NOKIA

NSP
Network Services Platform
Release 25.8

Network Automation Guide

Legal notice

Nokia is committed to diversity and inclusion. We are continuously reviewing our customer documentation and consulting with standards bodies to ensure that terminology is inclusive and aligned with the industry. Our future customer documentation will be updated accordingly.

This document includes Nokia proprietary and confidential information, which may not be distributed or disclosed to any third parties without the prior written consent of Nokia.

This document is intended for use by Nokia's customers ("You"/"Your") in connection with a product purchased or licensed from any company within Nokia Group of Companies. Use this document as agreed. You agree to notify Nokia of any errors you may find in this document; however, should you elect to use this document for any purpose(s) for which it is not intended, You understand and warrant that any determinations You may make or actions You may take will be based upon Your independent judgment and analysis of the content of this document.

Nokia reserves the right to make changes to this document without notice. At all times, the controlling version is the one available on Nokia's site.

No part of this document may be modified.

NO WARRANTY OF ANY KIND, EITHER EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO ANY WARRANTY OF AVAILABILITY, ACCURACY, RELIABILITY, TITLE, NON-INFRINGEMENT, MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE, IS MADE IN RELATION TO THE CONTENT OF THIS DOCUMENT. IN NO EVENT WILL NOKIA BE LIABLE FOR ANY DAMAGES, INCLUDING BUT NOT LIMITED TO SPECIAL, DIRECT, INDIRECT, INCIDENTAL OR CONSEQUENTIAL OR ANY LOSSES, SUCH AS BUT NOT LIMITED TO LOSS OF PROFIT, REVENUE, BUSINESS INTERRUPTION, BUSINESS OPPORTUNITY OR DATA THAT MAY ARISE FROM THE USE OF THIS DOCUMENT OR THE INFORMATION IN IT, EVEN IN THE CASE OF ERRORS IN OR OMISSIONS FROM THIS DOCUMENT OR ITS CONTENT.

Copyright and trademark: Nokia is a registered trademark of Nokia Corporation. Other product names mentioned in this document may be trademarks of their respective owners.

© 2025 Nokia.

Contents

About this document

Purpose

The Network Automation Guide introduces NSP network automation functions to operators and administrators by describing usage and features.

Scope

Network automation functions are available for OSS using programmable APIs. For general information about developer support, see the API page on the Network Developer Portal.

Document support

Customer documentation and product support URLs:

• Documentation Center
• Technical support

How to comment

Please send your feedback to documentation.feedback@nokia.com.

Part I: NSP Network Automation

Overview

Purpose

Describes the contents of the NSP Network Automation Guide.

Contents

1 Network automation in NSP

1.1 Network Automation functions

1.1.1 Functions included

Network Automation in NSP allows you to plan, perform, and audit device configuration at a network level rather than at an individual device level. Automating processes allows you to perform updates on multiple devices at one time.

Automation reduces or eliminates the need for manual intervention, which increases the productivity of operations teams across a wide range of tasks. These include provisioning new devices, configuring network services, monitoring and optimizing network and service performance, and troubleshooting problems.

The following, individually and in combination, are designed to assist operators in automating a variety of network management functions:

- Intents
- Workflows
- SDK for adaptors

1.2 What does this guide cover?

1.2.1 Guide scope

This guide provides an overview of many network automation topics in NSP.

The following table provides information about where to find additional network automation information.

Automation feature	Description	Information scope	Reference
Device Configuration	With Device Configuration, you can define reusable intent-based configuration templates covering physical configurations such as ports and services, and logical configurations such as QoS. These templates can be deployed to the network with fixed or flexible attributes.	Overview for operators	*NSP Device Management Guide*
		Detailed information for developers, including information about working with APIs.	Device Configuration tutorial on the Network Developer Portal

Automation feature	Description	Information scope	Reference
Operations	An operation is a series of executions, organized in phases, which are performed on a scope of NEs. You can use an operation to perform executions on large numbers of NEs concurrently; for example, upgrading all SR NEs in a network to the latest SR OS release.	Overview for operators	*NSP Device Management Guide*
		Detailed information for developers, including information about working with APIs.	Device Management tutorials on the Network Developer Portal
Service Management	Service Management allows for service provisioning and activation across networks accessible to the NSP. Through the application itself, or through the northbound interface (RESTCONF), Service Fulfillment enables users to make service requests that deploy services to the network using the NSP's mediation framework. A library exists with a predefined set of service models for both classic and model-mode SR OS networks. These service models can be installed and utilized by NSP to provide assurance that service configuration is as planned/requested, and also provides adaptability for custom service model requests.	Overview for operators	*NSP Service Management Guide*
		Detailed information for developers, including information about working with APIs.	Service Management tutorial on the Network Developer Portal
Resource Management	Resource Management manages resource pools and monitors pool usage. It maintains a list of available resource pools in a network, and allows the user to create and modify pools. Resource Management provides a quick way to assign IP addresses, numbers, or text strings to ports when you are setting up automated processes.	Overview for operators	*NSP System Administrator Guide*
		Detailed RESTCONF NBI information for developers	Resource Administrator tutorial on the Network Developer Portal

Part II: Artifacts

Overview

Purpose

Describes the use of NSP to manage artifacts.

Contents

3HE-21460-AAAB-TQZZA

15

2 Artifact management

Getting started using NSP Artifacts

2.1 What is an artifact?

2.1.1 Artifact overview

An artifact, in the context of an NSP function, is a piece of software that can be installed in a running NSP system to enable functionality for a type of device (for example, an SR OS router) or use case (for example, VPLS, VPRN, or software upgrade). NSP may require artifacts to be installed in order to perform specific functions. For example, the NE upgrade operation requires a series of workflows. The functional area within NSP that the artifact is designed to support is called the target application, or simply, target.

Artifacts are packaged for download as bundles, in zip format. See 2.2 "What is an artifact bundle?" (p. 20) for more information.

NSP facilitates the tracking and management of NSP artifacts and artifact bundles by providing a user interface for managing artifacts and artifact bundles in your system; namely, the views accessible under Artifacts in the NSP main menu. The installation and version management capabilities available in the Artifacts views are also supported through REST API calls.

Certain artifacts can be installed from the **Artifacts**, **Artifact Bundles** view, as shown in Table 2-1, "Artifacts supported for installation using the NSP Artifacts views" (p. 18), while other artifacts must be installed using legacy tools.

The following are examples of NSP artifacts:

- workflows, actions, and Jinja templates
- configuration, service, or ZTP intent types
- operation types
- attribute dependent alarm rules for Model Driven (SROS/SRL) devices
- adaptors for model-driven NE management

Artifacts are read-only in the NSP UI. You perform operations on artifact bundles, and the NSP manages the artifacts in the bundle accordingly.

Artifact dependency

An artifact may require the presence of other artifacts to work correctly. For example, an intent type may require NE adaptors to be available before it can be used to create a service. Network Intents can install the artifact, but the artifact is dependent on the NE adaptor to work. The dependency status parameter indicates whether dependencies exist and whether they are resolved, that is, whether the dependent artifact is present.

Artifact management
Getting started using NSP Artifacts
What is an artifact?

NSP

2.1.2 Adaptor artifacts

NSP supports a variety of Nokia and multi-vendor devices via pluggable adaptor artifacts, sometimes called "MDM adaptors", which are grouped by vendor and type/use case and bundled together for delivery.

Artifact guides are provided with the adaptors for each NE family and NSP release. For example, the Nokia SR OS Artifact Guide for Release 24.11 lists and describes the adaptor artifacts delivered to support management of Nokia SR OS devices by NSP Release 24.11 over model-driven interfaces. The artifact guides also contain information about the NSP functionality supported by the adaptors, NE compatibility with those NSP functions, NE commissioning information, and a list of active issues.

2.1.3 Obtaining artifacts

Artifacts evolve with the NSP product, and are delivered as part of one or several artifact bundles both with and between NSP releases. Depending on your feature packages, a set of artifacts may be included with your NSP installation. Additional or updated artifact bundles can be downloaded from the Nokia NSP software delivery site:

* Artifacts that can be installed using the Artifacts import functions in the NSP UI are located in the NSP/<release>/Artifacts or NSP/<release>/Beta Artifacts hierarchy. See 2.3 "How do I install an artifact bundle?" (p. 22) for installation information.

* Adaptors for NSP releases prior to 25.8 and their associated artifacts can be obtained from the NSP/<release>/Adaptors folder. For installation of older adaptors and other artifacts that cannot be installed using the Artifacts views, see "How do I install adaptor artifacts that are not supported in the Artifacts view?" in the *NSP System Administrator Guide*.

2.1.4 Compatible artifacts

The following table shows the artifact types that can be imported via NSP's Artifacts views.

| i | **Note:** If the target application for an artifact is not present when the bundle is installed, the installation of the artifact is skipped.

Table 2-1   Artifacts supported for installation using the NSP Artifacts views

Artifact type	Target application
Device mappings for MDM Server telemetry	MDM Telemetry Mappings
Resync device mappings for MDM managed NEs	MDM Resync Mappings
Resync device mappings for classically managed NEs	NFM-P Resync Mappings
YANG to YANG mappings for IETF	YANG-to-YANG Mappings
JSON ACT alarm rules mapping	NSP ACT Framework
Operation types	Device Management - Operations
Cloud native telemetry resources	Telemetry

Artifact management
Getting started using NSP Artifacts
What is an artifact?

NSP

Table 2-1   Artifacts supported for installation using the NSP Artifacts views   (continued)

Artifact type	Target application
Workflows, actions, and Jinja2 templates	workflow-manager These artifacts are found in the Workflows views after installation.
Intent types, except Service Management intent types	intent-manager These types intent are found in the Network Intents views after installation. To use intent types in other views, such as Device Configuration, you need to import them from the view they will be used in.
Service Management intent types	service-fulfillment The intent types are found in the Service Management, Intent Type Catalog view after installation.
Service Management metadata files for data sync	service-fulfillment
Model-driven adaptor artifacts[1]	MDM

Notes:

1. Adaptor artifact bundles with the prefix `nsp-adaptation` are supported for installation using the Artifacts views. If the adaptor bundle is not supported for installation from the Artifacts views, see "How do I install adaptor artifacts that are not supported in the Artifacts view?" in the *NSP System Administrator Guide* for information about installing it.

2.1.5 Access control

Artifact management requires an administrator role. Users with other roles can view artifact bundles and artifacts, but cannot install or uninstall artifact bundles.

Depending on your assigned role, some operations in the Artifacts views may not be available to you. Contact your system administrator if you can't access certain network resources or information.

2.1.6 Developer mode

If developer mode is disabled in the NSP, user actions are limited. Importing artifacts using the Artifacts views is supported for signed artifacts only.

⚠ **Note:** Disabling developer mode overrides access control settings. If developer mode is disabled, restricted functions are restricted for all users.

See the *NSP Installation and Upgrade Guide* for more information.

Artifact bundles

2.2 What is an artifact bundle?

2.2.1 Artifact bundle overview

An artifact bundle is a collection of artifacts that may make up a specific use case, for example, an NE backup, which may involve few artifacts, or a complex service rollout use case that involves many artifacts.

Artifacts are packaged for download as bundles, in zip format. You can obtain bundles from the Nokia NSP software download site and, in most cases, install them from the NSP Artifact Bundles view.

Artifacts are read-only in the NSP UI. You perform operations on artifact bundles, and the NSP manages the artifacts in the bundle accordingly.

When you import an artifact bundle from local storage to Artifacts and install it, the artifacts are installed by the dedicated deployers for the artifact type; for example, the Network Intents deployer installs intent type artifacts. After installation, the artifacts appear in the **Artifacts**, **All Artifacts** view.

Artifacts that were installed using legacy scripts are not displayed in the All Artifacts view.

2.2.2 Artifact bundle parameters

The following table describes the parameters that appear on the **Artifacts**, **Artifact Bundles** view.

Table 2-2   Artifact bundle parameters

Parameter	Predefined values	Notes
Columns in the Artifact Bundles view		
Artifact Bundle Name	—	If a bundle has a verified signature, the certificate is displayed, followed by the bundle's name.
Author	—	If a bundle has a verified signature, the author is displayed.
Artifact Bundle Version	—	Version number of the bundle. Artifacts in the bundle may have different version numbers.
Number of Artifacts	—	The number of artifacts in the bundle. This parameter is displayed when the bundle is installed.

Table 2-2 Artifact bundle parameters   (continued)

Parameter	Predefined values	Notes
Status		If applicable, the Status parameter in the Artifact Bundle Details panel will state if one or more artifacts in an imported bundle is installed by another bundle. View the list of artifacts for details about which bundles contain each artifact.
	Importing	The bundle is being imported.
	Imported	The bundle has been imported to NSP.
	Installing	At least one artifact in the artifact bundle is in an interim state: installation has not succeeded or failed yet.
	Installed	Installation has completed successfully for all artifacts in the bundle.
	Partially Installed	Installation of one or more artifacts in the bundle has succeeded, and one or more artifacts in the bundle has not been installed. An artifact may be partially installed if: • selective installation is available and not all artifacts in the bundle have been selected for installation • installation of one or more artifacts in the bundle has failed
	Installation Failed	Installation has failed for all artifacts in the bundle.
	Verification Failed	The bundle signature is not valid.
	Creation failed	One or more artifacts of the bundle have incompatible names. or there are syntax issues with the metadata.json file
	Uninstalling	The artifacts in the bundle are being uninstalled.
	Uninstalled	Uninstallation has completed successfully for all artifacts in the bundle.
	Partially Uninstalled	Uninstallation of one or more artifacts in the bundle has succeeded, and uninstallation of one or more artifacts has failed.

Release 25.8
September 2025
Issue 1

© 2025 Nokia.
Use subject to Terms available at: www.nokia.com/terms

3HE-21460-AAAB-TQZZA

21

Artifact management
Artifact bundles
How do I install an artifact bundle?

NSP

Table 2-2   Artifact bundle parameters   (continued)

Parameter	Predefined values	Notes
	Uninstallation Failed	Uninstallation has failed for all artifacts in the bundle.
	Waiting	The bundle is imported but the target has not yet responded to the installation request.
	Skipped	Artifact bundle installation was skipped because the target application responsible for installing the artifacts is not part of the deployment. View the Artifact List to see the target application; refer to "NSP product offerings" in the *NSP System Architecture Guide* to determine which feature package and installation options must be configured to install the target application.
	Obsolete	All the artifacts in the bundle have become obsolete.
Import Time	—	The time the artifact bundle was initially imported to NSP.
Additional parameters in the Artifact Bundle Details panel		
Selective Artifact Installation/ Uninstallation	Available, Not available	If selective installlation/uninstallation is available, artifacts in the bundle can be installed or uninstalled individually or at the bundle level. If selective installlation/uninstallation is not available, installation or uninstallation must be performed at the bundle level.

2.3   How do I install an artifact bundle?

2.3.1 Purpose

Use this procedure to install a bundle of artifacts using the NSP Artifact Bundles view. The artifact bundle must be compatible with the Artifacts views; see Table 2-1, "Artifacts supported for installation using the NSP Artifacts views" (p. 18). For installation of adaptors for model-driven NE management, see "How do I install adaptor artifacts that are not supported in the Artifacts view?" in the *NSP System Administrator Guide*.

Installing new versions of artifacts

It is not necessary to uninstall artifact bundles before installing new versions.

Artifact management
Artifact bundles
How do I install an artifact bundle?

NSP

If you install a bundle containing a new version of an artifact that is already installed in Artifacts, the new version is handled according to the version control policy of the target application. In many cases, the old version of the artifact remains in Artifacts, its status is changed to Obsolete, and the status of the bundle is changed to Partially Installed.

If you are installing a bundle whose contents may supersede artifacts you are using, such as an NSP service pack, ensure that you have performed any required backups; see 2.7.2 "Version management best practices" (p. 29).

Selective installation

Selective installation and uninstallation is available for some artifact bundles. With selective installation you can choose which artifacts in a bundle to install or uninstall. For example, you can install only the intent types you currently need.

Choose a bundle in the **Artifacts**, **Artifact Bundles** view to see the availability in the Artifact Bundle Details panel.

If selective installation/uninstallation is not available for a bundle, installation and uninstallation can only be performed on the entire bundle.

Importing and installing

There are three stages to installing an artifact bundle:

1. Obtain the artifact bundle from the Nokia NSP software download site and store it to your local system in .zip file format; see 2.1.3 "Obtaining artifacts" (p. 18).

2. Import the artifact bundle in the Artifacts views.

   Importing an artifact bundle transfers it from your local system to the NSP.

3. Install the artifacts in the artifact bundle.

   You can install the entire bundle at once, or, if selective artifact installation/uninstallation is available, choose one or more artifacts to install.

   When the bundle is installed, the contents of the zip file are extracted and made available to their target applications.

You can install a bundle at the time you import it, or later.

As the installation process completes for each artifact, the artifacts are added to the **Artifacts**, **All Artifacts** view. When the automatic reconcile operation occurs next, NSP checks the status of all artifacts, updates the artifact bundle status if possible, and retries any failed installations.

2.3.2 Before you begin

Before you begin, the artifact bundle must be downloaded from the Nokia NSP software download site and stored to your local system in .zip file format; see 2.1.3 "Obtaining artifacts" (p. 18).

2.3.3 Steps

1 ───────────────────────────────────

Open **Artifacts**, **Artifact Bundles**.

Release 25.8
September 2025
Issue 1

© 2025 Nokia.
Use subject to Terms available at: www.nokia.com/terms
3HE-21460-AAAB-TQZZA

23

Artifact management
Artifact bundles
How do I view artifact bundle contents?

NSP

2 ───────────────────────────────────────

Click **Import & Install**.

3 ───────────────────────────────────────

In the form that opens, drag and drop up to 10 zip files, or click **Browse** and navigate to the files on your system.

4 ───────────────────────────────────────

To install all artifacts in the artifact bundle immediately, click **Import & Install**. To import without installing, click **Import**.

The chosen operation is triggered immediately. The artifact bundle status is updated to Imported or Installed when NSP has confirmed the status of all artifacts in the artifact bundle.

5 ───────────────────────────────────────

To install the contents of a bundle in Imported status:

a. To install all artifacts in the bundle, choose **Install bundle** from the ⋮ (Table row actions) menu.

b. If selective installation is available for the bundle, choose artifacts to install:

1. Select a bundle and choose **View artifacts** from the ⋮ (Table row actions) menu.

2. In the Artifact List, select an artifact or Ctrl+click to select multiple artifacts.

3. Choose **Install** from the ⋮ (Table row actions) menu.

   Note: if the files list icon (⣿) appears instead of ⋮ (Table row actions), selective installation of artifacts is not available for the bundle.

The artifact bundle status is updated when NSP has confirmed the status of all artifacts in the artifact bundle.

Eɴᴅ ᴏꜰ sᴛᴇᴘs ───────────────────────────────

2.4 How do I view artifact bundle contents?

2.4.1 To open a list of artifacts in an artifact bundle

The artifact bundle must in Installed status to view the list of artifacts.

In the **Artifacts**, **Artifact Bundles** page, click on an installed bundle and choose **View artifacts** from the ⋮ (Table row actions) menu. A filtered page opens showing the list of artifacts in the bundle.

Click **Artifacts** to return to the **Artifacts**, **Artifact Bundles** view.

� **Tip:** Choose **Manage columns** ⋮ (Table setting & actions) menu in line with the column headers to hide or reveal columns, including columns that are hidden by default.

Artifact management
Artifact bundles
How do I uninstall an artifact bundle?

NSP

2.5 How do I uninstall an artifact bundle?

2.5.1 Purpose

Use this procedure to make the artifacts in a bundle unavailable to their target applications. You may choose to do this to perform a rollback of the artifacts in the bundle to the previous version, or as a first step towards completely deleting a bundle and its contents. Uninstallation of an artifact bundle changes the bundle status from Installed to Imported; if previous versions of any of the artifacts in the bundle exist in the system, they will be reinstalled according to the version management policy of the target, and their status updated to Installed. See 2.7 "What is artifact version management?" (p. 28) for more information about version management.

It is not necessary to uninstall artifact bundles before installing new versions. In most cases, when a new version of an existing artifact is installed (for example, a new version of an existing artifact is delivered in a different bundle), the older artifact version remains in Artifacts and its status is updated to Obsolete.

If an artifact appears in more than one bundle, uninstalling one of the bundles will not affect the other bundles. The artifact remains in the list and the Artifact Details panel shows the remaining bundles the artifact is in.

2.5.2 Steps

1 ─────────────────────────────────

Open **Artifacts**, **Artifact Bundles**.

2 ─────────────────────────────────

To uninstall the contents of a bundle in Installed status:

a. To uninstall all artifacts in the bundle, choose **Uninstall** from the ⋮ (Table row actions) menu.

b. If selective uninstallation is available for the bundle, choose artifacts to uninstall:

   1. Select a bundle and choose **View artifacts** from the ⋮ (Table row actions) menu.

   2. In the Artifact List, select an artifact or Ctrl+click to select multiple artifacts.

   3. Choose **Uninstall** from the ⋮ (Table row actions) menu.

     Note: if the files list icon (≗≡) appears instead of ⋮ (Table row actions), selective uninstallation of artifacts is not available for the bundle.

3 ─────────────────────────────────

In the confirmation form, click **Uninstall**.

The statuses of the bundle and its artifacts are updated to Uninstalled when the process succeeds, then revert to Imported.

When the automatic reconcile operation occurs next, NSP checks and updates the status of all artifacts and artifact bundles and retries any failed uninstallations.

Release 25.8
September 2025
Issue 1

© 2025 Nokia.
Use subject to Terms available at: www.nokia.com/terms
3HE-21460-AAAB-TQZZA

25

Artifact management
Artifact bundles
How do I uninstall an artifact bundle?

NSP

4

Optionally, proceed to delete the artifact bundle from the NSP. Select the bundle in the **Artifacts**, **Artifact Bundles** view and choose **Delete** from the ⋮ (Table row actions) menu.

Note: deleting the zip file removes the bundle and artifacts from NSP entirely, making them unavailable for rollback.

Eɴᴅ ᴏꜰ sᴛᴇᴘs

Artifact management
Artifacts
What does the All Artifacts view show?

NSP

Artifacts

2.6 What does the All Artifacts view show?

2.6.1 Artifact parameters

The following table describes the parameters that appear on the **Artifacts**, **All Artifacts** view.

Table 2-3   Artifact parameters

Parameter	Predefined values	Notes
Artifact Name	—	If an artifact has a verified signature, the certificate is displayed, followed by the artifact's name.
Author	—	If an artifact has a verified signature, the author is displayed.
Artifact Version	—	Version number of the artifact.
Dependency Status	Pending	NSP is checking for dependencies.
	No Dependencies Exist	—
	Resolved	All dependent artifacts are installed.
	Unresolved	At least one dependent artifact is not installed.
Retry	true	The artifact will be included in the next automatic reconcile operation. See the banner at the top of the Artifact Bundles view for the interval before the next reconcile.
	false	The artifact will not be included in the next automatic reconcile. There may be a problem with the artifact, or with the target.
Target Application	—	The functional area within NSP that the artifact is designed to support. The target also installs the artifact, therefore the target function must be installed in the NSP before the artifact can be installed.

Release 25.8
September 2025
Issue 1

© 2025 Nokia.
Use subject to Terms available at: www.nokia.com/terms

3HE-21460-AAAB-TQZZA

27

Table 2-3   Artifact parameters   (continued)

Parameter	Predefined values	Notes
Status	Installing	The artifact is being installed in NSP.
	Installed	Installation has completed successfully.
	Installation Failed	Installation has failed.
	Uninstalling	The artifact is being uninstalled.
	Uninstalled	Uninstallation has completed successfully.
	Uninstallation Failed	Uninstallation has failed.
	Verification Failed	The artifact digest is not valid.
	Creation Failed	The artifact has an incompatible name, or there are syntax issues with the metadata.json file.
	Waiting	The target has not yet responded to the installation request.
	Obsolete	The artifact has been superseded by a newer version of the artifact, installed by a different bundle.
	Skipped	Artifact installation was skipped because the target application responsible for installing the artifact is not part of the deployment. View the artifact parameters in the Artifact Details panel to see the target application; refer to "NSP product offerings" in the *NSP System Architecture Guide* to determine which feature package and installation options must be configured to install the target application.
Install Time	—	—
NSP Release Compatibility	—	The NSP releases the artifact is compatible with, for example, 25.0+ shows that the artifact is compatible with NSP 25.4 or later.

2.7   What is artifact version management?

2.7.1 Artifact version management

NSP supports version control for artifacts. Artifact versions are managed by the target deployer responsible for the artifact type.

For example, the unique identifier for a workflow is the workflow name. If multiple versions of the same workflow are available, such as 1.1.0 and 2.2.0, the deployer only installs the latest version, version 2.2.0.

In Network Intents, the unique identifier is intent type and major version, therefore if the artifact bundle contains version 1.1.0 and version 2.2.0 of the same intent type, the deployer installs both.

The version control policy of the target deployer also determines how new versions are handled when an existing version of an artifact is present. In many cases, the status of the lower version artifact is updated to Obsolete. The status of the higher version artifact is Installed.

2.7.2 Version management best practices

When you install an artifact bundle using the Artifact Bundles view, the artifacts in the bundle are installed in the target applications according to the version control policy of the target.

This means that new artifacts may supersede existing artifacts in use, especially for workflows or intent types as described in 2.7.1 "Artifact version management" (p. 28).

If you have updated artifacts in NSP after installation, for example, editing a workflow or creating intents for an intent type, you can export the changed artifacts to save your changes before installing new versions.

If you are installing a bundle, such as an NSP service pack, that may contain new versions of artifacts you have updated, the following steps are recommended:

*   Check the list of artifacts in the bundle.
    You can do this by opening the zip file in a zip file utility, or checking documentation that accompanies the bundle.
*   If the new bundle includes workflows or intent types you have modified, verify whether backups are needed.
    For example, if the bundle includes new service intent types with the same name and version as the ones you have modified, you can export your customized intent types in Network Intents. Similarly, if you have modified a workflow, you can export your modified workflow in Workflows.
*   If you need to save a local copy of your installed artifact bundles, although not required, you can export or download any bundles you have installed using the Artifacts views:
    1. Select **File Server** from the NSP main menu.
    2. Navigate through the folders: nokia, nsp, cam, artifacts, bundle. The zip files installed in NSP are displayed.
    3. Select a zip file and choose Download File from the ⋮ (Table row actions) menu.
    4. Save the zip file to your computer.
    The zip file is a local copy of the zip file as it was originally installed; it does not contain any modifications made in NSP.

    You can re-install the downloaded zip file to NSP if needed; see 2.3 "How do I install an artifact bundle?" (p. 22).

2.8 What is automatic reconcile of artifacts?

2.8.1 Automatic reconcile

The automatic reconcile operation retries failed installation and uninstallation operations. The reconcile operation sends eligible artifacts to the deployers for their targets to run the operations again. To be eligible for a reconcile, an artifact's status must be Installation Failed or Uninstallation Failed, and the retry parameter must be true.

Release 25.8
September 2025
Issue 1

© 2025 Nokia.
Use subject to Terms available at: www.nokia.com/terms

3HE-21460-AAAB-TQZZA

29

Artifact management
Artifacts
How do I retry a failed artifact operation?

NSP

By default, NSP performs a reconcile operation every three minutes.

2.9 How do I retry a failed artifact operation?

2.9.1 Retry parameter

NSP retries failed installation and uninstallation operations automatically; see 2.8 "What is automatic reconcile of artifacts?" (p. 29).

• To verify that a failed artifact is eligible for the next reconcile operation, verify that the retry parameter is true.

• To change the retry parameter, choose an artifact and click ⟳**Enable retry** or **Disable retry**.

• To check when the next automatic reconcile occurs, see the banner at the top of the **Artifacts**, **Artifact Bundles** view.

2.10 How do I edit or delete an artifact?

2.10.1 Modifying artifacts

Artifacts cannot be edited in the **Artifacts**, **All Artifacts** view. Modification, if possible, can be done in the target or using a developer tool.

Signed artifacts cannot be modified in NSP. Contact Nokia to request changes to a Nokia-signed artifact.

2.10.2 Deleting artifacts

Artifacts cannot be deleted individually. You can delete an artifact bundle with a Status of Imported, that is, a bundle that is not installed. Deleting an artifact bundle removes the bundle and artifacts from NSP entirely, making them unavailable for rollback.

Select the bundle and choose **Delete** from the ⋮ (Table row actions) menu.

See 2.5 "How do I uninstall an artifact bundle?" (p. 25) for information about uninstalling bundles that were installed using the NSP UI.

2.11 How do I roll back an artifact version?

2.11.1 Artifact version overwrites

When you install an artifact bundle, artifacts in the bundle are installed according to the version management policy of the target; see 2.7 "What is artifact version management?" (p. 28). In most cases, when a new version of an existing artifact is installed, the older version remains in Artifacts and the status is set to Obsolete.

To revert to the previous version, uninstall the new bundle. The older version of the artifact is reinstalled according to the version control policy of the target.

Part III: SDK Application

Overview

Purpose

Describes the SDK application for operators.

Contents

3 SDK

3.1 What is the SDK?

3.1.1 SDK overview

The software development kit (SDK) is a collection of software development tools in one installable package. They facilitate the creation of applications by having a compiler, debugger and sometimes a software framework. They are normally specific to a hardware platform and operating system combination.

The SDK can take the form of application programming interfaces (APIs) in the form of on-device libraries of reusable functions used to interface to a particular programming language, or it may be as complex as hardware-specific tools that can communicate with a particular embedded system.

The SDK facilitates adaptor development in the MDM environment.

3.1.2 Prerequisites

The following prerequisites must be met before installing and using the SDK.

- The RHEL 8 operating system must be installed on the host. The *NSP Installation and Upgrade Guide* provides detailed instructions for the RHEL OS installation.
- Docker-CE version 24.x with compatible Docker Compose is required for the SDK.
- To start the installation, log in as the root user on the SDK host.
- The deployer host must have following utilities: **Openssl**, **Curl**, and **Wget**.

Server size requirements

The deployer host must have at least 32 GB of memory and a minimum of 500 GB disk space available to run the SDK.

3.1.3 Port information

The following table describes the list of ports available on the SDK.

Table 3-1 Port information

Default Port	Type	Encryption Info	Description
80	TCP	There is no encryption.	This port redirects to port 443.
3002	TCP	Encryption provided by TLS. Strong ciphers are supported.	This port provides an HTTPS interface to the web application.
443	TCP	Encryption provided by TLS. Strong ciphers are supported.	This port provides an HTTPS interface to the web application.
22	TCP	Dynamic encryption is used. The cipher suite and strength as per RFC 4253.	This port provides remote access through the SSH/SCP/SFTP protocols.

Table 3-1 Port information (continued)

Default Port	Type	Encryption Info	Description
8101	TCP	Dynamic encryption is used. The cipher suite and strength as per RFC 4253.	This port provides remote access through the SSH/SCP/SFTP protocols.
8181	TCP	There is no encryption.	This is a Web console port.
5005	TCP	There is no encryption.	This port is for Java debugging.
162	UDP	When SNMPv3 is configured, static encryption is used. The cipher suite and strength are dependent on the network element.	This server port receives SNMP traps from network elements by default.
8282	TCP	Encryption provided by TLS. Strong ciphers are supported.	This port provides an HTTPS interface to the web application.

3.2 How do I install the SDK?

3.2.1 Before you begin

If the SDK is being installed on top of an existing SDK setup (upgrade scenario), use the command `./nsp-mdm-sdk.sh stop` to stop the SDK application and continue with the installation.

3.2.2 Steps

1

Download the MDM server and SDK application installer files from the NSP downloads page on the Nokia Support portal.

2

Open a console window and navigate to the folder where the installer files are located. Depending on your OS, you may need to change the permissions for each of the installer files.

For example, `chmod +x mdmsdk-*.sh`

3

Place the valid NSP license file in a folder containing the installer files before starting the installation.

> **i** **Note:** The NSP license file is the routing license obtained for the NSP system. If you do not already have a license, see "How do I install the NSP license?" in the *NSP System Administrator Guide* for instructions on how to obtain one. When submitting your license request, make sure to include the "Software Development Kit" feature package.
> If you already have an NSP system routing license but it does not include the "Software Development Kit" feature package, see "How do I add an NSP feature package?" in the *NSP System Administrator Guide* for instructions on how to do so.

Without a valid license file, the following prompt is displayed during the installation.

```
Please place the valid NSPLicense.zip in the current directory for SDK
installation and proceed.
```

4 ─────────────────────────────────

Run the downloaded MDM server installer file.

For example, **./mdmsdk-mdm-server-<*version*>.sh**

5 ─────────────────────────────────

Run the downloaded SDK application installer file.

For example, **./mdmsdk-installer-<*version*>.sh**

If the admin user is not already present, it is created with the default password. The following prompt is displayed.

```
Creating admin user
```

6 ─────────────────────────────────

The SDK installation begins.

> **i** **Note:** Python and Git are installed in the process. If the Python or Git download fails, see 3.18 "Web Proxy" (p. 47).

A prompt is displayed to specify the *<sdk-installation-directory>* where the SDK should be installed.

```
Input your desired installation directory
[Press Enter for default: /root/nsp-mdm-sdk]:
```

> **i** **Note:** When requested for an installation directory during upgrade scenario, enter the same value as in the previous installation directory.

7 ─────────────────────────────────

Perform one of the following

• Specify a new location for the installation directory.

• Specify an existing location for the installation directory. If the *<sdk-installation-directory>* already exists on the system, ensure that it has the necessary permissions to allow the SDK installer to write to it, otherwise, the installation will fail.

8 ─────────────────────────────────

When the installation is finished, close the console window.

> **i** **Note:** The ndk user and group are created as part of the installation process. User authentication and ownership of the SDK filesystem will be assigned to this user.

END OF STEPS ─────────────────────────────────

Release 25.8
September 2025
Issue 1

© 2025 Nokia.
Use subject to Terms available at: www.nokia.com/terms
3HE-21460-AAAB-TQZZA

35

3.3 What is user management?

3.3.1 SDK user management

To begin adaptor development with the SDK, you must create a new user in addition to the admin user created during SDK installation. The *<sdk-installation-directory>* includes the `user-management.sh` script for more feasible user management.

The following table lists the operations that can be performed using the `user-management.sh` script.

Table 3-2   User management Options

Option	Operation	Usage
1	List Users	To list all the available users. See 3.5 "How do I list all the available users?" (p. 37).
2	Create a New User	To create a new user. See 3.6 "How do I create a new user?" (p. 37).
3	Update Password	To update the password of the specific user. See 3.7 "How do I update the password of a user??" (p. 38).
4	Delete a User	To delete a user. See 3.8 "How do I delete a user?" (p. 39).
5	Troubleshoot	To list all user details. See 3.9 "How do I list all user details?" (p. 39). It lists the following information about each user. • User • Workspace • Downloads • Code-Server Extensions • Code-Server Status
6	List Docker Containers (code-server-*)	To list all available containers and their current status. See 3.10 "How do I list all the available containers?" (p. 40).
7	Update Pre-Existing System User(s) into SDK (Upgrade)	To update the SDK with a list of system users in the 'ndk' group. See 3.11 "How do I update the SDK with a list of system users?" (p. 40).
q	Quit	To exit from the user management menu.

3.4 How do I access the user management menu?

3.4.1 Steps

1

Navigate to the *<sdk-installation-directory>* directory and execute the command:

`./user-management.sh`

The following prompt is displayed.

```
Enter password for 'admin':
```

2

Enter the password. The following prompt is displayed.

```
Logging in as admin...
Login successful.
```

The user management menu with the supported operations is displayed.

```
Select an option:
1) List Users
2) Create a New User
3) Update Password
4) Delete a User
5) Troubleshoot
6) List Docker Containers (code-server-*)
7) Update Pre-Existing System User(s) into SDK (Upgrade)
q) Quit
Enter your choice (1/2/3/4/5/6/7/q):
```

END OF STEPS

3.5 How do I list all the available users?

3.5.1 Steps

1

Log in to the user management menu. See 3.4 "How do I access the user management menu?" (p. 36).

2

Enter 1. The following prompt is displayed.

```
Fetching list of users...
```

The users and their roles are displayed.

END OF STEPS

3.6 How do I create a new user?

3.6.1 Steps

1

Log in to the user management menu. See 3.4 "How do I access the user management menu?" (p. 36).

2 ───

Enter 2. The following prompt is displayed.

```
Enter new username:
```

3 ───

Enter the user name. The following prompt is displayed.

```
Enter password for new user '<user_name>':
```

4 ───

Enter the password. The following prompt is displayed.

```
Creating user '<user_name>'...
Message: <user_name> - user created successfully
```

The Code-Server container is created and associated with the newly created user.

```
Starting code-server containers...
code-server.sh executed successfully.
```

The docker container is restarted.

```
Restarting Docker container 'code-server-nginx'...
Docker container 'code-server-nginx' restarted successfully.
```

END OF STEPS ───

3.7 How do I update the password of a user??

3.7.1 Steps

1 ───

Log in to the user management menu. See 3.4 "How do I access the user management menu?" (p. 36).

2 ───

Enter 3. The following prompt is displayed.

```
Enter username for password update:
```

3 ───

Enter the user name. The following prompt is displayed.

```
Enter new password for user '<user_name>':
```

4 ───

Enter the password. The following prompt is displayed.

```
Updating password for user '<user_name>'...
```

```
       Message: <user_name> - password updated successfully
```

3.8 How do I delete a user?

3.8.1 Steps

1

Log in to the user management menu. See .

2

Enter 4. The following prompt is displayed.

```
Enter the username of the user to delete:
```

3

Enter the user name. The following prompt is displayed.

```
Deleting user '<user_name>'...
```
```
Message: <user_name> - user removed successfully
```
The Code-Server associated with the user is also deleted.
```
Removing Docker container 'code-server-<user_name>'...
```
```
Docker container 'code-server-<user_name>' removed successfully.
```
The docker container is restarted.
```
Restarting Docker container 'code-server-nginx'...
```
```
Docker container 'code-server-nginx' restarted successfully.
```

3.9 How do I list all user details?

3.9.1 Steps

1

Log in to the user management menu. See .

2

Enter 5. The following prompt is displayed.

```
Troubleshooting...
```

The information of each user is displayed in tabular format.

END OF STEPS

3.10 How do I list all the available containers?

3.10.1 Steps

1

Log in to the user management menu. See 3.4 "How do I access the user management menu?" (p. 36).

2

Enter 6. The following prompt is displayed.

```
--- Listing Docker Containers ---
```

The Code-Server containers and their current status are displayed.

END OF STEPS

3.11 How do I update the SDK with a list of system users?

3.11.1 Before you begin

This is a one-time action that should be performed only after upgrading the SDK. If this is a new installation, do not perform this operation.

3.11.2 Steps

1

Log in to the user management menu. See 3.4 "How do I access the user management menu?" (p. 36).

2

Enter 7. The following prompt is displayed.

```
ALERT!!! Updating the SDK with a list of system users in the 'ndk'
group.

This is a one-time action that should be performed only after
upgrading the SDK.

Do NOT perform this if this is a fresh installation.

Any system user other than admin in the 'ndk' user group will be
migrated as non-admin user in to the SDK after upgrade.
```

© 2025 Nokia.
Use subject to Terms available at: www.nokia.com/terms

40   3HE-21460-AAAB-TQZZA

Release 25.8
September 2025
Issue 1

System users are successfully added to the SDK and transformed into SDK users.

END OF STEPS

3.12 How do I backup the SDK?

3.12.1 Steps

1

Navigate to the *<sdk-installation-directory>* directory and execute the command:

`./nsp-mdm-sdk.sh backup`

The following prompt is displayed.

```
Backing up to the 'sdk-backup' folder.
```

```
Backup in progress (this may take 10 minutes or longer, depending on
the number of users and data size)...
```

Once the backup is complete, the following prompt is displayed.

```
Backup created successfully in folder: sdk-backup with filename:
backup_<timestamp>.tar.gz
```

> **i** **Note:** Back up file format: *backup_YYYY-MM-DD_HH*h-*mm*m-*ss*s.*tar.gz*

The SDK backup creates a backup of all repositories and other settings.

END OF STEPS

3.13 How do I restore the SDK?

3.13.1 Before you begin

Before you begin the restore process, place the valid backup file in the 'sdk-backup' folder in the *<sdk-installation-directory>* directory.

> **i** **Note:** If there is an issue with upgrading the current system OS (for example, from RHEL 7 to RHEL 8), it is recommended to create a fresh installation in a new system and then copy the backup file to the "sdk-backup" folder in the *<sdk-installation-directory>* directory of the new system and begin the restore process.

3.13.2 Steps

1

Navigate to the *<sdk-installation-directory>* directory and execute the command:

`./nsp-mdm-sdk.sh restore`

The following prompt is displayed, listing the available backup archives.

```
Enter the number of the backup archive to restore (or 'q' to quit):
```

Release 25.8
September 2025
Issue 1

© 2025 Nokia.
Use subject to Terms available at: www.nokia.com/terms

3HE-21460-AAAB-TQZZA

41

> **i** **Note:** If no backup files exist, the following prompt is displayed. `No backup archives`
>   `found in the backup archive folder: sdk-backup.`

2 —

Enter the number that corresponds to the backup archive you want to restore and press Enter.

The following prompt is displayed.

`Restoring backup: backup_<`*timestamp*`>.tar.gz ...`

`Backup 'backup_<`*timestamp*`>.tar.gz' restored successfully.`

> **i** **Note:** If a valid number is not entered, the following prompt is displayed. `Please try`
>   `again.`

Restore replaces the other settings and repositories with the same names as in the backup.

END OF STEPS —

3.14 Working with the SDK

3.14.1 Command line arguments

The **nsp-mdm-sdk.sh** file in the *<sdk-installation-directory>* directory is used to start, stop, or view the logs of the running SDK application.

Supported command line arguments are:

`./nsp-mdm-sdk.sh start`

`./nsp-mdm-sdk.sh stop`

`./nsp-mdm-sdk.sh logs`

`./nsp-mdm-sdk.sh logs_from_current`

`./nsp-mdm-sdk.sh mdm_console`

`./nsp-mdm-sdk.sh backup`

`./nsp-mdm-sdk.sh restore`

> **i** **Note:** The `logs` command populates all historical logs before populating the logs from the
>   current timeframe, whereas `logs_from_current` populates only the logs from the time the
>   command was supplied.

Once the SDK application has started, the running MDM server version can also be changed. To do this, run the following command and select a different MDM server version.

`./nsp-mdm-sdk.sh start`

3.14.2 Documentation

The following documents are available from the Help menu in the SDK application:

• user guide – documentation of the SDK application, modeling guides for the NSP FM AMI and

the NSP Equipment AMI, and detailed instructions for using the SDK Adaptor Designer Application to create and test adaptor suites

• troubleshooting guide – detailed description of the development environment (Gradle build and the MDM SDK Gradle plugins), troubleshooting information, and the online Javadoc API

3.15   How do I start the SDK application?

3.15.1 Before you begin

To successfully launch the SDK application, the necessary key and certificate for establishing an HTTPS connection must be available under `<sdk-installation-directory>` `/docker/nginx/ssl/` directory. See 3.16 "How do I generate a key and a certificate?" (p. 44).

Existing device modules must be reinstalled in the event of a restart.

3.15.2 Steps

1 ───────────────────────────────────────

Navigate to the *<sdk-installation-directory>* directory and execute the command:

`./nsp-mdm-sdk.sh start`

The following prompt is displayed requesting credentials for Git.

```
Input your desired user for GIT configuration or Press Enter for the
Default user [admin] to be configured:
```

```
Input your desired email for GIT configuration or Press Enter for the
Default email [admin@company.com] to be configured:
```

2 ───────────────────────────────────────

Gradle will be downloaded and installed.

> **i** **Note:** If there is issue in downloading gradle, you must manually place `gradle-4.8.1` and `gradle-5.6.4` under *<user-home-directory>* or try setting up a proxy. For more information on setting up the proxy, see 3.18 "Web Proxy" (p. 47), then return to Step 1 of this procedure.

3 ───────────────────────────────────────

The following prompt is displayed, listing the available MDM servers.

```
Please select one of the following MDM Server versions:
```

Enter the number that corresponds to the MDM server version you want to use for the SDK and press Enter.

4 ───────────────────────────────────────

The following prompt is displayed requesting to select build environment.

```
Please select one of the following build environments: (the generated
project build files will work in the selected build environment)
```

Enter the number that corresponds to the build environment you want to use for the SDK and press Enter.

E<small>ND OF STEPS</small>

3.16 How do I generate a key and a certificate?

3.16.1 Steps

1

Log in as the root user on the NSP, NFM-P VM, or alternative station, as required.

2

Open a console window.

3

Generate a private key. Enter the following command:

 # openssl genrsa -out key.pem *size_of_private_key*

where

size_of_private_key is the size of the private key file. The recommended size is 2048.

4

Generate a certificate signing request (CSR).

1. Enter the following:

   # openssl req -new -key key.pem -out csr.pem

   The following prompt is displayed:

   ```
   You are about to be asked to enter information that will be
   incorporated into your certificate request.

   What you are about to enter is what is called a Distinguished Name
   or a DN.

   There are quite a few fields but you can leave some blank.

   For some fields there will be a default value, If you enter '.',
   the field will be left blank.

   Country Name (2 letter code) [XX]:
   ```

2. Enter the country name. The following prompt is displayed:

   ```
   State or Province Name (full name) []:
   ```

3. Enter the state or province name. The following prompt is displayed:

   ```
   Locality Name (eg, city) [Default City]:
   ```

4. Enter the locality name. The following prompt is displayed:

Organization Name (eg, company) [Default Company Ltd]:

5. Enter the organization name. The following prompt is displayed:

Organizational Unit Name (eg, section) []:

6. Enter the organizational unit name. The following prompt is displayed:

Common Name (eg, your name or your server's hostname) []:

7. Enter the name identifier. The following prompt is displayed:

Email Address []:

8. Enter the E-mail. The following prompt is displayed:

Please enter the following 'extra' attributes to be sent with your certificate request

A challenge password []:

9. Enter the password. The folowing prompt is displayed:

An optional company name []:

10. Enter the company name. The utility generates a CSR file.

5 ──

Generate a certificate. Enter the following command:

openssl x509 -req -days *no_of_days* -in csr.pem -signkey key.pem -out cert.pem

where

no_of_days is the number of days for which the certificate is to be valid.

6 ──

Delete certificate signing request.

1. Enter the following:

   # rm csr.pem

   The following prompt is displayed:

   rm: remove regular file 'csr.pem'?

2. Enter the response: **Y**.

7 ──

Close the console window.

END OF STEPS ─────────────

3.17 Code-Server

3.17.1 What is Code-Server?

Code-Server is a Web-based IDE that simplifies project development. The SDK can run independently of the code-server, however the code-server requires the SDK to run.

3.17.2 How do I install the Code-Server?

Before beginning the Code-Server installation, ensure that the SDK is already installed.

1. Download the Code-Server installer from the NSP downloads page on the Nokia Support portal.

2. Open a console window and navigate to the folder where the Code-Server installer file is located. Depending on your OS, you may need to change the permissions for the installer file.

   For example, `chmod +x mdmsdk-code-server-<version>.sh`

3. Run the downloaded Code-Server installer file.

   For example, `./mdmsdk-code-server-<version>.sh`

3.17.3 How do I start the Code-Sever?

Navigate to the *<sdk-installation-directory>* directory and execute the command:

`./code-server.sh start`

The following prompt is displayed, if proxy is not set:

```
Web Proxy is not configured.

You can refer the SDK Installation guide for setting up Web proxy. Press
enter to proceed, (press q to exit):
```

After pressing Enter, set up the proxy. This is a one-time activity; once set, you won't be prompted to set it up again on future Code-Server startups. For more information on setting up the proxy, see 3.18 "Web Proxy" (p. 47).

After setting up the proxy, Code-Server starts up.

Execute the following command to see the Code-Server status:

`./code-server.sh logs`

3.17.4 Working with the Code-Server

Refer to the SDK application's User guide and Troubleshooting guide for more information.

3.17.5 How do I stop the Code-Sever?

Navigate to the *<sdk-installation-directory>* directory and execute the command:

`./code-server.sh stop`

3.18 Web Proxy

3.18.1 Why is Proxy needed?

If the server is in a private network and no proxy is configured, some features, such as installing Python and Git during SDK installation and installing new extensions through the Code-Server GUI, may not work.

3.18.2 How to set or unset Proxy?

Set Proxy:

export http_proxy=<*proxy*>

export https_proxy=<*proxy*>

Unset Proxy:

unset http_proxy

unset https_proxy

Part IV: Network Intents

Overview

Purpose

Describes the management of intent types and intents for operators.

Contents

4   Intent types

Overview

4.1   What are intent types?

4.1.1 Intent types

NSP allows you to create and execute intent-based automation flows to allow you to implement network-level planning and design. Intents translate high-level goals to the necessary network configuration, and the NSP generates and validates the configuration and continually verifies the state of the network.

With the use of intents, you can implement planning and design at a network level rather than at an individual device level. NSP translates the high-level goal from an intent to necessary network configuration. NSP generates and validates the configuration and continually verifies the state of the network and systems under its administrative control.

For example, you can have a goal to set up a service for a subscriber. In NSP, you define this goal using an intent type. The intent type defines the parameters for the service and provides mapping for device models. The intent type defines how an intent will work. When you create an intent using an intent type, you create an instance of the intent type with any required inputs provided. Network Intents maintains a catalogue of intent types created or imported by the user.

You can store service topology and configuration information in the database for later deployment and schedule audits of the network against the saved configuration requirements. Intents are customizable and can be deployed during runtime.

RESTCONF APIs are available for Network Intents; see the Intent Manager API page on the Network Developer Portal.

4.1.2 Intent types and intents

An intent type is a detailed specification for a desired network configuration. The intent type includes the YANG model and scripts and templates for the configuration to be performed. The intent type describes both what the creation of the intent looks like in the NSP, and what executing it does.

An intent is an instance of an intent type. The intent provides inputs to the intent type and executes the configuration.

Intent types can be modified or reused as often as needed, and can be deployed at any time for maximum flexibility.

4.1.3 Developer information

The Network Automation Guide provides an overview of network intents for operators.

For detailed information about the following topics for developers, see the Network Intents tutorial on the Network Developer Portal:

•   intent type design

Release 25.8
September 2025
Issue 1

© 2025 Nokia.
Use subject to Terms available at: www.nokia.com/terms
3HE-21460-AAAB-TQZZA   51

- resource file configuration
- schema form configuration

4.1.4 Developer mode

If developer mode is disabled in the NSP, users cannot create, import, modify, or delete intent types.

See the *NSP Installation and Upgrade Guide* for more information.

4.1.5 Access control

User groups are assigned access to menus, network resources, and Analytics resources by assigning roles in the NSP. Action permissions are assigned to roles.

> **i** **Note:** Disabling developer mode overrides access control settings. If developer mode is disabled, restricted functions are restricted for all users.

The following table describes scopes that are specific to intents.

Scope	Available operations
Manage Intent Types	Manage life cycle and delete intent types
Manage Intents	Create, edit, and delete intents, approve misalignments
Manage mediators	Perform all operations on the Mediators view
Import	Import all available file types
Operate-Intents	Synchronize, audit, mark as misaligned, run RPC actions, activate, migrate
Manage-Policies	All operations on the Policies and Policy Actions pages
Write Intent Types	Create and edit intent types

The scopes are combined to create the following roles:

- Developer:
  - Write Intent Types
  - Import
- Troubleshooter:
  - Manage Policies
  - Manage Mediators
  - Operate Intents
- Operator:
  - Manage Intents
  - Operate Intents
- Network Engineer
  - All scopes

Depending on your access settings, some of these pages or operations may not be available to you. See the *NSP System Administrator Guide* for more information.

Intent types
Overview
What are the components of an intent type?

NSP

4.2 What are the components of an intent type?

4.2.1 Intent type components

An intent type provides the logic for the creation of an intent, and for the execution of that intent. When a user creates an intent from the intent type, a Create Intent form launches requesting the required inputs. The YANG model, Target, and View components define the form. The Script and Resources components provide the logic required to realize the intent.

For detailed information about intent type design, see the intent types tutorial on the Network Developer Portal.

An intent type consists of the following components. The components appear as tabs in the Create and Edit forms.

> **i** **Note:** In the tabs with a list panel on the left, you can drag the split line to resize the panel if needed.

- General
- Target
- Yang
- Script
- Resources

4.2.2 General

The General tab includes the following panels.

General

The General panel shows the basic information about the intent type, for example, the name, version, labels and lifecycle state.

The **Mapping Engine** indicates the JavaScript engine used to create the intent type.

The **Labels** field lists the labels applied to the intent type. Labels are used to filter and group intent types. An intent type can have one or multiple labels.

If an intent type is compliant with a set of format requirements, such as Service Management, a label must be used to indicate what the intent type is configured for. For example, for an intent type to be imported by Service Management, it must have a ServiceFulfillment label.

The **Live state retrieval** check box dictates whether intent state attributes will be updated in the intent details on demand or when the intent is synchronized. Enable the check box to update state attributes on demand.

The **Composite** check box indicates a composite intent type.

The **Build** field displays the build number of the intent type from the intent type code. For Nokia-created intent types, the format is *release_x.y.z* where the numbers *x.y.z* are used for version control. The *x* number is the Version parameter in NSP. For example, if you install an intent type with the build number `22.9_2.1.1`, the intent type version will be 2, regardless of whether you have a version 1 in your system.

Intent types
Overview
What are the components of an intent type?

NSP

Policy

The Policy panel indicates the priority of the intent type and the targeted device. Policy configuration in the intent type helps NSP administer mass operations correctly.

The priority is used by NSP to determine the order of execution. An intent belonging to an intent type with a smaller priority number is executed by NSP before an intent belonging to an intent type with a bigger priority number. Setting the priority can ensure, for example, that port configurations are performed before the configuration of services that will require the ports.

The targeted device is used to allocate jobs so that only one job is executed at one time per device. This prevents delays and job timeouts due to a job attempting to run on a resource that is blocked by another job.

The targeted device is defined in JSON format using intent-type, targets, yang, inherit, and function names.

It can be defined in the following ways:

- target component of the current intent:

  ```
  "targetted-device": [ {
    "target-component": "device-name"
   }]
  ```

- target component of a dependent intent:

  ```
  "targetted-device": [ {
    "intent-type": "l2-infra", (empty or omitted indicates this intent type)
    "target-component": "device-name"
   }]
  ```

- leaf of the YANG of the current intent:

  ```
  "targetted-device": [ {
  "intent-type": "l2-infra",  (empty or omitted indicates this intent type)
    "yang": "fiber:fiber/pon-port/device-name"
   }]
  ```

- same devices as a dependent intent:

  ```
  "targetted-device": [ {
    "inherit": "true",
    "intent-type": "device-fx"
   }]
  ```

- in special cases, we can use special logic in JavaScript:

  ```
   "targetted-device": [ {
    "function": "getTargettedDevices"
   }]
  ```

Migration

The Migration panel is displayed for intent types with a version number greater than 1. The migration table shows the configured migration and rollback paths between versions of the intent type.

Intent types
Overview
What are the components of an intent type?

NSP

4.2.3 Target

The target attribute of the intent type is used to uniquely identify any intent created from the intent type. The definition must be in JSON format.

The target attribute is made up of one or more target components.

All attributes are optional: the only requirement is that the intent is uniquely defined.

When an intent is created, the target components are displayed as input fields in the intent creation dialog. NSP uses the user inputs to create the target value.

The following table describes the target attributes.

Attribute	Description	Example
name	Internal name of the component. This value is not displayed in the intent creation dialog.	device-id
value-type	Two types are supported: • STRING • NUMBER	STRING
i18n-text	The Create Intent form display name for the component	Device Name
order	The order in which this is displayed in the Create Intent form; 1 being the highest. You can give two components the same order; they appear side by side in the form.	1
pattern	Supported pattern check as supported by YANG pattern statement.	"AV-[a-zA-z0-9]+"
function-name	The name of the suggest function that provides a list of suggested inputs, if you want to use one. The suggest function is defined in the Script panel of the intent type.	"suggestDeviceNames" The function returns device names to populate the target attribute.
length	Supported length as supported by YANG length statement.	1..10

The following sample shows the structure of a target component definition.

Sample target

The following sample creates a target component:

```
{
  "target-component": [
    {
      "i18n-text": "Service Name",
```

Release 25.8
September 2025
Issue 1

© 2025 Nokia.
Use subject to Terms available at: www.nokia.com/terms
3HE-21460-AAAB-TQZZA

55

Intent types
Overview
What are the components of an intent type?

NSP

```
        "name": "service-name",

        "pattern": "[A-Za-z0-9_]{1,}",

        "value-type": "STRING",

        "order": 1

      }

   ]

}
```

You can include more than one target component, if needed. The target value separates the two components with a hash; for example, `service#port`.

The sample target component is displayed in the intent creation form as shown in . The user input becomes the target attribute.

Figure 4-1   Intent creation form with Sample target and Sample YANG

4.2.4 Yang

The Yang panel provides an abstraction for the instance of the intent type.

The following types of arguments are supported:

*   Configuration arguments: attributes where the user needs to provide a value

*   State arguments: attributes where the intent type reports relevant state information

The Yang model must extend the Nokia IBN Yang definition. The arguments must use the `ibn:configuration` container.

4.2.5 Script

The Script tab contains the realization logic of the intent type. This tab contains the programming required for the intent configuration to be performed.

Intent types
Overview
What are the components of an intent type?

NSP

The script contains an intentObject and needs to implement some predefined functions. When a user triggers an action, the predefined functions are called by the framework to fulfill the task. These methods have implementation specific to an Intent type.

The following functions are optional.

- Validate runs at the time an intent is synchronized, either at or after intent creation. If the intent input fails the validation condition, the intent cannot be synchronized.

- Reconcile is the opposite of a synchronize function. It will pull the configuration from the associated object and update the intent configurations to match it.

Script engines

NSP uses the GraalJS JavaScript engine by default. The JS JavaScript engine, also called Nashorn, is also available.

GraalJS supports some ECMAScript features and capabilities that are not available with Nashorn. For more details, see the Network Intents information on the NSP Developer Portal.

Sample script

The following sample shows a basic script structure.

```
var RuntimeException = Java.type('java.lang.RuntimeException');



var prefixToNsMap = {

    "ibn" : "http://www.nokia.com/management-solutions/ibn",

    "nc" : "urn:ietf:params:xml:ns:netconf:base:1.0",

    "device-manager" : "http://www.nokia.com/management-solutions/anv",

};



var nsToModule = {

    "http://www.nokia.com/management-solutions/anv-device-holders" :
"anv-device-holders"

};



function synchronize(input) {

  var target = input.getTarget();
```

Release 25.8
September 2025
Issue 1

© 2025 Nokia.
Use subject to Terms available at: www.nokia.com/terms
3HE-21460-AAAB-TQZZA

57

Intent types
Overview
What are the components of an intent type?

NSP

```
        var config = input.getIntentConfiguration();

        var topology = input.getCurrentTopology();

        //Retrieve the network state, it exists in the input argument.

        var networkState = input.getNetworkState().name();

        var customNetworkState = input.getCustomRequiredNetworkState();

        logger.info("######### NETWORK STATE "+ networkState);


        //If the network state is 'custom' then we must look in the
    customRequiredNetworkState property to get the actual state

        if(networkState === 'custom'){

          logger.info("######### CUSTOM NETWORK STATE "+ customNetworkState);


          result.setSuccess(true);

          return result;

    }


    function audit(input) {

       var report = auditFactory.createAuditReport(null, null);

       // Code to audit goes here


       return report

    }


    function validate(syncInput) {

      var contextualErrorJsonObj = {};
```

Intent types
Overview
What are the components of an intent type?

NSP

```
var intentConfig = syncInput.getJsonIntentConfiguration();


    // Code to validation here. Add errors to contextualErrorJsonObj.

// contextualErrorJsonObj["attribute"] = "Attribute must be set";

    if (Object.keys(contextualErrorJsonObj).length !== 0) {

        utilityService.throwContextErrorException(contextualErrorJsonObj);

    }

}

/*

RECONCILE

The return object that is expected from a reconcile is ReconcileOutput

ReconcileOutput POJO defined like below

  updatedIntentConfiguration - filled when the intent configuration can
be adapted

  updatedStateXML - filled when the Intent State can be adapted

  alignmentState - filled if intent needs to be aligned state after
reconcile

  updateDependents - whether dependent intents to be updated

  syncDependents - whether dependent intents to be synched

  topology - filled if there is change in Topology /  Intent Dependency ,
XtraInfo

*/

function reconcile (input){

  var reconcileOutput;

  var topologyXtraInfo = null;

  var topologyXtraInfo = null;
```

Release 25.8
September 2025
Issue 1

© 2025 Nokia.
Use subject to Terms available at: www.nokia.com/terms
3HE-21460-AAAB-TQZZA

59

Intent types
Overview
What are the components of an intent type?

NSP

```
var target = input.getTarget();

var topology = input.getCurrentTopology();

var targettedDevice = input.getTargettedDevices();

var networkState = input.getNetworkState();

var customRNS = input.getCustomRequiredNetworkState();

var intentConfigXml = input.getIntentConfiguration();



//Here we are changing the configuration

var testStringNode = intentConfigXml.getElementsByTagName
("test-string").item(0);



if(testStringNode.getTextContent() == 'reconcile'){

  testStringNode.setTextContent("reconcile");

}



var updatedConfig = domUtils.documentToString(intentConfigXml);

 if (networkState == "delete") {

logger.info("Reconcile intent with input networkState {}", networkState);

     throw new RuntimeException("Reconcile with required-network-state
as 'delete' is not possible");

  }

  if (networkState == "suspend") {

logger.info("Reconcile intent with input networkState {}", networkState);

     return new ReconcileOutput(null, null, "false", false, false,
null);

  }
```

```
      //Return for reconcile

      return new ReconcileOutput(updatedConfig, null, "true", false, false,
null);

   }
```

4.2.6 Resources

The Resources tab shows the list of resource files required to realize the intent. The resources are a library of files that the intent type can access.

Different intent types that target similar system configurations often use similar logic. The use of resource files allows the common logic pieces to be stored as framework and mapping files, and loaded when the intent runs.

The resource files can be in various formats, for example, YANG, JavaScript, or JSON.

The GraalJS script engine supports folders in the Resources library.

To configure resource libraries, click ⋮ **More**, **Resource Management** at the top of any view in the Network Intents path.

For detailed information about resource file creation, see the Network Intents tutorial on the Network Developer Portal.

4.3 What is a view?

4.3.1 View files

Views, or view files, are used to augment the intent type YANG to create the input form the user will see when creating an intent, or when using intent-based configuration in another NSP function.

Nokia-signed intent types, which are installed using Artifacts, cannot be edited except to manage views.

To see the list of views for an intent type, double-click on the intent type from the **Network Intents**, **Intent Types** view and click **Manage Views**.

The list of views includes:
- viewConfig files: a viewConfig file defines the changes to make to the attributes in the intent type YANG to define the fields that will appear in the input form and their properties. For example, if the YANG declares that the intent creation form will contain two string fields, a viewConfig file can provide names for the fields.
- schema forms: the schema form is automatically generated from the viewConfig file. NSP implements the schema form to create the input form.
  Choose a schema form in the views list to see a form preview.
- view.settings files: the view.settings file is automatically generated. It provides a link to the module and container defined in the intent type YANG to identify the YANG entry point.

For detailed information about View file configuration for developers, see the ViewConfig section of the Network Intents tutorial on the Network Developer Portal.

Release 25.8
September 2025
Issue 1

© 2025 Nokia.
Use subject to Terms available at: www.nokia.com/terms
3HE-21460-AAAB-TQZZA

61

Default viewConfig file

When an intent is created from the Intent Types view, NSP searches for a viewConfig file called `default.viewConfig` to render the intent creation form.

If there is no `default.viewConfig` file present, NSP renders a basic form from the intent type YANG. To customize the default intent creation form, configure a `default.viewConfig` file; see .

Other viewConfig files can be created for use from other NSP functions.

Parent and child viewConfig files

A viewConfig file can be attached to another as a child. The child file inherits all the attributes and rules configured in the parent file.

The child file can add attributes or overwrite attribute values and rules.

For example:

- The parent file augments one attribute: `attr1`. The title of `attr1` is set to `Parent`.
- The child file augments two attributes: `attr1` and `attr2`. The title of `attr1` is set to `child` and `attr2` is set to `child2`.

In the Form Preview, if you choose the parent schema form, one attribute is changed. The title is `Parent`. If you choose the child schema form, two attributes are changed. The titles are `child` and `child2`.

4.4 What is a composite intent type?

4.4.1 Composite intent type

A composite intent type is an intent type that encompasses multiple child intent types.

An intent type that is a child of a composite intent type can be managed on its own, or included in multiple composite intent types.

Combining several intent types into a composite intent type allows you to perform operations on the child intent types collectively. For example, you could use a composite intent type for managing several services as an end-to-end service: creating an intent from the composite intent type can configure all three services at once, and you can audit or sync the intent to monitor and maintain configuration on the end-to-end service as a whole.

A composite intent type has the same required components as other intent types, with the addition of an XML mapping file. The mapping file provides information about how to apply the parameters from the composite intent to the child intents. For example, the composite intent type may have a parameter called Service 2 Name. The mapping file tells NSP to apply the value of Service 2 name to the Service Name parameter of a specified child intent.

When the user creates an intent for the composite intent type, the user configures the parameters in the composite. NSP creates intents for the child intent types, passing the user inputs to the child intents as specified in the mapping file. You can configure dependency to the intents, for example, child 2 requires child 1. With this configuration, NSP creates the intent for child intent type 1 first, then creates the intent for child intent type 2.

If no order is configured, all the intents are created at the same time.

The following restrictions apply:

• The intent types that you want to add as child intent types must be created before the composite intent type.

• Composite parameters cannot be changed in edit mode. You can edit a composite intent to change the name, for example, but not to change the list of child intents or change child intent dependency.
If you need to change composite parameters, delete the composite intent type.

4.5　What is an intent type version?

4.5.1　Intent type version

Intent type versions are a tool to allow working with different intent type YANG models.

Changing certain attributes in an intent type can affect OSS integrations. If you need to change OSS-affecting attributes, you can create a new version of the intent type with the change, leaving earlier versions available to the OSS. Intents can be migrated between versions of the intent type when the OSS integration has been updated to work with the new version. Intents can be migrated from one version to another, avoiding the need to create them again.

If the new definition of the intent is incompatible with the previous version, for example if a new attribute has been added, you can configure migration handlers and rollback handlers as part of the script of the newer version.

Migration and rollback handler functions can also be used to indicate how approved misalignments are handled when intents are migrated.

The Migration table in the General tab of the new version shows the configured migration paths and the functions that support migration and rollback.

• Supported version: the Supported Version column in the Migration table refers to the "from" version in a migration path. When a version is created, for example, version 3, the migration table is populated with rows for versions 1 and 2. However, migration and rollback handler functions are not automatically assigned.

• Migration handler: the migration handler function ensures that your data instance remains in line with the updated YANG model, for example by adding a new attribute that was included in the newer version.

• Rollback handler: the rollback handler function manages migration from a newer version to an older version, for example, removing an attribute that was added by the new version.

See the Network Intents tutorial on the Network Developer Portal for details about scripting migration.

4.6　What is intent dependency?

4.6.1　Intent dependency

Many use cases involve multiple intent types where one intent can be dependent on one or many other intents, for example when both intents target the same network object.

Release 25.8
September 2025
Issue 1

© 2025 Nokia.
Use subject to Terms available at: www.nokia.com/terms
3HE-21460-AAAB-TQZZA

63

Network Intents supports two types of dependency:

- Existence dependency: An intent's existence is dependent on the existence of one or many other intents. When existence dependency is present, the direct dependents and the direct dependencies appear in the intent details page. If an intent has direct dependencies, it can't be deleted unless the depended intents are deleted first.

- Argument dependency: An intent's configuration logic could be dependent on the values of another intent's arguments.

Intent dependency is configured as part of the Script of an intent type. See the Network Intents tutorial on the Network Developer Portal for details.

4.7 What are intent type states?

4.7.1 Intent type states

The state of an intent type shows whether it is available for intent creation:

- Draft: intents cannot be created.
  When an intent type is cloned or a new version is created, the clone or new version is created in Draft state.

- Released: intents can be created.

- Phased-out: intents cannot be created. Existing intents can be synced and audited.

A user with the required permissions can edit an intent type in any state, including changing the state.

If intents have been created from an intent type, it is a best practice to create a new version of the intent type and edit the new version instead of editing the version with intents present.

Intent types
Intent type procedures
How do I create an intent type?

NSP

Intent type procedures

4.8 How do I create an intent type?

4.8.1 Purpose

Use this procedure to write an intent type in the Network Intents UI. To import a zip file of existing intent types from your computer, see 4.9 "How do I import an intent type from my computer?" (p. 67). To import intent types using Artifacts; see 2.3 "How do I install an artifact bundle?" (p. 22).

> **i** **Note:** GraalJS is the preferred script mapping engine for developing intent types. Intent types created using the Nashorn (JS) mapping engine can be edited in the UI but cannot be created.

4.8.2 Steps

1

Open **Network Intents**, **Intent Types**.

2

Click ＋**Intent Type**.

3

In the **Create Intent Type** form that opens, enter a name for the intent type.

> **i** **Important!** Intent Type names must meet the following requirements.
> • The name must begin with a letter.
> • The name of an intent type can only contain lowercase letters, dashes or underscores. If other characters are present, the intent type cannot be created.

4

Configure the parameters in the **General** pane as required:
• Enter labels in the **Labels** field. The intent type must have at least one label.
• Click **Live state retrieval** to enable updating intent state attributes in the intent details on demand.
• Click **Composite** to create a composite intent type.

5

Configure the parameters in the **Policy** pane.

If you are not creating a composite intent type, proceed to Step 7.

6

If applicable, configure the parameters in the **Composite Intent Types** pane:

Release 25.8
September 2025
Issue 1

© **2025 Nokia.**
Use subject to Terms available at: www.nokia.com/terms
3HE-21460-AAAB-TQZZA

65

Intent types
Intent type procedures
How do I create an intent type?

NSP

1. Click ＋**Add** to add child intent types.

2. In the **Name** field, enter an alias for the child intent type, to be used by the mapping file.

3. In the **Intent Type** and **Version** fields, enter the name and version of the intent type you are adding.

4. The **Requires Child Intent** parameter creates a dependency between child intent types. If the intent type you are adding is dependent on another child intent, enter the name of the other intent and click ＋.

5. Configure target component mapping for each target component in the child intent type:

   a. Click ＋**Add**

   b. Choose **From Target Component** or **From Data Component**.

   c. In the **Name** field, enter the To mapping, that is, the component in the child intent to configure.

   d. In the **From Target Component** or **From Constant Value** field, enter the From mapping, that is, the composite intent type component or data value to apply to the child intent.

      The following sample shows mapping of a component in the composite intent called Name to a child intent component called serviceName.

   e. Click **Add** to finish adding the target component mapping.

      Repeat this step as needed to configure mapping for all target components in the child intent type.

6. Click **Add** to finish adding the child intent type.

   Repeat this step as needed to add another child intent type.

7 ───────────────────────────────────────

In the **Target** tab, configure the identification information for intents created from the intent type.

Modify the default text, or enter, copy or paste as needed.

8 ───────────────────────────────────────

In the **YANG** tab, configure the form details for creation of an intent.

1. Choose the YANG module in the panel on the left.

2. Modify the default text, or enter, copy, or paste as needed.

9 ───────────────────────────────────────

Add additional YANG modules as required.

1. Click **Yang** ＋.

2. In the **Add Yang** form that opens, enter the name of the module and click **Add**.

Intent types
Intent type procedures
How do I import an intent type from my computer?

NSP

3. Enter or paste the YANG text in the panel.

4. Repeat as needed to add more modules.

10 ───

In the **Script** tab, configure the script for executing the intent.

Modify the default text, or enter, copy, or paste as needed.

11 ───

In the **Resources** tab, add or modify resources as required.

Note: Resource files in the framework directory should not be modified.

1. Click **Resource** +.

2. In the Add Resource form that opens, enter the name of the resource and click **Add**.

3. Enter or paste the resource file in the panel.

4. If the intent type has child intents, a generated mapping file called config-transformer is present. Verify the mapping information in the file and make changes if required.

Resource libraries can be configured using NSP Resource Management. To open Resource Management, click ⋮ **More**, **Resource Management** at the top of any Network Intents view.

12 ───

Click **Create**.

END OF STEPS ─────────────────────────────────

4.9 How do I import an intent type from my computer?

4.9.1 Purpose

This procedure describes importing a zip file of intent types from your computer to Network Intents. You can also import intent types using Artifacts; see 2.3 "How do I install an artifact bundle?" (p. 22). If your zip file includes artifacts other than intent types, you must use Artifacts to perform the import.

4.9.2 Steps

1 ───

Open **Network Intents**, **Intent Types**.

2 ───

Click **Import**.

3 ───

In the form that opens, navigate to the file you want and click **Open**.

Release 25.8
September 2025
Issue 1

© 2025 Nokia.
Use subject to Terms available at: www.nokia.com/terms
3HE-21460-AAAB-TQZZA

67

Intent types
Intent type procedures
How do I clone an intent type?

NSP

4

If the intent type and version are already present, a dialog opens:

a. Click **Overwrite** to overwrite the existing intent type.

b. Click the check box to import the intent type with a new version number, and click **Import**.

c. Click **Cancel** to cancel the import.

END OF STEPS

4.10 How do I clone an intent type?

4.10.1 Purpose

Use this procedure to create a copy of an intent type. You can edit the clone as needed and keep the original intent type intact.

> **i** **Note:** If you clone a signed intent type, the clone is not signed.

4.10.2 Steps

1

Open **Network Intents**, **Intent Types**.

2

Choose an intent type and choose ⋮ (Table row actions), **Clone Intent Type**.

3

In the **Clone Intent Type** form, enter a name for the clone.

4

Click **Clone**. The cloned intent type is added to the intent types list in draft state.

5

See to edit the clone.

END OF STEPS

4.11 How do I edit an intent type?

4.11.1 Purpose

Use this procedure to modify an intent type.

> **i** **Note:** GraalJS is the preferred script mapping engine for developing intent types.

Intent types
Intent type procedures
How do I add or change a View file?

NSP

4.11.2 Steps

1

Open the **Edit Intent Type** form:

a. From **Network Intents**, **Intent Types**, select an intent type and click ⋮ (Table row actions), **Edit**.

b. From the intents list for an intent type, click **Edit Intent Type**.

2

Make changes as needed.

3

Click **Update**.

Eɴᴅ ᴏꜰ ꜱᴛᴇᴘꜱ

4.12 How do I add or change a View file?

4.12.1 Purpose

Use this procedure to add or update a viewConfig file.

The ArtifactAdmin label is used to indicate that an intent type has been installed using Artifacts. For these intent types, if you want to make changes to views, Nokia recommends creating a new view instead of modifying views. This will allow you to export the views you create and save them to be added into any future versions of the intent type created.

ViewConfig Builder

The ViewConfig Builder form provides a UI for configuring a viewConfig file. You can customize how each attribute in the intent type YANG appears for users when they create intents, for example, updating the title of an attribute, how it is represented visually, and whether helper text or default values are presented. You can also update the behavior, such as setting the attribute to read only.

Attributes can also be hidden or deleted from the intent creation form:

• hidden: the attribute is invisible in the intent creation form but is included in the payload when the form is submitted.

For example, if you want to set a read-only value for an attribute that is different from the YANG file value, you can set a default value and hide the attribute. The updated value is submitted along with the form when an intent is created but is not shown to the user creating the intent.

• deleted: the attribute is invisible in the intent creation form and is not included in the payload.

For example, you can delete a value if you want to always use the YANG default value.

Release 25.8
September 2025
Issue 1

© 2025 Nokia.
Use subject to Terms available at: www.nokia.com/terms
3HE-21460-AAAB-TQZZA

69

Intent types
Intent type procedures
How do I add or change a View file?

NSP

4.12.2 Steps

1

From **Network Intents**, **Intent Types**, double click on an intent type. From the page that displays, click **Manage Views**.

The Manage Views form opens.

2

To add a View file:

1. In the Manage Views form, click **Views ＋**.

2. In the Add View form that opens, enter the name of the view and click **Add**.

    A *name*.viewConfig file is created in the list at the left of the page. A *name*.schemaForm file is automatically created.

3. Click on the viewConfig file in the list and click **Save Selected File**.

4. If the intent type includes an RPC action, repeat this step to add a View file called rpc_*schema*.viewConfig to provide user interface information for the RPC action.

    See the Network Intents tutorial on the Network Developer Portal for more information if needed.

Proceed to Step 3 to customize the viewConfig file.

3

Open the ViewConfig Builder form:

In the Manage Views form, select a viewConfig file and click ⋮ (Table row actions), **Build ViewConfig**.

The ViewConfig Builder form opens, displaying the attributes in the viewConfig file. Figure 4-2, "Example ViewConfig Builder form" (p. 71) shows an example.

Intent types
Intent type procedures
How do I add or change a View file?

NSP

Figure 4-2   Example ViewConfig Builder form

4

To hide or delete one or more attributes:

a. To hide or delete a single attribute, select the attribute and click ⋮ (Table row actions), **Hide** or **Delete**.

b. To hide or delete multiple attributes, click the check boxes to select two or more attributes. At the top of the attributes panel, click ⋮, **Hide** or **Delete**.

5

To show or undelete one or more attributes:

a. Select a single attribute and click ⋮ (Table row actions), **Hide** or **Delete**.

b. Click the check boxes to select two or more attributes. At the top of the attributes panel, click ⋮, **Show** or **Undelete**.

6

To make other adjustments to an attribute: .

1. Select the attribute and click ⋮ (Table row actions), **Adjust attribute**.

2. In the form that opens, configure the attribute parameters.

3. Click **Close**.

Figure 4-3, "Example Adjust Attribute form" (p. 72) shows an example.

Intent types
Intent type procedures
How do I add or change a View file?

NSP

Figure 4-3   Example Adjust Attribute form

7 ───────────────────────────────────────

To save your changes to the viewConfig file, click **Save ViewConfig**. A confirmation dialog appears if changes to the viewConfig file may impact another NSP function. Click **Save** to confirm.

8 ───────────────────────────────────────

Click **Close** to return to the Manage Views form.

9 ───────────────────────────────────────

To attach a viewConfig file as a child of another:

1. Select a viewConfig file and click ⋮ (Table row actions), **Attach to parent**.

2. In the form that opens, click in the **Parent ViewConfig** field, select a viewConfig file as the parent, and click **Select**.

3. Click **Add**.

The child viewConfig and its schema form appear as a child in the list, below the parent viewConfig file.

Click 🗐 to display the Change log. The change log shows the attributes that have been augmented and indicates which viewConfig implements the change.

Intent types
Intent type procedures
How do I delete an intent type?

NSP

10 ⎯⎯⎯⎯⎯⎯⎯⎯⎯⎯⎯⎯⎯⎯⎯⎯⎯⎯⎯⎯⎯⎯⎯⎯⎯⎯⎯⎯⎯⎯⎯⎯⎯⎯

To detach a child viewConfig file from its parent, select the child viewConfig file and click ⋮ (Table row actions), **Detach from parent**.

11 ⎯⎯⎯⎯⎯⎯⎯⎯⎯⎯⎯⎯⎯⎯⎯⎯⎯⎯⎯⎯⎯⎯⎯⎯⎯⎯⎯⎯⎯⎯⎯⎯⎯⎯

Click **Close**.

E<small>ND OF STEPS</small> ⎯⎯⎯⎯⎯⎯⎯⎯⎯⎯⎯⎯⎯⎯⎯⎯⎯⎯⎯⎯⎯⎯⎯⎯⎯⎯⎯⎯⎯⎯

4.13 How do I delete an intent type?

4.13.1 Steps

1 ⎯⎯⎯⎯⎯⎯⎯⎯⎯⎯⎯⎯⎯⎯⎯⎯⎯⎯⎯⎯⎯⎯⎯⎯⎯⎯⎯⎯⎯⎯⎯⎯⎯⎯

Open **Network Intents**, **Intent Types**.

2 ⎯⎯⎯⎯⎯⎯⎯⎯⎯⎯⎯⎯⎯⎯⎯⎯⎯⎯⎯⎯⎯⎯⎯⎯⎯⎯⎯⎯⎯⎯⎯⎯⎯⎯

Select an intent type.

3 ⎯⎯⎯⎯⎯⎯⎯⎯⎯⎯⎯⎯⎯⎯⎯⎯⎯⎯⎯⎯⎯⎯⎯⎯⎯⎯⎯⎯⎯⎯⎯⎯⎯⎯

Click ⋮ (Table row actions), **Delete**.

4 ⎯⎯⎯⎯⎯⎯⎯⎯⎯⎯⎯⎯⎯⎯⎯⎯⎯⎯⎯⎯⎯⎯⎯⎯⎯⎯⎯⎯⎯⎯⎯⎯⎯⎯

In the confirmation dialog box, click **Delete**.

E<small>ND OF STEPS</small> ⎯⎯⎯⎯⎯⎯⎯⎯⎯⎯⎯⎯⎯⎯⎯⎯⎯⎯⎯⎯⎯⎯⎯⎯⎯⎯⎯⎯⎯⎯

4.14 How do I change the state of an intent type?

4.14.1 Purpose

Use this procedure to update the deployment status of an intent; see 5.2 "What is a network state?" (p. 77).

4.14.2 Steps

1 ⎯⎯⎯⎯⎯⎯⎯⎯⎯⎯⎯⎯⎯⎯⎯⎯⎯⎯⎯⎯⎯⎯⎯⎯⎯⎯⎯⎯⎯⎯⎯⎯⎯⎯

Open **Network Intents**, **Intent Types**.

2 ⎯⎯⎯⎯⎯⎯⎯⎯⎯⎯⎯⎯⎯⎯⎯⎯⎯⎯⎯⎯⎯⎯⎯⎯⎯⎯⎯⎯⎯⎯⎯⎯⎯⎯

Click on an intent type and click ⋮ (Table row actions), **Edit**.

Release 25.8
September 2025
Issue 1

© 2025 Nokia.
Use subject to Terms available at: www.nokia.com/terms
3HE-21460-AAAB-TQZZA

73

Intent types
Intent type procedures
How do I create a new version of an intent type?

NSP

3 ───────────────────────────────────────

In the **General** panel, choose the required state from the **Lifecycle State** drop-down list.

4 ───────────────────────────────────────

Click **Update** to change the state.

END OF STEPS ───────────────────────────

4.15 How do I create a new version of an intent type?

4.15.1 Purpose

You can create a new version starting from any version of an intent type. For example, if there are three versions, you can create a new version of version 2. The new version is numbered 4, but it does not include any updates made in version 3.

The new version number is auto-assigned and cannot be changed.

4.15.2 Steps

1 ───────────────────────────────────────

Open **Network Intents**, **Intent Types**.

2 ───────────────────────────────────────

Select an intent type.

3 ───────────────────────────────────────

Click ⋮ (Table row actions), **Create new version**.

The new version is created in draft state.

4 ───────────────────────────────────────

Configure migration and rollback functions as needed.

1. Select the new version from the intent types list and click ⋮ (Table row actions), **Edit**.
2. Add the migration and rollback functions to the Script.

5 ───────────────────────────────────────

Configure migration paths.

1. In the General panel, click ✛**Add** in the Migration pane.
2. In the form that opens, configure the parameters and click **Add**.
3. Repeat steps 1 and 2 to configure additional migration paths as needed.

END OF STEPS ───────────────────────────

Intent types
Intent type procedures
How do I export an intent type or its view files?

NSP

4.16　How do I export an intent type or its view files?

4.16.1　Purpose

Exporting an intent type exports all components other than the view files. View files can be exported separately. The exported files can be repackaged together outside NSP; see the Network Intents tutorial on the Network Developer Portal.

Exported view files can also be reused and customized in another intent type with the same override properties.

4.16.2　Steps

1

Open **Network Intents**, **Intent Types**.

2

Select an intent type.

3

Perform the exports as needed:

a. Click ⋮ (Table row actions), **Export Intent Type**.

b. Click ⋮ (Table row actions), **Export Views**.

The files are exported in zip format.

END OF STEPS

4.17　How do I configure user access to an intent type?

4.17.1　Purpose

Use this procedure to grant or restrict access to an intent type for a user group. Before you can configure user access, user groups must be created; see the *NSP System Administrator Guide*.

This procedure requires an administrator role.

4.17.2　Steps

1

Log in to the NSP as an administrator.

2

Open **Network Intents**, **Intent Types**.

Release 25.8
September 2025
Issue 1

© 2025 Nokia.
Use subject to Terms available at: www.nokia.com/terms
3HE-21460-AAAB-TQZZA

75

Intent types
Intent type procedures
How do I configure user access to an intent type?

NSP

3 ⎯⎯⎯⎯⎯⎯⎯⎯⎯⎯⎯⎯⎯⎯⎯⎯⎯⎯⎯⎯⎯⎯⎯⎯⎯⎯⎯⎯⎯⎯⎯⎯⎯⎯⎯⎯⎯

Select one or more intent types.

Press and hold the Shift key to select multiple intent types.

4 ⎯⎯⎯⎯⎯⎯⎯⎯⎯⎯⎯⎯⎯⎯⎯⎯⎯⎯⎯⎯⎯⎯⎯⎯⎯⎯⎯⎯⎯⎯⎯⎯⎯⎯⎯⎯⎯

Open the User Access form:

- If you have selected one intent type, click ⋮ (Table row actions), **Configure user access**

- If you have selected multiple intent types, click **User Access** 🔓 at the top right of the view.

5 ⎯⎯⎯⎯⎯⎯⎯⎯⎯⎯⎯⎯⎯⎯⎯⎯⎯⎯⎯⎯⎯⎯⎯⎯⎯⎯⎯⎯⎯⎯⎯⎯⎯⎯⎯⎯⎯

In the **Configure User Access** form, configure access to the selected intent types:

1. Choose an action from the drop-down list at the top right of the form.

2. If you chose **Grant access to all user groups**, choose **Full access** or **No access** from the drop-down lists for each group in the panel at the right.

6 ⎯⎯⎯⎯⎯⎯⎯⎯⎯⎯⎯⎯⎯⎯⎯⎯⎯⎯⎯⎯⎯⎯⎯⎯⎯⎯⎯⎯⎯⎯⎯⎯⎯⎯⎯⎯⎯

Click **Save**. The user access is updated.

Eɴᴅ ᴏғ sᴛᴇᴘs ⎯⎯⎯⎯⎯⎯⎯⎯⎯⎯⎯⎯⎯⎯⎯⎯⎯⎯⎯⎯⎯⎯⎯⎯⎯⎯⎯⎯⎯⎯⎯

5 Intents

Overview

5.1 What is an intent?

5.1.1 Network intents

An intent is an instance of an intent type. The intent provides inputs to the intent type and executes the configuration.

For example, you can create an intent type for creation of a VPRN service. The intent type provides the parameters to set and the script to perform the configuration. When you are ready to create the VPRN, you create an intent to specify the parameter values and implement the service creation.

5.2 What is a network state?

5.2.1 Network state

The network state of an intent shows its deployment status to the network.

NSP offers several options for the network state of an intent. The intent type script declares the recommended implementation of the state. Synchronizing the intent performs the implementation.

When an intent is audited, the Audit operation compares the configuration on the target to the network state.

See the Network Intents tutorial on the Network Developer Portal for information about how network states are programmed, and how they are managed by the API.

The following table provides an example of the way the network states are recommended to be used, for an intent type that creates services.

Network State	Description based on a service creation example
Saved	The intent is created and the service entity name will be reserved. The user can edit the intent to resume creation when needed.
Planned	The service resources are reserved.
Suspended	The intent must be activated before it can be deployed.
Activate	Remove suspension and send the configuration to the network.
Deployed	The desired configuration is sent down to the network.
Not Present	The service is removed from the network, but its resources remain reserved.

5.3 What is an intent audit?

5.3.1 Identify misalignments

Perform an intent audit to identify any mismatch between the intended configuration and network state and the actual configuration in the network.

If a mismatch exists, the **Aligned** field is updated to **Misaligned**.

| **i** | **Note:** You can also choose ⋮ (Table row actions), **Mark as misaligned** to manually set the alignment status to Misaligned.

If a misalignment is found, the audit report form provides details. Misalignments are presented in the following categories:

- misaligned attributes: the value of an attribute in the network (actual value) is different from the value in the intent (expected value)
- misaligned objects: an object that is configured in the intent is missing in the network
- undesired objects: an object that is not configured in the intent is present in the network

You have several options to resolve misalignments:

- Synchronize: send the intent configuration to the misaligned target, updating the configuration to match the intent.

  If you have approved misalignments, the approved misalignments will not be changed when you synchronize.
- Reconcile: retrieve the target configuration and update the intent to match the configuration

5.4 What is the difference between synchronize and reconcile?

5.4.1 Source of truth

For the synchronize operation, the intent is the source of truth. Misalignments are corrected by updating the network to match the NSP.

If approved misalignments are present, the synchronize operation skips them. For example, if you have two misalignments and have approved one, the synchronize operation will only update the misalignment you have not approved.

For the reconcile operation, the network is the source of truth. The intent is updated in the NSP to match the network.

5.5 What are approved misalignments?

5.5.1 Partial synchronize

Approved misalignments are differences between the current configuration in the network and the intended configuration in the NSP, which the user has allowed. Approved misalignments are not flagged by a future audit or overwritten when the intent is synchronized.

You can approve changes and remove approvals from the audit report for an intent; see 5.10 "How do I approve misalignments?" (p. 82).

You can view the list of approved misalignments for all intents and remove approvals from the Network Intents, Approved Misalignments view; see 5.11 "How do I modify approved misalignments?" (p. 83).

Release 25.8
September 2025
Issue 1

© 2025 Nokia.
Use subject to Terms available at: www.nokia.com/terms

3HE-21460-AAAB-TQZZA

79

Procedures

5.6 How do I create an intent?

5.6.1 Steps

1 ───────────────────────────────────

Open **Network Intents**, **Intent Types**.

Open the Create Intent form:

a. Select an intent type and click ⋮ (Table row actions), **Create Intent**.

b. Double click on an intent type to open **Network Intents**, **Intent Type** *name_version*, **Intents**, and click ＋**Intent**.

2 ───────────────────────────────────

Choose the network state.

3 ───────────────────────────────────

Configure the parameters.

4 ───────────────────────────────────

Click the **Synchronize** check box if needed to synchronize the configuration with the network as soon as the intent is created.

5 ───────────────────────────────────

Click **Create**.

6 ───────────────────────────────────

Double click on the intent type to view the list of intents. The Aligned field shows that the intent parameters are aligned with the network: the intent has executed successfully.

E<small>ND OF STEPS</small> ───────────────────────────────────

5.7 How do I view intents?

5.7.1 Viewing intents for an intent type

From **Network Intents**, **Intent Types**, double-click on an intent type to view the list of intents in the network for the intent type. Double-click on an intent in the list to open an information page showing intent details and dependencies.

The intent details page shows the intent state (active or suspended), the intent type, the intent version, timestamps for the last audit and synchronization, alignment state, and reason for misalignment, if applicable.

The **Details** tab in the center panel shows the parameter values.

The **Intent Dependency** tab shows dependency relationships:

• Direct Dependents: intents that depend on this intent

• Direct Dependencies: intents that this intent depends on

5.8 How do I modify an intent?

 i **Note:** Target components can't be edited. To deploy an intent type to a different target, create another intent.

5.8.1 Steps

1

Open **Network Intents**, **Intent Types**.

2

Double-click on an intent type to open **Network Intents**, **Intent Type** *name_version*, **Intents**.

3

Select an intent and click ⋮ (Table row actions), **Edit**.

4

In the form that opens, update the parameters.

The **Synchronize** check box is checked by default. Uncheck this check box if you do not want the intent to synchronize after you edit it.

5

Click **Update**.

Eɴᴅ ᴏꜰ sᴛᴇᴘs

5.9 How do I audit an intent?

5.9.1 Steps

1

Open **Network Intents**, **Intent Types**.

2

Double-click on an intent type to open **Network Intents**, **Intent Type** *name_version*, **Intents**.

3

Select an intent and choose ⋮ (Table row actions) **Audit**.

4

When the audit completes, the alignment status is updated in the Aligned column. If the intent is misaligned, the Audit Report form opens, showing the misalignments.

Eɴᴅ ᴏꜰ ꜱᴛᴇᴘꜱ

5.10   How do I approve misalignments?

5.10.1 Steps

1

Open **Network Intents**, **Intent Types**.

2

Double-click on an intent type to open **Network Intents**, **Intent Type** *name_version*, **Intents**.

3

Double-click on an intent in misaligned status and choose ⋮ (Table row actions) **View Last Audit Report**.

The Audit Report form opens.

4

The Audit Report form displays the misalignments by category. An indicator appears beside the category name with the number of misalignments present in the category, for example, **Misaligned Attributes [2]**.

Click the category name to view a table of misalignments in the category.

5

To approve a change, click the check boxes in the row for the change you want to approve and click **Approve Selected**.

The change is added to the **Approved Misalignments** panel and the number indicator is updated.

6

Open other category panels and approve additional changes as needed.

7

View the list in the **Approved Misalignments** panel.

If you don't want to approve any change on the list, uncheck the check box and click **Remove Approval**.

8

Click **Apply**. The approved misalignments are added to the **Network Intents**, **Approved Changes** view.

The intent is audited automatically.

END OF STEPS

5.11 How do I modify approved misalignments?

5.11.1 Purpose

You can delete approved misalignments for any intent from the **Network Intents**, **Approved Changes** view. Deleting an approved misalignment removes the exception for the misalignment. That is, the change is declared misaligned when the intent is audited, and overwritten when the intent is synchronized.

To add approved misalignments, perform .

5.11.2 Steps

1

Open **Network Intents**, **Approved Changes**.

The view shows a list of intents with approved misalignments, and the number of approved misalignments for each.

2

Select an intent and click 🖺(Approved Misalignments) to view the list of approved misalignments for the intent.

3

In the form that opens, check the check box for the misalignment you want to delete and click **Remove Approval**

The approved misalignments are added to the **Network Intents**, **Approved Changes** view.

When you audit the intent again, the change will be marked as a misalignment.

END OF STEPS

5.12 How do I migrate an intent?

5.12.1 Purpose

Use this procedure to migrate a single intent. To perform a mass migration, create a policy.

If you migrate an intent with approved misalignments, the approved misalignments will be handled based on the migration settings in the parent intent type.

Intents
Procedures
How do I execute an action from an intent?

NSP

5.12.2 Steps

1 ────────────────────────────

Open **Network Intents**, **Intent Types**.

2 ────────────────────────────

Double-click on the intent type version that includes the intent you want to migrate.

3 ────────────────────────────

From the list of intents that appears, double-click on the intent you want to migrate.

4 ────────────────────────────

Click **Migrate Intent**.

5 ────────────────────────────

In the form that opens, choose the version to migrate the intent to.

6 ────────────────────────────

Click **Ok**.

End of steps ────────────────────

5.13   How do I execute an action from an intent?

5.13.1 Purpose

If your intent includes RPC actions, you can run the actions without auditing or synchronizing the intent.

5.13.2 Steps

1 ────────────────────────────

Open **Network Intents**, **Intent Types**.

2 ────────────────────────────

Double-click on the intent type that includes the RPC action you want to execute.

3 ────────────────────────────

From the list of intents that appears, double-click on an intent.

The intent details page opens with the list of available actions in a panel on the right of the screen.

Intents
Procedures
How do I execute an action from an intent?

NSP

4

Click on the action to execute.

a. If the action has no input parameters, it executes immediately.

b. If the action has input parameters, a form opens. Configure the parameters and click **Ok**.

E<small>ND OF STEPS</small>

Release 25.8
September 2025
Issue 1

© 2025 Nokia.
Use subject to Terms available at: www.nokia.com/terms
3HE-21460-AAAB-TQZZA

85

Intents
Procedures
How do I execute an action from an intent?

NSP

6 Intent Policies

6.1 What is a policy?

6.1.1 Mass operations for intents

Policies trigger and execute intents. You can create a policy to perform a mass operation on intents that fit specified criteria; for example, synchronize all intents that have not been synchronized in the last 10 days.

You can trigger policies manually, or configure them to execute automatically on a schedule. Scheduled policies help you to automatically manage your intents network-wide.

6.1.2 Editing and deleting policies

You can edit or delete a policy from the Table row actions menu; see "How do I navigate a list?" in the *NSP User Guide*.

6.2 How do I create an intent policy?

6.2.1 Steps

1

Open **Network Intents**, **Policies**.

2

Click ＋**Policy**.

3

In the **Create Policy** form that opens, enter a name for the policy.

4

In the Action panel, select the operation for the policy to perform from the drop-down.

- **Synchronize**:

  Synchronize the intents.

- **Audit**:

  Audit the intents.

- **Mark as Misaligned**:

  Mark target intents as misaligned.

- **Modify and Synchronize**:

  Modify and synchronize intents of a specified intent type and intent type version.

- **Migrate**:

Release 25.8
September 2025
Issue 1

© 2025 Nokia.
Use subject to Terms available at: www.nokia.com/terms

3HE-21460-AAAB-TQZZA

87

Migrate target intents to a new version of their intent types.

5 _____

In the Criteria table, configure the operation criteria:

1. Click ＋**Add**.

2. Enter an ID for the criterion.

3. Combine filters from the **Filter On** drop-down list with operators and values to create filters, for example `Synch Timestamp older-than 3 days`.

4. Add additional criteria as needed.

If the Action is Modify and Synchronize or Migrate, the criteria must include an Intent Type and Intent Type Version.

6 _____

If the Action you configured is Modify and Synchronize, configure the **Arguments** panel.

7 _____

In the Schedule panel, configure the parameters to create a schedule for the policy.

8 _____

Click **Create**. The policy is added to the list.

E<small>ND OF</small> <small>STEPS</small> _____

6.3 How do I trigger a policy?

6.3.1 Steps

1 _____

Open **Network Intents**, **Policies**.

2 _____

Select a policy.

3 _____

Click ⋮ (Table row actions), **Trigger**.

4 _____

In the confirmation dialog, click **Confirm**.

The policy is triggered.

E<small>ND OF</small> <small>STEPS</small> _____

6.4   What is a policy action?

6.4.1 Policy action

A policy action is an instance of a triggered policy.

Choose **Policy Actions** from the drop-down list at the top left of the screen to open the **Network Intents**, **Policy Actions** view.

The Policy Actions view shows the list of policy actions in Network Intents. The list shows the total job count, error and failed jobs along with status for all of the policies.

Double-click on an action to open a policy action details page, showing the job count, error count, and the list of jobs and their status.

Release 25.8
September 2025
Issue 1

© 2025 Nokia.
Use subject to Terms available at: www.nokia.com/terms
3HE-21460-AAAB-TQZZA

89

7 Mediators

7.1 What is a mediator?

7.1.1 Mediator

Mediators serve as communication proxies between Network Intents and other systems, such as NSP components that manage network devices, or external controllers. The use of a mediator removes the need for Network Intents to provide authentication credentials to reach the desired endpoint.

The information on the Mediators views may be useful to administrators for troubleshooting communication issues to the underlying components.

7.2 What mediators are available?

7.2.1 Mediator types

NSP is shipped with mediators for model-driven mediation (the MDC mediator) and for classic management (the NFM-P mediator). Depending on how a networking device is managed, the appropriate mediator is used.

The NFM-P mediator can be used to provide helper functions to the SAM-O XML API and to other applications that use intents, such as Service Management and Inventory Configuration Management. To allow this, the XML API username and password attributes in the helm chart values file (values.yaml) must match the NSP credentials of an active user with the OSS Management role.

Multiple NFM-P mediators may be used for multi-NFM-P deployment scenarios. These mediators, in addition to simplifying communication, provide a high-level inventory of the devices they manage. This allows NSP to provide information about which mediator is in use for communication with a specific device.

In addition to the NFM-P and MDC mediators, there is a common NSP mediator that can communicate with any NSP component, which allows for advanced use-cases like path control or telemetry collection controlled by an intent type.

Generic mediators can be configured to integrate with external controllers; see the *NSP System Administrator Guide* for configuration information. A generic mediator can be used with any REST or RESTCONF endpoint.

7.3 How do I edit a mediator?

7.3.1 Steps

1

Open **Network Intents**, **Mediators**.

2

Select a mediator in the left panel.

3

In the Attributes panel of the Info tab, configure the parameters as needed.

Parameters with an asterisk(*) are required.

4

Click **Update**.

END OF STEPS

7.4 How do I delete a mediator?

7.4.1 Steps

1

Open **Network Intents**, **Mediators**.

2

Select a mediator in the left panel.

3

To remove the mediator from the NSP GUI, click **Delete**.

4

To remove the mediator from the NSP system, perform a helm uninstallation; see "How do I configure a generic mediator?" in the *NSP System Administrator Guide*.

> **i** **Note:** If a mediator is deleted in error, restart the mediator pod to re-register it with the NSP system.

END OF STEPS

Part V: Workflows

Overview

Purpose

Describes the use of workflows for operators.

Contents

8 Workflows

8.1 What are Workflows?

8.1.1 Workflows

A workflow represents the process to do a job, for example, configure a service.

Using workflows in NSP, you can create automated procedures and closed loop automation using Nokia NSP or 3rd party APIs.

Some example use cases:

- Node software upgrades
- Service activation tests
- Service fulfillment invoking pre and post deployment workflows

- Customizable policy logic for other applications
- Mass migration of services from one tunnel type to another
- One place scheduling for all NSP REST APIs and node CLI
- Centralized orchestration for Nokia NSP and 3rd party vendor APIs

8.1.2 Requirements

NSP uses Mistral DSL v2 which is based on YAML. YAML defines expressions in workflow and action definitions. The formatting of the YAML must comply with Mistral DSL v2 specifications. For more information about Mistral DSL v2, the workflow hierarchy and associated attributes, refer to the Mistral DSL v2 specification online.

Certain functions within NSP workflows use Python, such as triggers and the nsp.python system action. NSP uses Python version 3.11. Any custom actions created by a user must be compliant with this version.

8.1.3 Developer information

The Network Automation Guide provides an overview for operators.

For detailed information about the following topics for developers, see the Workflows tutorial on the Network Developer Portal.

- Using Workflow APIs
- Creating workflows, including working with schema forms
- Working with Jinja2 templates
- Working with environments

- Working with ad-hoc actions
- Working with Kafka triggers
- Workflow signatures

Release 25.8
September 2025
Issue 1

© 2025 Nokia.
Use subject to Terms available at: www.nokia.com/terms
3HE-21460-AAAB-TQZZA

95

8.1.4 API support

NSP Workflow functions are available for OSS using programmable APIs. For general information about developer support, visit the Network Developer Portal

For specific documentation about APIs for Workflows, click on API Reference in the Workflow Manager row.

8.1.5 Developer mode

If developer mode is disabled in the NSP, users cannot create, import, modify, or delete workflows, actions, or Jinja2 templates.

See the *NSP Installation and Upgrade Guide* for more information.

8.1.6 Access control

User groups are assigned access to menus, network resources, and Analytics resources by assigning roles in the NSP. Action permissions are assigned to roles.

> **i** **Note:** Disabling developer mode overrides access control settings. If developer mode is disabled, restricted functions are restricted for all users.

The following table describes scopes that are specific to workflows.

Scope	Available operations
Import	Import all available file types
Write Workflow Artifacts	Create and Edit workflows
Publish Workflow Artifacts	Publish workflows and related artifacts
Manage Executions	Pause, resume, cancel and delete workflow and action executions
Manage Environments	Perform actions in the Environments view
Manage Triggers	Manage Kafka triggers and workflow schedules
Debug	Debug using Flow views, the YAQL evaluator, or code definition views
Execute Workflow Artifacts	Execute workflows and actions. Pause, resume, or cancel workflow executions created by the same user.

The scopes are combined to create the following roles:

- Developer:
  - Import
  - Write Workflow Artifacts
  - Publish Workflow Artifacts

- Troubleshooter:
  - Execute Workflow Artifacts
  - Manage Executions
  - Debug

- Operator:
  - Manage Executions
  - Manage Environments
  - Manage Triggers
  - Execute Workflow Artifacts
- Network Engineer
  - All scopes

Depending on your access settings, some of these pages or operations may not be available to you. See the *NSP System Administrator Guide* for more information.

9   Using workflows

9.1   What does a workflow look like?

9.1.1 Workflow structure

The following example workflow, `createNSPUser`, is available in the Create Workflow form for use as a workflow template. For detailed information about workflow design, see the Workflows tutorial on the Network Developer Portal.

The YAML definition of the workflow includes the following:

- General information
  - workflow name and description
  - tags

    Tags are used to filter and group objects.

    If a workflow is compliant with a set of format requirements, such as Service Fulfillment or Kafka triggers, a tag must be used to indicate what the workflow is configured for. For example, for a Kafka trigger to launch a workflow, the workflow must have a KafkaTrigger tag.
  - workflow metadata

    Metadata includes the workflow version number and dependencies
- input

  Input is the input parameters of a workflow or task, if required. The value of each parameter is a JSON-compliant type (number, string, etc), a dictionary, or a list.

  It can also be an expression to retrieve a value from a task context, or any of the mentioned types containing inline expressions.
- output (optional, not shown in the sample)

  Output is a data structure containing expressions that define workflow output.
- tasks

  A task is a logical step that can perform a Mistral action, for example, performing a REST request.

  Tasks can call other workflows. A task can also be iterated over a list of items performing bulk operations.

  Tasks can be chained together based on success or failure of the previous task, or on completion of the previous task (always run). On-success and on-error chaining run subsequent tasks at the same time. On-complete chaining runs subsequent tasks in sequence.
  - actions

    The actions performed by the task take input data and produce output data, as defined by the action. NSP is packaged with a set of system actions. For example, std.http sends an HTTP request.

    You can create ad-hoc actions to customize system actions. See What is an action? for more information.

Example workflow

```
version: '2.0'

createNSPUser:

  description: creates an NSP user with the given Group Name. Creates a
Group if the Group does not exist.


  type: direct


  tags:

    - NSP USER


  workflow_meta:

    title:   Create NSP user

    author:  WFM

    version: '2.0.0'

    dependencies:

      platform:

        nspOS: ['23.11']

  input:

    - username

    - password

    # password should contain at least one capital letter and one special
character

    - group

    - passwordUpdateRequired: false

    - accountEnabled: true

  tasks:

    getUserGroup:

      action: nsp.https

      input:

        url: 'https:
//rest-gateway/access-control-api/rest/api/v3/nspuac/group/<% $.group %>'

        method: GET

        contentType: 'application/json'
```

```
                      accept: 'application/json'
              publish:
                status:  <% task().result.status %>
              publish-on-error:
                status: <% task().result.status %>
              on-error:
                - createUserGroup
              on-success:
                - createUser
          createUserGroup:
            action: nsp.https
            input:
              url: 'https:
    //rest-gateway/access-control-api/rest/api/v3/nspuac/group/'
              method: POST
              body:
                userGroupName: <% $.username %>
                roles: []
              publish:
                status: <% task().result.status %>
              on-success:
                - createUser


        createUser:
          action: nsp.https
          input:
            url: 'https://<% locate_nsp()
    %>/nsp-keycloak-api/rest/api/v2/user-management/realm/Nokia/users/'
              method: POST
              body:
                userName: <% $.username %>
                password: <% $.password %>
                group: <% $.group %>
```

```
        accountEnabled: <% $.accountEnabled %>

        passwordUpdateRequired: <% $.passwordUpdateRequired %>

    publish:

      result: <% task().result.status %>
```

9.2 What are the components of a workflow?

9.2.1 Workflows in the NSP UI

The landing page for Workflows is the Dashboard. Dashlets in the Dashboard provide information about the status of recent, running, failed, and scheduled workflow executions. You can use the view selector to navigate to workflow-related views for the NSP. For example, the Workflow Executions view displays a list of executions for all workflows.

Choose Workflows in the view selector to display the list of workflows created in or imported to NSP. Double-click on a workflow to navigate to a set of views specific to the selected workflow:

- Info
- Executions
- Action Executions Waiting On User Input
- Definition
- Flow
- Input Form
- Schedules
- Kafka Triggers

9.2.2 Info

The Info page provides a dashboard for the selected workflow, displaying execution status, workflow metadata, and, if available, a workflow readme.

9.2.3 Executions

In the executions list, you can see the list of executions of the workflow that have been run since the last cleanup, including their status (Idle, Running, Success, Error, or Canceled), their run time, and how they were triggered.

> **i** **Note:** The executions list on the Dashboard does not include subworkflows.

Double-click on an execution to drill down for further information about tasks, actions executed, input/output, or the execution flow diagram. From the flow diagram, you can debug action executions, or evaluate YAQL expressions.

9.2.4 Action Executions Waiting On User Input

If an action in the workflow requires user input during execution, such as a confirmation message, the action enters Waiting On User Input status. The Action Executions Waiting On User Input view shows the list of action executions executed by the selected workflow, filtered by Waiting On User Input status.

To open the user input form and complete the execution, choose the action execution and click ⋮ (Table row actions), **Update**.

9.2.5 Definition

The definition view shows the YAML script of the workflow. See 9.1 "What does a workflow look like?" (p. 99) for more information about workflow structure. For detailed information about workflow design, see the Workflows tutorial on the Network Developer Portal.

9.2.6 Flow

A Flow diagram is a graphical representation of a workflow or workflow execution. Tasks are shown as boxes, and dependencies as lines between the boxes.

The Flow diagram of a workflow shows all of the tasks that could be run by the workflow and their dependencies. For example: on success of Task 1, perform Task 2; on failure of Task 1, perform Task 3. For this example, the dependency is indicated by a solid line, colored green for success or red for failure. If the dependency is conditional, the diagram shows a dotted green line. Hover your mouse over the word **Condition** to see the condition.

Tasks are all in Preview status because the diagram is showing a workflow, not a completed execution.

Choose a task to see the path to the task; for example, the success of two previous tasks. With a task selected, click **Details** ⓘon the right of the page to open the Details panel and view information about the task, including the action performed, the task definition, and the information published by the action, if any. See 9.13 "What is a flow?" (p. 114) for more details.

9.2.7 Input form

The input form defines the UI for executing the workflow, that is, the Input and Parameters panel of the Create Execution form.

The required input is configured in the workflow's YAML definition; see 9.1 "What does a workflow look like?" (p. 99). For example, the workflow may require a username to be entered. The input form defines the appearance of the fields in the Input and Parameters panel. For example, the input form defines the width of the username field, and whether the field for the next parameter appears on a new line in the form.

If the developer has included standard input attributes in the workflow script, the input form can include execution mode options; see the Workflows tutorial on the Network Developer Portal. The following options can be included:

• Dry Run: execute the workflow without pushing any information to the network.

• Compare: execute the workflow and provide a summary of what has changed, or, if Dry Run is also enabled, what would be changed if you set Dry Run to false

• Force: execute the workflow with predefined user input values.

See 9.10 "How do I update an input form?" (p. 111) to update the input form.

9.2.8 Schedules

The Schedules view shows the list of configured schedules that execute the selected workflow, including information about remaining executions. You can create a schedule for the workflow from this view if needed.

9.2.9 Kafka triggers

The Kafka Triggers view shows the list of configured Kafka triggers that execute the selected workflow, including the Kafka topic and event that trigger execution. See 14.1 "What is a Kafka trigger?" (p. 143) for more details.

> **i** **Note:** You can delete a workflow from the Table row actions menu in the Workflows view; see "How do I navigate a list?" in the *NSP User Guide*. If there is a Kafka trigger associated with a workflow, the Kafka trigger must be removed before the workflow can be deleted.

9.3 How do I add a workflow?

9.3.1 Workflow creation options

You have several options for adding workflows to NSP.

• Create a workflow. This procedure provides general steps. See the Network Automation tutorial on the Network Developer Portal for detailed information.

• Import a workflow from your computer.

• Import a workflow from Git.

• Install a workflow using Artifacts; see 2.3 "How do I install an artifact bundle?" (p. 22).
  Note: if an installed workflow appears with a red signature icon, there is a problem with the workflow. Try downloading and installing again. Contact Nokia if the problem persists.

• Clone a workflow and edit the clone as needed.

> **i** **Note:** Here are some things to know when setting up workflows.
>
> • A workflow must pass validation before it can be saved in NSP. If you import a workflow that is not valid or clone a workflow and make changes that don't pass validation, you must update it to fix the errors before you can add it to the workflow list.
>
> • Workflows are imported in the DRAFT state. The state must be changed to PUBLISHED before the workflow is available to all users.

9.3.2 Workflow examples

Workflow examples are available in the creation form. You can choose the best example for your needs and use it as a guide, or delete it and insert your own YAML code.

The example workflows `cleanup_WFM` and `createNspUser` are available by default. Any workflow with the `example` tag will also be available in the creation form.

9.4　How do I create a workflow?

| **i** | **Attention:** Workflow names must only contain alphanumeric characters.

9.4.1 Steps

1

Open **Workflows**, **Workflows**.

2

Click ＋**Workflow**.

3

In the **Create Workflow** form that opens, enter your code in the YAML panel:

1. If necessary, click in the Select Workflow field to change the 9.3.2 "Workflow examples" (p. 104) provided.
2. Update the YAML as needed.

4

Click **Validate & Update Flow**. Resolve any validation errors that appear.

After the YAML has been validated, you can click on the **Flow** tab to see the Flow diagram if needed.

5

Click **Create**.

The workflow is created in Draft status.

END OF STEPS

9.5　How do I import workflow-related files from my computer?

9.5.1 Purpose

Use this procedure to import one or more files from your computer to the Workflows views.

You can import any supported file type from the views in the Main or Templates section of the view selector:

• **Workflows**

• **Actions**

• **Environments**

• **Jinja2**

Release 25.8
September 2025
Issue 1

© 2025 Nokia.
Use subject to Terms available at: www.nokia.com/terms
3HE-21460-AAAB-TQZZA

105

Using workflows
How do I import workflow-related files from my computer?

NSP

Multiple files of different types can be imported in the same operation.

The following file types can be imported:

*   packages (.zip)
*   workflows (.yaml)
    Workflow zip files can contain input files in JSON format. The JSON file should have the same name as the workflow, for example, workflow1.json.
    Workflows are imported in DRAFT status.
*   environment files (.env)
*   actions (.action)
*   Jinja2 templates (.jinja)

9.5.2 Steps

1

Open one of the views in the Main or Templates section of the view selector.

2

Open the Import form:

a. From **Workflows**, **Workflows**, click the **Import** drop-down list and choose **File System**.

b. From another view, click **Import**.

3

In the **Import** form, perform one of the following steps to select your file.

a. Navigate to a file or files on your computer and drag them to the panel at the top of the form.

b. Click in the panel at the top of the form to navigate to the file you want. Click **Open**.

The file names appear in the Files to Import panel.

4

Click **Import**.

The Import Results panel displays the status of the import of each file, with explanations for failed imports.

5

Click **Close**.

END OF STEPS

9.6 How do I export workflow-related files?

9.6.1 Purpose

Use this procedure to export a file to your computer.

Configured files can be exported:

*   workflows (.yaml or .zip)
*   environment files (.env)
*   ad-hoc actions (.action)
*   Jinja2 templates (.jinja)

You can also export the Flow diagram of a workflow or workflow execution as a .png file.

9.6.2 Steps

1

Open the view that includes the object you need to export.

2

Choose ⋮ (Table row actions), **Export** or ⋮ (Table row actions), **Export Zip**.

Depending on your browser settings, the file is downloaded or a confirmation dialog opens.

3

To export the Flow diagram of a workflow:

1. Open **Workflows**, **Workflows**.
2. Double-click on a workflow to open the Info page
3. Choose Flow from the drop-down list.
4. Click **Screenshot Flow** ⌂ to download the Flow diagram as a .png file.

4

To export the Flow diagram of a workflow execution:

1. Open **Workflows**, **Workflow Executions**.
2. Double-click on a workflow execution to open the Info page
3. Choose Flow from the drop-down list.
4. Click **Screenshot Flow** ⌂ to download the Flow diagram as a .png file.

5

Save the file as needed.

E<small>ND OF STEPS</small>

Release 25.8
September 2025
Issue 1

© 2025 Nokia.
Use subject to Terms available at: www.nokia.com/terms

3HE-21460-AAAB-TQZZA

107

Using workflows
How can I interact with the Nokia Git repository from NSP?

NSP

9.7 How can I interact with the Nokia Git repository from NSP?

9.7.1 Purpose

The Nokia nsp-workflow repository on GitHub hosts sample workflows to be used with NSP. Use these workflows to get to know workflows in NSP, or as blueprints for your own workflows. You can also check for updates to workflows you have imported, or open the Nokia nsp-workflow repository.

The default environment uses the Nokia repository on GitHub. You can update the default environment to point to your own Git if needed.

9.7.2 Prerequisite

To import from the Nokia GitHub repository, you must:

• Add the following URLs to the allowed hosts list on the NSP server:
  − *raw.githubusercontent.com*
  − *api.github.com*
  You can update the allowed hosts via postman, or using `curl` from the command line on the NSP server:

  ```
  # curl -kv --request POST https://VM
  IP/session-manager/api/v1/whitelist/allowedHosts --header
  'Content-Type: application/json' --data-raw '{ "host": "url"
  }' --header "Authorization: Bearer authorization token ↵
  ```

  where *VM IP* is the IP address of the NSP server, *url* is the URL you are adding, and *authorization token* is a bearer token requested from NSP.

• Add GitHub to the `cspWhitelist` setting in the NSP configuration file, `nsp-config.yml`, as shown in the following example:

  ```
  nspUISettings:
      cspWhitelist: "https://raw.githubusercontent.com:* https://api.
  github.com:*"
  ```

9.7.3 Steps

1 ────────────────────────────────

Open **Workflows**, **Workflows**.

2 ────────────────────────────────

Click the **Import** drop-down list and choose **GitHub** to open the **Import from Git** form.

3 ────────────────────────────────

To import a workflow from Git:

  1. In the **Import from Git** form, choose 🗑**Import Packages** from the panel at the left of the screen.

  2. Explore the list of workflows, or filter the list using the filter field.

     If a workflow has not been added to your NSP, the **Import** button is available. If updates are available for a workflow you have imported, the **Update** button is available.

3. Click ⬈(Open Repository) to view the workflow in the Nokia repository, or click **More actions···** for details.

4. Click **Import**. The workflow is imported and the **Import** button changes to an indicator that your copy of the workflow is up to date.

4 ───

To import updates to workflows you have saved:

1. In the **Import from Git** form, choose ⬇**Updates** from the panel at the left of the screen.

2. Click **Check for Updates** to refresh the list of available updates.

3. Explore the list of updates, or filter the list using the filter field.

4. Choose an update from the list and click **Update**, or click **Update All** at the top of the form.

    The updates are completed.

5 ───

To open the repository, in the **Import from Git** form, choose ⬈**Open Repository** from the panel at the left of the form.

The Nokia nsp-workflow repository opens in a new browser tab.

Eɴᴅ ᴏꜰ ꜱᴛᴇᴘꜱ ─────────────────────────

9.8　How do I clone a workflow?

9.8.1　Purpose

Use this procedure to copy a workflow, for example, if you need a new workflow that is similar to an existing one but with a different value for an attribute., or if you want to test a change to a workflow.

⬛ⁱ **Note:** You can clone a signed workflow, but the clone will not be signed.

9.8.2　Steps

1 ───

Open **Workflows**, **Workflows**.

2 ───

Choose a workflow and choose ⋮ (Table row actions), **Clone**.

3 ───

In the form that opens, enter a name for the cloned workflow and click **Clone**.

Release 25.8
September 2025
Issue 1

© 2025 Nokia.
Use subject to Terms available at: www.nokia.com/terms

3HE-21460-AAAB-TQZZA

109

4 _____

See 9.9 "How do I edit a workflow?" (p. 109) to edit the clone.

E<small>ND OF STEPS</small> _____

9.9 How do I edit a workflow?

9.9.1 Purpose

Use this procedure to modify a workflow.

When you put a workflow into DRAFT status to edit it, it is locked for your edits. Other users can't modify the workflow, or see the changes being made. When the workflow is PUBLISHED, it's no longer locked.

If any schedules or executions are running at the time you put the workflow into Draft status, they are completed according to the last published version of the workflow. That is, the executions are not affected by your changes.

Workflows that are imported from GitHub and signed workflows cannot be edited. You can clone these workflows and make changes but the cloned versions are not signed.

You must have write privileges for the workflow you want to modify.

9.9.2 Steps

1 _____

Open **Workflows**, **Workflows**.

2 _____

Open the Definition page of the workflow:

a. Click on a workflow and choose ⋮ (Table row actions), **Update definition**.

b. Double-click on a workflow and choose **Definition** from the **Info** drop-down list.

3 _____

Change the workflow state to Draft:

1. Click ↑↓**Modify state**.
2. In the form that opens, choose Draft and click Save.

4 _____

In the **YAML (Draft)** tab, make changes as needed.

You can click on the **Published** tab at any time to see the previous published YAML description.

5 _____

Click **Validate & Update Flow**.

If the validation fails, fix any problems and validate again.

6 ───────────────────────────────────

When the validation succeeds, click **Publish** to publish the edited workflow, or **Save Draft** to keep it in Draft status.

7 ───────────────────────────────────

When you complete your edits and publish the workflow, check to see if any schedules or Kafka triggers are affected by your changes. For example, if you added a parameter to a workflow, any schedules that run the workflow must be updated to provide input for the new parameter.

END OF STEPS ───────────────────────────

9.10 How do I update an input form?

9.10.1 Purpose

Use this procedure to configure the UI form for executing the workflow.

The required input is configured in the workflow's YAML definition; see 9.1 "What does a workflow look like?" (p. 99). For example, the workflow may require a username to be entered.

When the workflow is manually executed, the Create Execution form has two panels: General and Input and Parameters. The input form defines the appearance of the fields in the Input and Parameters panel. For example, the input form defines the width of the username field, and whether the field for the next parameter appears on a new line in the form. You can use the JSON schema to create form properties such as lists, choices, and fields with autocomplete.

i | **Tip:** You can add or change the form input of a workflow in either DRAFT or PUBLISHED status.

9.10.2 Steps

1 ───────────────────────────────────

Open **Workflows**, **Workflows**.

2 ───────────────────────────────────

Double-click on a workflow.

3 ───────────────────────────────────

Choose **Input Form** from the **Info** drop-down list.

4 ───────────────────────────────────

Choose a task from the following options.

a. To update the input form manually, enter JSON or YAML details in the editor and click **Update Input Form**.

Release 25.8
September 2025
Issue 1

© 2025 Nokia.
Use subject to Terms available at: www.nokia.com/terms

3HE-21460-AAAB-TQZZA

111

b. To auto-generate JSON, click **Auto Generate UI**. Click **Ok** to confirm.

The input form is auto-generated. The auto-generated form overwrites any existing form input.

c. To import JSON, click ⬇**Import JSON from Filesystem** at the top of the screen. Navigate to the file you want and click **Open**.

The JSON details appear in a JSON form. Click **Update Input Form**.

The input form is updated in the workflow.

END OF STEPS

9.11 How do I change the status of a workflow?

i **Tip:** When multiple workflows are selected, **Modify State** ↑↓and **Export All** become available at the top of the screen.

9.11.1 Steps

1

Open **Workflows**, **Workflows**.

2

Open the Update Workflow State form:

a. Click on a workflow and choose ⋮ (Table row actions), **Modify Status**.

b. From the **Definition** page of a workflow, click ↑↓**Modify state**.

3

Select the radio button for the new state.

4

Click **Save**.

END OF STEPS

9.12 How do I execute a workflow?

9.12.1 Purpose

Use this procedure to run a workflow manually. Executing a workflow creates a workflow execution.

9.12.2 Steps

1

Open the Create Execution form:

a. From **Workflows**, **Workflows**, choose a workflow and click ⋮ (Table row actions), **Execute** .

b. From **Workflows**, **Workflow Executions**, click ＋**Execution**.

2

In the **Create Execution** form, verify that the environment is correct.

Click in the Environment field to change the environment if needed.

3

Enter a description if necessary.

The description appears in the **Workflows**, **Workflow Executions** view when the execution is created.

4

Enable the **Kafka Notification** check box as needed to enable Kafka notifications.

If Kafka notifications are enabled, when the workflow enters a new state (for example, Running, Success, or Waiting) a Kafka event is created on the WFM topic with details.

5

From the **Input Format** drop-down list, choose the format and provide inputs.

Note: File inputs cannot exceed 1 MB in size.

a. **Direct Input**

b. **File Input**

Click ⬚ and choose the file to import. The Input Type is detected.

c. **URL Input**

Type or paste the URL and click Enter ↵.

The Input panel is populated.

6

Configure input parameters as needed.

7

Click **Execute** to execute the workflow.

8

If an action in the workflow requires user input during execution, such as a confirmation message, the action enters Waiting On User Input status.

1. Choose **Executions**, **Workflow Executions** from the drop-down list at the top left of the page and double-click the workflow execution you created.

2. Click **Action Executions Waiting On User Input**.

3. Choose the action execution and click ⋮ (Table row actions), **Update** . The user input form opens.

4. Configure the parameters.

The action execution status is updated and the workflow execution proceeds.

9

Return to the **Workflows**, **Executions**, **Workflow Executions** view to view the workflow execution status.

END OF STEPS

9.13 What is a flow?

9.13.1 Flow diagrams

A Flow diagram is a graphical representation of a workflow or workflow execution. Tasks are shown as boxes, and dependencies as lines between the boxes.

To open a Flow page, double-click on a workflow or workflow execution to open an information page. Choose Flow from the drop-down list, beside the workflow name.

When available, icons are shown on the task boxes to indicate the category of action the task performs:

- File Action ▯: The task performs a file operation, such as delete or create.
- Folder Action ▯: The task performs a folder operation, such as delete or create.
- Template Action ▯: The task implements a Jinja2 template.
- Message Action ▯: The task makes REST API calls or sends a message, including email or Kafka.
- Generic System Action ▯
- CLI Action ▯: The task uses CLI.
- User Action ▯: The task requires user input.
- Javascfript Action ▯: The task includes JavaScript.
- Python Action ▯: The task includes Python.
- Checksum Action ▯: The task returns the checksum value of a filename or path.
- Delay Action ▯: The task incorporates a delay, for example, "perform this action, then wait two s".
- Subworkflow ▯: The task calls another workflow.

Click **Screenshot Flow** ▯to download the Flow diagram as a .png file.

9.13.2 The Flow diagram of a workflow

The **Flow** diagram of a workflow shows all of the tasks that could be run by the workflow and their dependencies. For example: on success of Task 1, perform Task 2; on failure of Task 1, perform Task 3. For this example, the dependency is indicated by a solid line, colored green for success or

Using workflows
How do I configure user access to a workflow?

NSP

red for failure. If the dependency is conditional, the diagram shows a dotted green line. Hover your mouse over the word **Condition** to see the condition.

Tasks are all in Preview status because the diagram is showing a workflow, not a completed execution.

Choose a task to see the path to the task; for example, the success of two previous tasks. With a task selected, click **Details** ⓘon the right of the page to open the Details panel and view information about the task, including the action performed, the task definition, and the information published by the action, if any.

9.13.3 The Flow diagram of a workflow execution

The Flow diagram of a workflow execution shows the tasks that ran for the chosen execution, including their status.

For detailed information about a task, choose the task and click **Details** ⓘon the right of the page to see the information panels.

The information panels show the following:

- **Details** ⓘ: name of the task, the action performed, information published, and time stamps
- **Action Executions**☑: the list of action executions performed when the task ran, with their action IDs, status, input and output
  From the Action Executions panel, click **Debug** to test or debug an action.
- **Run Time**🕒: the list of tasks performed in the workflow execution, the run time of each, and their timespan

From the flow diagram, you can debug action executions, or evaluate YAQL expressions.

9.14 How do I configure user access to a workflow?

9.14.1 Purpose

Use this procedure to grant or restrict access to a workflow for a user group. Before you can configure user access, user groups must be created; see the *NSP System Administrator Guide*.

This procedure requires an administrator role.

9.14.2 Steps

1 ───────────────────────────────

Log in to the NSP as an administrator.

2 ───────────────────────────────

Open **Workflows**, **Workflows**.

3 ───────────────────────────────

Select one or more workflows.

Release 25.8
September 2025
Issue 1

© 2025 Nokia.
Use subject to Terms available at: www.nokia.com/terms
3HE-21460-AAAB-TQZZA

115

Press and hold the Shift key to select multiple workflows.

4 ───

Open the User Access form:

- If you have selected one workflow, click ⋮ (Table row actions), **User Access**

- If you have selected multiple workflows, click **User Access** 🖻 at the top right of the page.

5 ───

In the User Access form, configure access to the selected workflows:

1. Choose an action from the drop-down list at the top right of the form.

2. If you chose **Grant access to all user groups**, configure the drop-down lists for each group in the panel at the right.

6 ───

Click **Save**. The user access is updated.

E<small>ND OF STEPS</small> ─────────────────────────────

9.15 Sample procedure: importing and executing a workflow from Git to NSP

9.15.1 Purpose
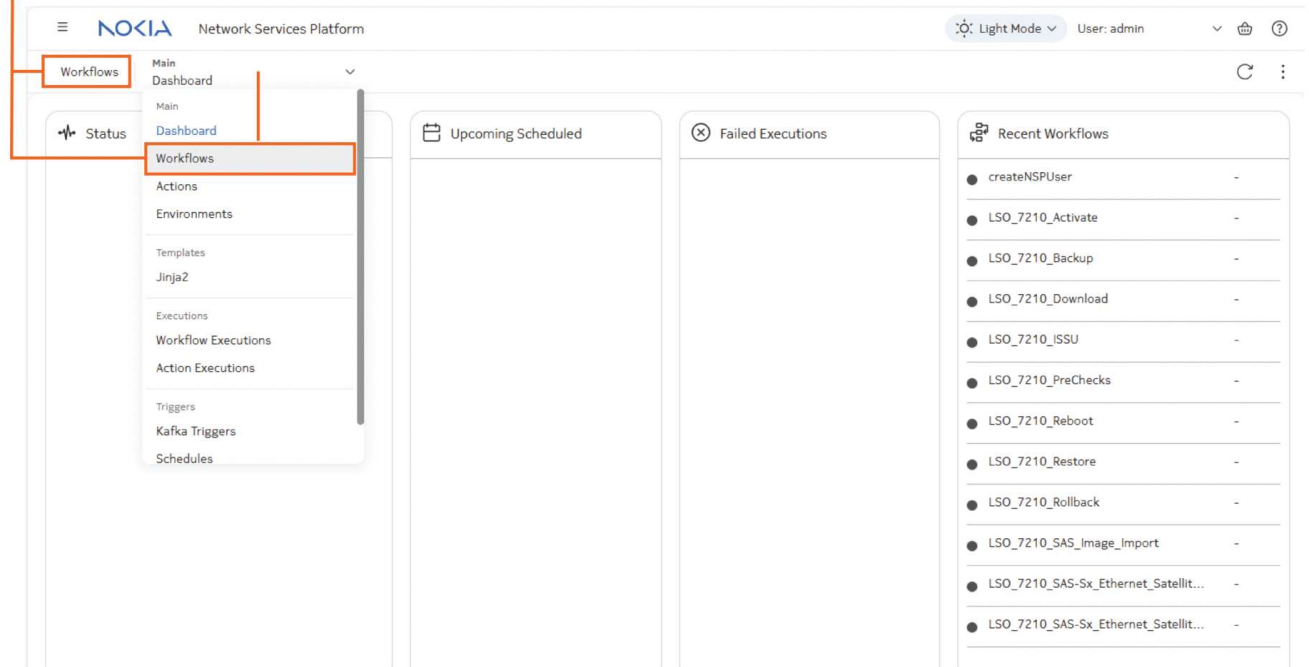
This article shows how to import a workflow from the Nokia GitHub repository, publish and execute the workflow.

9.15.2 Steps

1 ───

Open **Workflows**, **Workflows**.

1 **Choose Workflows from the main menu and from the view selector**

40428

2 ─────────────────────────────────

At the top of the view, choose **Import**, **GitHub**.
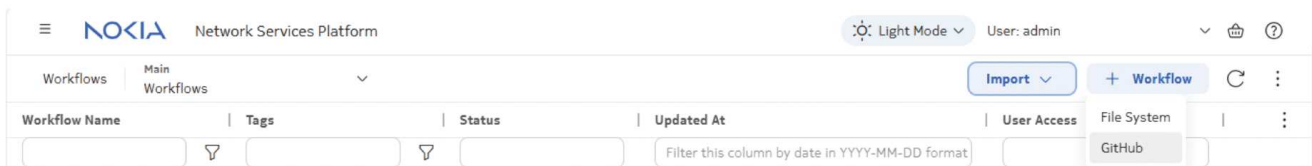
40427

3 ─────────────────────────────────

The Import from Git form shows the list of available workflows in the Nokia NSP repository. We'll import Cleanup WFM Results. Use the filter field to filter the workflows and click Import.

Click **Updates** to verify that the imported workflow is up to date.

Release 25.8
September 2025
Issue 1

© 2025 Nokia.
Use subject to Terms available at: www.nokia.com/terms

3HE-21460-AAAB-TQZZA

117

40430
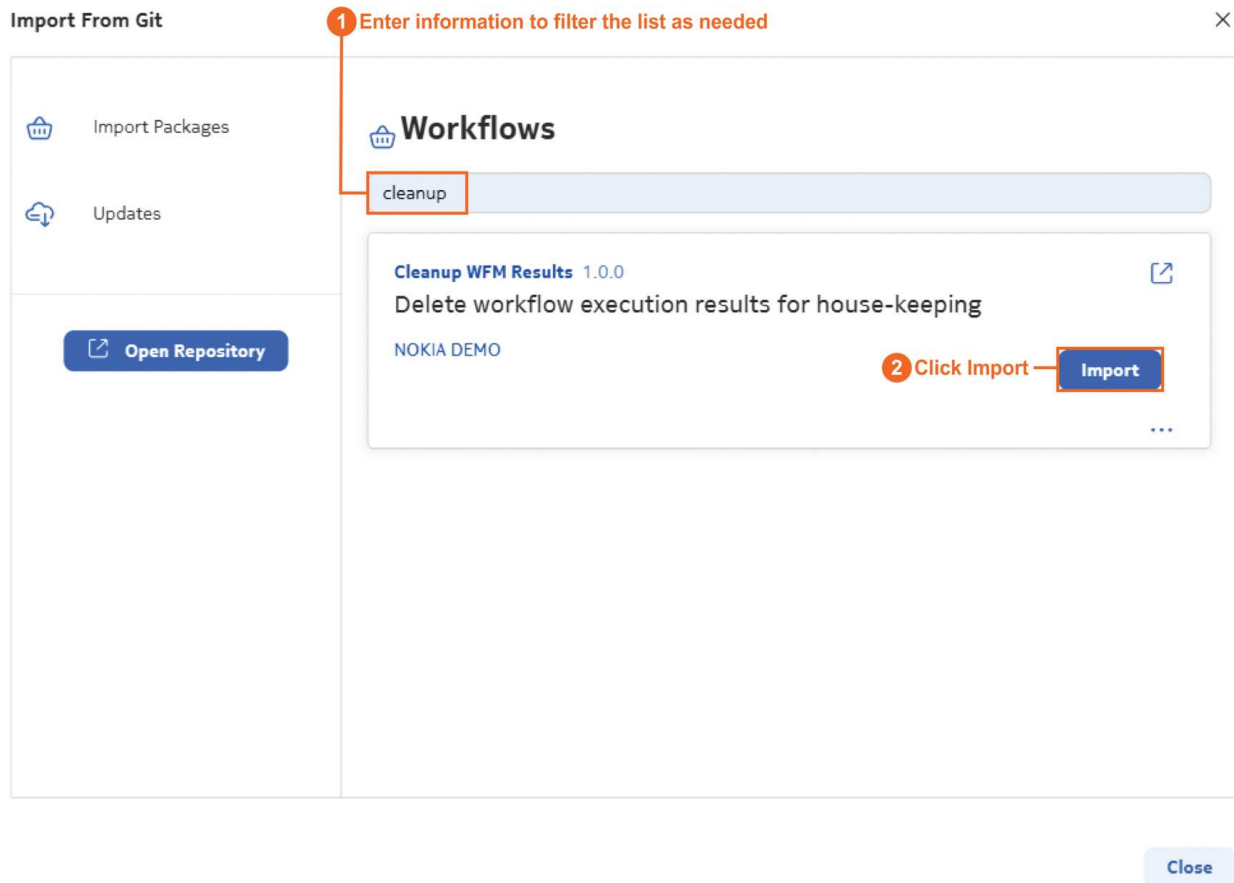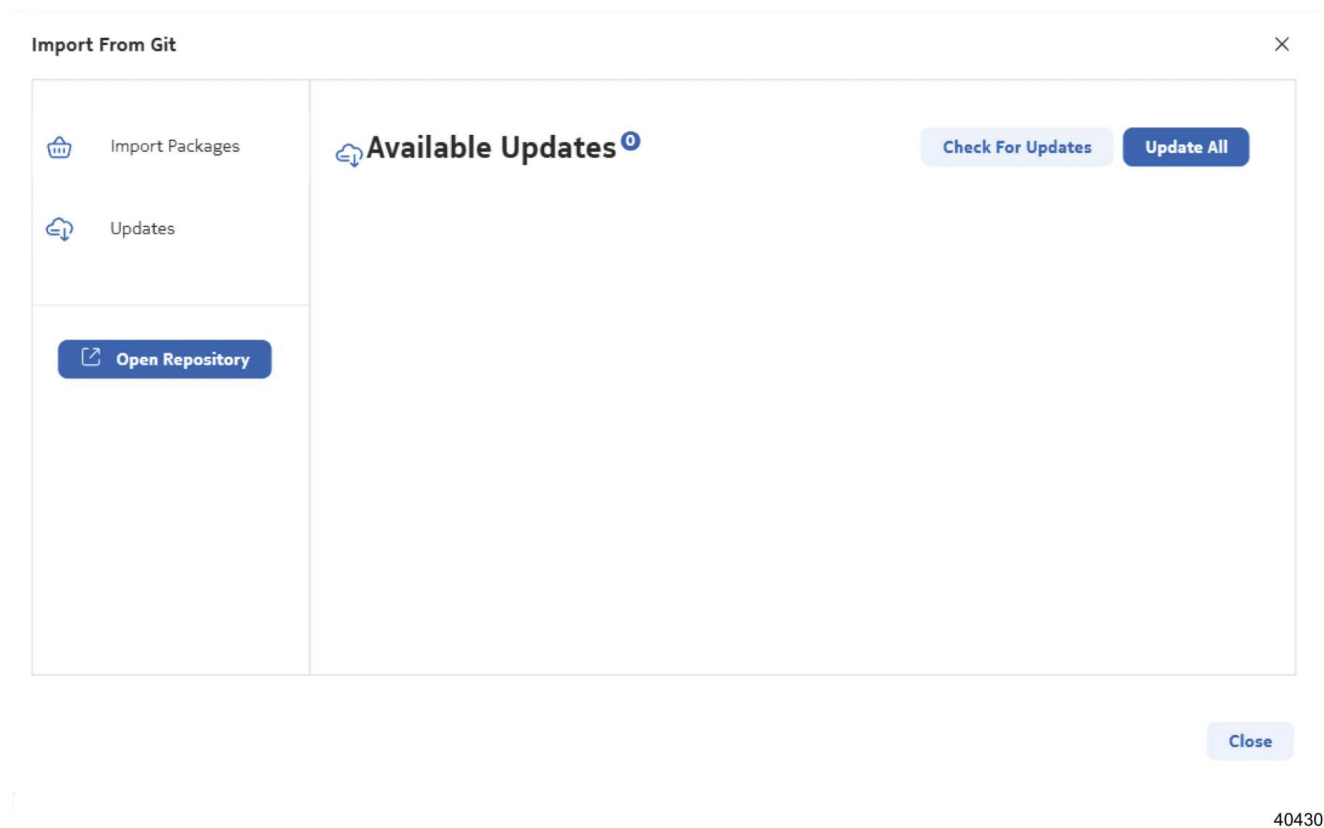
4 ⎯⎯

The imported workflow appears in the **Workflows**, **Workflows** view, with a status of Draft. To execute the workflow, we'll need to change the status to Published.

Release 25.8
September 2025
Issue 1

© 2025 Nokia.
Use subject to Terms available at: www.nokia.com/terms
3HE-21460-AAAB-TQZZA

119
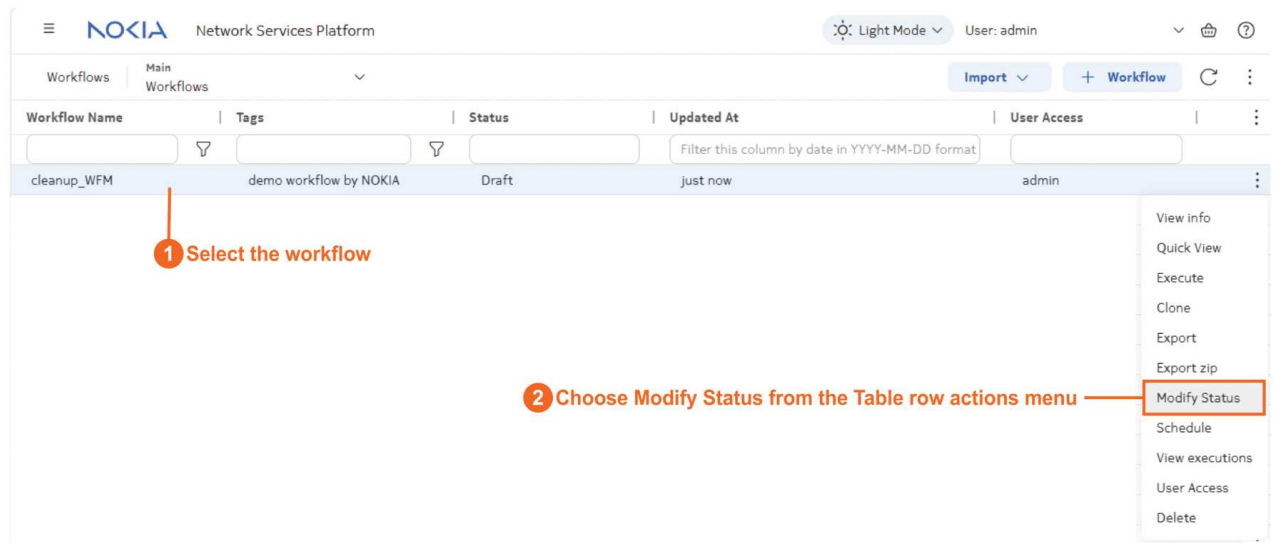
40432
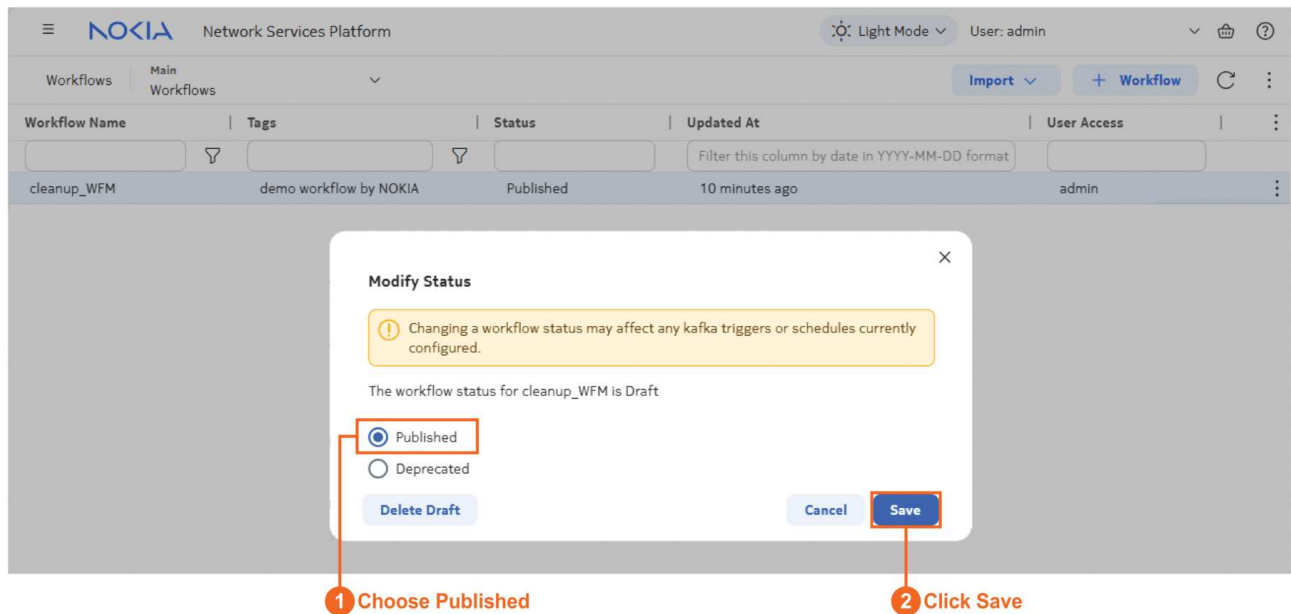
40433

5

Now we can execute the workflow:

40434

Release 25.8
September 2025
Issue 1

© 2025 Nokia.
Use subject to Terms available at: www.nokia.com/terms

3HE-21460-AAAB-TQZZA

121

40435

Result

Let's switch to the **Workflows**, **Workflow Executions** view to view the status. The green checkmark indicates that the workflow has executed successfully.

© 2025 Nokia.
Use subject to Terms available at: www.nokia.com/terms

122   3HE-21460-AAAB-TQZZA

Release 25.8
September 2025
Issue 1

40436

10 Actions and action executions

10.1 Overview

10.1.1 What is an action?

An action is the smallest unit of instruction in a workflow.

A set of actions is included with the NSP. Actions can also be installed using Artifacts; see 2.3 "How do I install an artifact bundle?" (p. 22).

System actions are included with Mistral or downloaded to the NSP, and ad-hoc actions are customized by the user. For example, file.write is a system action: it dumps data to a file. If this action is incorporated into a task, the task must specify the filename to write the data to. The action also has an optional parameter to specify the mode: create a file or append an existing file. An ad-hoc action can be created that incorporates the filename and the append mode into the action. A task that incorporated this ad-hoc action would always append the information to the specified file.

The Actions view shows the actions saved to NSP. Both system actions and ad-hoc actions are listed. You can create, edit, or delete an ad-hoc action. System actions are read-only.

Double-click on an action to open an Action Detail page, which provides a description of the action including parameters and usage examples.

Example

The "Example workflow" (p. 100) shown in 9.1 "What does a workflow look like?" (p. 99) includes several tasks, each of which executes the std.https action. The std.https action is a system action: it sends an HTTPS request. The details of the request are specified in the input.

The following task from the example workflow creates an NSP user group. The task specifies the action and provides the input, that is, the action parameters.

In this example, the workflow input form requires the user to provide a username when the workflow is executed. The createUserGroup task executes the std.https action to send a POST request to the Access Control API, adding the username as a group name.

```
    createUserGroup:
      action: nsp.https
      input:
       url: 'https:
//rest-gateway/access-control-api/rest/api/v3/nspuac/group/'
        method: POST
        body:
          userGroupName: <% $.username %>
          roles: []
```

Release 25.8
September 2025
Issue 1

© 2025 Nokia.
Use subject to Terms available at: www.nokia.com/terms
3HE-21460-AAAB-TQZZA

125

Actions and action executions
How do I create an ad-hoc action?

NSP

```
        publish:

          status: <% task().result.status %>

        on-success:

          - createUser
```

10.1.2 What is an action execution?

An action execution is a record of an executed action. The Action Executions view shows the list of actions that were executed by workflows, the names of the tasks that initiated them, and their status.

Choose **Action Executions** from the drop-down list at the top left of the page to open the **Action Executions** view.

10.1.3 What can I do with actions or action executions?

Viewing the lists of actions and action executions can be useful to developers for troubleshooting and evaluating whether and how workflows might need to be modified.

Double-click on an action on the Actions page to open an Action Detail page. The Action Details page provides documentation about the action, including a description, identifying details, and, the YAML definition of the action.

See the Workflows tutorial on the Network Developer Portal for more information.

To evaluate a YAQL expression, click **.*** to open the Yaqulator.

10.2 How do I create an ad-hoc action?

10.2.1 Steps

1

Open **Workflows**, **Actions**.

2

Click +**Ad-Hoc Action**.

3

In the **Create Action** form that opens, enter, copy or paste your code in the YAML panel.

4

Click **Validate**. Resolve any validation errors that appear.

5

Click **Create**.

END OF STEPS

Actions and action executions
How do I create an action execution?

NSP

10.3   How do I create an action execution?

10.3.1 Steps

1 ───────────────────────────────────

Open **Workflows**, **Actions**.

2 ───────────────────────────────────

Choose an action and click ⋮ (Table row actions), **Run**.

The Run Action Execution form opens.

3 ───────────────────────────────────

Choose the Action Name and provide input as needed.

If an example is provided in the documentation for the action, you can select the example from the Select Example drop-down list. The example code is provided in the input field.

4 ───────────────────────────────────

Click **Run**.

The action executes.

END OF STEPS ───────────────────────────

Actions and action executions
How do I create an action execution?

NSP

11 Environments

11.1 What is an environment?

11.1.1 Environment

An environment file is a set of environment variables for use by workflows. When a workflow is executed, the environment variables are fetched and applied to the workflow where needed.

For example, you can set frequently used values such as hostnames as environment variables to reduce the amount of user input required by workflows.

See the Workflows tutorial on the Network Developer Portal for details about working with environments.

NSP is installed with a default environment, DefaultEnv. The default environment provides the information that NSP needs to reach the Git repository. Access to the NSP GitHub repository is configured by default.

If you want the **Import from Git** button to point to a different location, update the DefaultEnv environment.

> **i** **Note:** The prerequisites for accessing the Git repository must also be in place; see 9.7 "How can I interact with the Nokia Git repository from NSP?" (p. 108).

11.2 How do I create an environment?

> **i** **Attention:** Environment names must only contain alphanumeric characters.

11.2.1 Steps

1 ───────────────────────────────

Open **Workflows**, **Environments**.

2 ───────────────────────────────

Click ＋**Environment**

3 ───────────────────────────────

In the **Create Environment** form that opens, enter a name in the Environment Name field.

4 ───────────────────────────────

a. If **FORM** is selected in the **Edit Format** field:
1. Click ＋**Add**
2. Enter the variable name and value and click **Add**.

3. Repeat for each variable you need to add.

b. If **YAML** is selected in the **Edit Format** field:

1. Enter the variables in the panel.

2. Click **Validate** and resolve any validation errors.

5 ——

Click **Create**.

E<small>ND OF STEPS</small> ————————————————————————————————

11.3 How do I clone an environment?

11.3.1 Steps

1 ——

Open **Workflows**, **Environments**.

2 ——

Choose an environment and click ⋮ (Table row actions), **Clone**. NSP creates a copy of the environment called *environment_name*_CLONE and adds it to the list.

3 ——

See to edit the clone.

E<small>ND OF STEPS</small> ————————————————————————————————

11.4 How do I modify an environment?

11.4.1 Steps

1 ——

Open **Workflows**, **Environments**.

2 ——

Choose an environment and click ⋮ (Table row actions), **Update**.

3 ——

a. If **FORM** is selected in the **Edit Format** field:

1. Click ＋**Add**

2. Enter the variable name and value and click **Add**.

3. Repeat for each variable you need to add.

b. If **YAML** is selected in the **Edit Format** field:

1. Enter the variables in the panel.

2. Click **Validate** and resolve any validation errors.

4 ───

Click **Update**.

E<small>ND OF STEPS</small> ──────────────────────────────

Release 25.8
September 2025
Issue 1

© 2025 Nokia.
Use subject to Terms available at: www.nokia.com/terms
3HE-21460-AAAB-TQZZA

131

© 2025 Nokia.
Use subject to Terms available at: www.nokia.com/terms

132

3HE-21460-AAAB-TQZZA

Release 25.8
September 2025
Issue 1

12 Jinja2 templates

12.1 What is a Jinja2 template?

12.1.1 Template engine

Jinja2 is a template engine for the Python programming language. A Jinja2 template is a structured file containing template tag blocks, or variables, and instructions for how to replace the variables with data. The template engine uses a Python script to read a YAML data file, insert the data into the template tag blocks, and generate the file. For example, Jinja2 templates can be used to convert a data set to a more reader-friendly format such as an HTML document.

Jinja2 templates can be created in NSP, imported, or installed using Artifacts; see 2.3 "How do I install an artifact bundle?" (p. 22).

See the Workflows tutorial on the Network Developer Portal for details about working with Jinja2 templates.

12.1.2 Sample template

The Jinja2 template below is a simple template that creates an HTML page.

```
name: myJinja
tags:
  - a
  - b
description: Sample Jinja2 Template
template: |
  <!doctype html>
  <title>{% block title %}{% endblock %}</title>
  <ul>
    {% for user in users %}
    <li><a href="{{ user.url }}">{{ user.username }}</a></li>
    {% endfor %}
  </ul>
  <p>
    Refer to https://jinja.palletsprojects.com/en/2.10.x/api/
  </p>
```

12.2   How do I create a Jinja2 template?

> **i**   **Attention:** Jinja2 template names must only contain alphanumeric characters.

12.2.1 Steps

1

Open **Workflows**, **Jinja2**.

2

Click ╋**Jinja Template**.

3

In the **Create Jinja Template** form, enter a name for the template.

4

Add tags as needed:

1. Enter text in the **Tags** field and click ╋.
2. Repeat to add additional tags.

5

In the **Template** panel, enter or paste the template YAML.

6

Turn off the **Show Form** toggle at the top right of the form.

7

Click **Validate** and resolve any validation errors that appear.

8

Click **Create**. The Jinja2 template is created.

END OF STEPS

12.3   How do I clone a Jinja2 template?

12.3.1 Steps

1

Open **Workflows**, **Jinja2**.

2

Choose a Jinja2 template and click ⋮ (Table row actions), **Clone**. NSP creates a copy of the environment called *template_name_*CLONE and adds it to the list.

3

See to edit the clone.

Eₙᴅ ᴏꜰ ꜱᴛᴇᴘꜱ

12.4   How do I edit a Jinja2 template?

12.4.1 Steps

1

Open **Workflows**, **Jinja2**.

2

Select a Jinja2 template and click ⋮ (Table row actions), **Update** .

3

Enter your updates in the YAML panel and click **Validate**. Resolve any validation errors.

4

Click **Update**.

Eₙᴅ ᴏꜰ ꜱᴛᴇᴘꜱ

3HE-21460-AAAB-TQZZA

13   Workflow executions

13.1   What is a workflow execution?

13.1.1   Workflow execution

A workflow execution is an instance of an executed workflow.

In **Workflows**, **Workflow Executions**, you can see the list of executions that have been run since the last cleanup, including the inputs provided, their status (Idle, Running, Success, Error, or Canceled), their run time, and how they were triggered. The list can show you the status of scheduled and running workflows, and serve as a record of succeeded and failed executions.

Enable the **Subworkflows** toggle to include executions of subworkflows in the list. If a workflow includes subworkflows, you can expand the workflow execution in the list to see the executions of its subworkflows.

A subworkflow is executed by its parent workflow, therefore the **Executed By** column is blank.

The list shows the 1000 most recent executions by default. Statistics showing the status of workflows are based on the 4000 most recent executions.

Double-click on an execution to open an information page that shows the input entered, tasks that succeeded, any error messages that were produced, and a Flow diagram showing the tasks that executed.

13.1.2   Workflow tasks

A task defines a specific logical step in a workflow. Each task can take input data and produce output data.

A single task is a logical step that can perform an action; for example, performing a REST request using std.http or CLI request using std.ssh. A task can call other workflows. A task can also be iterated over a list of items performing bulk operations.

Tasks can be chained together based on the following criteria:

*   on-success (runs only if the task ran successfully)
*   on-error (runs only if the task failed)
*   on-complete (always runs)

Tasks that are not chained are executed at the same time.

13.1.3   Execution workers

Execution workers are queues used by NSP to maximize the number of operations that can be executed at one time.

A workflow execution is automatically assigned one of the following workers, based on how the execution was initiated:

*   default: executions initiated from the GUI or by an OSS

- trigger: executions initiated by a Kafka trigger or schedule
- LSO: executions initiated by large-scale operations performed by an NSP operator or Device Administrator APIs

Assigning executions to execution workers ensures that , for example, the execution of a large-scale operation will not prevent a scheduled execution from running on time.

Execution worker information is read-only.

13.1.4 Execution status

You can view the status of an execution from the list in **Workflows**, **Workflow Executions**. Select an execution and click ⋮ (Table row actions).

- Choose **Quick View** to view a summary of the execution, including input, output, and flow.
- Choose **Go to Tasks** to see the list of task executions.

The status of the execution can be any of the following:

- Idle: waiting to be run

  If the execution represents a child workflow and the parent workflow or task stops before the child runs, the child execution will remain in this state.

- Running

- Error: an error has been encountered by this execution or by a child workflow.

  If a child workflow is in an error state but the error is handled by the parent task, the parent workflow will not enter an error state.

  An error encountered by a related workflow execution will not cause another execution to enter an error state. For example, if Task 1 requires both Workflow 1 and Workflow 2 to succeed, and Workflow 2 encounters an error, Task 1 will fail, however, the status of Workflow 1 will be based only on the progress of that workflow.

- Success: completed successfully

- Failed

- Canceled: canceled by a user

13.2   How do I rerun a workflow execution?

13.2.1 Purpose

Rerun an execution to execute the workflow again with the same inputs.

13.2.2 Steps

1 ──

To rerun a single execution manually:

1. Open **Workflows**, **Workflow Executions**.
2. Choose a workflow execution and click ⋮ (Table row actions), **Rerun** .

Workflow executions
How do I update the description of a workflow execution?

NSP

2 ———————————————————————————————————

To automate running a workflow repeatedly with the same inputs, create a schedule.

END OF STEPS ———————————————————————————

13.3 How do I update the description of a workflow execution?

13.3.1 Purpose

Use this procedure to add notes to the Description field of a workflow execution. You can filter on the Description field in the Executions list.

13.3.2 Steps

1 ———————————————————————————————————

Open **Workflows**, **Workflow Executions**.

2 ———————————————————————————————————

Choose a workflow execution and click ⋮ (Table row actions), **Update** .

3 ———————————————————————————————————

In the form that opens, make changes as needed.

4 ———————————————————————————————————

Click **Update**.

END OF STEPS ———————————————————————————

13.4 How do I stop a workflow execution?

13.4.1 Purpose

Use this procedure to cancel a workflow execution manually. The execution can also be canceled by an API action or by a policy; see 16.2 "How do I create a workflow execution cleanup policy?" (p. 149).

 i **Note:**

- When a workflow execution is canceled, no new actions are executed. Actions that are already running at the time of cancellation will complete.

- If a workflow contains a child workflow, canceling either the child or the parent will cancel the execution of both workflows.

Release 25.8
September 2025
Issue 1

© **2025 Nokia.**
Use subject to Terms available at: www.nokia.com/terms
3HE-21460-AAAB-TQZZA

139

Workflow executions
How do I change the state of a running or failed workflow execution?

NSP

13.4.2 Steps

1 ─────────────────────────────────────

Open **Workflows**, **Workflow Executions**.

2 ─────────────────────────────────────

Select the execution you need to cancel and click ⋮ (Table row actions), **Cancel**.

END OF STEPS ─────────────────────────────

13.5 How do I change the state of a running or failed workflow execution?

13.5.1 Steps

1 ─────────────────────────────────────

Open **Workflows**, **Workflow Executions**.

2 ─────────────────────────────────────

Choose a workflow execution and click ⋮ (Table row actions), **Modify State** .

3 ─────────────────────────────────────

Choose the new state and click **Update** to confirm.

END OF STEPS ─────────────────────────────

13.6 How do I evaluate a YAQL expression or JSON xpath?

13.6.1 Purpose

Use this procedure to evaluate a YAQL expression or JSON xpath. The Yaqlator evaluates the expression and context to generate a result. You can verify that the result is correct for your requirements.

13.6.2 Steps

1 ─────────────────────────────────────

Open the Yaqlator form:

a.

1. From **Workflows**, **Workflow Executions**, double-click on a workflow execution to open the information page.
2. Choose **Flow** from the drop-down list beside the workflow execution name.
3. From the Flow page, click **Yaqulator** ⁎.

b. From **Workflows**, **Actions**, click **Yaqulator** .***** .

2 ───────────────────────────────────────

Enter or paste the YAML or JSON context for the expression to evaluate in the YAML/JSON Context field.

Verify that the radio button for the correct format is selected.

3 ───────────────────────────────────────

Click **Evaluate**.

The Result field is populated with the results of your query.

E<small>ND OF STEPS</small> ───────────────────────

13.7 How do I debug an action?

13.7.1 Steps

1 ───────────────────────────────────────

Open the information page for a workflow or a workflow execution:

a. Open **Workflows**, **Workflows** and double-click on a workflow.

b. Open **Workflows**, **Workflow Executions** and double-click on a workflow execution.

2 ───────────────────────────────────────

Choose **Flow** from the drop-down list beside the workflow name.

3 ───────────────────────────────────────

From the Flow page, choose an action execution:

1. Choose a task and click **Action Executions**⬚ at the right of the screen to open the Action Executions panel.

2. Select an action execution from the list to show the execution information.

4 ───────────────────────────────────────

Click **Debug**.

5 ───────────────────────────────────────

In the **Run Action Execution** form that opens, update the contents as needed.

6 ───────────────────────────────────────

Click **Run** to run the action with the updated details.

7 ⎯⎯⎯⎯⎯⎯⎯⎯⎯⎯⎯⎯⎯⎯⎯⎯⎯⎯⎯⎯⎯⎯⎯⎯⎯⎯⎯⎯

Click **Result** to verify the results of the updated action.

8 ⎯⎯⎯⎯⎯⎯⎯⎯⎯⎯⎯⎯⎯⎯⎯⎯⎯⎯⎯⎯⎯⎯⎯⎯⎯⎯⎯⎯

As needed, click **Copy** to copy the updated action to the clipboard, and modify the workflow to paste in the copied details.

END OF STEPS ⎯⎯⎯⎯⎯⎯⎯⎯⎯⎯⎯⎯⎯⎯⎯⎯⎯⎯⎯⎯⎯⎯⎯⎯⎯

14 Kafka triggers

14.1 What is a Kafka trigger?

14.1.1 Kafka trigger

A Kafka trigger is a method of event-driven automation. When a Kafka trigger is configured, a workflow is executed in response to a Kafka event notification. For example, if a Kafka notification is received indicating that a port is down, a workflow execution can be triggered to bring up another port and move the services from the down port to the new port.

A Kafka trigger consists of a Kafka topic and event type to monitor, and a trigger rule. The trigger rule is a statement in JSONPath syntax that declares the matching criteria for the workflow to run.

Restrictions apply on topics that can be used for triggering; see the Workflow Manager APIs tutorial on the Network Developer Portal.

By default, the Kafka trigger is limited to 10 executions per minute. This can be changed by updating the Kafka trigger.

i | **Note:** If the **Limit per minute** parameter is set to zero, executions are unlimited.

See the Network Automation tutorial on the Network Developer Portal for details about working with Kafka triggers.

14.1.2 Sample Kafka trigger

The Kafka trigger below is configured to execute the workflow called ExampleWorkflow if an alarm creation event is received on the nsp-db-fm (fault management) Kafka topic for a specified NE.

Create Kafka Trigger ✕

Workflow

ExampleWorkflow ✕ ⓘ

Edit Format*

FORM ⌄ 🗁

Kafka Topic*

nsp-db-fm

Trigger Name*

Alarm Creation

Trigger Rule*

$[?(@.neId == '192.02.0')]

Kafka Event*

CREATE ⌄ 🗁

☑ Enabled

Limit per Minute

10

Cancel   Create

14.2   How do I create a Kafka trigger?

14.2.1 Before you begin

Before a Kafka trigger can be created to execute a workflow, the KafkaTrigger tag must be applied to the workflow definition.

[i]   **Attention:** Kafka trigger names must only contain alphanumeric characters.

14.2.2 Steps

1

Open **Workflows**, **Kafka Triggers**.

2

Click +**Kafka Trigger**.

3

In the **Create Kafka Trigger** form that opens, configure the parameters.

4

Click **Create**.

E<small>ND OF STEPS</small>

14.3   How do I edit a Kafka trigger?

14.3.1 Steps

1

To update the Kafka trigger definition:

1.   Open **Workflows**, **Kafka Triggers**.
2.   Choose a Kafka trigger and click ⋮ (Table row actions), **Update** .
3.   In the Update Kafka Trigger page, make changes as needed.
4.   Click **Update**.

2

To reset the Times Matched and Times Executed counters, select a Kafka trigger and click **Reset Counters** ↺.

E<small>ND OF STEPS</small>

15 Schedules

15.1 What is a schedule?

15.1.1 Schedule

A schedule allows you to configure a workflow to execute in the future, either once or repeatedly.

When you create a schedule you will specify the input and parameters. If you want the input to change from one execution to another, you need multiple schedules. For example, if you would like a cleanup workflow to clean up failed workflow executions once a week and all workflow executions once a month, create a weekly schedule and a monthly schedule.

15.2 How do I schedule a workflow?

> **i** **Attention:** Schedule names must only contain alphanumeric characters.

15.2.1 Steps

1

Open the **Create Schedule** form:

a. Open **Workflows**, **Workflows** view, choose a workflow and click ⋮ (Table row actions), **Schedule**. The Create Schedule form opens for the workflow.

b. Open **Workflows**, **Schedules**, and click ＋**Schedule**. The Create Schedule form opens with no workflow specified.

2

Configure the general parameters.

1. Select a workflow and environment as needed.
2. Enter a schedule name.
3. For NSP to notify Kafka when the schedule runs the workflow, select Kafka Notification.

3

Update the information in the **Input and Parameters** area.

When the schedule runs, it executes the workflow with the input you enter here.

4

In the **Schedule** area, configure the number of executions and the scheduling details.

1. Configure the Start Time parameter. If you only want the schedule to execute once, this is the only parameter you need to configure.

2. Configure repetition if needed:
   - To set a simple repetition pattern, such as once a week, choose **Basic** from the **Mode** drop-down and configure the parameters.
   - To set a custom repetition, such as specific days of the month, choose **Advanced** from the **Mode** drop-down and configure the parameters.

5 ────────────────────────────────

Click **Create**.

END OF STEPS ────────────────────────────

16 Workflow policies

16.1 What is a workflow policy?

16.1.1 Workflow policies

A policy allows you to configure a workflow execution removal operation to execute on a perpetual schedule. You can configure a simple schedule, or create a custom schedule using a cron expression.

The policy can perform a delete operation to clean up the list of executions, or a cancel operation to stop running executions.

Choose **Policies** from the drop-down list at the top left of the page to open the **Policies** view.

When you create a policy you specify the schedule, age of executions to clean up, and filters. You can filter based on tags and/or execution state. For example, you can configure one policy to delete failed workflow executions weekly, and another to delete successful workflow executions with the `testing` tag daily.

The default policy is pre-loaded with installation of NSP. By default, executions are retained for 30 days.

You can also manage policies using the Workflow Manager API; see the Network Automation tutorial on the Network Developer Portal.

16.1.2 Canceling running executions

You can configure a cancel policy to terminate hanging workflow executions. The policy defines the length of time you want to allow execution tasks to remain in a running state.

For example, if each task in a workflow is expected to take an hour or less to run, you can configure a policy to cancel an execution of this workflow when a task has been running for two hours. Alternatively, you can create a policy to cancel all workflow executions with tasks that have been running for longer than a specified length of time.

The duration parameter, or retention period, for cancel policies is based on the time since a task was completed, that is, since the workflow execution was updated. For example, if the workflow has two tasks and the retention period is three hours, the policy is triggered if either task is running for three hours.

16.2 How do I create a workflow execution cleanup policy?

16.2.1 Steps

1 ────────────────────────────────────

Open **Workflows**, **Policies**.

Workflow policies
How do I create a workflow execution cleanup policy?

NSP

2 —————————————————————————————

Click ┿**Policy**. The Create Policy form opens.

3 —————————————————————————————

Enter a policy name.

4 —————————————————————————————

Specify the duration type; days or hours.

5 —————————————————————————————

Define the workflow executions to be cleaned up by the policy:

1. Enter a number in the **Older Than** field.

   The number entered is combined with the Duration Type to configure the retention duration. By default, executions are retained for 30 days.

2. Select tags from the **Filter** field to filter on as needed. The **Filter** field includes the tags that are currently in use for workflows.

   The policy applies to executions of workflows with the selected tags.

3. Select workflows from the **Workflow Name** to filter on as needed.

   The policy applies to executions of the selected workflows.

If the **Filter** and **Workflow Name** fields are both configured, the policy applies when either condition is fulfilled; that is, when the tag is present or the execution belongs to a specified workflow.

6 —————————————————————————————

Choose the operation type.

The delete operation removes executions that are no longer running. The cancel operation stops running executions.

7 —————————————————————————————

Select an execution state.

If the operation parameter is configured as **Cancel executions**, the only supported state is Running.

8 —————————————————————————————

Select the Enabled check box to enable the policy to begin running according to the schedule you configure.

9 —————————————————————————————

Configure the schedule:

a. Click **Schedule** and configure the schedule parameters.

    b. Click **Custom** and configure a cron expression in the format *minute hour day-of-month month day-of-week*.

    Cron expression wildcards are supported. For example, the cron expression `0 0 1 * ?` indicates midnight on the first day of the month.

> **i** **Note:** Nokia recommends that schedules not be set to run more than once a day.

10 ───────────────────────────────────────

Click **Create**.

END OF STEPS ───────────────────────────────

16.3 How do I edit a policy?

16.3.1 Steps

1 ───────────────────────────────────────

Open **Workflows**, **Policies**.

2 ───────────────────────────────────────

Choose a policy and choose ⋮ (Table row actions), **Update**. The Update Policy form opens.

3 ───────────────────────────────────────

Edit the parameters as needed and click **Update**.

END OF STEPS ───────────────────────────────