
]
},
"device": {
"host": "spitfire250.cisco.com",
"ip": "172.22.141.250",
"label": "spitfire250.cisco.com",
"name": "spitfire250",
"productSeries": "Cisco 8000 Series Routers",
"productType": "Cisco 8201 Router",
"resource": "nso-173",
"softwareType": "IOS XR",
"softwareVersion": "24.4.2",
"uuid": "9c10abb5-1396-40ff-ba5c-9f7250f791d1",
"vendor": "Cisco Systems"
},
"executionMode": 1,
"phase": "",
"resource": "nso-173",
"stage": "pre"
},
"operations": [
{
"expressions": [
{
"condition": ".\"tailf-ned-cisco-ios-xr:HundredGigE\" | length > 0",
```

```
"message": "The interfaces are provided in the output",
"status": "success"
}
],
"id": 1,
"method": "get",
"path":
"restconf/data/tailf-ncs:devices/device=${.\"app-data\".device.name}/config/tailf-ned-cisco-ios-xr:interface/HundredGigE"
},
{
"command": "${[.stash[].stash[] | .\"tailf-ned-cisco-ios-xr:HundredGigE\"[] |
 select(.shutdown[] == null) | \"show interface HundredGigE\" + .id]}",
"expressions": [
{
"condition": ".[] | select(. | keys != [\"app-data\"]) as $data |
$data.INTERFACE != null and $data.ADMIN_STATUS != null and $data.OPER_STATUS
!= null",
"message": "The interfaces details have been captured properly",
"status": "success"
}
],
"id": 2,
"loop-enabled": true,
"method": "execCli",
"template": "Value Required INTERFACE (\\S+)\nValue ADMIN_STATUS
(up|down|administratively down)\nValue OPER_STATUS (up|down|administratively
down|line protocol is up|line protocol is down)\n\nStart\n ^${INTERFACE} is
${ADMIN_STATUS}, line protocol is ${OPER_STATUS}\n"
}
]
}
Output
{
"message": "All commands executed and evaluated successfully",
"processedResource": [
"9c10abb5-1396-40ff-ba5c-9f7250f791d1"
],
"stash": [
{
"id": 1,
"stash": [
{
"tailf-ned-cisco-ios-xr:HundredGigE": [
{
"id": "0/0/0/24",
"shutdown": [
null
]
},
{
"id": "0/0/0/25",
"shutdown": [
null
]
},
{
"id": "0/0/0/26",
"shutdown": [
null
]
},
{
```

```
"id": "0/0/0/27",
"shutdown": [
null
]
},
{
"id": "0/0/0/28",
"shutdown": [
null
]
},
{
"id": "0/0/0/29",
"shutdown": [
null
]
},
{
"id": "0/0/0/30",
"shutdown": [
null
]
},
{
"id": "0/0/0/31",
"shutdown": [
null
]
},
{
"id": "0/0/0/32",
"shutdown": [
null
]
},
{
"id": "0/0/0/33",
"shutdown": [
null
]
},
{
"id": "0/0/0/34",
"shutdown": [
null
]
},
{
"id": "0/0/0/35",
"shutdown": [
null
]
}
]
}
]
},
{
"id": "2-1",
"stash": [
{
"ADMIN_STATUS": "administratively down",
```

```
"INTERFACE": "HundredGigE0/0/0/24",
"OPER_STATUS": "administratively down"
}
]
},
{
"id": "2-2",
"stash": [
{
"ADMIN_STATUS": "administratively down",
"INTERFACE": "HundredGigE0/0/0/25",
"OPER_STATUS": "administratively down"
}
]
},
{
"id": "2-3",
"stash": [
{
"ADMIN_STATUS": "administratively down",
"INTERFACE": "HundredGigE0/0/0/26",
"OPER_STATUS": "administratively down"
}
]
},
{
"id": "2-4",
"stash": [
{
"ADMIN_STATUS": "administratively down",
"INTERFACE": "HundredGigE0/0/0/27",
"OPER_STATUS": "administratively down"
}
]
},
{
"id": "2-5",
"stash": [
{
"ADMIN_STATUS": "administratively down",
"INTERFACE": "HundredGigE0/0/0/28",
"OPER_STATUS": "administratively down"
}
]
},
{
"id": "2-6",
"stash": [
{
"ADMIN_STATUS": "administratively down",
"INTERFACE": "HundredGigE0/0/0/29",
"OPER_STATUS": "administratively down"
}
]
},
{
"id": "2-7",
"stash": [
{
"ADMIN_STATUS": "administratively down",
"INTERFACE": "HundredGigE0/0/0/30",
"OPER_STATUS": "administratively down"
```

```
}
]
},
{
"id": "2-8",
"stash": [
{
"ADMIN_STATUS": "administratively down",
"INTERFACE": "HundredGigE0/0/0/31",
"OPER_STATUS": "administratively down"
}
]
},
{
"id": "2-9",
"stash": [
{
"ADMIN_STATUS": "administratively down",
"INTERFACE": "HundredGigE0/0/0/32",
"OPER_STATUS": "administratively down"
}
]
},
{
"id": "2-10",
"stash": [
{
"ADMIN_STATUS": "administratively down",
"INTERFACE": "HundredGigE0/0/0/33",
"OPER_STATUS": "administratively down"
}
]
},
{
"id": "2-11",
"stash": [
{
"ADMIN_STATUS": "administratively down",
"INTERFACE": "HundredGigE0/0/0/34",
"OPER_STATUS": "administratively down"
}
]
},
{
"id": "2-12",
"stash": [
{
"ADMIN_STATUS": "administratively down",
"INTERFACE": "HundredGigE0/0/0/35",
"OPER_STATUS": "administratively down"
}
]
}
],
"status": "success"
}
3. loop-enabled is false and method is post.
Input
{
"app-data": {
"data": {
"imageSize": 14412823,
```

```
"imageVersion": "",
"softwareImages": [
"cisco8k.tar"
]
},
"device": {
"host": "spitfire250.cisco.com",
"ip": "172.22.141.250",
"label": "spitfire250.cisco.com",
"name": "spitfire250",
"productSeries": "Cisco 8000 Series Routers",
"productType": "Cisco 8201 Router",
"resource": "nso-173",
"softwareType": "IOS XR",
"softwareVersion": "24.4.2",
"uuid": "9c10abb5-1396-40ff-ba5c-9f7250f791d1",
"vendor": "Cisco Systems"
},
"executionMode": 1,
"phase": "",
"resource": "nso-173",
"stage": "activate"
},
"operations": [
{
"expressions": [
{
"condition": ".\"tailf-ned-cisco-ios-xr:HundredGigE\" | length > 0",
"message": "The interfaces are provided in the output",
"status": "success"
}
],
"id": 1,
"method": "get",
"path":
"restconf/data/tailf-ncs:devices/device=${.\"app-data\".device.name}/config/tailf-ned-cisco-ios-xr:interface/HundredGigE"
},
{
"expressions": [
{
"condition": ". != null and . != \"\"",
"message": "The interfaces details have been captured properly",
"status": "success"
}
],
"id": 3,
"method": "post",
"path":
"restconf/operations/devices/device=${.\"app-data\".device.name}/live-status/exec/any",
"payload": "{\"args\":
\"${.stash.\\\"tailf-ned-cisco-ios-xr-stats:output\\\".result |
split(\\\"HundredGigE0/0/0/\\\")[1] | split(\\\" \\\")[0] | tonumber + 1 |
tostring as $id | \\\"show interface HundredGigE0/0/0/\\\" + $id}\"}"
},
{
"expressions": [
{
"condition": ". != null and . != \"\"",
"message": "The interfaces details have been captured properly",
"status": "success"
}
],
```

```
"id": 2,
"loop-enabled": false,
"method": "post",
"path":
"restconf/operations/devices/device==${.\"app-data\".device.name}/live-status/exec/any",
"payload": "{\"args\":
\"${[.stash.\\\"tailf-ned-cisco-ios-xr:HundredGigE\\\"[].id as $id | \\\"show
 interface HundredGigE\\\" + $id][0]}\"}"
}
]
}
Output
{
"message": "All commands executed and evaluated successfully",
"processedResource": [
"9c10abb5-1396-40ff-ba5c-9f7250f791d1"
],
"stash": [
{
"id": 1,
"stash": [
{
"tailf-ned-cisco-ios-xr:HundredGigE": [
{
"id": "0/0/0/24",
"shutdown": [
null
]
},
{
"id": "0/0/0/25",
"shutdown": [
null
]
},
{
"id": "0/0/0/26",
"shutdown": [
null
]
},
{
"id": "0/0/0/27",
"shutdown": [
null
]
},
{
"id": "0/0/0/28",
"shutdown": [
null
]
},
{
"id": "0/0/0/29",
"shutdown": [
null
]
},
{
"id": "0/0/0/30",
"shutdown": [
```

```
null
]
},
{
"id": "0/0/0/31",
"shutdown": [
null
]
},
{
"id": "0/0/0/32",
"shutdown": [
null
]
},
{
"id": "0/0/0/33",
"shutdown": [
null
]
},
{
"id": "0/0/0/34",
"shutdown": [
null
]
},
{
"id": "0/0/0/35",
"shutdown": [
null
]
}
]
}
]
},
{
"id": 2,
"stash": {
"tailf-ned-cisco-ios-xr-stats:output": {
"result": "\r\n\rFri Oct 17 05:29:40.805 PDT\r\nHundredGigE0/0/0/24 is
administratively down, line protocol is administratively down \r\n  Interface
 state transitions: 0\r\n  Hardware is HundredGigE, address is 6c13.d5d2.10c0
 (bia 6c13.d5d2.10c0)\r\n  Internet address is Unknown\r\n  MTU 1514 bytes,
BW 100000000 Kbit (Max: 100000000 Kbit)\r\n     reliability 255/255, txload
0/255, rxload 0/255\r\n  Encapsulation ARPA,\r\n  Full-duplex, 100000Mb/s,
link type is force-up\r\n  output flow control is off, input flow control is
off\r\n  Carrier delay (up) is 200 msec\r\n  loopback not set,\r\n  Last input
 never, output never\r\n  Last clearing of \"show interface\" counters never\r\n
  5 minute input rate 0 bits/sec, 0 packets/sec\r\n  5 minute output rate 0
bits/sec, 0 packets/sec\r\n     0 packets input, 0 bytes, 0 total input
drops\r\n     0 drops for unrecognized upper-level protocol\r\n     Received
0 broadcast packets, 0 multicast packets\r\n              0 runts, 0 giants,
0 throttles, 0 parity\r\n     0 input errors, 0 CRC, 0 frame, 0 overrun, 0
ignored, 0 abort\r\n     0 packets output, 0 bytes, 0 total output drops\r\n
    Output 0 broadcast packets, 0 multicast packets\r\n     0 output errors,
0 underruns, 0 applique, 0 resets\r\n     0 output buffer failures, 0 output
buffers swapped out\r\n     0 carrier
transitions\r\n\r\n\r\nRP/0/RP0/CPU0:spitfire250#"
}
}
```

```
},
{
"id": 3,
"stash": {
"tailf-ned-cisco-ios-xr-stats:output": {
"result": "\r\n\rFri Oct 17 05:29:41.345 PDT\r\nHundredGigE0/0/0/25 is
administratively down, line protocol is administratively down \r\n  Interface
 state transitions: 0\r\n  Hardware is HundredGigE, address is 6c13.d5d2.10c4
 (bia 6c13.d5d2.10c4)\r\n  Internet address is Unknown\r\n  MTU 1514 bytes,
BW 100000000 Kbit (Max: 100000000 Kbit)\r\n     reliability 255/255, txload
0/255, rxload 0/255\r\n  Encapsulation ARPA,\r\n  Full-duplex, 100000Mb/s,
link type is force-up\r\n  output flow control is off, input flow control is
off\r\n  Carrier delay (up) is 200 msec\r\n  loopback not set,\r\n  Last input
 never, output never\r\n  Last clearing of \"show interface\" counters never\r\n
  5 minute input rate 0 bits/sec, 0 packets/sec\r\n  5 minute output rate 0
bits/sec, 0 packets/sec\r\n     0 packets input, 0 bytes, 0 total input
drops\r\n     0 drops for unrecognized upper-level protocol\r\n     Received
0 broadcast packets, 0 multicast packets\r\n              0 runts, 0 giants,
0 throttles, 0 parity\r\n     0 input errors, 0 CRC, 0 frame, 0 overrun, 0
ignored, 0 abort\r\n     0 packets output, 0 bytes, 0 total output drops\r\n
    Output 0 broadcast packets, 0 multicast packets\r\n     0 output errors,
0 underruns, 0 applique, 0 resets\r\n     0 output buffer failures, 0 output
buffers swapped out\r\n     0 carrier
transitions\r\n\r\n\r\nRP/0/RP0/CPU0:spitfire250#"
}
}
}
],
"status": "success"
}
4. loop-enabled is false and method is get.
Input
{
"app-data": {
"data": {
"imageSize": 14412823,
"imageVersion": "",
"softwareImages": [
"cisco8k.tar"
]
},
"device": {
"host": "spitfire250.cisco.com",
"ip": "172.22.141.250",
"label": "spitfire250.cisco.com",
"name": "spitfire250",
"productSeries": "Cisco 8000 Series Routers",
"productType": "Cisco 8201 Router",
"resource": "nso-173",
"softwareType": "IOS XR",
"softwareVersion": "24.4.2",
"uuid": "9c10abb5-1396-40ff-ba5c-9f7250f791d1",
"vendor": "Cisco Systems"
},
"executionMode": 1,
"phase": "",
"resource": "nso-173",
"stage": "commit"
},
"operations": [
{
"expressions": [
```

```
{
"condition": ".\"tailf-ned-cisco-ios-xr:interface\" | length > 0",
"message": "The interfaces are provided in the output",
"status": "success"
}
],
"id": 1,
"method": "get",
"path":
"restconf/data/tailf-ncs:devices/device=${.\"app-data\".device.name}/config/tailf-ned-cisco-ios-xr:interface"
},
{
"expressions": [
{
"condition": ". != null and . != \"\"",
"message": "The interfaces details have been captured properly",
"status": "success"
}
],
"id": 3,
"loop-enabled": false,
"method": "get",
"path": "${(. as $data | $data.stash | keys[0] | split(\":\")[0] as $intName
| (\"restconf/data/tailf-ncs:devices/device=\" + $data.\"app-data\".device.name
 + \"/config/\" + $intName + \":interface/MgmtEth\"))}"
},
{
"expressions": [
{
"condition": ". != null and . != \"\"",
"message": "The interfaces details have been captured properly",
"status": "success"
}
],
"id": 2,
"loop-enabled": false,
"method": "get",
"path": "${. as $data | .stash.\"tailf-ned-cisco-ios-xr:interface\" | keys[0]
 as $intName | \"restconf/data/tailf-ncs:devices/device=\" +
$data.\"app-data\".device.name + \"/config/tailf-ned-cisco-ios-xr:interface/\"
 + $intName}"
}
]
}
Output
{
"message": "All commands executed and evaluated successfully",
"processedResource": [
"9c10abb5-1396-40ff-ba5c-9f7250f791d1"
],
"stash": [
{
"id": 1,
"stash": [
{
"tailf-ned-cisco-ios-xr:interface": {
"FourHundredGigE": [
{
"id": "0/0/0/0",
"shutdown": [
null
]
```

```
},
{
"id": "0/0/0/1",
"shutdown": [
null
]
},
{
"id": "0/0/0/10",
"shutdown": [
null
]
},
{
"id": "0/0/0/11",
"shutdown": [
null
]
},
{
"id": "0/0/0/12",
"shutdown": [
null
]
},
{
"id": "0/0/0/13",
"shutdown": [
null
]
},
{
"id": "0/0/0/14",
"shutdown": [
null
]
},
{
"id": "0/0/0/15",
"shutdown": [
null
]
},
{
"id": "0/0/0/16",
"shutdown": [
null
]
},
{
"id": "0/0/0/17",
"shutdown": [
null
]
},
{
"id": "0/0/0/18",
"shutdown": [
null
]
},
{
```

```
"id": "0/0/0/19",
"shutdown": [
null
]
},
{
"id": "0/0/0/2",
"shutdown": [
null
]
},
{
"id": "0/0/0/20",
"shutdown": [
null
]
},
{
"id": "0/0/0/21",
"shutdown": [
null
]
},
{
"id": "0/0/0/22",
"shutdown": [
null
]
},
{
"id": "0/0/0/23",
"shutdown": [
null
]
},
{
"id": "0/0/0/3",
"shutdown": [
null
]
},
{
"id": "0/0/0/4",
"shutdown": [
null
]
},
{
"id": "0/0/0/5",
"shutdown": [
null
]
},
{
"id": "0/0/0/6",
"shutdown": [
null
]
},
{
"id": "0/0/0/7",
"shutdown": [
```

```
null
]
},
{
"id": "0/0/0/8",
"shutdown": [
null
]
},
{
"id": "0/0/0/9",
"shutdown": [
null
]
}
],
"HundredGigE": [
{
"id": "0/0/0/24",
"shutdown": [
null
]
},
{
"id": "0/0/0/25",
"shutdown": [
null
]
},
{
"id": "0/0/0/26",
"shutdown": [
null
]
},
{
"id": "0/0/0/27",
"shutdown": [
null
]
},
{
"id": "0/0/0/28",
"shutdown": [
null
]
},
{
"id": "0/0/0/29",
"shutdown": [
null
]
},
{
"id": "0/0/0/30",
"shutdown": [
null
]
},
{
"id": "0/0/0/31",
"shutdown": [
```

```
null
]
},
{
"id": "0/0/0/32",
"shutdown": [
null
]
},
{
"id": "0/0/0/33",
"shutdown": [
null
]
},
{
"id": "0/0/0/34",
"shutdown": [
null
]
},
{
"id": "0/0/0/35",
"shutdown": [
null
]
}
],
"MgmtEth": [
{
"id": "0/RP0/CPU0/0",
"ipv4": {
"address": {
"ip": "172.22.141.250",
"mask": "255.255.255.0"
}
},
"vrf": "MGMT"
}
],
"preconfigure": {
"HundredGigE": [
{
"id": "0/0/0/3/0"
}
]
}
}
}
]
},
{
"id": 2,
"stash": {
"tailf-ned-cisco-ios-xr:FourHundredGigE": [
{
"id": "0/0/0/0",
"shutdown": [
null
]
},
{
```

```
"id": "0/0/0/1",
"shutdown": [
null
]
},
{
"id": "0/0/0/10",
"shutdown": [
null
]
},
{
"id": "0/0/0/11",
"shutdown": [
null
]
},
{
"id": "0/0/0/12",
"shutdown": [
null
]
},
{
"id": "0/0/0/13",
"shutdown": [
null
]
},
{
"id": "0/0/0/14",
"shutdown": [
null
]
},
{
"id": "0/0/0/15",
"shutdown": [
null
]
},
{
"id": "0/0/0/16",
"shutdown": [
null
]
},
{
"id": "0/0/0/17",
"shutdown": [
null
]
},
{
"id": "0/0/0/18",
"shutdown": [
null
]
},
{
"id": "0/0/0/19",
"shutdown": [
```

```
null
]
},
{
"id": "0/0/0/2",
"shutdown": [
null
]
},
{
"id": "0/0/0/20",
"shutdown": [
null
]
},
{
"id": "0/0/0/21",
"shutdown": [
null
]
},
{
"id": "0/0/0/22",
"shutdown": [
null
]
},
{
"id": "0/0/0/23",
"shutdown": [
null
]
},
{
"id": "0/0/0/3",
"shutdown": [
null
]
},
{
"id": "0/0/0/4",
"shutdown": [
null
]
},
{
"id": "0/0/0/5",
"shutdown": [
null
]
},
{
"id": "0/0/0/6",
"shutdown": [
null
]
},
{
"id": "0/0/0/7",
"shutdown": [
null
]
```

```
},
{
"id": "0/0/0/8",
"shutdown": [
null
]
},
{
"id": "0/0/0/9",
"shutdown": [
null
]
}
]
}
},
{
"id": 3,
"stash": {
"tailf-ned-cisco-ios-xr:MgmtEth": [
{
"id": "0/RP0/CPU0/0",
"ipv4": {
"address": {
"ip": "172.22.141.250",
"mask": "255.255.255.0"
}
},
"vrf": "MGMT"
}
]
}
}
],
"status": "success"
}
```


5. deviceSelection is not null, loop-enabled is true, method is get.

Input


```
{
"app-data": {
"data": {},
"devices": {
"source": [
{
"host": "spitfire250.cisco.com",
"ip": "172.22.141.250",
"label": "source",
"name": "spitfire250",
"productSeries": "Cisco 8000 Series Routers",
"productType": "Cisco 8201 Router",
"resource": "nso-173",
"softwareType": "IOS XR",
"softwareVersion": "24.4.2",
"uuid": "2c611a0a-31fc-4a11-80bf-170c9417014c",
"vendor": "Cisco Systems"
}
]
},
"executionMode": 2,
"phase": "",
"resource": "nso-173",
"stage": "pre-provision"
```

```
},
"deviceSelection": {
"jsonPath": ".app-data.devices.source[0]"
},
"operations": [
{
"Expression": [
{
"Condition": "",
"Message": "",
"Status": ""
}
],
"command": "",
"expressions": [
{
"condition": ".\"tailf-ned-cisco-ios-xr:HundredGigE\" | length > 0",
"message": "The interfaces are provided in the output",
"status": "success"
}
],
"filter": "",
"id": 1,
"loop-enabled": false,
"method": "get",
"path":
"restconf/data/tailf-ncs:devices/device=${.\"app-data\".device.name}/config/tailf-ned-cisco-ios-xr:interface/HundredGigE",
"payload": "\"\"",
"template": ""
},
{
"Expression": [
{
"Condition": "",
"Message": "",
"Status": ""
}
],
"command": "${[.stash[].stash[] | .\"tailf-ned-cisco-ios-xr:HundredGigE\"[] |
 select(.shutdown[] == null) | \"show interface HundredGigE\" + .id]}",
"expressions": [
{
"condition": ".[] | select(. | keys != [\"app-data\"]) as $data |
$data.INTERFACE != null and $data.ADMIN_STATUS != null and $data.OPER_STATUS
!= null",
"message": "The interfaces details have been captured properly",
"status": "success"
}
],
"filter": "",
"id": 2,
"loop-enabled": true,
"method": "execCli",
"path": "",
"payload": "\"\"",
"template": "Value Required INTERFACE (\\S+)\nValue ADMIN_STATUS
(up|down|administratively down)\nValue OPER_STATUS (up|down|administratively
down|line protocol is up|line protocol is down)\n\nStart\n ^${INTERFACE} is
${ADMIN_STATUS}, line protocol is ${OPER_STATUS}\n"
}
]
}
```

```
Output
{
"message": "All commands executed and evaluated successfully",
"processedResource": [
"2c611a0a-31fc-4a11-80bf-170c9417014c"
],
"stash": [
{
"id": 1,
"stash": [
{
"tailf-ned-cisco-ios-xr:HundredGigE": [
{
"id": "0/0/0/24",
"shutdown": [
null
]
},
{
"id": "0/0/0/25",
"shutdown": [
null
]
},
{
"id": "0/0/0/26",
"shutdown": [
null
]
},
{
"id": "0/0/0/27",
"shutdown": [
null
]
},
{
"id": "0/0/0/28",
"shutdown": [
null
]
},
{
"id": "0/0/0/29",
"shutdown": [
null
]
},
{
"id": "0/0/0/30",
"shutdown": [
null
]
},
{
"id": "0/0/0/31",
"shutdown": [
null
]
},
{
"id": "0/0/0/32",
```

```
"shutdown": [
null
]
},
{
"id": "0/0/0/33",
"shutdown": [
null
]
},
{
"id": "0/0/0/34",
"shutdown": [
null
]
},
{
"id": "0/0/0/35",
"shutdown": [
null
]
}
]
}
]
},
{
"id": "2-1",
"stash": [
{
"ADMIN_STATUS": "administratively down",
"INTERFACE": "HundredGigE0/0/0/24",
"OPER_STATUS": "administratively down"
}
]
},
{
"id": "2-2",
"stash": [
{
"ADMIN_STATUS": "administratively down",
"INTERFACE": "HundredGigE0/0/0/25",
"OPER_STATUS": "administratively down"
}
]
},
{
"id": "2-3",
"stash": [
{
"ADMIN_STATUS": "administratively down",
"INTERFACE": "HundredGigE0/0/0/26",
"OPER_STATUS": "administratively down"
}
]
},
{
"id": "2-4",
"stash": [
{
"ADMIN_STATUS": "administratively down",
"INTERFACE": "HundredGigE0/0/0/27",
```

```
"OPER_STATUS": "administratively down"
}
]
},
{
"id": "2-5",
"stash": [
{
"ADMIN_STATUS": "administratively down",
"INTERFACE": "HundredGigE0/0/0/28",
"OPER_STATUS": "administratively down"
}
]
},
{
"id": "2-6",
"stash": [
{
"ADMIN_STATUS": "administratively down",
"INTERFACE": "HundredGigE0/0/0/29",
"OPER_STATUS": "administratively down"
}
]
},
{
"id": "2-7",
"stash": [
{
"ADMIN_STATUS": "administratively down",
"INTERFACE": "HundredGigE0/0/0/30",
"OPER_STATUS": "administratively down"
}
]
},
{
"id": "2-8",
"stash": [
{
"ADMIN_STATUS": "administratively down",
"INTERFACE": "HundredGigE0/0/0/31",
"OPER_STATUS": "administratively down"
}
]
},
{
"id": "2-9",
"stash": [
{
"ADMIN_STATUS": "administratively down",
"INTERFACE": "HundredGigE0/0/0/32",
"OPER_STATUS": "administratively down"
}
]
},
{
"id": "2-10",
"stash": [
{
"ADMIN_STATUS": "administratively down",
"INTERFACE": "HundredGigE0/0/0/33",
"OPER_STATUS": "administratively down"
}
```

```
]
},
{
"id": "2-11",
"stash": [
{
"ADMIN_STATUS": "administratively down",
"INTERFACE": "HundredGigE0/0/0/34",
"OPER_STATUS": "administratively down"
}
]
},
{
"id": "2-12",
"stash": [
{
"ADMIN_STATUS": "administratively down",
"INTERFACE": "HundredGigE0/0/0/35",
"OPER_STATUS": "administratively down"
}
]
}
],
"status": "success"
}
```


6. deviceSelection is not null, loop-enabled is false, method is get.

Input


```
{
"app-data": {
"data": {},
"devices": {
"source": [
{
"host": "spitfire250.cisco.com",
"ip": "172.22.141.250",
"label": "source",
"name": "spitfire250",
"productSeries": "Cisco 8000 Series Routers",
"productType": "Cisco 8201 Router",
"resource": "nso-173",
"softwareType": "IOS XR",
"softwareVersion": "24.4.2",
"uuid": "2c611a0a-31fc-4a11-80bf-170c9417014c",
"vendor": "Cisco Systems"
}
]
},
"executionMode": 2,
"phase": "",
"resource": "nso-173",
"stage": "pre-migration"
},
"deviceSelection": {
"jsonPath": ".app-data.devices.source[0]"
},
"operations": [
{
"expressions": [
{
"condition": ".\"tailf-ned-cisco-ios-xr:interface\" | length > 0",
"message": "The interfaces are provided in the output",
"status": "success"
```

```
}
],
"id": 1,
"method": "get",
"path":
"restconf/data/tailf-ncs:devices/device=${.\"app-data\".device.name}/config/tailf-ned-cisco-ios-xr:interface"
},
{
"expressions": [
{
"condition": ". != null and . != \"\"",
"message": "The interfaces details have been captured properly",
"status": "success"
}
],
"id": 3,
"loop-enabled": false,
"method": "get",
"path": "${(. as $data | $data.stash | keys[0] | split(\":\")[0] as $intName
| (\"restconf/data/tailf-ncs:devices/device=\" + $data.\"app-data\".device.name
 + \"/config/\" + $intName + \":interface/MgmtEth\"))}"
},
{
"expressions": [
{
"condition": ". != null and . != \"\"",
"message": "The interfaces details have been captured properly",
"status": "success"
}
],
"id": 2,
"loop-enabled": false,
"method": "get",
"path": "${. as $data | .stash.\"tailf-ned-cisco-ios-xr:interface\" | keys[0]
 as $intName | \"restconf/data/tailf-ncs:devices/device=\" +
$data.\"app-data\".device.name + \"/config/tailf-ned-cisco-ios-xr:interface/\"
 + $intName}"
}
]
}
Output
{
"message": "All commands executed and evaluated successfully",
"processedResource": [
"2c611a0a-31fc-4a11-80bf-170c9417014c"
],
"stash": [
{
"id": 1,
"stash": [
{
"tailf-ned-cisco-ios-xr:interface": {
"FourHundredGigE": [
{
"id": "0/0/0/0",
"shutdown": [
null
]
},
{
"id": "0/0/0/1",
"shutdown": [
```

```
null
]
},
{
"id": "0/0/0/10",
"shutdown": [
null
]
},
{
"id": "0/0/0/11",
"shutdown": [
null
]
},
{
"id": "0/0/0/12",
"shutdown": [
null
]
},
{
"id": "0/0/0/13",
"shutdown": [
null
]
},
{
"id": "0/0/0/14",
"shutdown": [
null
]
},
{
"id": "0/0/0/15",
"shutdown": [
null
]
},
{
"id": "0/0/0/16",
"shutdown": [
null
]
},
{
"id": "0/0/0/17",
"shutdown": [
null
]
},
{
"id": "0/0/0/18",
"shutdown": [
null
]
},
{
"id": "0/0/0/19",
"shutdown": [
null
]
```

```
},
{
"id": "0/0/0/2",
"shutdown": [
null
]
},
{
"id": "0/0/0/20",
"shutdown": [
null
]
},
{
"id": "0/0/0/21",
"shutdown": [
null
]
},
{
"id": "0/0/0/22",
"shutdown": [
null
]
},
{
"id": "0/0/0/23",
"shutdown": [
null
]
},
{
"id": "0/0/0/3",
"shutdown": [
null
]
},
{
"id": "0/0/0/4",
"shutdown": [
null
]
},
{
"id": "0/0/0/5",
"shutdown": [
null
]
},
{
"id": "0/0/0/6",
"shutdown": [
null
]
},
{
"id": "0/0/0/7",
"shutdown": [
null
]
},
{
```

```
"id": "0/0/0/8",
"shutdown": [
null
]
},
{
"id": "0/0/0/9",
"shutdown": [
null
]
}
],
"HundredGigE": [
{
"id": "0/0/0/24",
"shutdown": [
null
]
},
{
"id": "0/0/0/25",
"shutdown": [
null
]
},
{
"id": "0/0/0/26",
"shutdown": [
null
]
},
{
"id": "0/0/0/27",
"shutdown": [
null
]
},
{
"id": "0/0/0/28",
"shutdown": [
null
]
},
{
"id": "0/0/0/29",
"shutdown": [
null
]
},
{
"id": "0/0/0/30",
"shutdown": [
null
]
},
{
"id": "0/0/0/31",
"shutdown": [
null
]
},
{
```

```
"id": "0/0/0/32",
"shutdown": [
null
]
},
{
"id": "0/0/0/33",
"shutdown": [
null
]
},
{
"id": "0/0/0/34",
"shutdown": [
null
]
},
{
"id": "0/0/0/35",
"shutdown": [
null
]
}
],
"MgmtEth": [
{
"id": "0/RP0/CPU0/0",
"ipv4": {
"address": {
"ip": "172.22.141.250",
"mask": "255.255.255.0"
}
},
"vrf": "MGMT"
}
],
"preconfigure": {
"HundredGigE": [
{
"id": "0/0/0/3/0"
}
]
}
}
}
]
},
{
"id": 2,
"stash": {
"tailf-ned-cisco-ios-xr:FourHundredGigE": [
{
"id": "0/0/0/0",
"shutdown": [
null
]
},
{
"id": "0/0/0/1",
"shutdown": [
null
]
```

```
},
{
"id": "0/0/0/10",
"shutdown": [
null
]
},
{
"id": "0/0/0/11",
"shutdown": [
null
]
},
{
"id": "0/0/0/12",
"shutdown": [
null
]
},
{
"id": "0/0/0/13",
"shutdown": [
null
]
},
{
"id": "0/0/0/14",
"shutdown": [
null
]
},
{
"id": "0/0/0/15",
"shutdown": [
null
]
},
{
"id": "0/0/0/16",
"shutdown": [
null
]
},
{
"id": "0/0/0/17",
"shutdown": [
null
]
},
{
"id": "0/0/0/18",
"shutdown": [
null
]
},
{
"id": "0/0/0/19",
"shutdown": [
null
]
},
{
```

```
"id": "0/0/0/2",
"shutdown": [
null
]
},
{
"id": "0/0/0/20",
"shutdown": [
null
]
},
{
"id": "0/0/0/21",
"shutdown": [
null
]
},
{
"id": "0/0/0/22",
"shutdown": [
null
]
},
{
"id": "0/0/0/23",
"shutdown": [
null
]
},
{
"id": "0/0/0/3",
"shutdown": [
null
]
},
{
"id": "0/0/0/4",
"shutdown": [
null
]
},
{
"id": "0/0/0/5",
"shutdown": [
null
]
},
{
"id": "0/0/0/6",
"shutdown": [
null
]
},
{
"id": "0/0/0/7",
"shutdown": [
null
]
},
{
"id": "0/0/0/8",
"shutdown": [
```

```
null
]
},
{
"id": "0/0/0/9",
"shutdown": [
null
]
}
]
}
},
{
"id": 3,
"stash": {
"tailf-ned-cisco-ios-xr:MgmtEth": [
{
"id": "0/RP0/CPU0/0",
"ipv4": {
"address": {
"ip": "172.22.141.250",
"mask": "255.255.255.0"
}
},
"vrf": "MGMT"
}
]
}
}
],
"status": "success"
}
```


7. deviceSelection is not null, loop-enabled is true, method is execCli.

Input


```
{
"app-data": {
"data": {},
"devices": {
"source": [
{
"host": "spitfire250.cisco.com",
"ip": "172.22.141.250",
"label": "source",
"name": "spitfire250",
"productSeries": "Cisco 8000 Series Routers",
"productType": "Cisco 8201 Router",
"resource": "nso-173",
"softwareType": "IOS XR",
"softwareVersion": "24.4.2",
"uuid": "2c611a0a-31fc-4a11-80bf-170c9417014c",
"vendor": "Cisco Systems"
}
]
},
"executionMode": 2,
"phase": "",
"resource": "nso-173",
"stage": "migration"
},
"deviceSelection": {
"jsonPath": ".app-data.devices.source[0]"
},
```

```
"operations": [
{
"expressions": [
{
"condition": ".\"tailf-ned-cisco-ios-xr:HundredGigE\" | length > 0",
"message": "The interfaces are provided in the output",
"status": "success"
}
],
"id": 1,
"method": "get",
"path":
"restconf/data/tailf-ncs:devices/device=${.\"app-data\".device.name}/config/tailf-ned-cisco-ios-xr:interface/HundredGigE"
},
{
"command": "${[.stash[].stash[] | .\"tailf-ned-cisco-ios-xr:HundredGigE\"[] |
 select(.shutdown[] == null) | \"show interface HundredGigE\" + .id]}",
"expressions": [
{
"condition": ".[] | select(. | keys != [\"app-data\"]) as $data |
$data.INTERFACE != null and $data.ADMIN_STATUS != null and $data.OPER_STATUS
!= null",
"message": "The interfaces details have been captured properly",
"status": "success"
}
],
"id": 2,
"loop-enabled": true,
"method": "execCli",
"template": "Value Required INTERFACE (\\S+)\nValue ADMIN_STATUS
(up|down|administratively down)\nValue OPER_STATUS (up|down|administratively
down|line protocol is up|line protocol is down)\n\nStart\n ^${INTERFACE} is
${ADMIN_STATUS}, line protocol is ${OPER_STATUS}\n"
}
]
}
Output
{
"message": "All commands executed and evaluated successfully",
"processedResource": [
"2c611a0a-31fc-4a11-80bf-170c9417014c"
],
"stash": [
{
"id": 1,
"stash": [
{
"tailf-ned-cisco-ios-xr:HundredGigE": [
{
"id": "0/0/0/24",
"shutdown": [
null
]
},
{
"id": "0/0/0/25",
"shutdown": [
null
]
},
{
"id": "0/0/0/26",
```

```
"shutdown": [
null
]
},
{
"id": "0/0/0/27",
"shutdown": [
null
]
},
{
"id": "0/0/0/28",
"shutdown": [
null
]
},
{
"id": "0/0/0/29",
"shutdown": [
null
]
},
{
"id": "0/0/0/30",
"shutdown": [
null
]
},
{
"id": "0/0/0/31",
"shutdown": [
null
]
},
{
"id": "0/0/0/32",
"shutdown": [
null
]
},
{
"id": "0/0/0/33",
"shutdown": [
null
]
},
{
"id": "0/0/0/34",
"shutdown": [
null
]
},
{
"id": "0/0/0/35",
"shutdown": [
null
]
}
]
}
]
},
```

```
{
"id": "2-1",
"stash": [
{
"ADMIN_STATUS": "administratively down",
"INTERFACE": "HundredGigE0/0/0/24",
"OPER_STATUS": "administratively down"
}
]
},
{
"id": "2-2",
"stash": [
{
"ADMIN_STATUS": "administratively down",
"INTERFACE": "HundredGigE0/0/0/25",
"OPER_STATUS": "administratively down"
}
]
},
{
"id": "2-3",
"stash": [
{
"ADMIN_STATUS": "administratively down",
"INTERFACE": "HundredGigE0/0/0/26",
"OPER_STATUS": "administratively down"
}
]
},
{
"id": "2-4",
"stash": [
{
"ADMIN_STATUS": "administratively down",
"INTERFACE": "HundredGigE0/0/0/27",
"OPER_STATUS": "administratively down"
}
]
},
{
"id": "2-5",
"stash": [
{
"ADMIN_STATUS": "administratively down",
"INTERFACE": "HundredGigE0/0/0/28",
"OPER_STATUS": "administratively down"
}
]
},
{
"id": "2-6",
"stash": [
{
"ADMIN_STATUS": "administratively down",
"INTERFACE": "HundredGigE0/0/0/29",
"OPER_STATUS": "administratively down"
}
]
},
{
"id": "2-7",
```

```
"stash": [
{
"ADMIN_STATUS": "administratively down",
"INTERFACE": "HundredGigE0/0/0/30",
"OPER_STATUS": "administratively down"
}
]
},
{
"id": "2-8",
"stash": [
{
"ADMIN_STATUS": "administratively down",
"INTERFACE": "HundredGigE0/0/0/31",
"OPER_STATUS": "administratively down"
}
]
},
{
"id": "2-9",
"stash": [
{
"ADMIN_STATUS": "administratively down",
"INTERFACE": "HundredGigE0/0/0/32",
"OPER_STATUS": "administratively down"
}
]
},
{
"id": "2-10",
"stash": [
{
"ADMIN_STATUS": "administratively down",
"INTERFACE": "HundredGigE0/0/0/33",
"OPER_STATUS": "administratively down"
}
]
},
{
"id": "2-11",
"stash": [
{
"ADMIN_STATUS": "administratively down",
"INTERFACE": "HundredGigE0/0/0/34",
"OPER_STATUS": "administratively down"
}
]
},
{
"id": "2-12",
"stash": [
{
"ADMIN_STATUS": "administratively down",
"INTERFACE": "HundredGigE0/0/0/35",
"OPER_STATUS": "administratively down"
}
]
}
],
"status": "success"
}
```


8. deviceSelection is not null, loop-enabled is false, method is post.

```
Input
{
"app-data": {
"data": {},
"devices": {
"source": [
{
"host": "spitfire250.cisco.com",
"ip": "172.22.141.250",
"label": "source",
"name": "spitfire250",
"productSeries": "Cisco 8000 Series Routers",
"productType": "Cisco 8201 Router",
"resource": "nso-173",
"softwareType": "IOS XR",
"softwareVersion": "24.4.2",
"uuid": "2c611a0a-31fc-4a11-80bf-170c9417014c",
"vendor": "Cisco Systems"
}
]
},
"executionMode": 2,
"phase": "",
"resource": "nso-173",
"stage": "post-migration"
},
"deviceSelection": {
"jsonPath": ".app-data.devices.source[0]"
},
"operations": [
{
"expressions": [
{
"condition": ".\"tailf-ned-cisco-ios-xr:HundredGigE\" | length > 0",
"message": "The interfaces are provided in the output",
"status": "success"
}
],
"id": 1,
"method": "get",
"path":
"restconf/data/tailf-ncs:devices/device=${.\"app-data\".device.name}/config/tailf-ned-cisco-ios-xr:interface/HundredGigE"
},
{
"expressions": [
{
"condition": ". != null and . != \"\"",
"message": "The interfaces details have been captured properly",
"status": "success"
}
],
"id": 3,
"method": "post",
"path":
"restconf/operations/devices/device=${.\"app-data\".device.name}/live-status/exec/any",
"payload": "{\"args\":
\"${.stash.\\\"tailf-ned-cisco-ios-xr-stats:output\\\".result |
split(\\\"HundredGigE0/0/0/\\\")[1] | split(\\\" \\\")[0] | tonumber + 1 |
tostring as $id | \\\"show interface HundredGigE0/0/0/\\\" + $id}\"}"
},
{
"expressions": [
```

```
{
"condition": ". != null and . != \"\"",
"message": "The interfaces details have been captured properly",
"status": "success"
}
],
"id": 2,
"loop-enabled": false,
"method": "post",
"path":
"restconf/operations/devices/device==${.\"app-data\".device.name}/live-status/exec/any",
"payload": "{\"args\":
\"${[.stash.\\\"tailf-ned-cisco-ios-xr:HundredGigE\\\"[].id as $id | \\\"show
 interface HundredGigE\\\" + $id][0]}\"}"
}
]
}
Output
{
"message": "All commands executed and evaluated successfully",
"processedResource": [
"2c611a0a-31fc-4a11-80bf-170c9417014c"
],
"stash": [
{
"id": 1,
"stash": [
{
"tailf-ned-cisco-ios-xr:HundredGigE": [
{
"id": "0/0/0/24",
"shutdown": [
null
]
},
{
"id": "0/0/0/25",
"shutdown": [
null
]
},
{
"id": "0/0/0/26",
"shutdown": [
null
]
},
{
"id": "0/0/0/27",
"shutdown": [
null
]
},
{
"id": "0/0/0/28",
"shutdown": [
null
]
},
{
"id": "0/0/0/29",
"shutdown": [
```

```
null
]
},
{
"id": "0/0/0/30",
"shutdown": [
null
]
},
{
"id": "0/0/0/31",
"shutdown": [
null
]
},
{
"id": "0/0/0/32",
"shutdown": [
null
]
},
{
"id": "0/0/0/33",
"shutdown": [
null
]
},
{
"id": "0/0/0/34",
"shutdown": [
null
]
},
{
"id": "0/0/0/35",
"shutdown": [
null
]
}
]
}
]
},
{
"id": 2,
"stash": {
"tailf-ned-cisco-ios-xr-stats:output": {
"result": "\r\n\rThu Oct 23 23:56:32.351 PDT\r\nHundredGigE0/0/0/24 is
administratively down, line protocol is administratively down \r\n  Interface
 state transitions: 0\r\n  Hardware is HundredGigE, address is 6c13.d5d2.10c0
 (bia 6c13.d5d2.10c0)\r\n  Internet address is Unknown\r\n  MTU 1514 bytes,
BW 100000000 Kbit (Max: 100000000 Kbit)\r\n     reliability 255/255, txload
0/255, rxload 0/255\r\n  Encapsulation ARPA,\r\n  Full-duplex, 100000Mb/s,
link type is force-up\r\n  output flow control is off, input flow control is
off\r\n  Carrier delay (up) is 200 msec\r\n  loopback not set,\r\n  Last input
 never, output never\r\n  Last clearing of \"show interface\" counters never\r\n
  5 minute input rate 0 bits/sec, 0 packets/sec\r\n  5 minute output rate 0
bits/sec, 0 packets/sec\r\n     0 packets input, 0 bytes, 0 total input
drops\r\n     0 drops for unrecognized upper-level protocol\r\n     Received
0 broadcast packets, 0 multicast packets\r\n              0 runts, 0 giants,
0 throttles, 0 parity\r\n     0 input errors, 0 CRC, 0 frame, 0 overrun, 0
ignored, 0 abort\r\n     0 packets output, 0 bytes, 0 total output drops\r\n
```

```
    Output 0 broadcast packets, 0 multicast packets\r\n     0 output errors,
0 underruns, 0 applique, 0 resets\r\n     0 output buffer failures, 0 output
buffers swapped out\r\n     0 carrier
transitions\r\n\r\n\r\nRP/0/RP0/CPU0:spitfire250#"
}
}
},
{
"id": 3,
"stash": {
"tailf-ned-cisco-ios-xr-stats:output": {
"result": "\r\n\rThu Oct 23 23:56:32.872 PDT\r\nHundredGigE0/0/0/25 is
administratively down, line protocol is administratively down \r\n  Interface
 state transitions: 0\r\n  Hardware is HundredGigE, address is 6c13.d5d2.10c4
 (bia 6c13.d5d2.10c4)\r\n  Internet address is Unknown\r\n  MTU 1514 bytes,
BW 100000000 Kbit (Max: 100000000 Kbit)\r\n     reliability 255/255, txload
0/255, rxload 0/255\r\n  Encapsulation ARPA,\r\n  Full-duplex, 100000Mb/s,
link type is force-up\r\n  output flow control is off, input flow control is
off\r\n  Carrier delay (up) is 200 msec\r\n  loopback not set,\r\n  Last input
 never, output never\r\n  Last clearing of \"show interface\" counters never\r\n
  5 minute input rate 0 bits/sec, 0 packets/sec\r\n  5 minute output rate 0
bits/sec, 0 packets/sec\r\n     0 packets input, 0 bytes, 0 total input
drops\r\n     0 drops for unrecognized upper-level protocol\r\n     Received
0 broadcast packets, 0 multicast packets\r\n              0 runts, 0 giants,
0 throttles, 0 parity\r\n     0 input errors, 0 CRC, 0 frame, 0 overrun, 0
ignored, 0 abort\r\n     0 packets output, 0 bytes, 0 total output drops\r\n
    Output 0 broadcast packets, 0 multicast packets\r\n     0 output errors,
0 underruns, 0 applique, 0 resets\r\n     0 output buffer failures, 0 output
buffers swapped out\r\n     0 carrier
transitions\r\n\r\n\r\nRP/0/RP0/CPU0:spitfire250#"
}
}
}
],
"status": "success"
}
```


9. loop-enabled is false and method is post with payload.

Input


```
{
"app-data": {
"data": {},
"devices": {
"source": [
{
"host": "spitfire250.cisco.com",
"ip": "172.22.141.250",
"label": "source",
"name": "spitfire250",
"productSeries": "Cisco 8000 Series Routers",
"productType": "Cisco 8201 Router",
"resource": "nso-173",
"softwareType": "IOS XR",
"softwareVersion": "24.4.2",
"uuid": "2c611a0a-31fc-4a11-80bf-170c9417014c",
"vendor": "Cisco Systems"
}
]
},
"executionMode": 2,
"phase": "",
"resource": "nso-173",
"stage": "post-migration"
```

```
},
"deviceSelection": {
"jsonPath": ".app-data.devices.source[0]"
},
"operations": [
{
"expressions": [
{
"condition": ".\"tailf-ned-cisco-ios-xr:HundredGigE\" | length > 0",
"message": "The interfaces are provided in the output",
"status": "success"
}
],
"id": 1,
"method": "get",
"path":
"restconf/data/tailf-ncs:devices/device=${.\"app-data\".device.name}/config/tailf-ned-cisco-ios-xr:interface/HundredGigE"
},
{
"expressions": [
{
"condition": ". != null and . != \"\"",
"message": "The interfaces details have been captured properly",
"status": "success"
}
],
"id": 3,
"method": "post",
"path":
"restconf/operations/devices/device=${.\"app-data\".device.name}/live-status/exec/any",
"payload": "{\n\"args\":
\"${.stash.\\\"tailf-ned-cisco-ios-xr-stats:output\\\".result |
split(\\\"HundredGigE0/0/0/\\\")[1] | split(\\\" \\\")[0] | tonumber + 1 |
tostring as $id | \\\"show interface HundredGigE0/0/0/\\\" + $id}\"\n}"
},
{
"expressions": [
{
"condition": ". != null and . != \"\"",
"message": "The interfaces details have been captured properly",
"status": "success"
}
],
"id": 2,
"loop-enabled": false,
"method": "post",
"path":
"restconf/operations/devices/device=${.\"app-data\".device.name}/live-status/exec/any",
"payload": "{\n\"args\":
\"${[.stash.\\\"tailf-ned-cisco-ios-xr:HundredGigE\\\"[].id as $id | \\\"show
 interface HundredGigE\\\" + $id][0]}\"\n}"
}
]
}
Output
{
"message": "All commands executed and evaluated successfully",
"processedResource": [
"2c611a0a-31fc-4a11-80bf-170c9417014c"
],
"stash": [
{
```

```
"id": 1,
"stash": [
{
"tailf-ned-cisco-ios-xr:HundredGigE": [
{
"id": "0/0/0/24",
"shutdown": [
null
]
},
{
"id": "0/0/0/25",
"shutdown": [
null
]
},
{
"id": "0/0/0/26",
"shutdown": [
null
]
},
{
"id": "0/0/0/27",
"shutdown": [
null
]
},
{
"id": "0/0/0/28",
"shutdown": [
null
]
},
{
"id": "0/0/0/29",
"shutdown": [
null
]
},
{
"id": "0/0/0/30",
"shutdown": [
null
]
},
{
"id": "0/0/0/31",
"shutdown": [
null
]
},
{
"id": "0/0/0/32",
"shutdown": [
null
]
},
{
"id": "0/0/0/33",
"shutdown": [
null
```

```
]
},
{
"id": "0/0/0/34",
"shutdown": [
null
]
},
{
"id": "0/0/0/35",
"shutdown": [
null
]
}
]
}
]
},
{
"id": 2,
"stash": {
"tailf-ned-cisco-ios-xr-stats:output": {
"result": "\r\n\rFri Oct 31 01:31:48.019 PDT\r\nHundredGigE0/0/0/24 is
administratively down, line protocol is administratively down \r\n  Interface
 state transitions: 0\r\n  Hardware is HundredGigE, address is 6c13.d5d2.10c0
 (bia 6c13.d5d2.10c0)\r\n  Internet address is Unknown\r\n  MTU 1514 bytes,
BW 100000000 Kbit (Max: 100000000 Kbit)\r\n     reliability 255/255, txload
0/255, rxload 0/255\r\n  Encapsulation ARPA,\r\n  Full-duplex, 100000Mb/s,
link type is force-up\r\n  output flow control is off, input flow control is
off\r\n  Carrier delay (up) is 200 msec\r\n  loopback not set,\r\n  Last input
 never, output never\r\n  Last clearing of \"show interface\" counters never\r\n
  5 minute input rate 0 bits/sec, 0 packets/sec\r\n  5 minute output rate 0
bits/sec, 0 packets/sec\r\n     0 packets input, 0 bytes, 0 total input
drops\r\n     0 drops for unrecognized upper-level protocol\r\n     Received
0 broadcast packets, 0 multicast packets\r\n              0 runts, 0 giants,
0 throttles, 0 parity\r\n     0 input errors, 0 CRC, 0 frame, 0 overrun, 0
ignored, 0 abort\r\n     0 packets output, 0 bytes, 0 total output drops\r\n
    Output 0 broadcast packets, 0 multicast packets\r\n     0 output errors,
0 underruns, 0 applique, 0 resets\r\n     0 output buffer failures, 0 output
buffers swapped out\r\n     0 carrier
transitions\r\n\r\n\r\nRP/0/RP0/CPU0:spitfire250#"
}
}
},
{
"id": 3,
"stash": {
"tailf-ned-cisco-ios-xr-stats:output": {
"result": "\r\n\rFri Oct 31 01:31:48.468 PDT\r\nHundredGigE0/0/0/25 is
administratively down, line protocol is administratively down \r\n  Interface
 state transitions: 0\r\n  Hardware is HundredGigE, address is 6c13.d5d2.10c4
 (bia 6c13.d5d2.10c4)\r\n  Internet address is Unknown\r\n  MTU 1514 bytes,
BW 100000000 Kbit (Max: 100000000 Kbit)\r\n     reliability 255/255, txload
0/255, rxload 0/255\r\n  Encapsulation ARPA,\r\n  Full-duplex, 100000Mb/s,
link type is force-up\r\n  output flow control is off, input flow control is
off\r\n  Carrier delay (up) is 200 msec\r\n  loopback not set,\r\n  Last input
 never, output never\r\n  Last clearing of \"show interface\" counters never\r\n
  5 minute input rate 0 bits/sec, 0 packets/sec\r\n  5 minute output rate 0
bits/sec, 0 packets/sec\r\n     0 packets input, 0 bytes, 0 total input
drops\r\n     0 drops for unrecognized upper-level protocol\r\n     Received
0 broadcast packets, 0 multicast packets\r\n              0 runts, 0 giants,
0 throttles, 0 parity\r\n     0 input errors, 0 CRC, 0 frame, 0 overrun, 0
```

```
ignored, 0 abort\r\n     0 packets output, 0 bytes, 0 total output drops\r\n
    Output 0 broadcast packets, 0 multicast packets\r\n     0 output errors,
0 underruns, 0 applique, 0 resets\r\n     0 output buffer failures, 0 output
buffers swapped out\r\n     0 carrier
transitions\r\n\r\n\r\nRP/0/RP0/CPU0:spitfire250#"
}
}
}
],
"status": "success"
}
```


Parent topic

Generic MOP workflows

nso-service-deployment-cwm-sol Workflow

This workflow automates the deployment of Network Service Orchestrator (NSO) services with built-in safety mechanisms, dry-run capabilities, and error handling for both new service creation and existing service updates.

Refer to this topic to review the schema and input/output examples for nso-service-deployment-cwm-sol.

Workflow details
Table 60: nso-service-deployment-cwm-sol details

Field	Value
Sr. No.	4
MOP action name	nso-service-deployment-cwm-sol
Description	This workflow automates the deployment of Network Service Orchestrator (NSO) services with built-in safety mechanisms, dry-run capabilities, and error handling for both new service creation and existing service updates.

Schema

```
{
"inputData": "string (required) - JSON payload for NSO RESTCONF service",
"path": "string (required) - NSO RESTCONF URI path",
"waitForDryRun": "boolean (optional, default: true) - pause for user approval
 after dry-run"
}
```

Input and Output Example

```
1. waitForDryRun is not True.
Input
{
"inputData": {
"application": [
{
"device": "ncs540-79",
"jinja-template": {
```

```
"template": "xr-native-int",
"version": "1"
},
"name": "test-native-xr-app8",
"variable": [
{
"name": "ADDR",
"value": "192.168.200.200"
},
{
"name": "MASK",
"value": "255.255.255.0"
}
]
}
]
},
"path": "restconf/data/golden-config:golden-config",
"resource": "nso-173"
}
Output
{
"dryRunResult": {
"data": " devices {\n     device ncs540-79 {\n         config {\n
 interface {\n+               TenGigE 0/0/0/1 {\n+                    ipv4
{\n+                      address {\n+                              ip
192.168.200.200;\n+                              mask 255.255.255.0;\n+
                }\n+                      }\n+                }\n
}\n         }\n     }\n }\n golden-config {\n+    application
test-native-xr-app8 {\n+        device ncs540-79;\n+        jinja-template
{\n+            template xr-native-int;\n+            version 1;\n+
}\n+        variable ADDR {\n+            value 192.168.200.200;\n+
}\n+        variable MASK {\n+            value 255.255.255.0;\n+        }\n+
    }\n }\n"
},
"message": "The execution completed successfully",
"status": "success"
}
2. waitForDryRun is True but creation of new service was rejected by the user
 using form.
Input
{
"inputData": {
"application": [
{
"device": "ncs540-79",
"jinja-template": {
"template": "xr-native-int",
"version": "1"
},
"name": "test-native-xr-app9",
"variable": [
{
"name": "ADDR",
"value": "192.168.200.200"
},
{
"name": "MASK",
"value": "255.255.255.0"
}
]
}
```

```
]
},
"path": "restconf/data/golden-config:golden-config",
"resource": "nso-173",
"waitForDryRun": true
}
Output
{
"message": "The execution has been terminated by the user",
"status": "terminated"
}
3. waitForDryRun is True but creation of new service was accepted by the user
 using form.
Input
{
"inputData": {
"application": [
{
"device": "ncs540-79",
"jinja-template": {
"template": "xr-native-int",
"version": "1"
},
"name": "test-native-xr-app9",
"variable": [
{
"name": "ADDR",
"value": "192.168.200.200"
},
{
"name": "MASK",
"value": "255.255.255.0"
}
]
}
]
},
"path": "restconf/data/golden-config:golden-config",
"resource": "nso-173",
"waitForDryRun": true
}
Output
{
"dryRunResult": {
"data": " golden-config {\n+    application test-native-xr-app9 {\n+
device ncs540-79;\n+        jinja-template {\n+            template
xr-native-int;\n+            version 1;\n+        }\n+        variable ADDR
{\n+            value 192.168.200.200;\n+        }\n+        variable MASK
{\n+            value 255.255.255.0;\n+        }\n+    }\n }\n"
},
"message": "The execution completed successfully",
"status": "success"
}
4. waitForDryRun is True but modification of existing service was accepted by
 the user using form.
Input
{
"inputData": {
"application": [
{
"device": "ncs540-79",
"jinja-template": {
```

```
"template": "xr-native-int",
"version": "1"
},
"name": "test-native-xr-app9",
"variable": [
{
"name": "ADDR",
"value": "192.168.200.201"
},
{
"name": "MASK",
"value": "255.255.255.0"
}
]
}
]
},
"path": "restconf/data/golden-config:golden-config",
"resource": "nso-173",
"waitForDryRun": true
}
Output
{
"dryRunResult": {
"data": " devices {\n     device ncs540-79 {\n         config {\n
 interface {\n                TenGigE 0/0/0/1 {\n                    ipv4
{\n                  address {\n-                              ip
192.168.200.200;\n+                           ip 192.168.200.201;\n
                }\n                       }\n                 }\n
}\n         }\n     }\n }\n golden-config {\n     application
test-native-xr-app9 {\n         variable ADDR {\n-            value
192.168.200.200;\n+           value 192.168.200.201;\n         }\n+
conflict-dev ncs540-79 {\n+           conflict-node
\"/ncs:devices/device{ncs540-79}/config/cisco-ios-xr:interface/TenGigE{0/0/0/1}/ipv4/address/ip\"
 {\n+                old-value 192.168.200.200;\n+                new-value
192.168.200.201;\n+                refcount 2;\n+            }\n+        }\n
    }\n }\n"
},
"message": "The execution completed successfully",
"status": "success"
}
```


Parent topic

Generic MOP workflows

pause-mop-cwm-sol Workflow

This workflow provides a manual checkpoint mechanism for Method of Procedure (MOP) execution, allowing users to pause automation workflows for review, verification, and decision-making before continuing with subsequent operations.

Refer to this topic to review the schema and input/output examples for pause-mop-cwm-sol.

Workflow details
Table 61: pause-mop-cwm-sol details

Field	Value
Sr. No.	5
MOP action name	pause-mop-cwm-sol

Field	Value
Description	This workflow provides a manual checkpoint mechanism for Method of Procedure (MOP) execution, allowing users to pause automation workflows for review, verification, and decision-making before continuing with subsequent operations.

Schema

N/A - orchestrator-injected app-data only

Input and Output Example

```
Input
{}
Output
{
"status": "success",
"message": "Workflow execution has been continued by the user."
}
```

Parent topic

Generic MOP workflows

sleep-cwm-sol. Workflow

This is a workflow to sleep for a specified amount of time. The user needs to provide the number of seconds the workflow needs to sleep as input and the workflow would remain idle for that period of time.

Refer to this topic to review the schema and input/output examples for sleep-cwm-sol.

Workflow details
Table 62: sleep-cwm-sol. details

Field	Value
Sr. No.	6
MOP action name	sleep-cwm-sol.
Description	This is a workflow to sleep for a specified amount of time. The user needs to provide the number of seconds the workflow needs to sleep as input and the workflow would remain idle for that period of time.

Schema

```
{
"sleepDuration": "integer (required, 1-86400) - sleep time in seconds"
}
```

Input and Output Example

```
Input
{
"sleepDuration": 2
}
Output
{
"sleepDuration": 2
}
```

Parent topic

Generic MOP workflows