

Open Programmability User Guide
V100R20C10

Open Programmability User Guide

Issue 01
Date 2021-02-23



Copyright © Huawei Technologies Co., Ltd. 2021. All rights reserved.

No part of this document may be reproduced or transmitted in any form or by any means without prior written consent of Huawei Technologies Co., Ltd.

Trademarks and Permissions



HUAWEI and other Huawei trademarks are trademarks of Huawei Technologies Co., Ltd.

All other trademarks and trade names mentioned in this document are the property of their respective holders.

Notice

The purchased products, services and features are stipulated by the contract made between Huawei and the customer. All or part of the products, services and features described in this document may not be within the purchase scope or the usage scope. Unless otherwise specified in the contract, all statements, information, and recommendations in this document are provided "AS IS" without warranties, guarantees or representations of any kind, either express or implied.

The information in this document is subject to change without notice. Every effort has been made in the preparation of this document to ensure accuracy of the contents, but all statements, information, and recommendations in this document do not constitute a warranty of any kind, express or implied.

Huawei Technologies Co., Ltd.

Address: Huawei Industrial Base
Bantian, Longgang
Shenzhen 518129
People's Republic of China

Website: <https://www.huawei.com>

Email: support@huawei.com

About This Document

Purpose

This document describes how to use the functions of the NCE open programmability system.






Intended Audience

This document is intended for:

- System administrator
- Maintenance engineers
- Technical support engineers

Symbol Conventions

The symbols that may be found in this document are defined as follows.

Symbol	Description
 DANGER	Indicates an imminently hazardous situation which, if not avoided, will result in death or serious injury.
 WARNING	Indicates a potentially hazardous situation which, if not avoided, could result in death or serious injury.
 CAUTION	Indicates a potentially hazardous situation which, if not avoided, could result in minor or moderate injury.
 NOTICE	Indicates a potentially hazardous situation which, if not avoided, could result in equipment damage, data loss, performance deterioration, or unanticipated results. NOTICE is used to address practices not related to personal injury.
 NOTE	Calls attention to important information, best practices, and tips. NOTE is used to address information not related to personal injury, equipment damage, and environment deterioration.

GUI Conventions

The GUI elements that may be found in this document are defined as follows.

Format	Description
""	Buttons, menus, parameters, tabs, windows, and dialog titles are in boldface . For example, click OK .
>	Multi-level menus are in boldface and separated by the ">" signs. For example, choose File > Create > Folder .

Command Conventions

The command conventions that may be found in this document are defined as follows.

Format	Description
Boldface	Command keywords are in boldface .
<i>Italic</i>	Command arguments are in <i>Italic</i> .
[]	Items (keywords or arguments) in square brackets [] are optional.
{x y ...}	Optional items are grouped in braces and separated by vertical bars. One is selected.
[x y ...]	Optional items are grouped in square brackets and separated by vertical bars. One or none is selected.
{ x y ... } *	Optional items are grouped in braces and separated by vertical bars. A minimum of one or a maximum of all can be selected.
[x y ...] *	Optional items are grouped in square brackets and separated by vertical bars. A maximum of all or none can be selected.

Change History

Issue	Date	Description
02	2021-07-30	This issue is the second official release.
01	2020-12-31	This issue is the first official release.

Contents

About This Document.....	ii
1 Overview.....	1
2 Operation Procedure.....	2
3 System Login.....	4
4 Package Repository Management.....	6
4.1 Repository Management.....	6
4.2 Package Management.....	8
4.2.1 Creating a Software Package.....	9
4.2.2 Importing a Software Package.....	10
4.2.3 Deploying a Software Package.....	12
4.2.4 Viewing Software Packages.....	15
4.3 Public Key Management.....	16
4.3.1 Uploading a Public Key.....	16
4.4 Task Management.....	18
4.4.1 Managing Software Packages.....	18
4.4.2 Product Package Management.....	19
5 Device Configuration.....	20
5.1 Configuring a Device on the Web UI.....	20
5.2 Configuring a Device Using the Northbound CLI.....	21
5.3 Verifying Device Data Consistency.....	22
5.3.1 Comparing Configurations.....	23
5.3.2 Synchronizing Data from a Device.....	24
5.3.2.1 Synchronizing Data to the Controller.....	24
5.3.2.2 Configuring Scheduled Synchronization.....	24
5.3.3 Performing Data Consistency Verification.....	25
5.3.4 Saving Southbound Device Configurations.....	25
5.3.5 Managing Device Group Data Consistency.....	27
5.3.6 Managing Configuration Records.....	27
5.4 Device Group Management.....	31
5.4.1 Creating a Device Group.....	31
5.5 Applying a Template.....	31

5.5.1 Applying a Template to a Device.....	32
5.5.2 Applying a Template to a Device Group.....	35
5.5.3 Applying a Template to a Network-Level Service.....	37
6 Service Configuration.....	39
6.1 Configuring Services on the Web UI.....	39
6.2 Configuring Services Using the Northbound CLI.....	46
6.3 Verifying Service Data Consistency.....	46
6.3.1 Performing Service Inconsistency Discovery.....	47
6.3.2 Performing In-Depth Service Inconsistency Discovery.....	47
6.3.3 Checking Southbound Configuration Records of Service Instances.....	49
7 Template Management.....	50
7.1 Managing NE Templates.....	50
7.1.1 Exporting an NE Template Based on a YANG File.....	50
7.1.2 Importing an NE Template.....	51
7.1.3 Creating an NE Template.....	53
7.1.4 Deleting an NE Template.....	54
7.1.5 Exporting an NE Template.....	55
7.2 Managing Service Templates.....	55
7.2.1 Exporting a Service Template Based on a YANG File.....	56
7.2.2 Importing a Service Template.....	56
7.2.3 Creating a Service Template.....	57
7.2.4 Deleting a Service Template.....	59
7.2.5 Exporting a Service Template.....	60
7.3 Managing Template Groups.....	60
7.3.1 Creating a Template Group.....	61
7.3.2 Deleting a Template Group.....	63
7.3.3 Exporting a Template Group.....	64
7.4 Managing Parameter Sets.....	64
7.4.1 Creating a Parameter Set.....	64
7.4.2 Importing a Parameter Set.....	65
7.4.3 Deleting a Parameter Set.....	67
7.4.4 Exporting a Parameter Set.....	68
8 Configuration Management.....	69
8.1 Task Management.....	69
8.2 Commit History	70
9 CLI.....	72
9.1 General Guidelines.....	72
9.2 applications.....	73
9.3 commit.....	73
9.4 display Commands.....	73
9.4.1 display this.....	74

9.4.2 display this-configuration.....	74
9.4.3 display inconsistency-discovery.....	74
9.4.4 display dry-run preview.....	75
9.4.5 display dry-run compare.....	75
9.5 dry-run.....	75
9.6 nes Commands.....	75
9.6.1 nes ne [xxx].....	76
9.6.2 nes inconsistency-discovery [A] [B].....	76
9.6.3 nes sync-to [A] [B].....	76
9.6.4 nes sync-from [A] [B].....	76
9.7 quit.....	76
9.8 reset.....	76
9.9 return.....	76
9.10 clear Command.....	77
9.11 undo.....	77
9.12 rpc Command.....	77
9.13 recompute.....	78
10 Alarm/Event.....	79
10.1 Alarms Reported by Devices.....	79
10.1.1 Third-Party Package of SNMP Traps.....	79
10.1.2 NETCONF Notification.....	80
10.2 Customized Events.....	81
10.3 Alarm Synchronization.....	81
10.4 Alarm Clearance.....	82
11 Service Security.....	83

1 Overview

After a function package is compiled locally, you can import it to the NCE open programmability system (OPS) for online commissioning and use, implementing device management, device configuration delivery, and service configuration delivery.

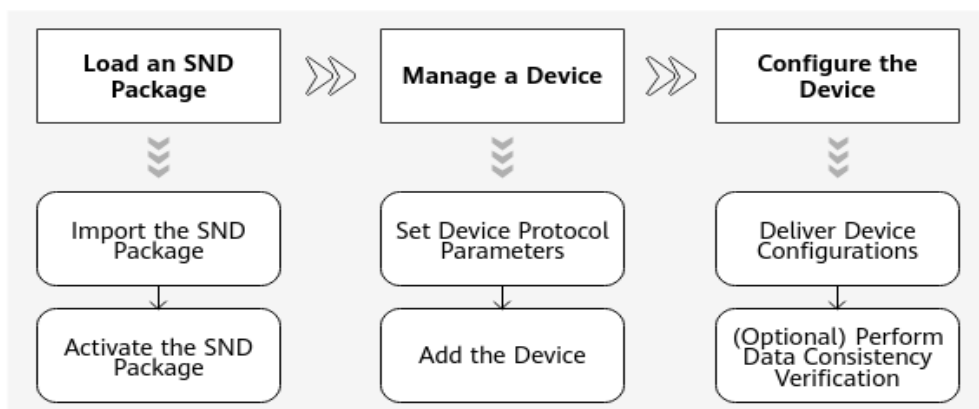
The OPS is applied in the following scenarios:

- **Managing and configuring a device**
The OPS automatically generates the device configuration page based on the device YANG model (compiling and loading an SND package) for you to manage and configure a device.
- **Configuring network services across devices**
The OPS automatically generates the service configuration page based on the service YANG model (compiling and loading an SSP package) for you to deliver service configurations to devices.
- **Verifying data consistency between a device and the OPS**
You can verify data consistency between a device and the OPS. If the data is inconsistent, you can perform data consistency comparison, synchronization, and verification.

2 Operation Procedure

You can develop a function package (such as an SND or SSP package) based on site requirements by referring to the development guide and then configure devices and services by referring to the following process.

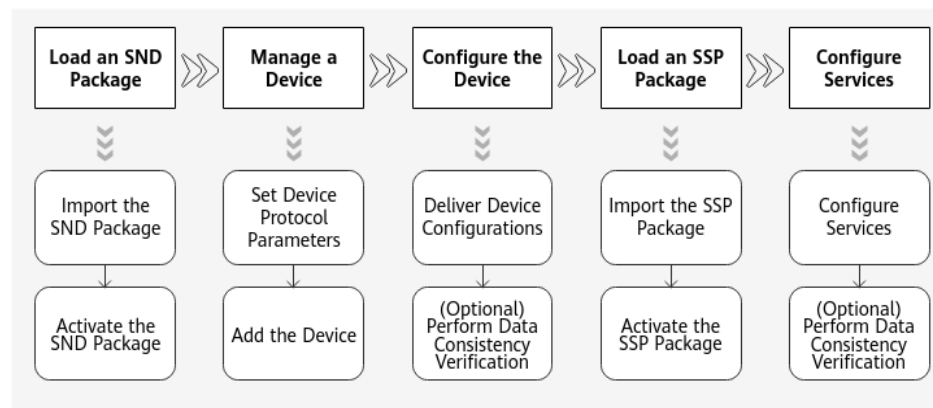
Figure 2-1 Process of managing and configuring a device



1. Load an SND package:
 - a. Import the compiled SND package to the system.
 - b. Activate the SND package.
2. Manage a device:
 - a. Set protocol parameters, including the user name and password.
 - b. Add a device. After the device is added, the controller automatically initiates a connection to the device.
3. Configure the device using either of the following methods:
 - Method 1: After all device data is collected, synchronize data based on the data on NCE or device data.
 - Method 2: Configure the device on the device configuration page.

If the configuration data in the system is modified by a third party or is lost after a device restarts unexpectedly, the configuration data in the system will be inconsistent with that on the device. In this case, you can perform data consistency verification on the device.

Figure 2-2 Process of configuring network services across devices



If a device is not managed, manage the device by referring to the preceding flowchart and then configure services. If the device has been managed, you do not need to perform the first three steps.

1. Load an SND package:
 - a. Import the compiled SND package to the system.
 - b. Activate the SND package.
2. Manage a device:
 - a. Set protocol parameters, including the user name and password.
 - b. Add a device. After the device is added, the controller automatically initiates a connection to the device.
3. Configure the device using either of the following methods:
 - a. Method 1: After all device data is collected, synchronize data based on the data on NCE or device data.
 - b. Method 2: Configure the device on the device configuration page.
4. Load an SSP package:
 - a. Import the compiled SSP package to the system.
 - b. Activate the SSP package.
5. Configure services.

3 System Login

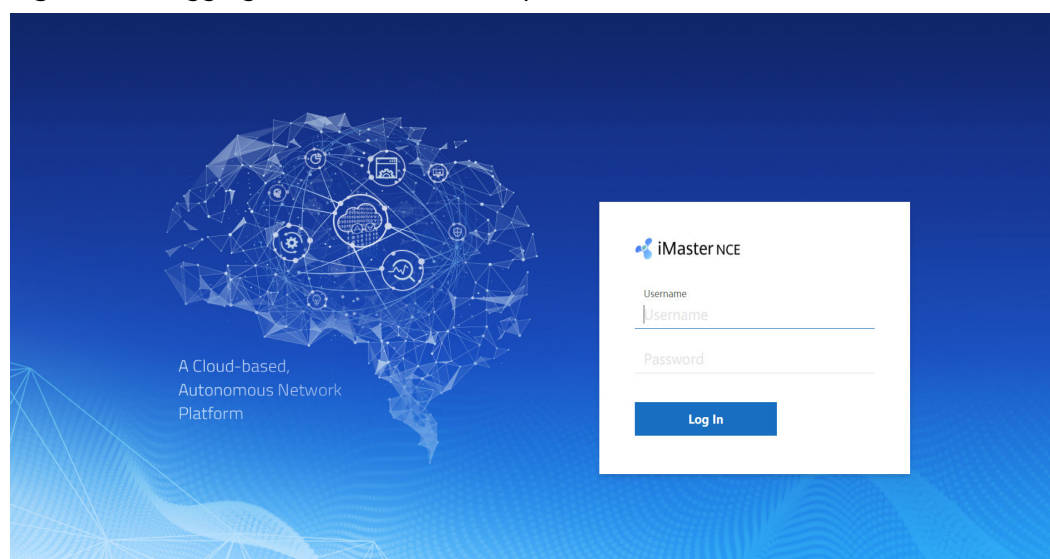
This section describes how to log in to the system using a browser. The table below lists the browsers that are supported.

Browser	Version
Microsoft Edge (Chromium Inside)	MicrosoftEdge_89.0.774.27_Beta
Chrome Stable Channel	Chrome 73.0.3683 and later
Firefox	<ul style="list-style-type: none">• Firefox 65 and later• Firefox 68 ESR and later

Logging In to the Agile Open Container App

Step 1 Log in to the NCE O&M plane. Access the O&M plane at **https://IP address of the O&M plane:31943**. Enter the user name and password and click **Log In**.

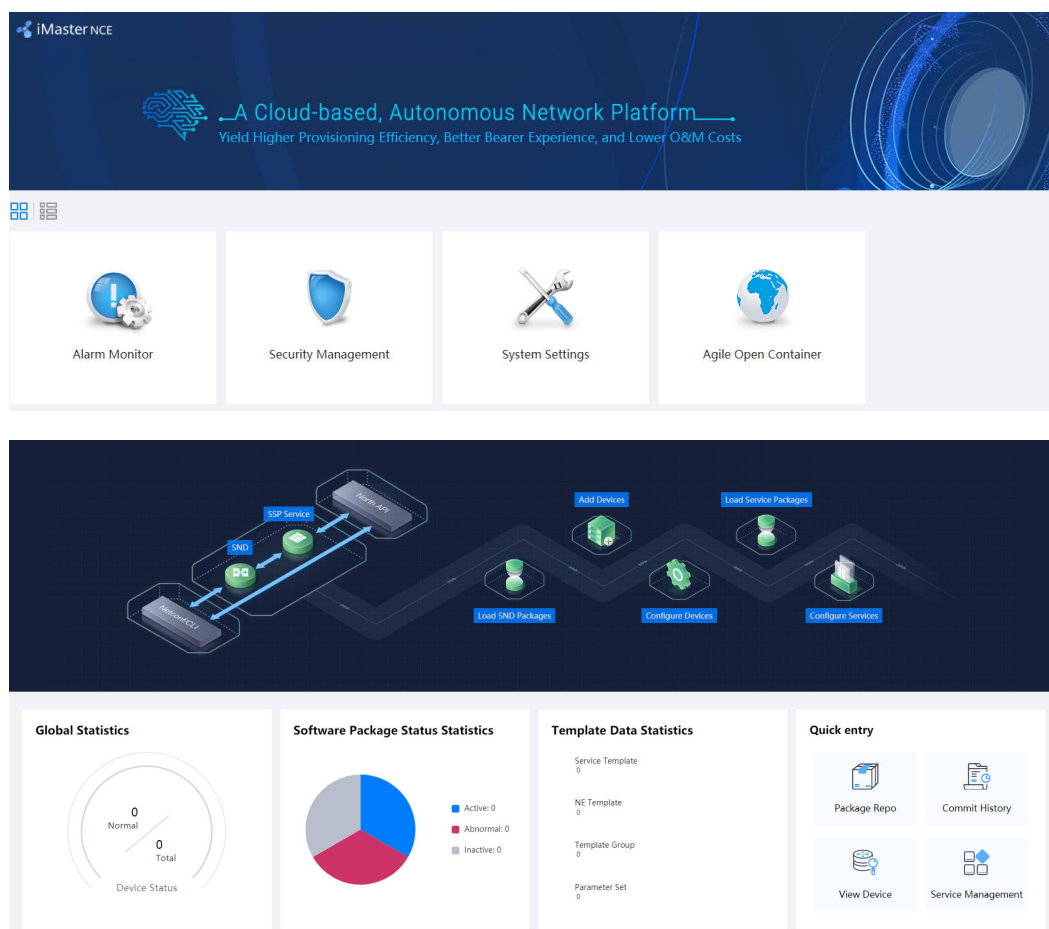
Figure 3-1 Logging in to the NCE O&M plane



 NOTE

- You need to change the password upon the first login. Keep the new password properly. To improve system security, you are advised to periodically change the password to prevent security risks such as brute force cracking.
- The IP address of the O&M plane is the client login IP address configured on the Common_Service node. If the Common_Service node is deployed in a cluster, the IP address is set to the floating IP address of the cluster. If the Common_Service node is deployed in single-node mode, the IP address is the client login IP address of the node.

Step 2 Log in to the system and click the **Agile Open Container** app. The Agile Open Container home page is displayed.



Step 3 On the home page, click the corresponding shortcut entry or click any shortcut entry based on the actual application scenario to access the main menu.

----End

4 Package Repository Management

Package repository management includes repository management, package management, public key management, and task management.

Repository management is used to create, modify, and delete repositories where software packages are stored. Package management is used to import, deploy, update, and delete software packages. Public key management is used to upload and delete public keys. Task management is used to check software package tasks.

[4.1 Repository Management](#)

[4.2 Package Management](#)

[4.3 Public Key Management](#)

[4.4 Task Management](#)

4.1 Repository Management

Scenario Description

A repository stores software packages. Before creating or importing a software package, you must create a repository. You can create repositories of different types or authentication modes as required.

Procedure

- Step 1** Choose **Package Repo** from the main menu.
- Step 2** Choose **Repo** from the navigation pane, and click **New Repo** in the upper right corner of the page. In the **New Repo** dialog box that is displayed, set required parameters.

Table 4-1 Key parameters

Parameter	Description
RepoType	<p>FS: Packages are stored in the disk of the node where the package repository management service is deployed, delivering low reliability.</p> <p>HOFS: Packages are stored in the HOFS file system, implementing high reliability. If you have high requirements on package reliability, HOFS is strongly recommended.</p>
CheckType	<p>SHA256: Integrity verification is performed only on packages stored in the repository.</p> <p>ASC: Signature verification is performed on packages stored in the repository to ensure that the packages are not tampered with.</p> <p>CMS: Signature verification is performed on packages stored in the repository to ensure that the packages are not tampered with.</p> <p>NOTE</p> <p>The security level of SHA256, ASC, and CMS increases in ascending order. To ensure secure storage of packages in the repository, you are advised to select ASC or CMS.</p> <p>If two or more authentication modes are selected, the system performs authentication based on the selected authentication modes in sequence when you upload a package. As long as one authentication mode is passed, the package can be successfully uploaded to the repository.</p>

Figure 4-1 Creating a repository

The screenshot shows a 'New Repo' dialog box with the following fields and values:

- RepoName:** HOFSPUB1
- RepoType:** HOFS
- CheckType:** CMS
- LinkRepo:** (empty dropdown)

Buttons: Cancel, OK

----End

Follow-up Procedure

If **CheckType** is set to **ASC** or **CMS**, you need to import the public key for signature verification. The public key is used to verify the signature of each package managed by the repository. Otherwise, signature verification cannot be performed.

Related Operations

You can perform the following operations as required:

- To modify a repository, click **Modify** in the **Operation** column of the repository to be modified. You can modify the repository name and authentication mode.
- To delete a repository, click **Delete** in the **Operation** column of the repository to be deleted.

4.2 Package Management

4.2.1 Creating a Software Package

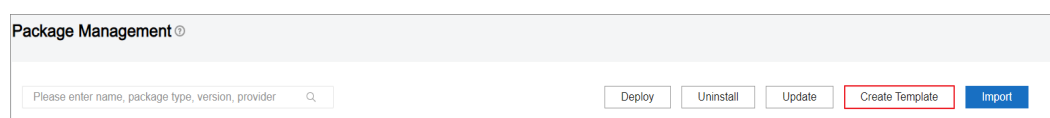
Scenario Description

When a software package is created, the default software template directory and sample code are generated for you to develop the software package.

Procedure

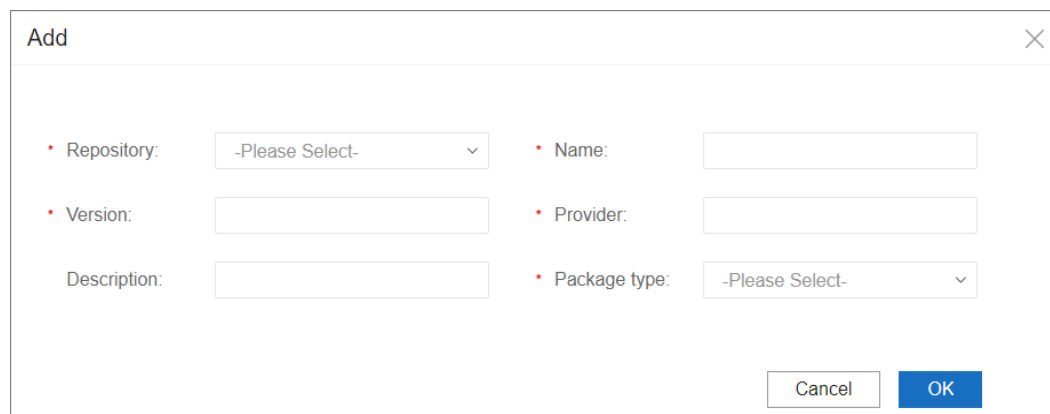
- Step 1** Choose **Package Repo** from the main menu.
- Step 2** Choose **Package Management** from the navigation pane, and click **Create Template** in the upper right corner of the page.

Figure 4-2 Creating a software package



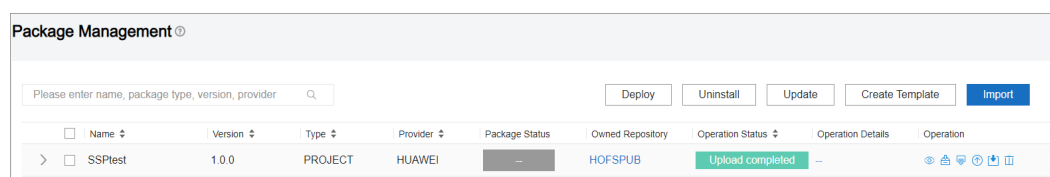
- Step 3** In the **Add** dialog box that is displayed, set mandatory parameters, such as **Name**, **Version**, **Provider**, and **Package type**. The name and version uniquely identify a software package. The provider refers to the provider of a software package.

Figure 4-3 Parameters for creating a software package



- Step 4** After the software package is created, you can view the new software package on the **Package Management** page.

Figure 4-4 Newly created software package template







----End

Follow-up Procedure

You need to export the new software package locally for programming. After the development is complete, import the software package to the system.

Related Operations

- To view a software package, click  in the **Operation** column.
- To uninstall a software package, click  in the **Operation** column.
- To delete a software package, click  in the **Operation** column.
- To export a software package, click  in the **Operation** column.

4.2.2 Importing a Software Package

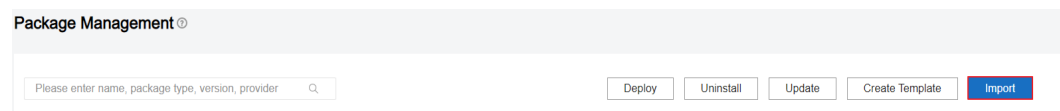
Scenario Description

You can import a software package that has been programmed or obtained from other developers to the system. The imported software package can be used only after being deployed as required.

Procedure

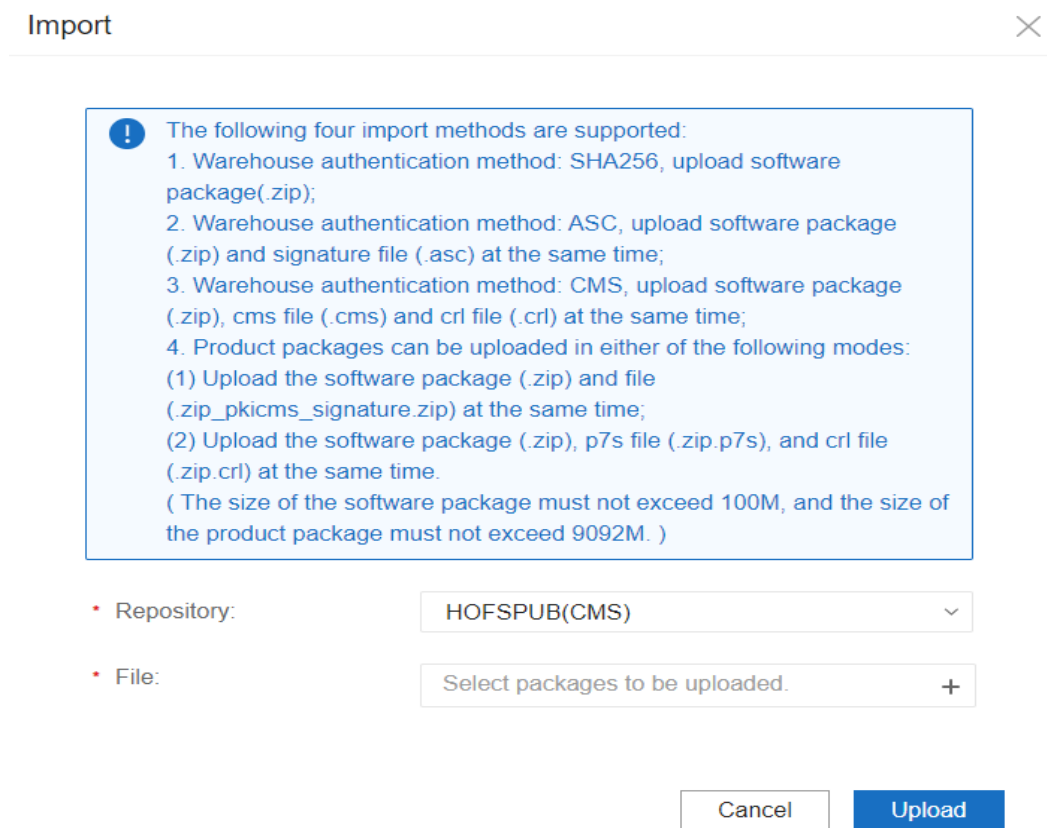
- Step 1** Choose **Package Repo** from the main menu. Then choose **Package Management** from the navigation pane, and click **Import** on the displayed page.

Figure 4-5 Importing a Software Package



- Step 2** In the **Import** dialog box that is displayed, select the software package to be imported.

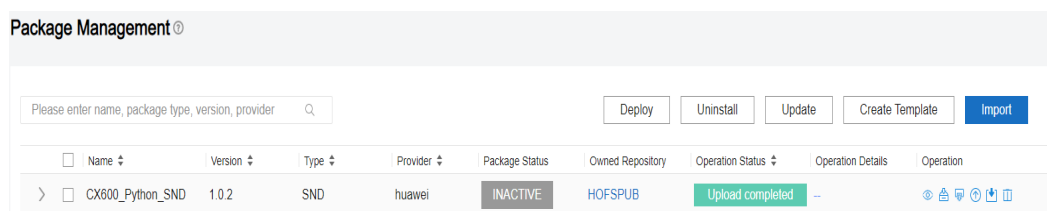
Figure 4-6 Uploading a software package



Step 3 Click **Upload**.

After the software package is imported, you can view the imported software package on the **Package Management** page.

Figure 4-7 Successfully importing a software package






-----End

Follow-up Procedure

You need to deploy the software package after being imported to ensure that software package functions take effect in the system.

Related Operations

- To view a software package, click in the **Operation** column.
- To uninstall a software package, click in the **Operation** column.

- To delete a software package, click  in the **Operation** column.
- To export a software package, click  in the **Operation** column.
- To deploy a software package, click  in the **Operation** column.

4.2.3 Deploying a Software Package

Prerequisites

The software package has been successfully uploaded or uninstalled, and all dependent software packages have been successfully deployed.

Scenario Description

You need to deploy a software package to make it take effect in the system.

Procedure


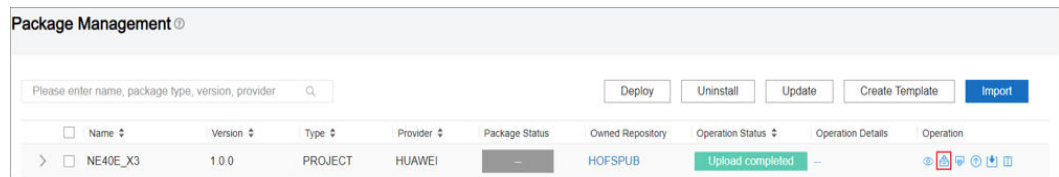
- Step 1** Choose **Package Repo** from the main menu. Then choose **Package Management** from the navigation pane. On the **Package Management** page, click  in the **Operation** column to deploy the software package.


Figure 4-8 Deploying a software package



- Step 2** Wait for a few minutes until the software package is deployed. Software package deployment takes a period of time, especially when there are a large number of YANG files in the installation package.


When a software package is deployed, the system loads the corresponding code, script, and model in the package, and generates database entries based on the model.

Table 4-2 Status change after deployment

Deployment Status	Package Status	Operation Status	Description
DEPLOY_SUCCESS	ACTIVE	DEPLOY_SUCCESS	The software package is successfully deployed. (When a third-party programming component package is activated, the ExtendedPkgRTService process virtualizes a local private IP address 192.168.xxx.xxx or 172.16.xxx.xxx and binds port 30122 as well as ports in the range from 30254 to 30274 to the virtual private IP address. This port is used for internal communication of ExtendedPkgRTService. This IP address cannot be accessed by external systems and is used only for local communication within the open programmability framework.)
DEPLOY_FAILURE	INACTIVE	UPLOAD_SUCCESS (same as the operation status before the deployment)	As long as the software package fails to be deployed, the system performs a rollback and the rollback succeeds. For details about the deployment failure cause, see operation details.
	ABNORMAL	DEPLOY_FAILURE	<ul style="list-style-type: none"> As long as the software package fails to be deployed, the system performs a rollback and the rollback fails. For details about the deployment failure cause, see operation details. If the status of a software package is abnormal, click  to enable the system to automatically restore the software package to the inactive state.

----End

Related Operations

- To update a software package:
 - Click  in the **Operation** column to update the software package to the target version.

 **NOTE**

- Before updating the software package, ensure that the original software package is successfully activated. When the software package is updated, the system automatically uninstalls the original software package and deploys the new software package (that is, the software package of the target version).

Table 4-3 Status change after update

Update Status	Package Status	Operation Status	Description
UPDATE_SUCCESS	Software package of the target version: ACTIVE Original software package: INACTIVE	Software package of the target version: DEPLOY_SUCCESS Original software package: UNINSTALL_SUCCESS	The software package is successfully updated. NOTE When an SSP package is updated and some YANG model objects of the original package are deleted, the system automatically deletes data of the deleted objects during the update. Therefore, exercise caution when you perform this operation.
UNINSTALL_FAILURE	Software package of the target version: INACTIVE Original software package: ACTIVE	Software package of the target version: UPLOAD_SUCCESS Original software package: DEPLOY_SUCCESS	As long as the software package fails to be updated, the system performs a rollback and the rollback succeeds. For details about the update failure cause, see operation details.
	Software package of the target version: ABNORMAL Original software package: ABNORMAL	Software package of the target version: DEPLOY_FAILURE Original software package: UNINSTALL_FAILURE	As long as the software package fails to be updated, the system performs a rollback and the rollback fails. For details about the uninstallation failure cause, see operation details.



- To uninstall a software package, click  in the **Operation** column.

Table 4-4 Status change after uninstallation


Uninstall Status	Package Status	Operation Status	Description
UNINSTALL_SUCCESS	INACTIVE	UNINSTALL_SUCCESS	The software package is successfully uninstalled.
UNINSTALL_FAILURE	ACTIVE	DEPLOY_SUCCESS	As long as the software package fails to be uninstalled, the system performs a rollback and the rollback succeeds. For details about the uninstallation failure cause, see operation details.
	ABNORMAL	UNINSTALL_FAILURE	As long as the software package fails to be uninstalled, the system performs a rollback and the rollback fails. For details about the uninstallation failure cause, see operation details.

- To delete a software package, click  in the **Operation** column.

 **NOTE**

SND packages and preset packages cannot be deleted.

If a software package is abnormal, it may fail to be forcibly deleted. In this case, you can upgrade the software package to an available version and then delete it.

- To export a software package, click  in the **Operation** column.

 **NOTE**


Preset packages can be deployed, uninstalled, and updated in batches.

4.2.4 Viewing Software Packages

Scenario Description

You can view the software packages that exist in the system on the page.

Procedure

Step 1 Choose **Package Repo** from the main menu. Then choose **Package Management** from the navigation pane, and click  on the displayed page.

Step 2 In the file view displayed on the **check** page, click a file. The file content then is displayed on the right area.

Figure 4-9 Software package file view



----End

4.3 Public Key Management

4.3.1 Uploading a Public Key

Scenario Description

If a package signed using a private key is uploaded to the repository, you need to use the public key to verify the signature of the package so as to ensure that the package is not tampered with. In this case, you need to import the corresponding public key before uploading the package.

NOTE

A private key supports only the RSA algorithm, and the private key length must be in the range from 3072 to 4096.

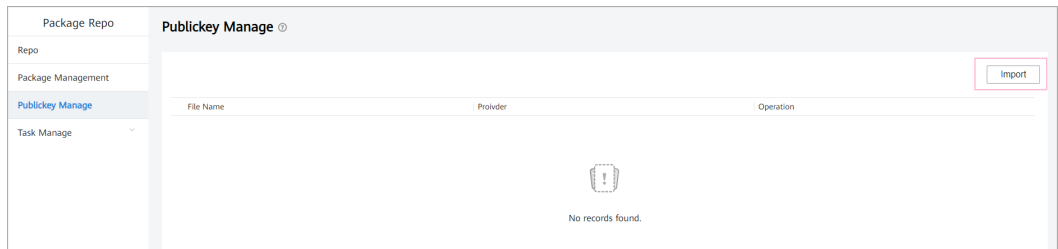
Context

Keys are generated in pairs, including a public key and a private key. The private key is used to sign packages, and the public key is used to verify signatures. If the signature passes the verification, the package is not tampered with.

Procedure

- Step 1** Choose **Package Repo** from the main menu. Choose **Publickey Manage** from the navigation pane, and click **Import** on the displayed page.

Figure 4-10 Importing a public key



Step 2 In the **Import** dialog box that is displayed, set **Provider** to the software package provider, and select the public key file.

Figure 4-11 Entering the public key provider

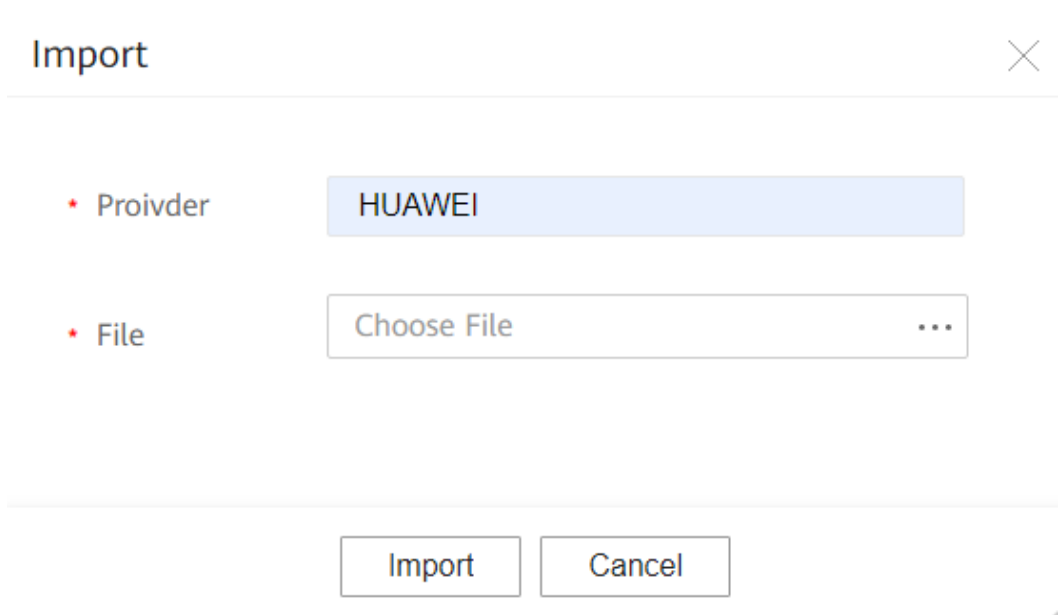
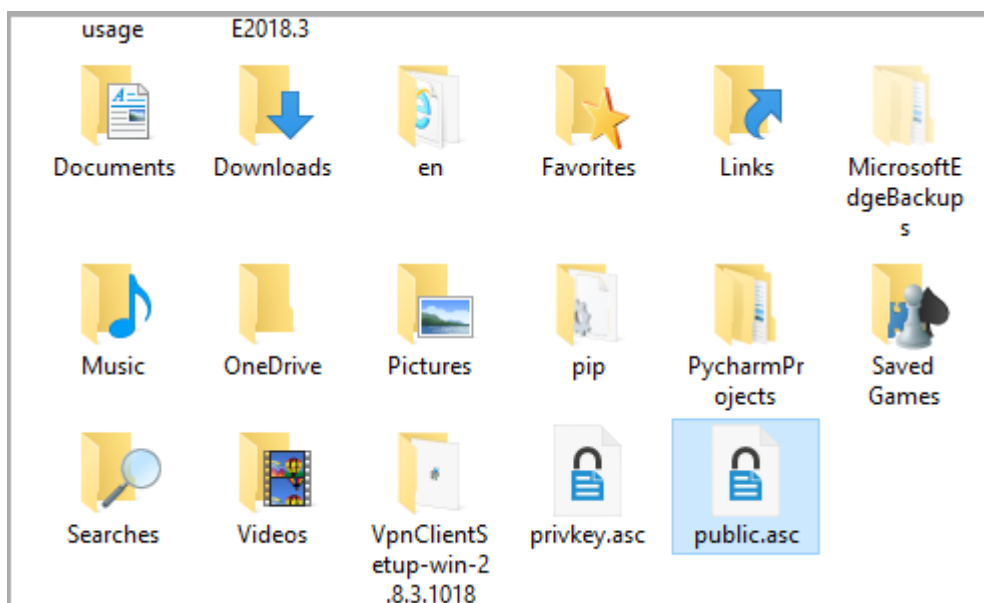
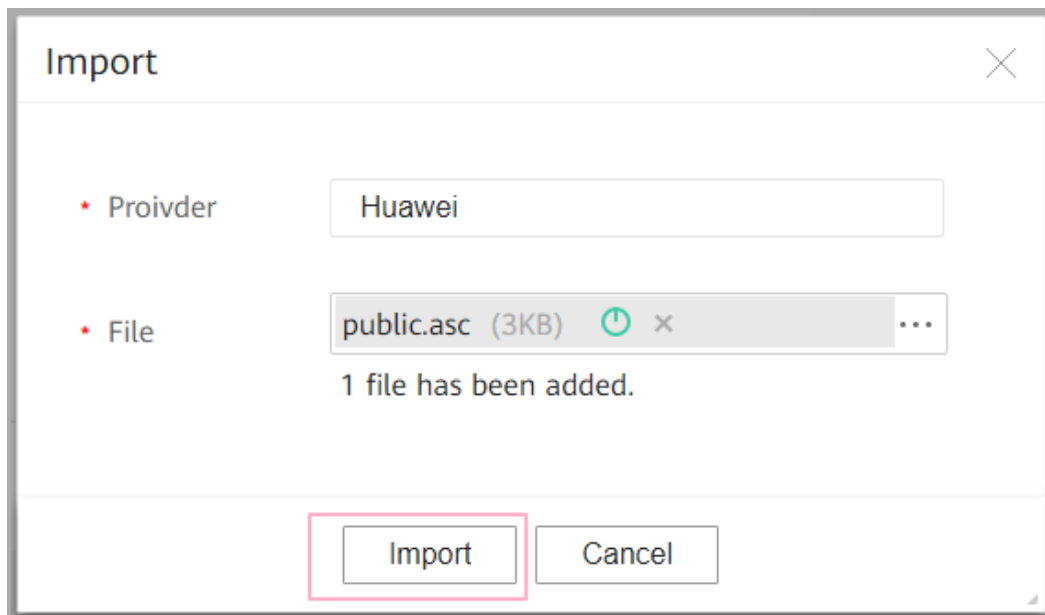


Figure 4-12 Selecting the public key file



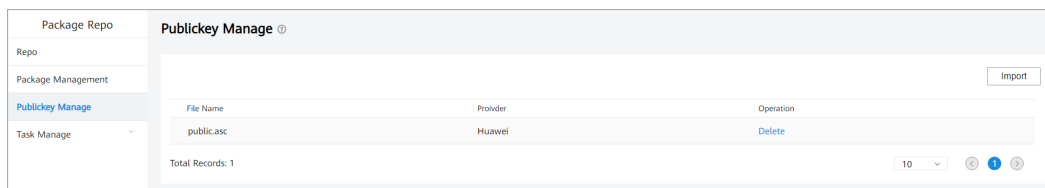
Step 3 Click **Import**.

Figure 4-13 Importing a public key



Step 4 After the public key is imported, you can view it on the **Publickey Manage** page.

Figure 4-14 Successfully importing a public key



----End

Related Operations

To delete a public key, click **Delete** in the **Operation** column.

4.4 Task Management

4.4.1 Managing Software Packages

Scenario Description

You can view the progress of each software package task on the page.

Procedure

Step 1 Choose **Package Repo** from the main menu. Then choose **Task Manage > Package Management** from the navigation pane.

Step 2 Click the collapse icon on the left of the software package to be viewed to expand the software package task details.

Figure 4-15 Checking software package task details

The screenshot shows a 'Software Package Management' window with a 'Refresh' button. Below it is a table with columns: Task ID, Software Package, Workflow, Procedure, and Progress. A task with ID '2605a676-7e71-0e0e-5a8d-b1afd6578def' is selected, showing a workflow of 'ONESTEPACTIVE' and a procedure of 'D:ACTIVE:POST' with 100% progress. Below this is a detailed table for the package 'MigrateChangeValue_python-1.0.4'.

Task ID	Software Package	Workflow	Procedure	Progress
<input checked="" type="checkbox"/> 2605a676-7e71-0e0e-5a8d-b1afd6578def	MigrateChangeValue_python-1.0.4	ONESTEPACTIVE	D:ACTIVE:POST	100%

Package Name	Package Version	Module ID	Operation Type	Procedure	Index Progress	Index Message
MigrateChangeValue_python	1.0.4	ExtendedPkgRTService	PRE_INSTALL	D:INSTALL-PRE	100%	
MigrateChangeValue_python	1.0.4	aocservice	PRE_INSTALL	D:INSTALL-PRE	100%	
MigrateChangeValue_python	1.0.4	ExtendedPkgRTService	INSTALL	D:INSTALL	100%	

----End

4.4.2 Product Package Management

A product package is a collection of product software packages and can be used to manage software packages in batches.

Scenario Description

You can view the progress of each product package task on the GUI.

Procedure

Step 1 Choose **Package Repo** from the main menu. Then choose **Task Management > Product Package Management** from the navigation pane.

Figure 4-16 Product package view

The screenshot shows a 'Product Package Management' window with a search bar and a 'Refresh' button. Below is a table with columns: Name, Transmission Status, Repository ID, Procedure, and Process Result. The table is currently empty, displaying 'No records found.'.

Name	Transmission Status	Repository ID	Procedure	Process Result
No records found.				

Step 2 Collapse a product package to check the product package task details.

----End

5 Device Configuration

- [5.1 Configuring a Device on the Web UI](#)
- [5.2 Configuring a Device Using the Northbound CLI](#)
- [5.3 Verifying Device Data Consistency](#)
- [5.4 Device Group Management](#)
- [5.5 Applying a Template](#)

5.1 Configuring a Device on the Web UI

Procedure

- Step 1** Choose **Device Configuration** > **Device Configuration** from the main menu. The **Device Configuration** page is displayed.
- Step 2** In the **Operation** column of the device to be configured, click **Edit**.

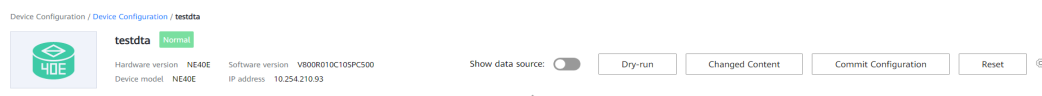
Figure 5-1 Accessing the device configuration page



Icon	Device Name	IP Address	Device Model	Software Version	Status	Vendor	Sync	Sync Status	NE Model Status	Operation
	test	192.168.3.76	NE40E-X3	V800R009C10SPC1...	Normal	huawei	No	Init	Load successful	Edit View Inconsist... More

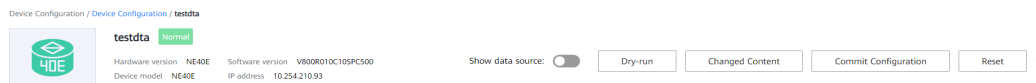
- Step 3** On the page that is displayed, select the corresponding model and edit the model data.
- Step 4** Click **Changed Content** and check the configuration changes.

Figure 5-2 Viewing the changed content



- Step 5** Click **Dry-run** to preview the packets to be delivered.
 - If the result meets expectations, click **Commit Configuration**.

Figure 5-3 Committing the configuration



- If the result does not meet expectations or reconfiguration is required, click **Reset**.

NOTE

- After a user opens a web page, the foreground applies for a transaction ID from the background. If the user does not edit the page for a long time, the background will age the transaction ID. The default validity period of a transaction ID is 5 days. If the third-party package used for editing the configuration is forcibly uninstalled, the background clears the transaction ID that is being used. If the user continues to use the old transaction ID to edit the configuration, a dialog box is displayed, indicating that the local transaction ID has expired. In this case, click **OK** and then click **Reset** to edit the configuration again.
- If multiple users modify the same data at the same time, a transaction conflict may occur. In this case, click **OK** and then click **Reset** to edit the configuration again.

Specifications of transaction conflicts

- In the following scenarios, no conflict will occur:
 - Multiple users modify irrelevant data nodes at the same time.
 - Multiple users modify the same leaf node at the same time, and commit the configurations in sequence.
 - User 1 and user 2 perform operations at the same time. Node A functions as a container or list node. User 1 deletes Node A, and user 2 adds Subnode B under Node A. User 1 commits the configuration first, then user 2 commits the configuration of adding Subnode B without refreshing the page. In this scenario, no conflict will occur, and Node A and Subnode B are retained.
- In the following scenarios, a conflict will occur:

Two different SSP service instances need to be delivered to the same leaf node, but the data is different.

For example, service 1 is decomposed to node A on NE 1, and the decomposed data is 1; service 2 is decomposed to node A on NE 1, and the decomposed data is 2. In this case, a conflict will occur.

----End

5.2 Configuring a Device Using the Northbound CLI

Step 1 Choose **CLI** from the main menu. The CLI is displayed.

```
Welcome to NCE CLI
[NCE]>
```

Step 2 **Optional:** Run a command to enter the device view.

```
[NCE]> nes ne ?
[ipi-zba900-r-bn-01]
[NCE]> nes ne ipi-zba900-r-bn-01
[nes:nes/ne[ipi-zba900-r-bn-01]]>
```

Step 3 Edit device features.

```
[nes:nes/ne[ipi-zba900-r-bn-01]]> system systemInfo sysName Huawei1
[nes:nes/ne[ipi-zba900-r-bn-01]]/huawei-system/system:systemInfo>
```

Step 4 Conduct a dry-run on the device features.

```
[nes:nes/ne[ipi-zba900-r-bn-01]/huawei-system/system:system/systemInfo]> dry-run  
[OK]
```

Step 5 Preview the configuration of device features.

```
[nes:nes/ne[ipi-zba900-r-bn-01]/huawei-system/system:system/systemInfo]> display dry-run preview  
### ipi-zba900-r-bn-01 ###  
<system xmlns="https://www.huawei.com/netconf/vrp/huawei-system">  
<systemInfo>  
<sysName>Huawei1</sysName>  
</systemInfo>  
</system>  
[nes:nes/ne[ipi-zba900-r-bn-01]/huawei-system/system:system/systemInfo]>
```

- If the displayed result in **Step 5** meets expectations, commit the configuration.

```
[nes:nes/ne[ipi-zba900-r-bn-01]/huawei-system/system:system/systemInfo]> commit
```

- To delete the configuration, reset the configuration.

```
[nes:nes/ne[ipi-zba900-r-bn-01]/huawei-system/system:system/systemInfo]> return  
[NCE]> reset
```

----End

5.3 Verifying Device Data Consistency

Prerequisites

A southbound device has been imported to NCE, and the network connection between the southbound device and NCE is working properly.

Constraints and Precautions

1. The AOC does not synchronize sensitive data from devices or compare sensitive data during consistency comparison.
2. For data synchronized from devices, sensitive data fields need to be reconfigured to prevent configuration delivery failures.

Scenario Description

- Configuration comparison
If a southbound device is configured offline or data is lost after it restarts unexpectedly, the configurations of the southbound device will be inconsistent with those on NCE.
To learn the differences between the NE configurations on NCE and running configurations on southbound devices, click **Discover Inconsistencies**.
- If the NE configurations on NCE and running configurations on forwarders are inconsistent:
 - If forwarder configurations are considered correct, select **Synchronize from Device** to synchronize inconsistent configurations from forwarders to NCE.
 - If NE configurations on NCE are considered correct, select **Synchronize to Device** to synchronize inconsistent configurations from NCE to forwarders.

In normal cases, the NE configurations on NCE and forwarder configurations must be consistent.

5.3.1 Comparing Configurations

Procedure

Step 1 Choose **Device Configuration > Device Configuration** from the main menu. The **Device Configuration** page is displayed.

Step 2 Select a southbound device and click **Discover Inconsistencies**.

After the inconsistency discovery is complete, the value of **Sync Status** changes to **Discovered**.

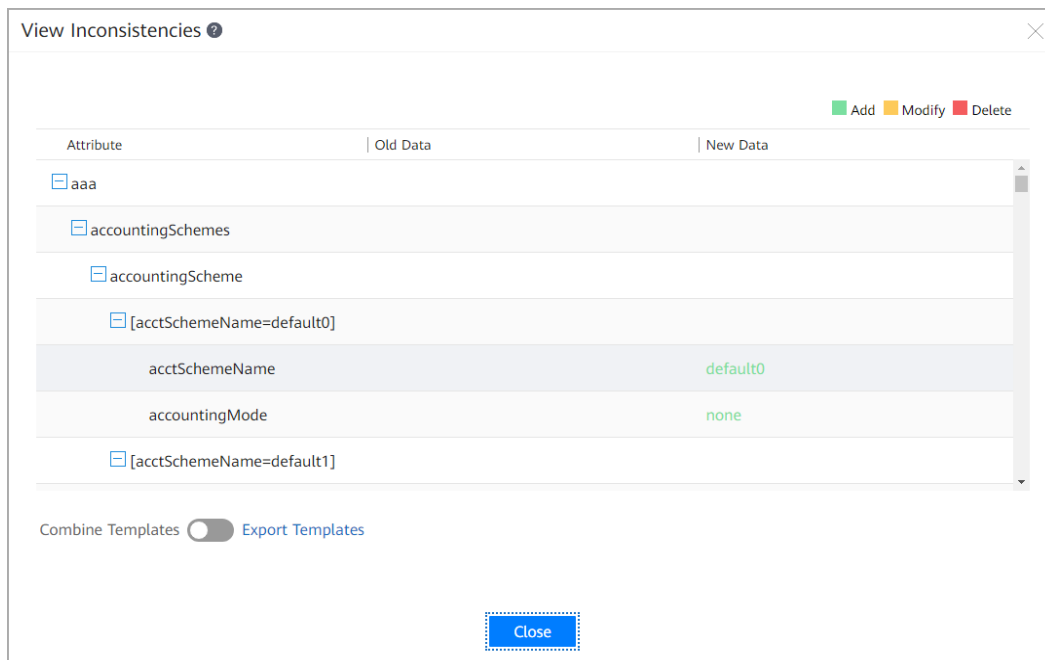
Step 3 Click **View Inconsistencies** in the **Operation** column to access the inconsistency display page and view the inconsistent configurations.

NOTE

The value of **Sync** indicates whether the forwarder and NCE configurations are consistent. If the value is **No**, the forwarder and NCE configurations may be inconsistent, and the actual result depends on the SND implementation. To determine whether the configurations are consistent, use the inconsistency discovery function.

Step 4 On the inconsistency display page, the data in the **Old Data** column indicates the NCE data, and the data in the **New Data** column indicates the forwarder data.

Figure 5-4 Inconsistency display page



 NOTE

When developing a Jinja2 template, you can deliver related configurations to devices and export the corresponding device configuration segments using the template export function in inconsistency discovery.

----End

5.3.2 Synchronizing Data from a Device

5.3.2.1 Synchronizing Data to the Controller

Procedure

- Step 1** Choose **Device Configuration > Device Configuration** from the main menu. The **Device Configuration** page is displayed.
- Step 2** Select a southbound device and choose **Synchronize > Synchronize from Device**.
After data is successfully synchronized, the value of **Sync Status** changes to **Synchronized**.
- Step 3** Click **View Inconsistencies** in the **Operation** column to access the inconsistency display page and view the inconsistent configurations between NCE and the forwarder before the synchronization.

----End

5.3.2.2 Configuring Scheduled Synchronization

Procedure


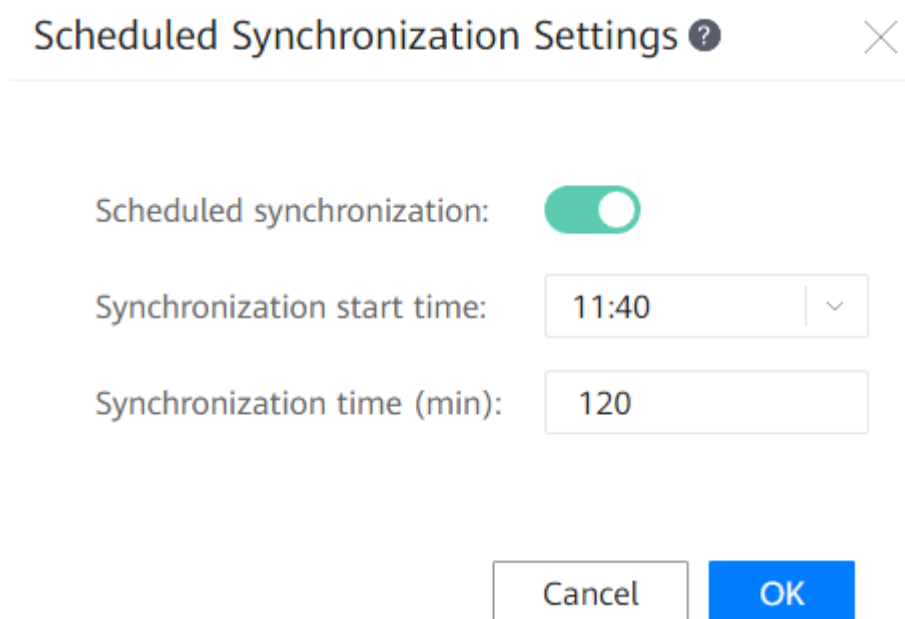
- Step 1** Choose **Device Configuration > Device Configuration** from the main menu. The **Device Configuration** page is displayed.
- Step 2** Select a southbound device, click  next to **Reset**, and click **Scheduled Synchronization**.

Figure 5-5 Configuring scheduled synchronization



- Step 3** Enable **Scheduled synchronization** and set synchronization parameters.

Figure 5-6 Enabling scheduled synchronization



Scheduled Synchronization Settings ?

Scheduled synchronization:

Synchronization start time: 11:40

Synchronization time (min): 120

Cancel OK

Step 4 Click **OK**.

----End

5.3.3 Performing Data Consistency Verification

Device configurations may be deleted or modified during data consistency verification. Therefore, you are advised to perform inconsistency discovery before data consistency verification and verify that the left and right values of the discovered inconsistencies meet your expectations to prevent misoperations on device configurations.

Procedure

Step 1 Choose **Device Configuration** > **Device Configuration** from the main menu. The **Device Configuration** page is displayed.

Step 2 Select a southbound device and choose **Synchronize** > **Synchronize to Device**.

After data consistency verification is complete, the value of **Sync Status** changes to **Reconciled**.

Step 3 Click **View Inconsistencies** in the **Operation** column to access the inconsistency display page and view the inconsistent configurations between NCE and the forwarder before the data consistency verification.

----End

5.3.4 Saving Southbound Device Configurations

Context

After NCE delivers configurations (including data consistency verification) to a southbound device, the southbound device needs to periodically save the

configurations to prevent configuration loss caused by device restarts. Theoretically, devices should automatically save the configurations. However, some devices cannot automatically save configurations. To prevent this problem, NCE provides the function of manually saving southbound device configurations.

Procedure

Step 1 Choose **Device Configuration > Device Configuration** from the main menu. Select a southbound device, and choose **More > Save**.


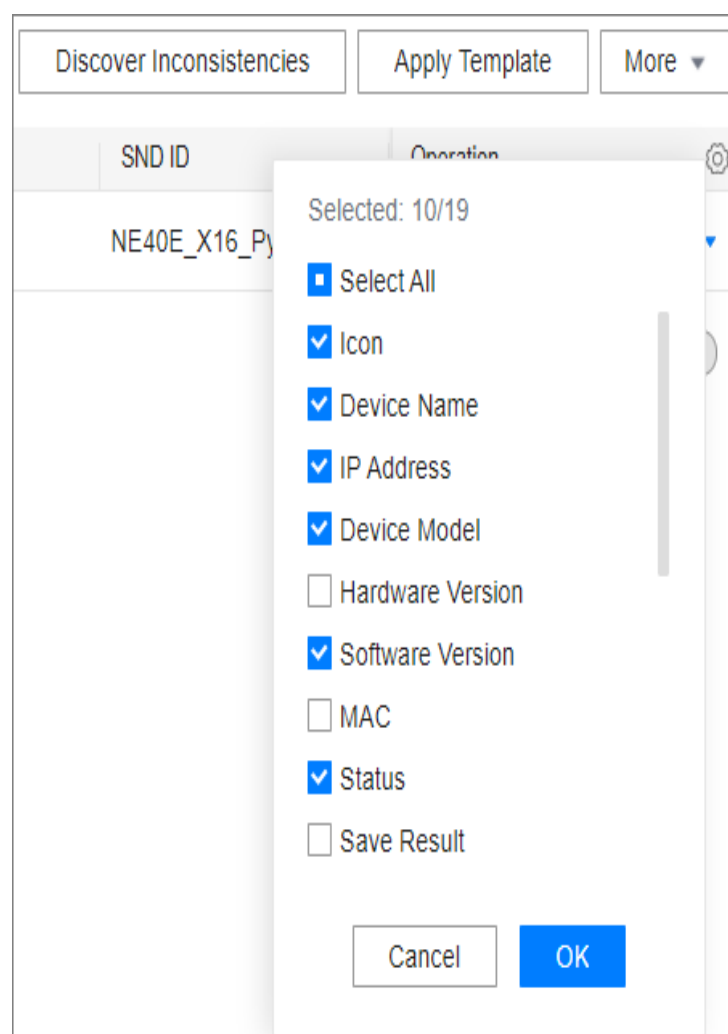
Step 2 Click  and select related options, such as **Save Result** and **Save Time**.

Figure 5-7 Setting device display options



Step 3 Check the saving result.

After southbound device configurations are saved successfully, the saving result is **Success**.

----End

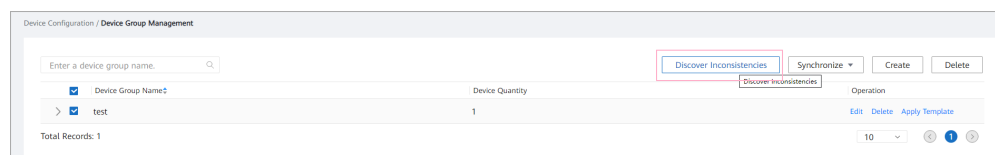
5.3.5 Managing Device Group Data Consistency

Procedure

Step 1 Choose **Device Configuration > Device Group Management** from the main menu. Then select a device group.

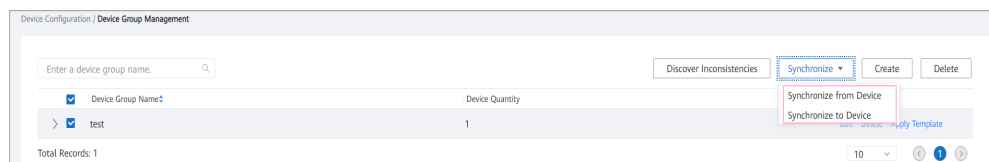
- Click **Discover Inconsistencies** to trigger inconsistency discovery for the device group.

Figure 5-8 Triggering inconsistency discovery for a device group



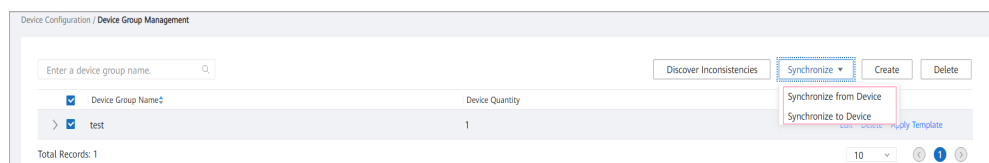
- Click **Synchronize** and select **Synchronize from Device** to trigger synchronization for the device group.

Figure 5-9 Triggering synchronization for a device group



- Click **Synchronize** and select **Synchronize to Device** to trigger data consistency verification for the device group.

Figure 5-10 Triggering data consistency verification for a device group



Step 2 After inconsistency discovery, synchronization, or data consistency verification is triggered for a device group, the **Information** dialog box is displayed. Click the hyperlink in the dialog box to access the **Device Configuration** page and view the operating status of specific devices.

----End

5.3.6 Managing Configuration Records

Context

NCE provides the function of recording device configurations. You can view device configuration records on the **Device View** and **Record View** pages. In addition, you can view the configuration records delivered to southbound devices after service instances are decomposed.

Procedure

Step 1 Choose **Device Configuration > Device Configuration History** from the main menu.

The **Device Configuration History** page displays configuration records by device or record sequence.

Figure 5-11 Device view

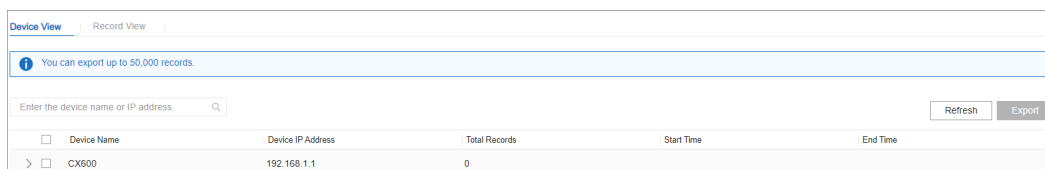
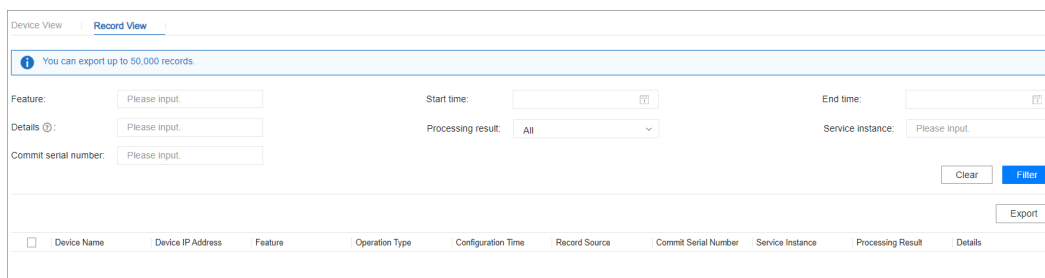


Figure 5-12 Record view

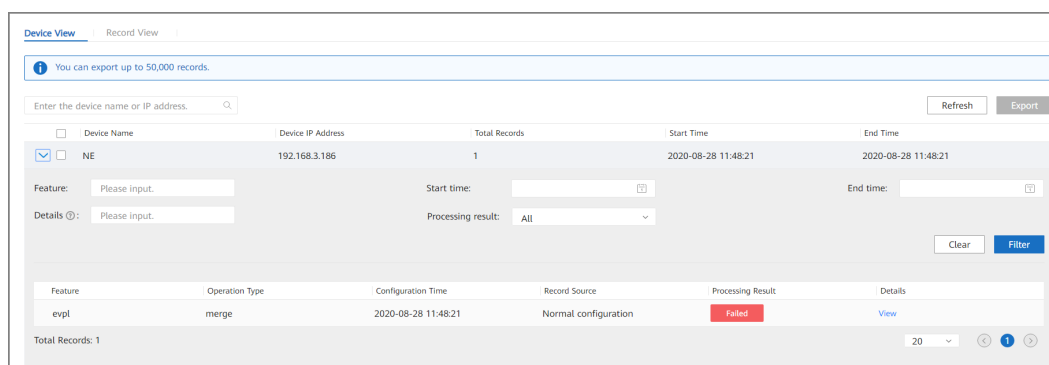


Step 2 View historical configuration records.

- Device view

In the device view, click > to view the configuration record details. Click **View** in the **Details** column to view the configuration data delivered by NCE to the southbound device.

Figure 5-13 Configuration record details in the device view



NOTE

If too many records exist, filter them by **Feature**, **Details**, **Start time**, **End time**, and **Processing result**.

Table 5-1 Configuration record parameters

Parameter	Description
Device Name	Name of a southbound device.
Device IP Address	IP address of a southbound device.
Total Records	Number of configuration records delivered by NCE to a southbound device.
Start Time	Start time of configuration delivery.
End Time	End time of configuration delivery.
Feature	Name of a feature.
Operation Type	Operation type of delivered packets.
Configuration Time	Configuration time of a feature.
Record Source	There are two record sources: <ul style="list-style-type: none"> - Normal configuration - Rollback configuration
Processing Result	There are two processing results: <ul style="list-style-type: none"> - Succeed - Failed
Details	Check the configuration data delivered by NCE to a southbound device.

- Record view
 Compared with the device view, the record view has two more fields: **Transaction ID** and **Service Instance**.

Table 5-2 Fields specific to the record view

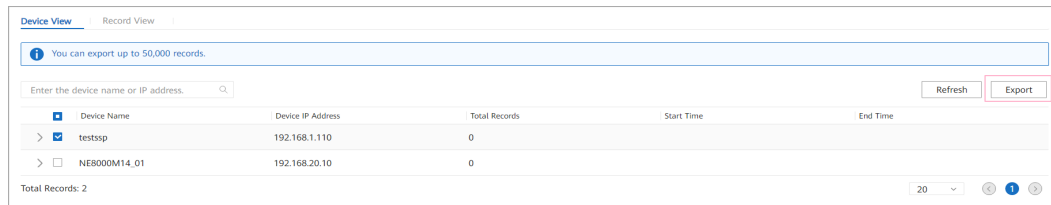
Parameter	Description
Transaction ID	Transaction ID used when NCE delivers configurations to a southbound device. This field is empty when configurations are delivered in non-transaction mode.
Service Instance	Service instance path configured for a southbound device.

Step 3 Export historical configuration records.

- Device view
 In the device view, select a southbound device and click **Export** to export the southbound device configuration records of the selected device. After the

records have been exported, the exported records are automatically downloaded.

Figure 5-14 Export operation in the device view



- Record view

In the record view, select configuration records, and click **Export**. In the dialog box that is displayed, click **Export all data** or **Export selected data** to export specific device configuration records.

The exported records are automatically downloaded.

Figure 5-15 Export operation in the record view

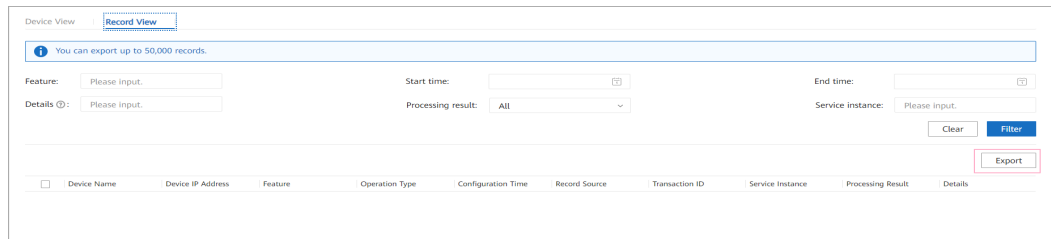
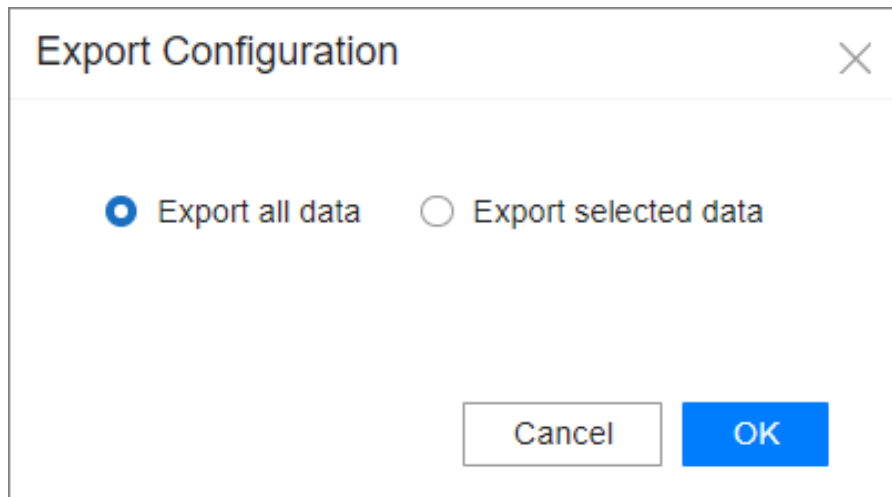


Figure 5-16 Export settings



----End

5.4 Device Group Management

Scenario Description

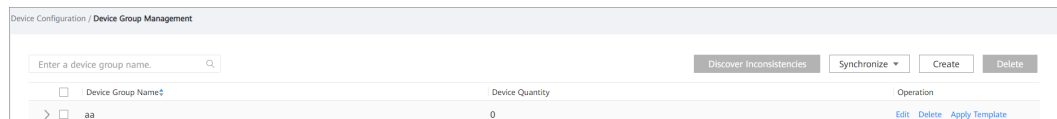
Before applying a template or template group to multiple devices, add these devices to the same device group. A device group may contain multiple devices and device groups.

5.4.1 Creating a Device Group

Procedure

- Step 1** Choose **Device Configuration > Device Group Management** from the main menu. The **Device Group Management** page is displayed.
- Step 2** Click **Create**. In the dialog box that is displayed, enter the device group name and click **OK**.
- Step 3** On the page that is displayed, add a device or sub device group.
 - On the **Device** tab page, click **Add** to add a device.
 - On the **Device Group** tab page, click **Add** to add a sub device group.
- Step 4** In the **Add** dialog box that is displayed, select the device or device group to be added and click **OK**.
- Step 5** Click **Device Group Management** in the upper left corner to return to the **Device Group Management** page and view the created device group.

Figure 5-17 Checking the device group



----End

Related Operations

- To edit a device group, choose **Device Configuration > Device Group Management**. On the page that is displayed, click **Edit** in the **Operation** column.
- To delete a device group, choose **Device Configuration > Device Group Management**. On the page that is displayed, click **Delete** in the **Operation** column.

5.5 Applying a Template

Scenario Description

To orchestrate and deliver configurations to devices or networks in batches, you can select devices or device groups or access the **Application Template** page from

the **Service Management** page to orchestrate configuration templates and deliver configurations.

5.5.1 Applying a Template to a Device

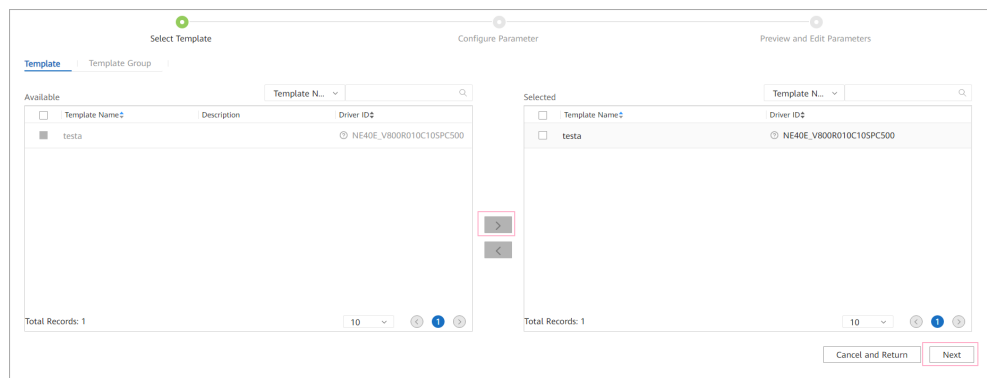
Prerequisites

- A device is properly connected to NCE.
- A template has been created according to [7.1.3 Creating an NE Template](#).
- (Optional) The parameter set required by the template has been created according to [7.4.1 Creating a Parameter Set](#).

Procedure

- Step 1** Choose **Device Configuration** > **Device Configuration** from the main menu.
- Step 2** On the **Device Configuration** page that is displayed, select a device and click **Apply Template**. Alternatively, click **More** in the **Operation** column of a device, and select **Apply Template**.
- Step 3** On the **Template** tab page, select a template in the available template area, click **>**, and click **Next**. Alternatively, click the **Template Group** tab, and select a template group in the available template group area. Then click **>** and click **Next**.

Figure 5-18 Adding a template

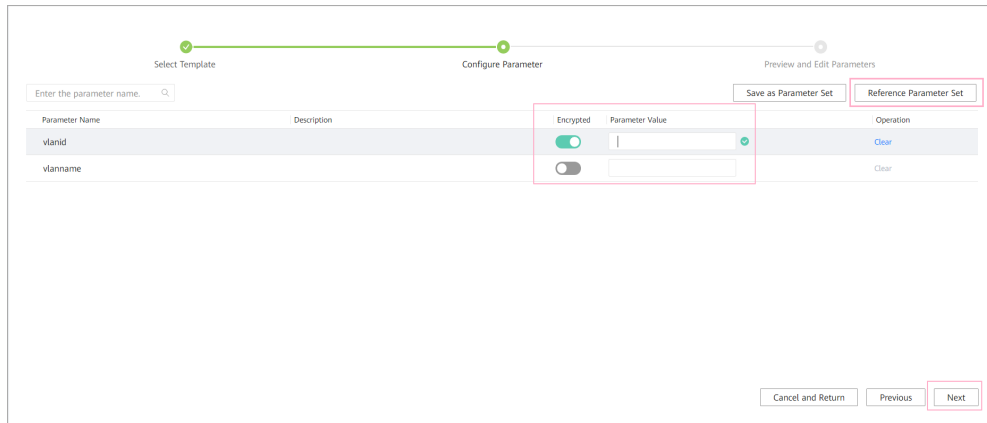


- Step 4** On the **Configure Parameter** page that is displayed, modify parameter values and click **Next**.

You can modify the parameters as follows:

- Click **Reference Parameter Set** to reference the parameter content in a parameter set.
- Enable the **Encrypted** function if a parameter needs to be displayed in ciphertext on the GUI.
- Directly enter the parameter value.

Figure 5-19 Configuring parameters



Step 5 Preview and edit parameters.

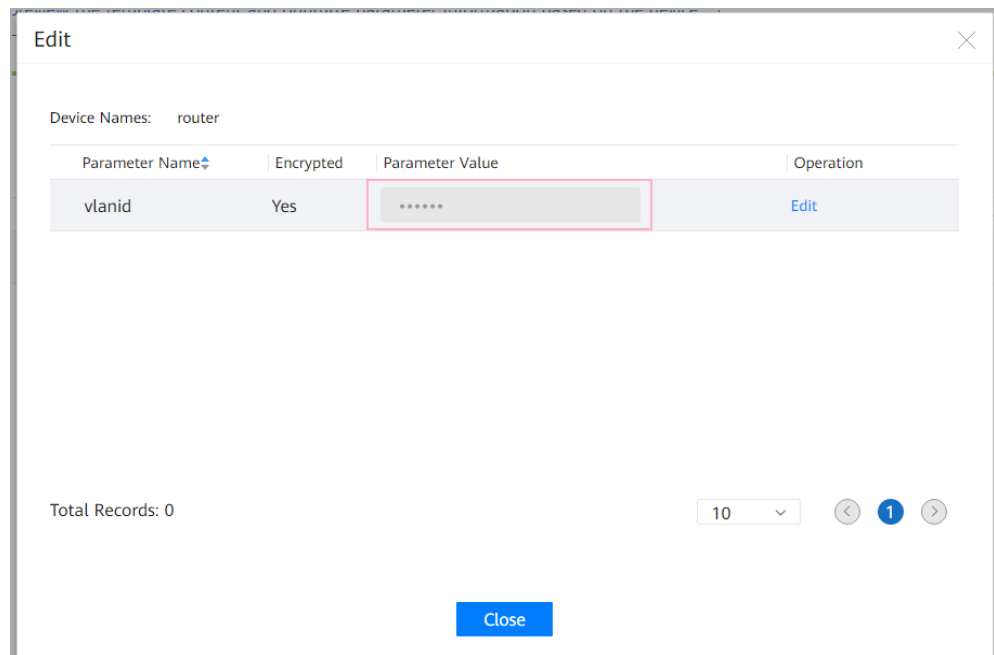
1. On the **Preview and Edit Parameters** page that is displayed, click **View** in the **Operation** column of a device.
2. View the template content in the preview dialog box that is displayed.

Figure 5-20 Viewing the template content



3. Click **Edit** in the **Operation** column of a device.
4. In the edit dialog box that is displayed, click **Edit** to modify a parameter value. After completing the modification, click **Save**.

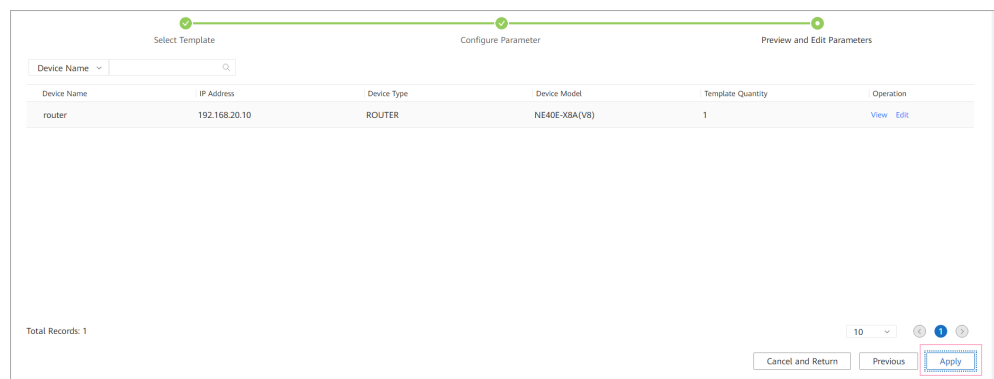
Figure 5-21 Setting parameters



Step 6 Deliver a template.

1. Click **Apply**.

Figure 5-22 Applying a template



2. In the dialog box that is displayed, click **Submit and deliver** and click **OK**.

NOTE

- To commit the configuration to the cache of NCE and deliver the configuration to the device, click **Submit and deliver**.
- Click **Submit** to commit the configuration to the NCE database without delivering the configuration to the device. In this mode, you need to manually commit the configuration.
- If there are a large number of devices with huge amounts of configurations, you are advised to select **Submit and deliver**. A maximum of 2000 devices are supported in a single task, and the task execution timeout interval is 45 minutes.

----End

5.5.2 Applying a Template to a Device Group

Procedure

- Step 1** Choose **Device Configuration > Device Group Management** from the main menu.
- Step 2** On the **Device Group Management** page that is displayed, click **Apply Template** in the **Operation** column of a device group.

Figure 5-23 Accessing the **Apply Template** page from the **Device Group Management** page

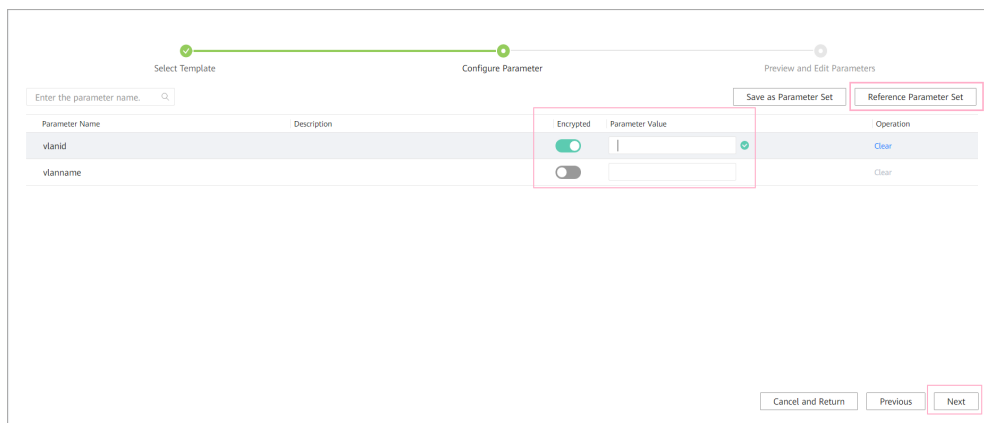


- Step 3** On the **Template** tab page, select a template in the available template area, click **>**, and click **Next**. Alternatively, click the **Template Group** tab, and select a template group in the available template group area. Then click **>** and click **Next**.
- Step 4** On the **Configure Parameter** page that is displayed, modify parameter values and click **Next**.

You can modify the parameters as follows:

- Click **Reference Parameter Set** to reference the parameter content in a parameter set.
- Enable the **Encrypted** function if a parameter needs to be displayed in ciphertext on the GUI.
- Directly enter the parameter value.

Figure 5-24 Configuring parameters



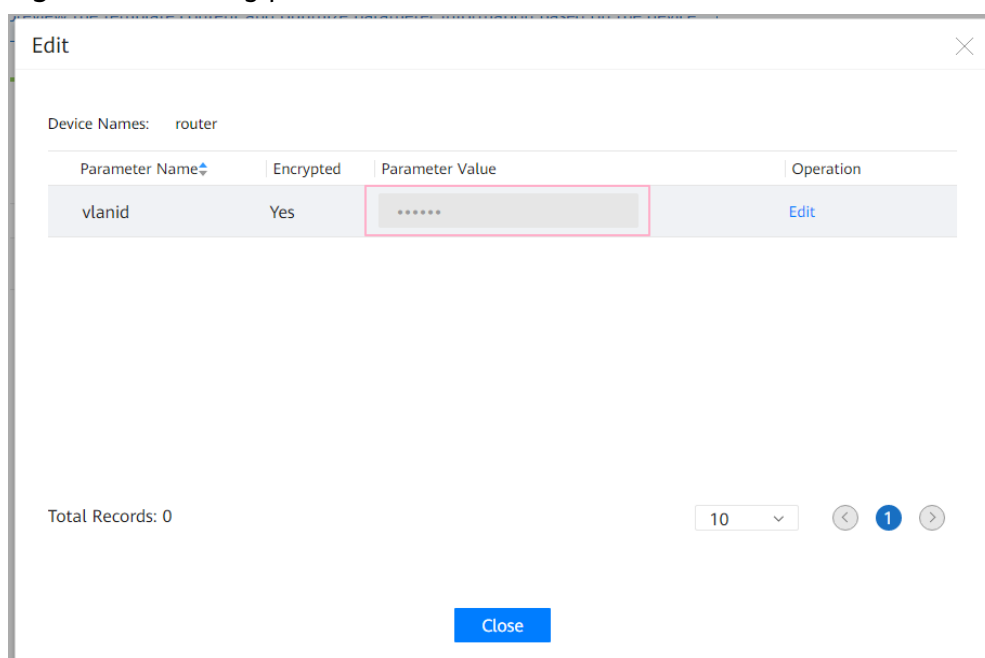
- Step 5** Preview and edit parameters.
 1. On the **Preview and Edit Parameters** page that is displayed, click **View** in the **Operation** column of a device.
 2. View the template content in the preview dialog box that is displayed.

Figure 5-25 Viewing the template content

```
Device Names:  router
1  <vlan xmlns="http://www.huawei.com/netconf/vrp/huawei-vlan">
2  <vlans>
3  <vlan>
4  <vlanId>*****</vlanId>
5  </vlan>
6  </vlans>
7  </vlan>
8
```

3. Click **Edit** in the **Operation** column of a device.
4. In the edit dialog box that is displayed, click **Edit** to modify a parameter value. After completing the modification, click **Save**.

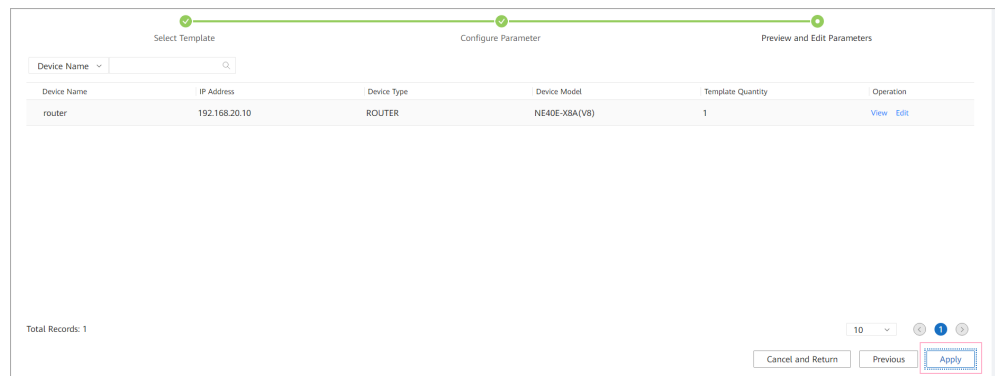
Figure 5-26 Setting parameters



Step 6 Deliver a template.

1. Click **Apply**.

Figure 5-27 Applying a template



2. In the dialog box that is displayed, click **Submit and deliver** and click **OK**.

NOTE

- To commit the configuration to the cache of NCE and deliver the configuration to the device, click **Submit and deliver**.
- Click **Submit** to commit the configuration to the NCE database without delivering the configuration to the device. In this mode, you need to manually commit the configuration.
- If there are a large number of devices with huge amounts of configurations, you are advised to select **Submit and deliver**. A maximum of 2000 devices are supported in a single task, and the task execution timeout interval is 45 minutes.

----End

5.5.3 Applying a Template to a Network-Level Service

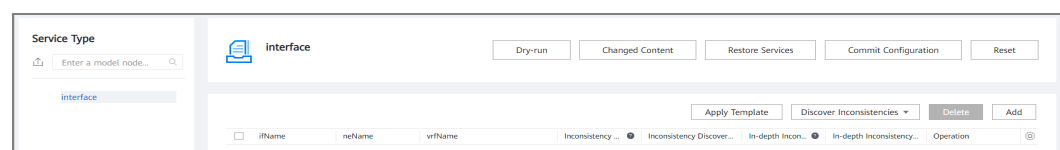
Prerequisites

- A device is properly connected to NCE.
- A template has been created according to [7.2.3 Creating a Service Template](#).
- (Optional) The parameter set required by the template has been created according to [7.4.1 Creating a Parameter Set](#).

Procedure

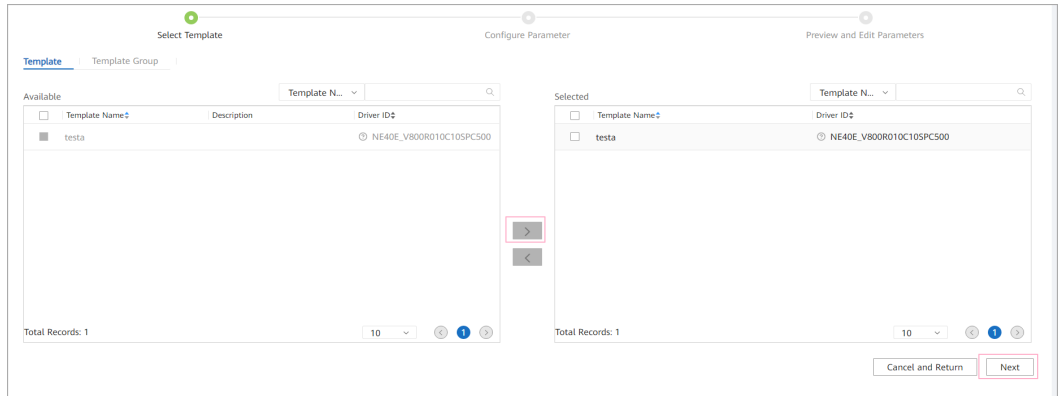
- Step 1** Choose **Service Management** from the main menu. The **Service Management** page is displayed.
- Step 2** On the **Service Management** page that is displayed, click **Apply Template** to access the **Apply Template** page.

Figure 5-28 Accessing the **Apply Template** page from the **Service Management** page



Step 3 On the **Template** tab page, select a template in the available template area, click **>**, and click **Next**. Alternatively, click the **Template Group** tab, and select a template group in the available template group area. Then click **>** and click **Next**.

Figure 5-29 Adding a template



Step 4 On the **Configure Parameter** page that is displayed, modify parameter values and click **Next**.

You can modify the parameters as follows:

- Click **Reference Parameter Set** to reference the parameter content in a parameter set.
- Toggle on **Encrypted** if a parameter needs to be displayed in ciphertext on the GUI.
- You can directly enter a parameter value.

Step 5 Preview the template.

Step 6 Click **Apply**.

----End

6 Service Configuration

This chapter describes how to configure services on multiple devices using the service management function. You can configure services on the service management page or using commands.

Prerequisites

- Devices are online.
- Service packages have been activated.

Limitations

The same command is used for the create and modify operations on the CLI. For the same command, the last configuration that is delivered is the latest configuration of the device. Therefore, you need to check whether related configurations exist on the device before configuring services. This problem also occurs on NETCONF devices that do not support the create operation.

Scenario Description

This scenario applies if you need to perform end-to-end service configurations across devices on NCE.

[6.1 Configuring Services on the Web UI](#)

[6.2 Configuring Services Using the Northbound CLI](#)

[6.3 Verifying Service Data Consistency](#)

6.1 Configuring Services on the Web UI

Context

Before delivering service configurations, NCE detects the following conflict scenarios:

Table 6-1 Service delivery conflict detection

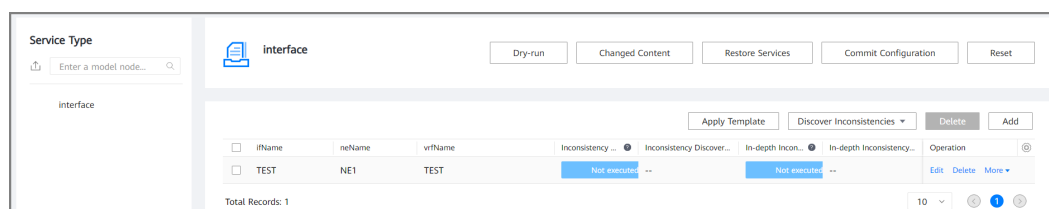
Conflict Scenario	Scenario Description	Solution
Multi-terminal operation conflict	Assume that the same service instance service 1 (x=1, y=2) is configured for both terminal 1 and terminal 2. The service instance is then edited, submitted, and updated to service 1 (x=2, y=2) for terminal 1, while terminal 2 performs operations based on the original service 1 (x=1, y=2). In this case, the system displays a message indicating a conflict when terminal 2 submits data.	<ol style="list-style-type: none"> 1. Roll back to the previous configuration. 2. Re-configure the data.
Device configuration conflict	<ul style="list-style-type: none"> • NCE manages a device for the first time and does not synchronize the device data. • NCE saves the sequence IDs of device configuration changes. (The IDs can be obtained from the device or based on the checksum for calculating the full device configuration on NCE.) Before redelivering configurations, NCE obtains the sequence IDs again. If an ID change is detected, NCE considers that the configuration changes. • When a device is disconnected from NCE during configuration delivery, NCE attempts to roll back the configurations. If the device is offline for a long time, the NCE data is inconsistent with the device data. 	<ul style="list-style-type: none"> • Commit the device configurations forcibly. In this case, NCE does not check for conflicts. • Synchronize data from the device to NCE, and edit and commit the configurations again.

Conflict Scenario	Scenario Description	Solution
Service configuration conflict	Before delivering service configurations, NCE checks whether the configurations delivered to a device conflict with each other. Assume that both service 1 and service 2 deliver the description configurations of VLAN 2 to device 1. In this case, NCE checks whether the two configurations are the same. If not, NCE displays a message indicating that a configuration conflict occurs.	<ul style="list-style-type: none"> Commit the service configurations forcibly. In this case, NCE does not check for conflicts. Delete the previous service configurations and attributes related to the service configurations to avoid conflicts with the current service configurations.

Procedure

Step 1 Choose **Service Management** from the main menu. On the **Service Management** page that is displayed, click the required service model.


Figure 6-1 Selecting a service model



NOTE

The service type information in the left pane of the **Service Management** page derives from the instances defined in the service package.

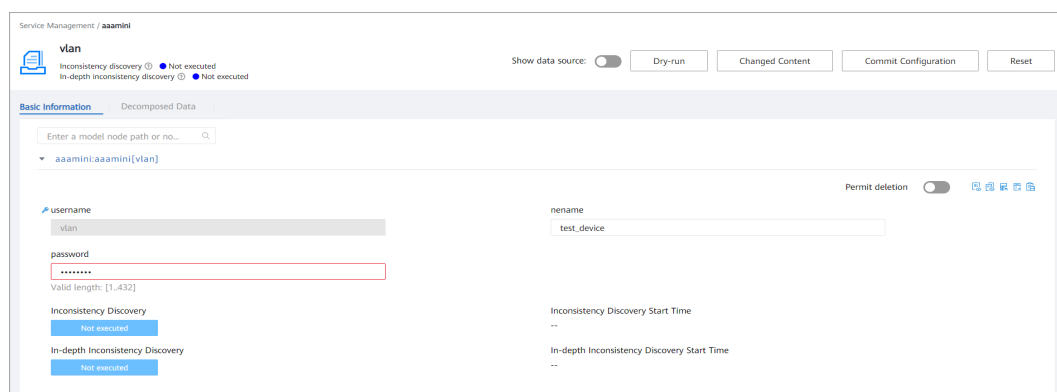
Step 2 Click **Edit** to edit the model data.

Click  to recompute the data of a service instance that is not modified.

 NOTE

- Full service recomputing: The system performs full service recomputing when configurations are created or updated.
- Minimum delivery: The system delivers the minimum configuration based on the difference between the NE configuration decomposed from the current service configuration and the NE configuration stored in the system.
- The system does not support the verification of objects that depend on each other. You need to ensure that the dependency between objects is correct.
- The system does not limit the service volume of a transaction. However, if the service volume is too large, the transaction may time out (timeout interval: 5 minutes). To avoid this, you need to control the size of configurations in a single transaction.
- When you create duplicate service or NE configuration instances, the system reports an error.
- During service decomposition, the existing configurations on the device may be merged. The verification capability for the create operation is not supported.
- Editing sequence: Only the simple editing sequence is supported, for example, configuration based on interfaces in sequence. The template content sequence in a random order is not supported. For example, the system does not support automatic sequencing when you configure interfaces out of order.

Figure 6-2 Recomputing data



 NOTE

On the **Service Management** page, you can trigger recomputing of a service instance that is not modified. When this function is used, the service instance is recomputed and converted to logic codes for delivery.

When a service instance is modified, the internal code logic is invoked to recompute and deliver configurations. Therefore, you can update or create a service instance to deliver modified configurations to devices.

In some scenarios, the service instance is not updated, but the internal code logic needs to be invoked to recompute and deliver service configurations. For example:

1. If the service processing code changes while the service instance configurations do not change, the configurations need to be recomputed and delivered again.
2. If the device configuration is modified in other modes, for example, through the CLI or synchronization on NCE, the service instance needs to be deployed again to overwrite the synchronized data.

Click  to undeploy a service instance.

 **NOTE**

On the **Service Management** page, you can trigger undeployment of a service instance. When this function is used, the service instance deletes decomposed NE configurations.

If services of devices are affected after decomposed NE configurations are deleted by the service instance and the original network configurations needs to be retained, roll back the NE configurations and wait for the configurations to be delivered again. (Recomputing must be triggered before configuration redelivery.)

Step 3 Click **Changed Content** to view service data changes.

Figure 6-3 Clicking **Changed Content**

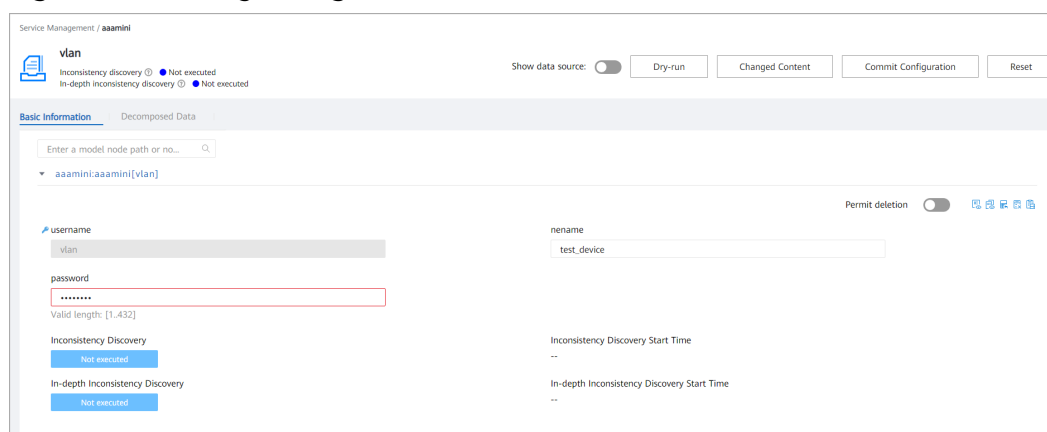
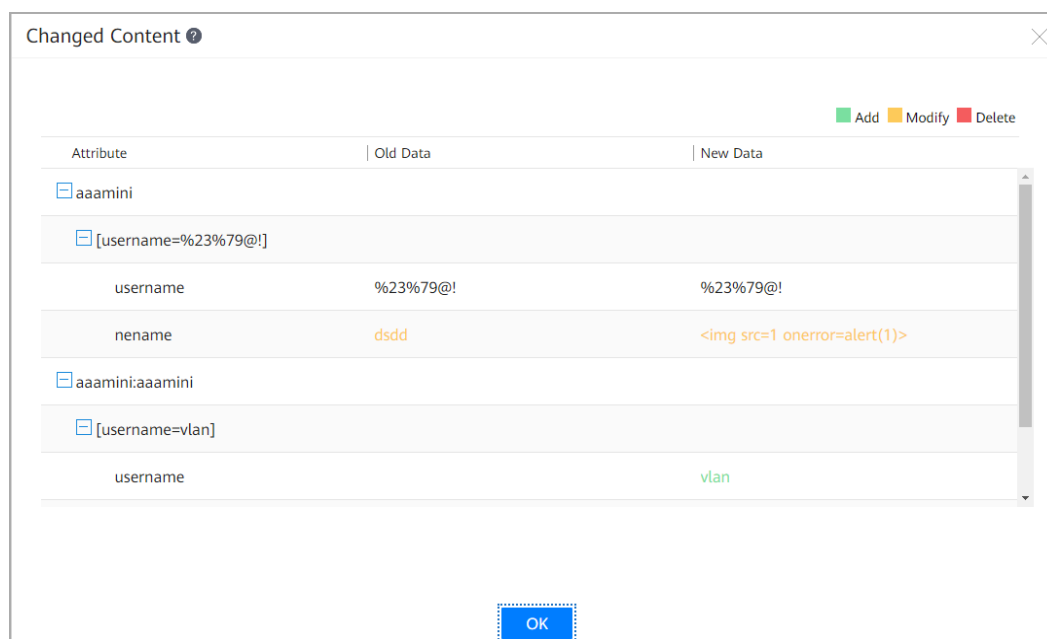
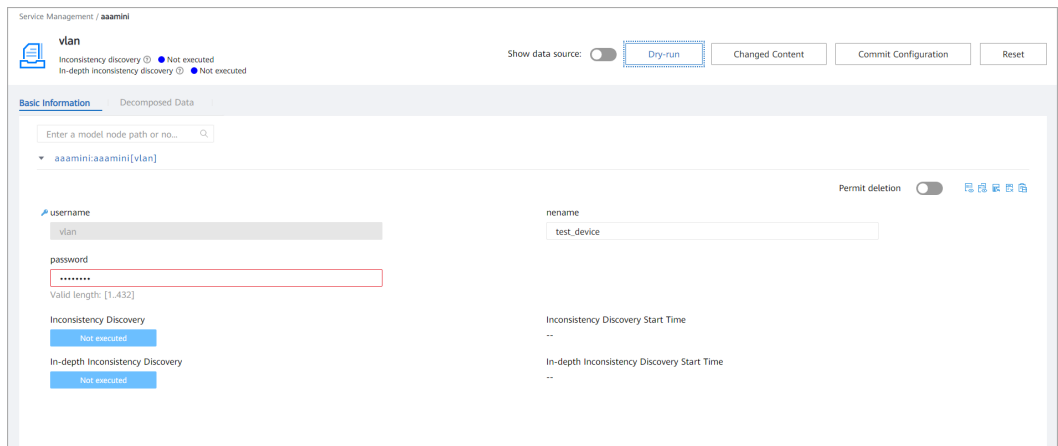


Figure 6-4 Viewing service data changes



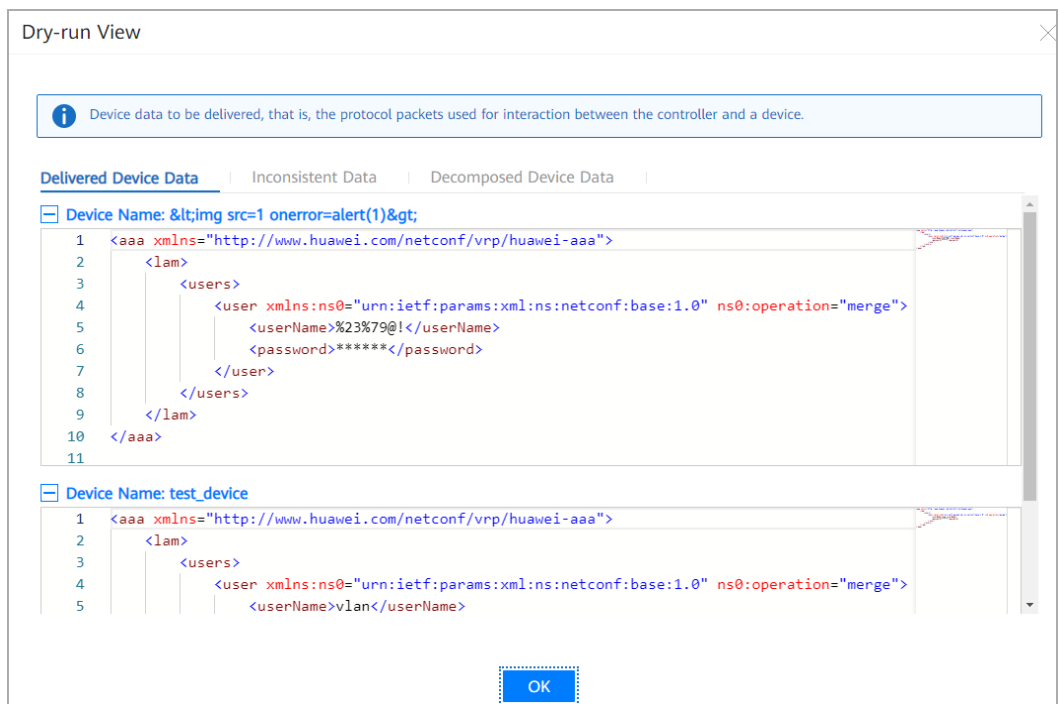
Step 4 Click **Dry-run** to check whether the configuration is correct.

Figure 6-5 Clicking Dry-run



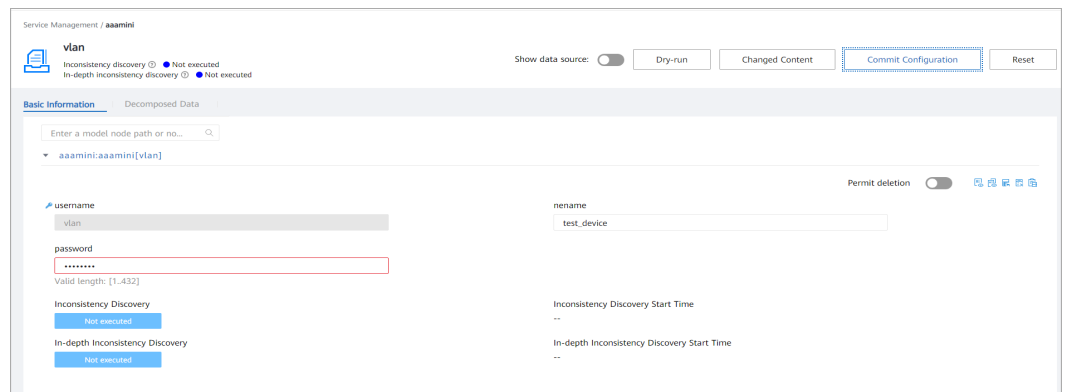
Step 5 After the dry run is complete, the **Dry-run View** page is displayed. You can view device data to be delivered and inconsistent data.

Figure 6-6 Viewing device data to be delivered on the Dry-run View page



- If the result meets expectations, click **Commit Configuration**.

Figure 6-7 Committing the configuration



NOTE

The configuration delivery timeout interval is 5 minutes. If the configuration volume is too large, configuration delivery may time out.

ACID of configuration delivery transactions: Currently, the system provides isolation capabilities only at the read committed level. For concurrent services, issues such as non-repeatable read and phantom read may occur. You are advised to control the data volume serially in the northbound API.

Before delivering configurations, you are advised to perform synchronization to ensure data consistency between the controller and devices. This prevents device configurations from being overwritten upon configuration delivery.

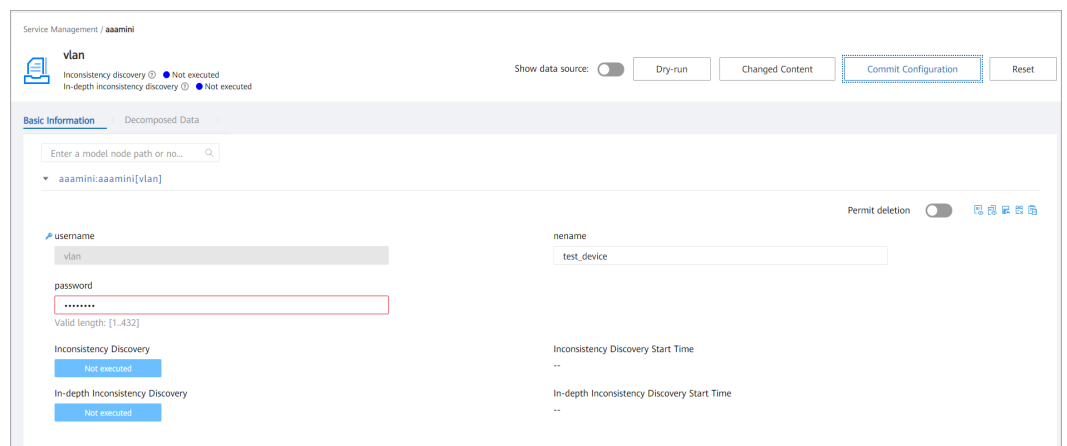
Transactions are characterized by atomicity. When some configurations in the transaction fail to be delivered, the system initiates a rollback. If the rollback fails, the error message does not contain the rollback failure information. In this case, you can view the configuration difference through data consistency verification.

No networking mode

When configurations are delivered through NCE, the configurations will be written to the NCE database and delivered to devices. When configurations are delivered in no networking mode, the configurations will only be written to the NCE database without being delivered to devices.

- If the result does not meet expectations or reconfiguration is required, click **Reset**.

Figure 6-8 Resetting the configuration



- If the configuration is not edited for a long time or the third-party package is forcibly deleted, the system displays a message indicating that the transaction has expired when you perform operations on the GUI. In this case, click **OK** to refresh the page and edit the configuration again.

NOTE

After you open a web page, the foreground applies for a transaction ID from the background. If you do not edit the page for a long time, the background will age the transaction ID. The default validity period of a transaction ID is 5 days. If the third-party package used for editing the configuration is forcibly uninstalled, the background clears the transaction ID that is being used.

If you still use the old transaction ID to edit the configuration, a dialog box is displayed, indicating that the local transaction ID has expired.

----End

6.2 Configuring Services Using the Northbound CLI

Step 1 Choose **CLI** from the main menu. The CLI is displayed.

```
Welcome to NCE CLI
[NCE]>
```

Step 2 **Optional:** Run the command to enter the service view.

```
[NCE]> applications
[app:applications]>
```

Step 3 Edit the service.

```
[app:applications]> hbng hbng_ins
[app:applications/hbng:hbng[hbng_ins]]> core_id 4033
[app:applications/hbng:hbng[hbng_ins]]> pool_soo ZBB
[app:applications/hbng:hbng[hbng_ins]/hbng:pool_soo[ZBB]]> soo 501
[app:applications/hbng:hbng[hbng_ins]/hbng:pool_soo[ZBB]]> quit
[app:applications/hbng:hbng[hbng_ins]]>
```

Step 4 Conduct a dry run for the service configuration.

```
[app:applications/hbng:hbng[hbng_ins]]> dry-run
[OK]
```

Step 5 Preview the configuration.

```
[app:applications/hbng:hbng[hbng_ins]]> display dry-run preview
### ipi-zba900-r-bn-01 ###
<system xmlns="https://www.huawei.com/netconf/vrp/huawei-system">
<systemInfo>
<sysName>Huawei1 </sysName>
</systemInfo>
</system>
[app:applications/hbng:hbng[hbng_ins]]>
```

- If the displayed result meets expectations, commit the configuration.
[NCE]>commit
- To delete the configuration, reset the configuration.
[NCE]> reset

----End

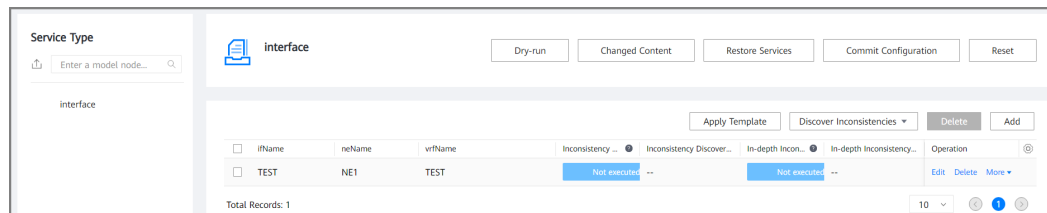
6.3 Verifying Service Data Consistency

6.3.1 Performing Service Inconsistency Discovery

Procedure

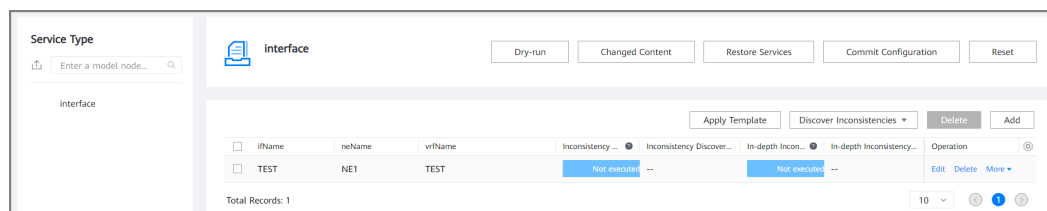
- Step 1** Choose **Service Management** from the main menu. On the **Service management** page that is displayed, select a service instance. Then click **Discover Inconsistencies** and select **Discover Inconsistencies**.

Figure 6-9 Triggering inconsistency discovery



- Step 2** After service inconsistency discovery is complete, the value of **Inconsistency Discovery Status** changes to **Succeeded (inconsistencies found)**.

Figure 6-10 Inconsistency discovery completed



- Step 3** The southbound device list is displayed on the left of the service inconsistency display page.

Select a service instance, click **More**, and select **View Inconsistencies**. The detailed inconsistent data is displayed on the right of the service inconsistency display page. The data in the **Old Data** column indicates the southbound device configurations saved in the NCE database, and the data in the **New Data** column indicates the southbound device configurations generated after service instances are recomputed.

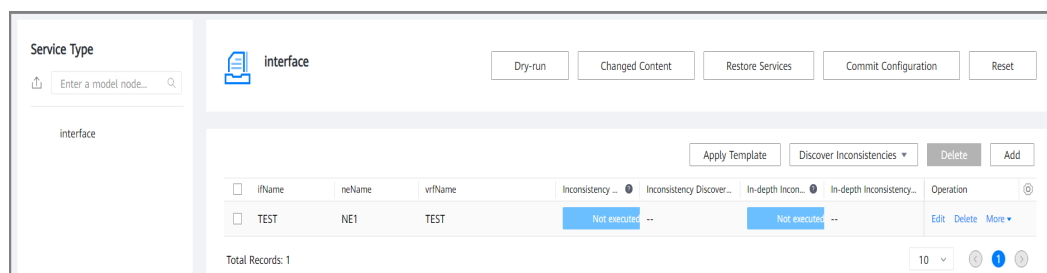
----End

6.3.2 Performing In-Depth Service Inconsistency Discovery

Procedure

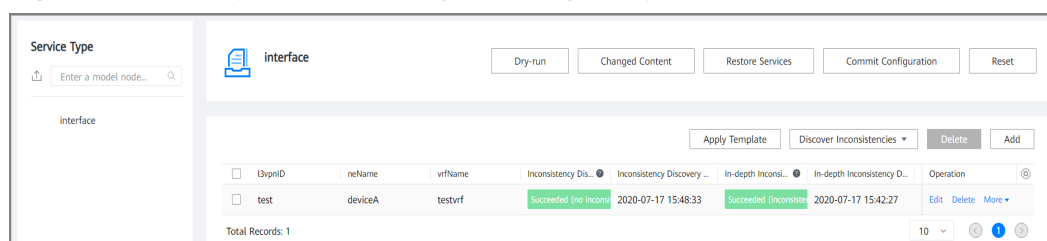
- Step 1** Choose **Service Management** from the main menu. On the **Service management** page that is displayed, select a service instance. Then click **Discover Inconsistencies** and select **Discover Inconsistencies (In-depth)**.

Figure 6-11 Triggering in-depth inconsistency discovery



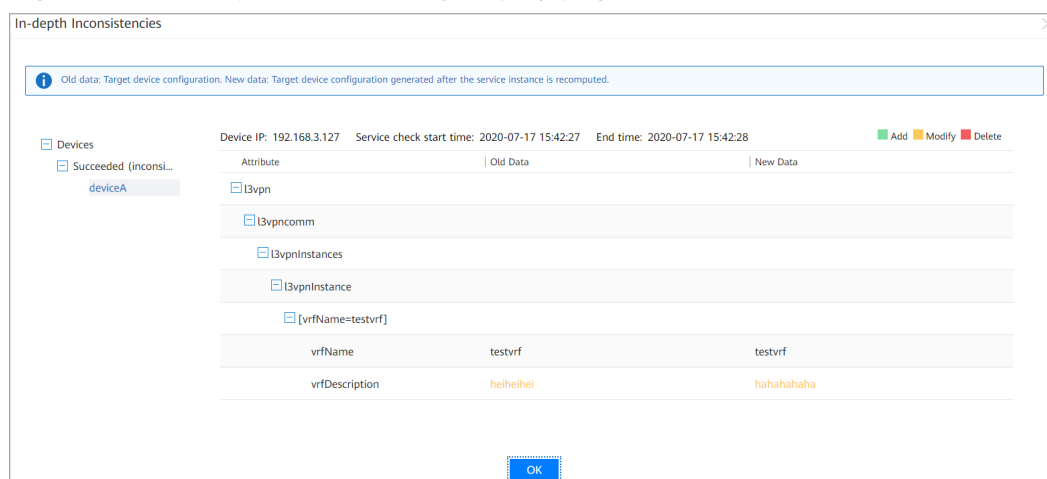
Step 2 After service inconsistency discovery is complete, the value of **In-depth Inconsistency Discovery Status** changes to **Succeeded (inconsistencies found)**.

Figure 6-12 In-depth inconsistency discovery completed



Step 3 Choose **Service Management** from the main menu. On the **Service Management** page that is displayed, choose **More > View In-depth Inconsistencies**. The southbound device list is displayed in the left pane of the in-depth service inconsistency display page, and the in-depth service inconsistency result is displayed on the right. The following figure shows an example, which indicates that **vrfDescription** of the L3VPN service configured on device A changes.

Figure 6-13 In-depth inconsistency display page



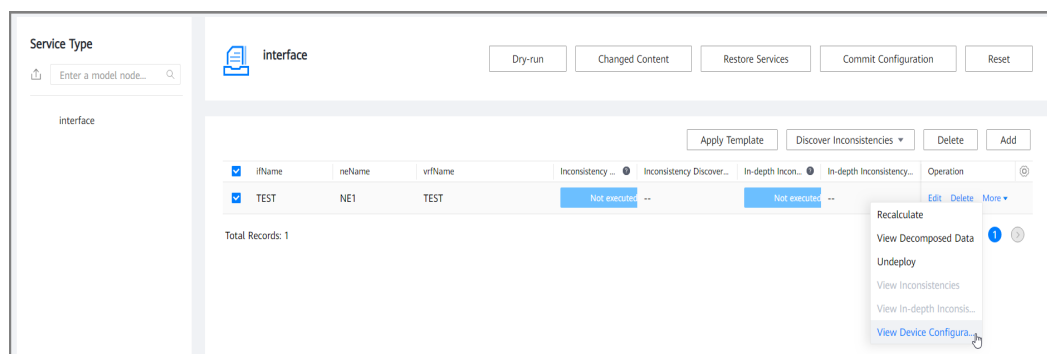
----End

6.3.3 Checking Southbound Configuration Records of Service Instances

Procedure

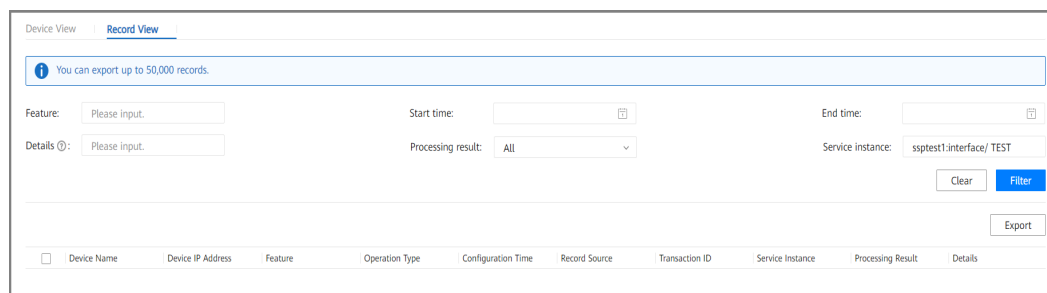
- Step 1** Choose **Service Management** from the main menu. On the **Service Management** page that is displayed, click **More** and select **View Device Configuration History**. The **Record View** tab page is displayed.

Figure 6-14 Record View tab page



- Step 2** On the **Record View** tab page, the configuration records of the service instance are automatically filtered and displayed.

Figure 6-15 Device configuration record management page



----End

7 Template Management

Scenario Description

Template management allows you to manage NE templates and service templates. To deliver configurations, select a template from available templates and orchestrate as well as deliver the template.

[7.1 Managing NE Templates](#)

[7.2 Managing Service Templates](#)

[7.3 Managing Template Groups](#)

[7.4 Managing Parameter Sets](#)

7.1 Managing NE Templates

Scenario Description

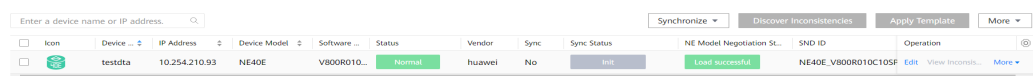
You can use the NE template management function to manage device-level templates.

7.1.1 Exporting an NE Template Based on a YANG File

Procedure

- Step 1** Choose **Device Configuration** > **Device Configuration** from the main menu.
- Step 2** On the **Device Configuration** page that is displayed, click **Edit** in the **Operation** column of a device to access the **Configuration Information** page.

Figure 7-1 Editing device information



Icon	Device	IP Address	Device Model	Software	Status	Vendor	Sync	Sync Status	NE Model Negotiation St.	SND ID	Operation
	testdta	10.254.210.93	NE40E	V800R010...	Normal	huawei	No	Init	Load successful	NE40E_V800R010C10SF	Edit View Incons... More


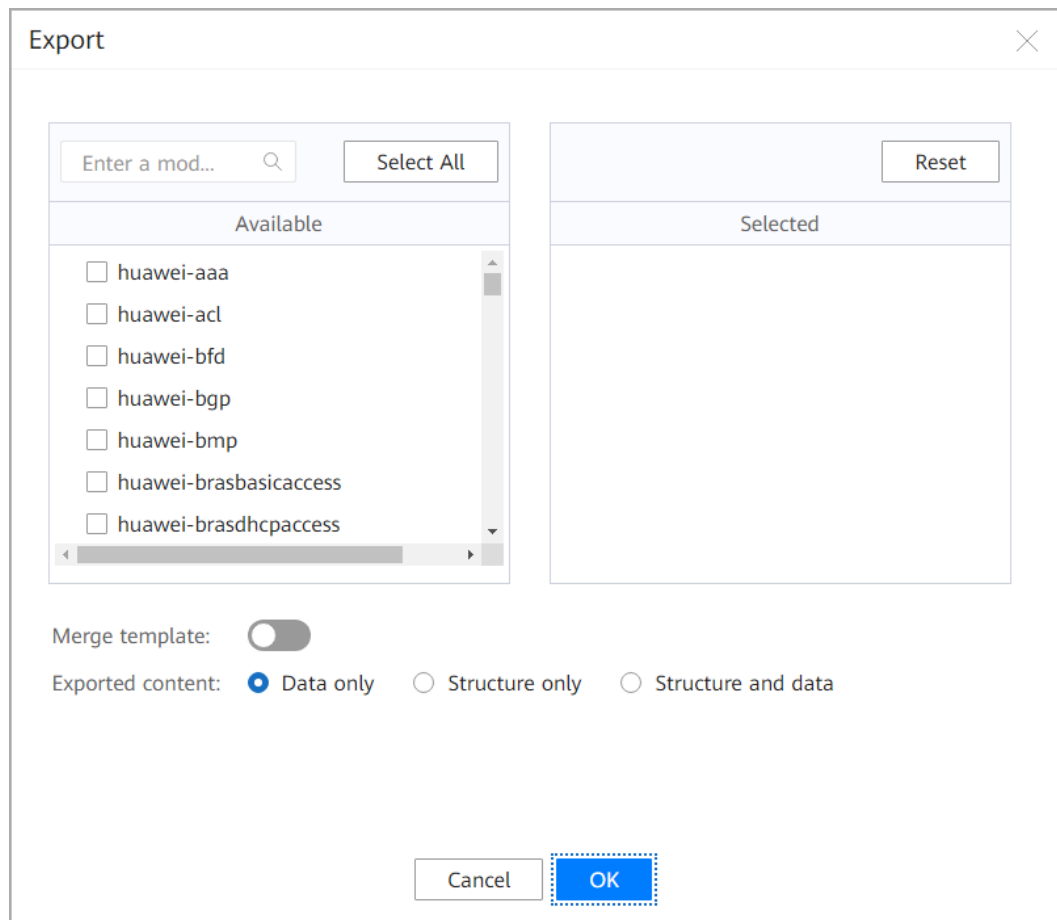
- Step 3** On the **Configuration Information** page that is displayed, click  in the upper left corner.

Figure 7-2 Exporting the YANG file



Step 4 In the **Export** dialog box that is displayed, select the YANG file to be exported as a template, set **Merge template** and **Exported content**, and click **OK**.

NOTE

- You can enable **Merge template** to merge the template content generated by multiple YANG files.
- A maximum of 100 instances can be exported under the same module.

----End

7.1.2 Importing an NE Template

Prerequisites

An NE template has been successfully exported according to [7.1.1 Exporting an NE Template Based on a YANG File](#).

Procedure

Step 1 Edit the template exported based on a YANG file. The following example describes the SNMP configuration. Delete the unnecessary objects and define the objects that require parameter transfer as parameters.

NOTE

The defined parameter name must be enclosed in {{}}.

Figure 7-3 Edited template

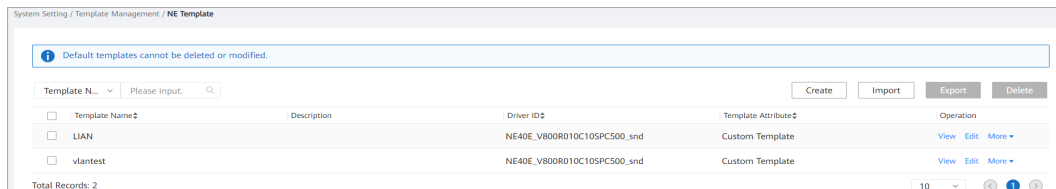
```

1 <snmp xmlns="http://www.huawei.com/netconf/vrp/huawei-snmp">
2   <agentCfg>
3     <agentEnable>true</agentEnable>
4   </agentCfg>
5   <engine>
6     <version>{{version}}</version>
7   </engine>
8 </snmp>
9

```

Step 2 Choose **System Setting > Template Management > NE Template** from the main menu. On the **NE Template** page that is displayed, click **Import**.

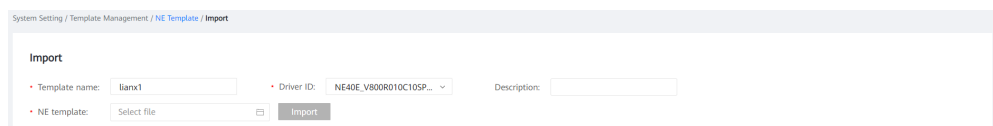
Figure 7-4 NE template import page



Step 3 In the displayed dialog box, set the template name and import a template file.

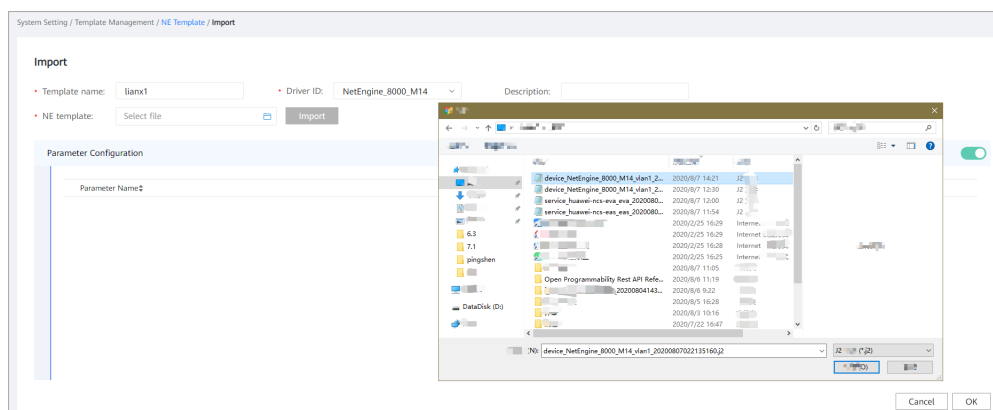
1. Set **Template name**, **Driver ID**, **Description**, and **NE template**.

Figure 7-5 Setting the template name, driver ID, description, and NE template



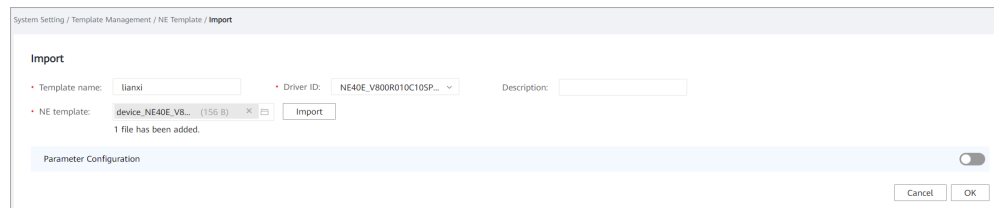
2. Click , select the edited template, and click **Open**.

Figure 7-6 Selecting a file



3. Click **Import**.

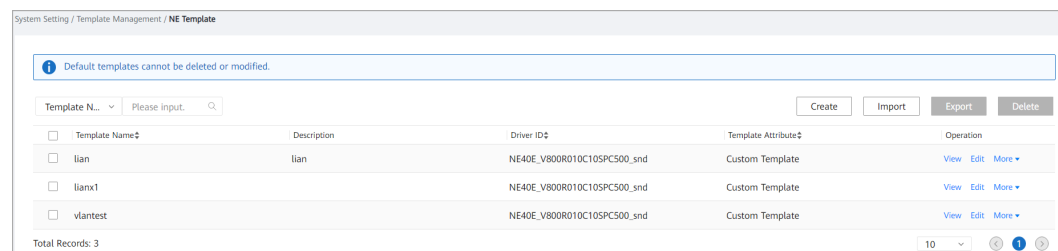
Figure 7-7 Uploading an NE template



4. Click **OK** after the template is imported.

Step 4 View the template list. The imported template is displayed in the template list.

Figure 7-8 Viewing the imported template



----End

7.1.3 Creating an NE Template

Procedure

Step 1 Edit the template content. The following example describes the VLAN configuration. Define the objects that require parameter transfer as parameters.

NOTE

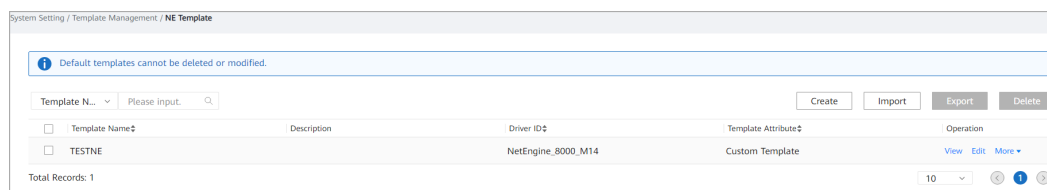
The defined parameter name must be enclosed in `{{}}`.

Figure 7-9 Edited template



Step 2 Choose **System Setting > Template Management > NE Template** from the main menu.

Figure 7-10 Creating an NE Template

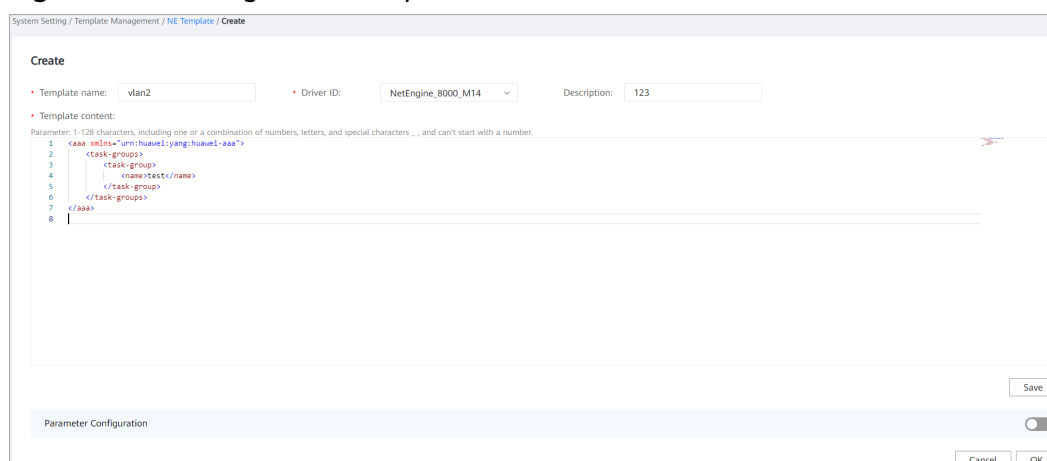


Step 3 Click **Create**. In the displayed dialog box, set **Template name**, **Description**, **Driver ID** and **Template content**, and click **OK**.

NOTE

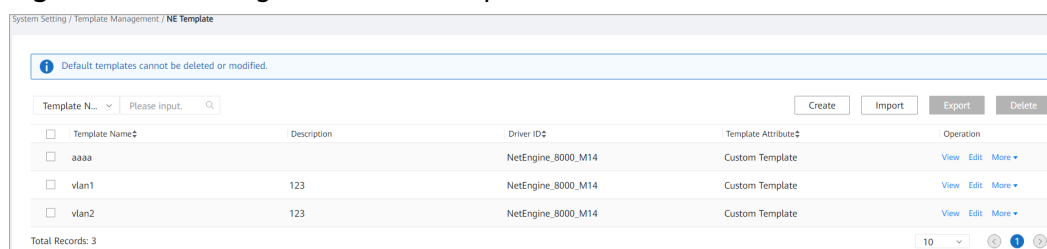
Template content sequence: Only the simple template content sequence is supported, for example, configuration based on interfaces in sequence. The template content sequence in a random order is not supported. For example, the system does not support automatic sequencing when you configure interfaces out of order.

Figure 7-11 Editing an NE template



Step 4 View the template list. The created template is displayed in the template list.

Figure 7-12 Viewing the created template



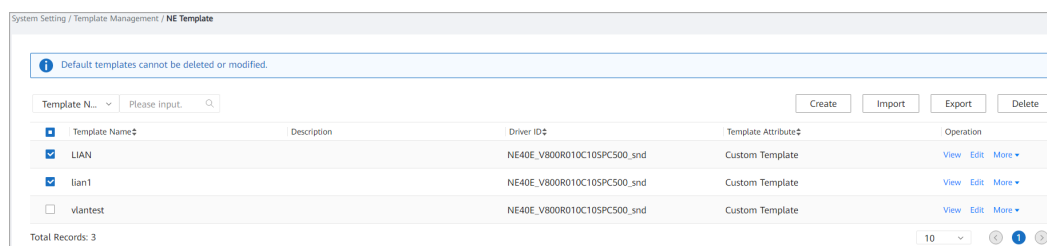
----End

7.1.4 Deleting an NE Template

Procedure

Step 1 Choose **System Setting > Template Management > NE Template** from the main menu. On the **NE Template** page that is displayed, select the templates to be deleted and click **Delete**.

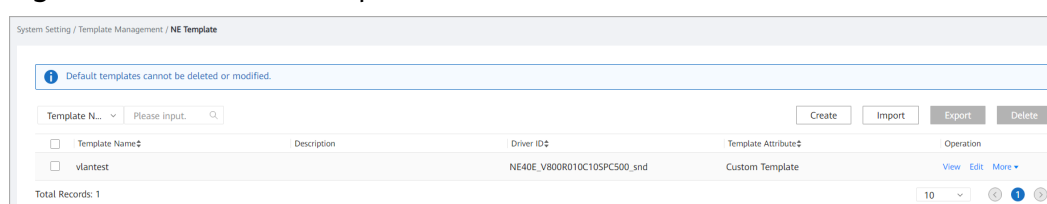
Figure 7-13 Deleting NE templates in batches



Step 2 In the **Warning** dialog box, click **OK**.

Step 3 View the template list. The templates have been deleted.

Figure 7-14 Successful template deletion



----End

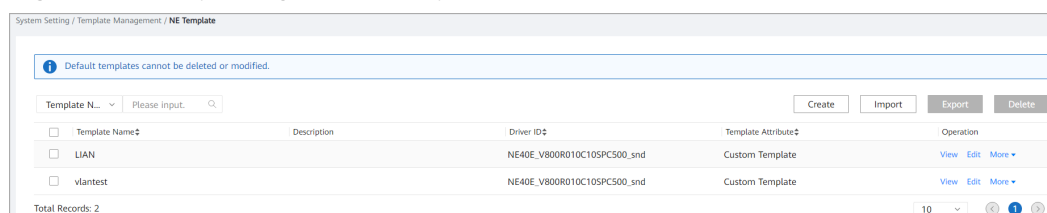
7.1.5 Exporting an NE Template

Procedure

Step 1 Choose **System Setting > Template Management > NE Template** from the main menu.

Step 2 On the **NE Template** page that is displayed, select the template to be exported and click **Export**.

Figure 7-15 Exporting an NE template



----End


7.2 Managing Service Templates

Scenario Description

You can use the service template management function to manage network-level templates.

7.2.1 Exporting a Service Template Based on a YANG File

Procedure

- Step 1** Choose **Service Management** from the main menu. The **Service Management** page is displayed.
- Step 2** On the **Service Management** page that is displayed, click  and select the YANG file to be exported as a template.
- Step 3** In the **Export** dialog box that is displayed, set **Merge template** and **Exported content**, and click **OK**.

NOTE

You can enable **Merge template** to merge the template content generated by multiple YANG files.

----End

7.2.2 Importing a Service Template

Prerequisites

A service template has been exported according to [7.2.1 Exporting a Service Template Based on a YANG File](#).

Procedure

- Step 1** Edit the template exported based on a YANG file. The following example describes the VLAN mapping configuration. Delete the unnecessary objects and define the objects that require parameter transfer as parameters.

NOTE

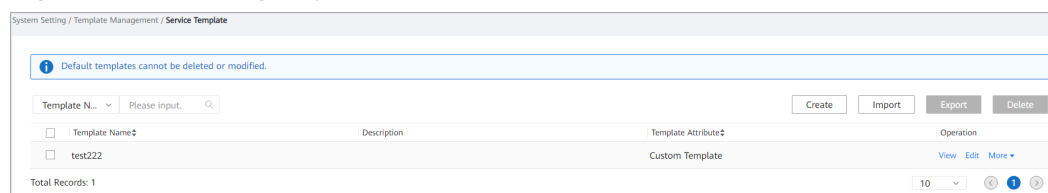
The defined parameter name must be enclosed in {{}}.

Figure 7-16 Edited template

```
<vlan_mapping xmlns="http://example.com/vlan_mapping">
  <pe_id>{{pe_id}}</pe_id>
  <trunk_id>{{trunk_id}}</trunk_id>
  <mapping_id>{{mapping_id}}</mapping_id>
</vlan_mapping>
```

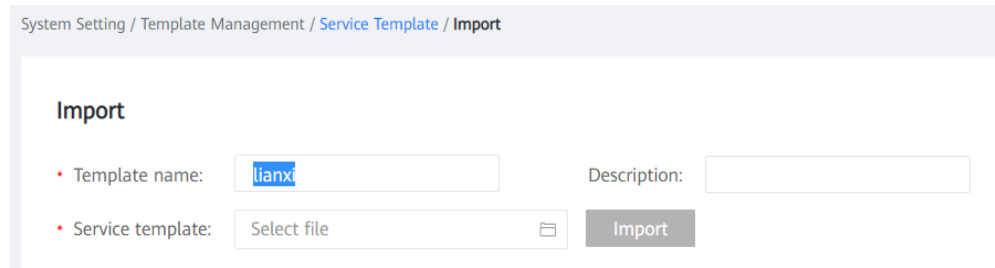
- Step 2** Choose **System Setting > Template Management > Service Template** from the main menu. On the **Service Template** page that is displayed, click **Import**.

Figure 7-17 Clicking Import



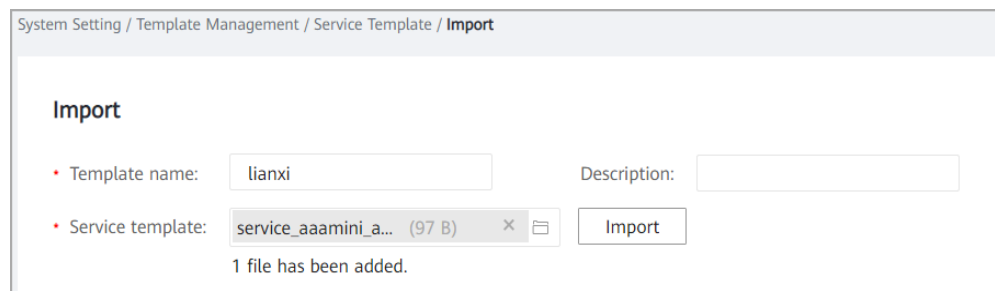
- Step 3** In the displayed dialog box, set the template name and import a template file.
1. Set **Template name**, **Description**, and **Service template**.

Figure 7-18 Setting **Template name**, **Description**, and **Service template**



2. Click , select the edited template, and click **Import**.

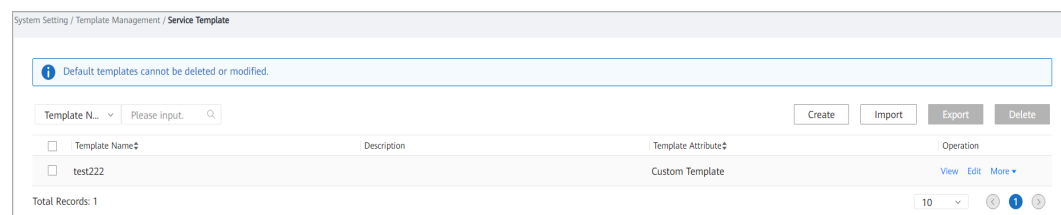
Figure 7-19 Importing a service template



3. Click **OK** after the template is imported.

- Step 4** View the template list. The imported template is displayed in the template list.

Figure 7-20 Viewing the imported service template



----End

7.2.3 Creating a Service Template

Procedure

- Step 1** Edit the template content. The following example describes the configuration of bngVrfsPairs. Define the objects that require parameter transfer as parameters.

 **NOTE**

The defined parameter name must be enclosed in `{{}}`.

Figure 7-21 Edited template

```
<bngVrfsPairs xmlns="http://swisscom.com/bngVrfsPairs">  
  <bngVrfsPairsName>{{bngVrfsPairsName}}</bngVrfsPairsName>  
  <bngServiceName>{{bngServiceName}}</bngServiceName>  
  <pairName>{{pairName}}</pairName>  
</bngVrfsPairs>
```

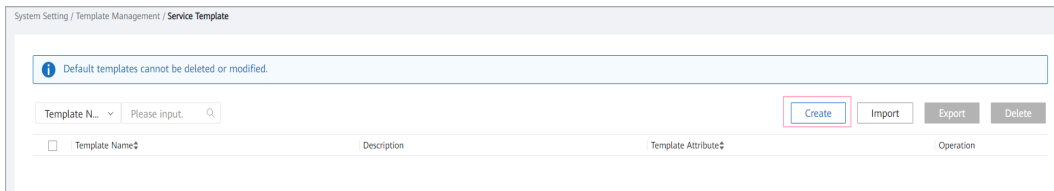
Step 2 Choose **System Setting > Template Management > Service Template** from the main menu.

Step 3 On the **Service Template** page that is displayed, click **Create**.

 **NOTE**

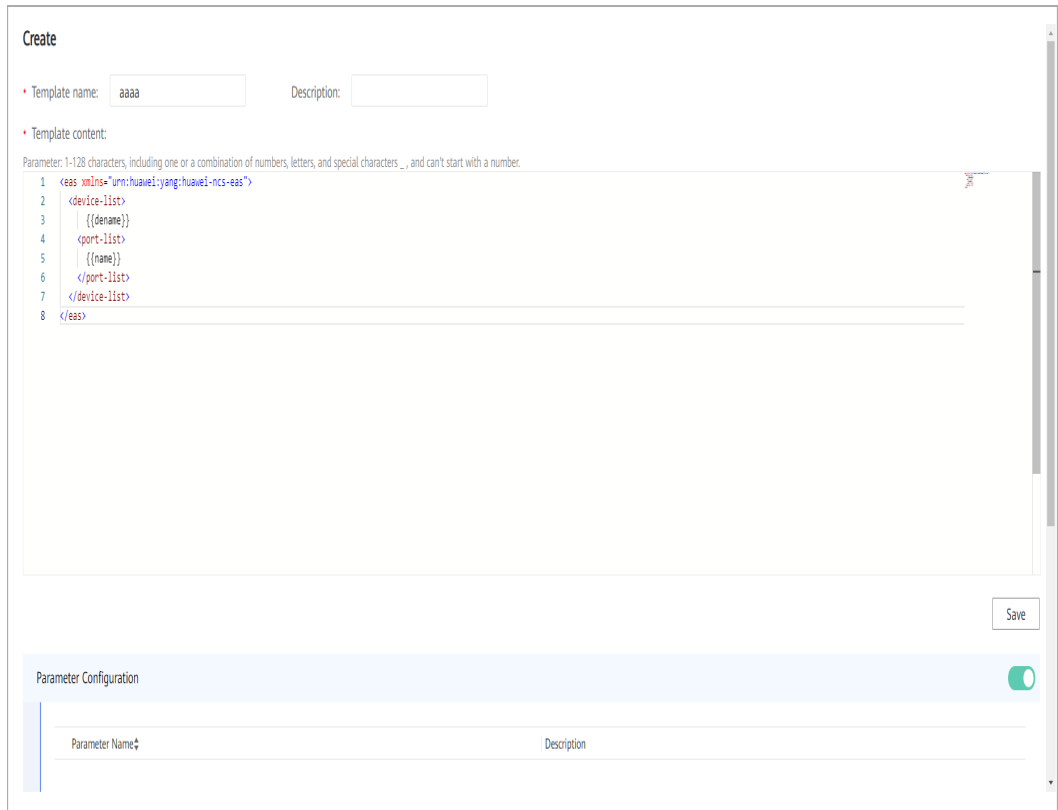
Template content sequence: Only the simple template content sequence is supported, for example, configuration based on interfaces in sequence. The template content sequence in a random order is not supported. For example, the system does not support automatic sequencing when you configure interfaces out of order.

Figure 7-22 Creating a template



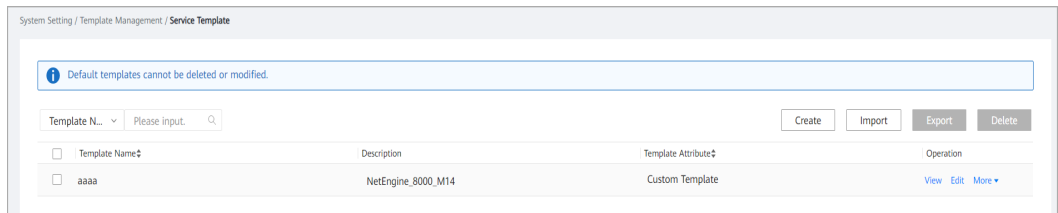
Step 4 In the displayed dialog box, set **Template name**, **Description**, and **Template content**, and click **OK**.

Figure 7-23 Editing a template



Step 5 View the template list. The created template is displayed in the template list.

Figure 7-24 Viewing the created template



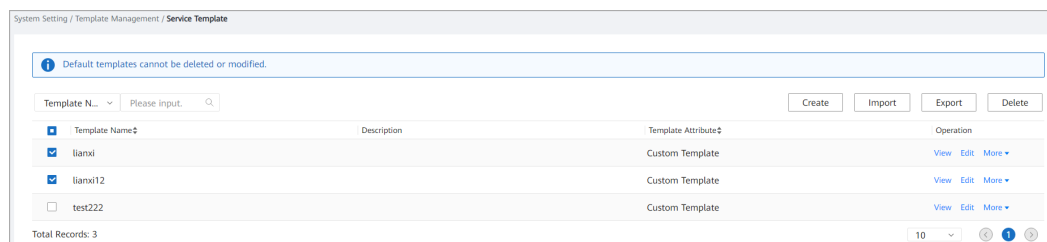
----End

7.2.4 Deleting a Service Template

Procedure

- Step 1** Choose **System Setting > Template Management > Service Template** from the main menu.
- Step 2** On the **Service Template** page that is displayed, select the templates to be deleted and click **Delete**.

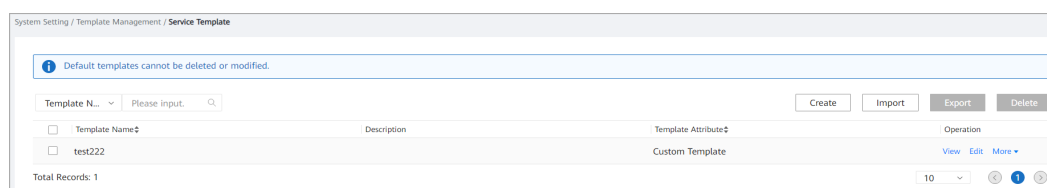
Figure 7-25 Deleting service templates in batches



Step 3 In the **Warning** dialog box, click **OK**.

Step 4 View the template list. The templates have been deleted.

Figure 7-26 Successful template deletion



----End

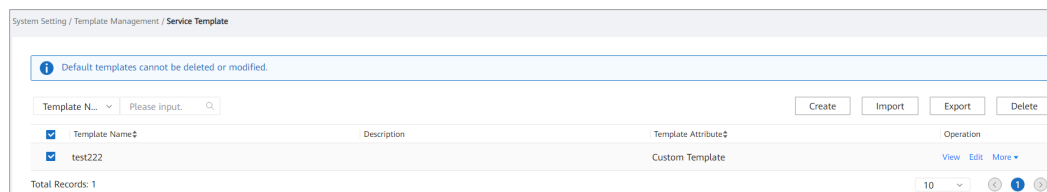
7.2.5 Exporting a Service Template

Procedure

Step 1 Choose **System Setting** > **Template Management** > **Service Template** from the main menu.

Step 2 On the **Service Template** page that is displayed, select the template to be exported and click **Export**.

Figure 7-27 Exporting a service template



----End

7.3 Managing Template Groups

Scenario Description

A template group is a collection of service configurations. You can use template groups for preliminary service orchestration.

7.3.1 Creating a Template Group

Procedure

- Step 1** Choose **System Setting > Template Management > Template Group Management** from the main menu.
- Step 2** On the **Template Group Management** page that is displayed, click **Create**.

Figure 7-28 Template Group Management page

Template Group Name	Template Group Type	Description	Template Quantity	Operation
a123	Service template group		1	Edit Delete Export
t123	NE template group		1	Edit Delete Export
t222	NE template group		0	Edit Delete Export

NOTE

There are two types of template groups: NE template group and service template group. The following example describes how to create a service template group.

- Step 3** On the **Create** page that is displayed, set **Template group name**, **Template group type**, and **Description**, and click **Create**.

Figure 7-29 Creating a template group

Create

• Template group name: liany2

• Template group type: Service template group

Description:

Create

- Step 4** Under **Add template**, click **Add**.

Figure 7-30 Adding a template

Edit

1 • Template group name: liany2

• Template group type: Service template group

Description:

2 Add template

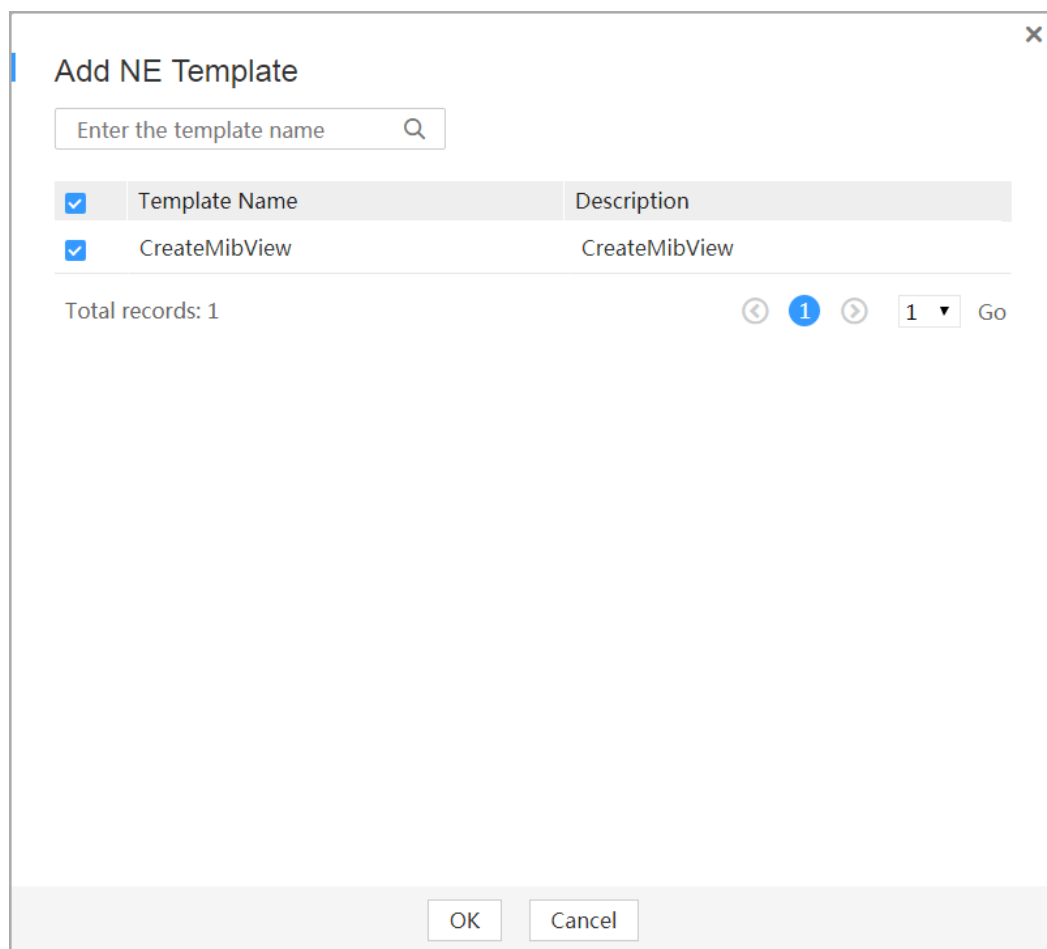
Template N... Please input.

Template Name	Description	Operation
---------------	-------------	-----------

Add Delete

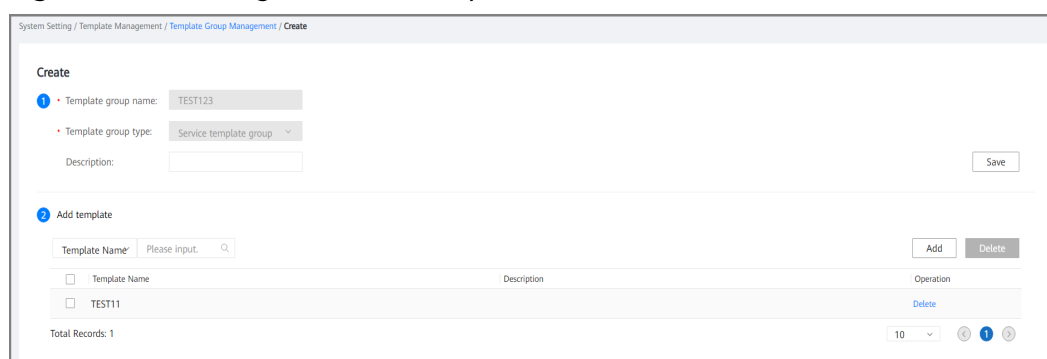
- Step 5** Under **Add template**, select the template to be added and click **OK**.

Figure 7-31 Adding a service template



Step 6 Under **Add template**, you can view the added template.

Figure 7-32 Viewing the added template



Step 7 Click **Template Group** on the menu to return to the **Template Group** page and view the created template group.

----End

7.3.2 Deleting a Template Group

Procedure

 **NOTE**

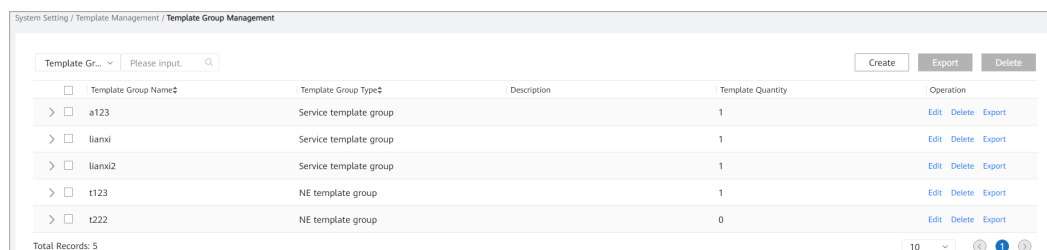
- To delete a template group, click **Delete** in the **Operation** column.
- To delete template groups in batches, select them and click **Delete**.

The following example deletes template groups in batches.

Step 1 Choose **System Setting > Template Management > Template Group Management** from the main menu. The **Template Group Management** page is displayed.

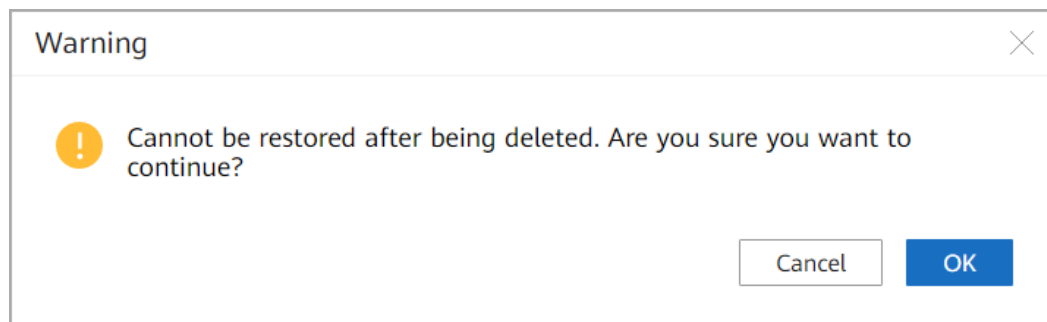
Step 2 On the **Template Group Management** page that is displayed, select the template groups to be deleted and click **Delete**.

Figure 7-33 Deleting template groups



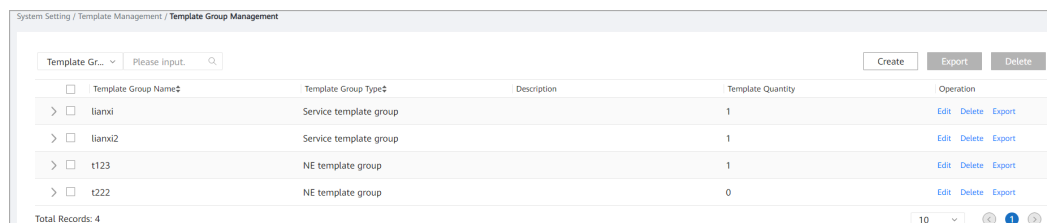
Step 3 In the displayed dialog box, click **OK**.

Figure 7-34 Warning message displayed after the deletion of template groups



Step 4 View the template group list. The template groups have been deleted.

Figure 7-35 Successful deletion of template groups



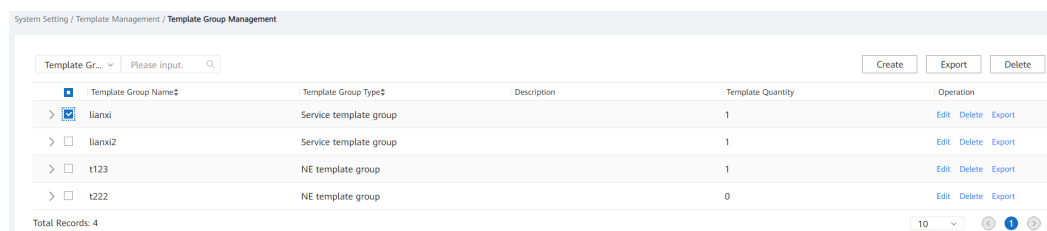
----End

7.3.3 Exporting a Template Group

Procedure

- Step 1** Choose **System Setting > Template Management > Template Group Management** from the main menu.
- Step 2** On the **Template Group Management** page that is displayed, select the template group to be exported and click **Export**.

Figure 7-36 Exporting a template group



----End

7.4 Managing Parameter Sets

Scenario Description

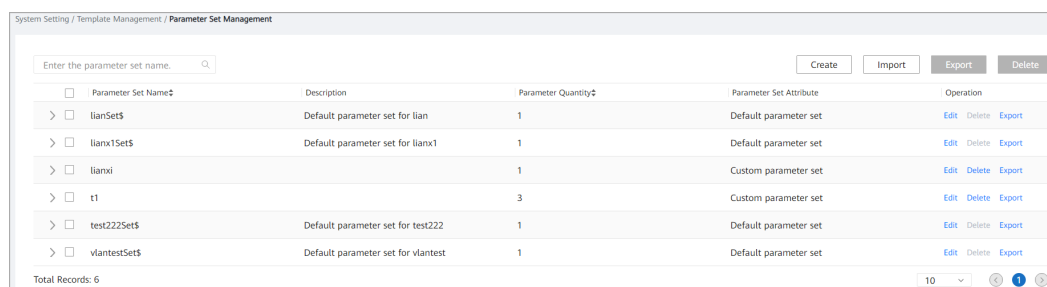
You can use the parameter set management function to manage parameters defined in templates. You can preconfigure parameter sets and directly select a template and the corresponding parameter set for delivery.

7.4.1 Creating a Parameter Set

Procedure

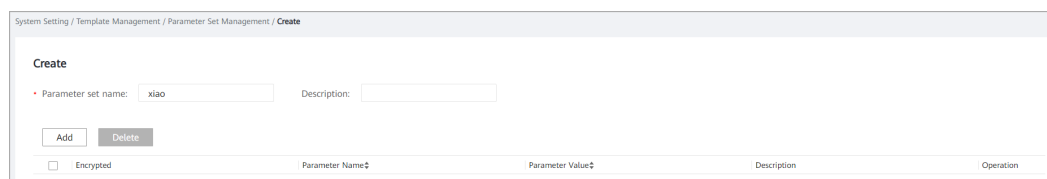
- Step 1** Choose **System Setting > Template Management > Parameter Set Management** from the main menu.
- Step 2** On the **Parameter Set Management** page that is displayed, click **Create**.

Figure 7-37 Creating a parameter set



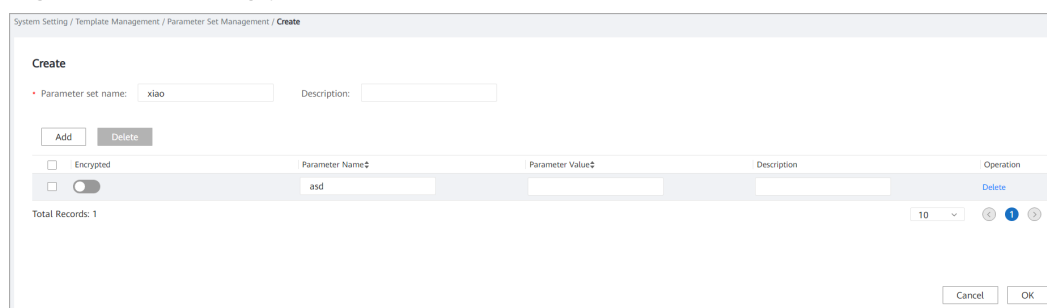
- Step 3** On the **Create** page that is displayed, set **Parameter set name** and click **Add**.

Figure 7-38 Setting Parameter set name



Step 4 In the displayed list, set the parameters and click **OK**. On the **Parameter Set Management** page that is displayed, you can view the created parameter set.

Figure 7-39 Setting parameters



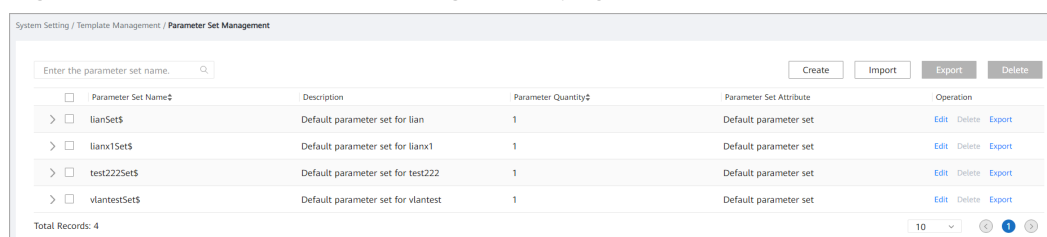
----End

7.4.2 Importing a Parameter Set

Procedure

- Step 1** Choose **System Setting > Template Management > Parameter Set Management** from the main menu.
- Step 2** On the **Parameter Set Management** page that is displayed, click **Import**.

Figure 7-40 Parameter Set Management page



Step 3 In the **Import** dialog box that is displayed, set **Parameter set name**.

Figure 7-41 Setting Parameter set name

Import

* Parameter set name:

Description:

Download template: [Template.xls](#)

Upload parameter list:

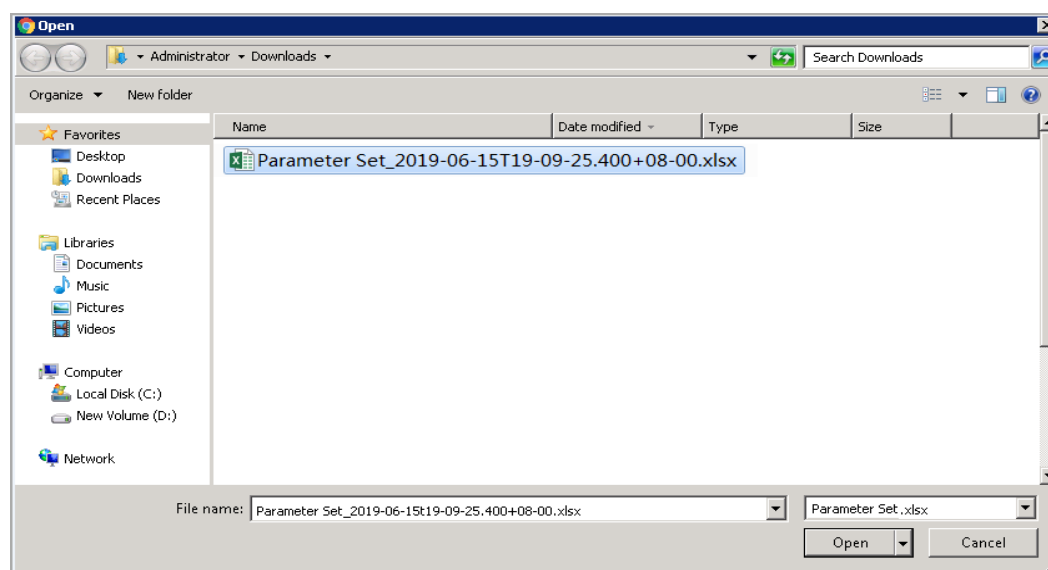
NOTE

Two file sources are available for setting **Upload parameter list**:

1. In the **Import** dialog box, download the parameter template, then edit and save it as a new parameter template.
2. Use the parameter template exported in [7.4.4 Exporting a Parameter Set](#).

Step 4 Click and select the edited template.

Figure 7-42 Selecting a file



Step 5 Click **Upload**. Then click **OK** after the template is uploaded.

Figure 7-43 Uploading a parameter set

----End

7.4.3 Deleting a Parameter Set

Context

You can delete parameter sets using either of the following methods:

- Click **Delete** in the **Operation** column to delete a specific parameter set.
- Select multiple parameter sets and click **Delete** to delete these parameter sets in a batch.

Procedure

- Step 1** Choose **System Setting > Template Management > Parameter Set Management** from the main menu.

Figure 7-44 Parameter Set Management page

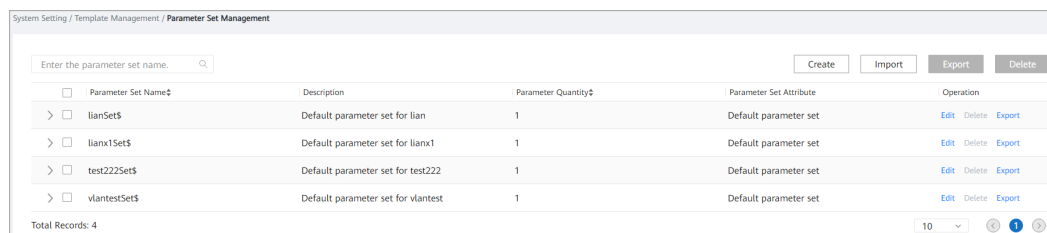
Parameter Set Name	Description	Parameter Quantity	Parameter Set Attribute	Operation
liantest	Default parameter set for lian	1	Default parameter set	Edit Delete Export
liantest1	Default parameter set for lianx1	1	Default parameter set	Edit Delete Export
t1	Default parameter set for test222	3	Custom parameter set	Edit Delete Export
test222test	Default parameter set for test222	1	Default parameter set	Edit Delete Export
vlanTest	Default parameter set for vlantest	1	Default parameter set	Edit Delete Export

- Step 2** On the **Parameter Set Management** page that is displayed, select the parameter set to be deleted and click **Delete**.

- Step 3** In the **Warning** dialog box, click **OK**.

Step 4 View the parameter set list. The parameter sets have been deleted.

Figure 7-45 Successful deletion of parameter sets



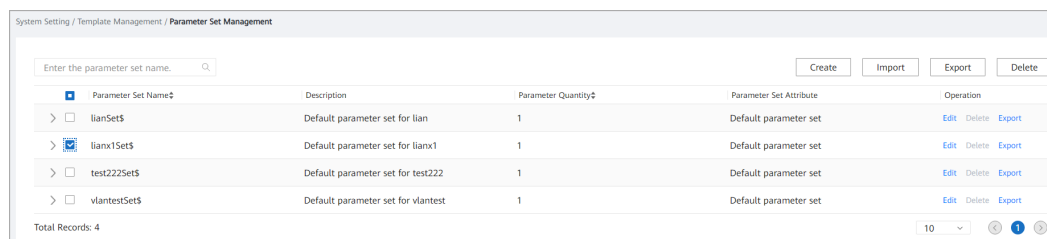
----End

7.4.4 Exporting a Parameter Set

Procedure

- Step 1** Choose **System Setting > Template Management > Parameter Set Management** from the main menu.
- Step 2** On the **Parameter Set Management** page that is displayed, select the parameter set to be exported and click **Export**.

Figure 7-46 Exporting a parameter set



----End

8 Configuration Management

Rollback refers to the process of performing reverse operations for all configurations delivered from the current time to a specified commitment point. Revert refers to the process of performing reverse operations for a single or multiple configurations.

8.1 Task Management

8.2 Commit History

The **Commit History** page records the commitment history of service management, NE management, CLI, and consistency management during synchronization based on forwarder data. On this page, you can view historical commitment records and roll back specific commitment points. The system does not limit the service volume of a transaction. However, if the service volume is too large, the transaction may time out (timeout interval: 5 minutes). To avoid this, you need to control the size of historical configurations to be rolled back.

8.1 Task Management

Scenario Description

Task management allows you to view the execution progress and result of a configuration task. Currently, only NE configuration tasks can be viewed.

Procedure

- Step 1** Choose **Config Management > Task Management** from the main menu. On the **Task Management** page that is displayed, filter and view tasks by task status.

Figure 8-1 Filtering tasks by task status

Task Type	Creator	Task Status	End Time	Task Progress	Task Status
> Dry-run	admin	Successful	2021-08-03 17:18:04	100%	Successful
> Dry-run	admin	Failed	2021-08-02 20:23:32	100%	Successful
> Dry-run	admin	To be performed	2021-08-02 20:23:18	100%	Successful

- Step 2** View the task status and execution result based on the filtering result.

Figure 8-2 Task details display

Task Type	Creator	Start Time	End Time	Task Progress	Task Status
> Dry-run	admin	2021-08-03 17:18:04	2021-08-03 17:18:04	100%	Successful
> Dry-run	admin	2021-08-02 20:23:31	2021-08-02 20:23:32	100%	Successful

NOTE

Tasks that have been completed for more than half an hour are periodically cleared and are not displayed on the page.

----End

8.2 Commit History

The **Commit History** page records the commitment history of service management, NE management, CLI, and consistency management during synchronization based on forwarder data. On this page, you can view historical commitment records and roll back specific commitment points. The system does not limit the service volume of a transaction. However, if the service volume is too large, the transaction may time out (timeout interval: 5 minutes). To avoid this, you need to control the size of historical configurations to be rolled back.

- Step 1** Choose **Config Management > Commit History** from the main menu.
- Step 2** On the **Commit History** page, commitment records are displayed in order of time. You can click **>** on the left to view the detailed configurations.

Figure 8-3 Viewing the committed information



NOTE

- The OPS framework tunes the committed information to facilitate storage and computation.
- No commitment history will be recorded for configurations that fail to be committed. A commitment point is generated only for configurations that are successfully committed.

- Step 3** Roll back the commitment history. The following rollback methods are available:
 - Rolling back a single transaction: Click **Rollback** in the **Operation** column and click **OK**.
 - Reverting a single transaction: Click **Revert** in the **Operation** column, and click **OK** to roll back all configurations between the current time and a specified commitment point.
 - Rolling back multiple transactions: Select the records to be rolled back, click **Rollback**, and click **OK**.

Figure 8-4 Selecting multiple commitment points for rollback

Date	Transaction ID	Service	Device	Operation
<input checked="" type="checkbox"/> 2021-08-03 15:16:37	71c0049b-6250-4171-819c-92d224c5c857	I3vpm	--	Roll Back Roll Back to Sp... View Device Co...
<input checked="" type="checkbox"/> 2021-08-03 15:16:07	e641eb0d-d5ad-4fd2-83d9-4f93f05d9fe9	I3vpm	--	Roll Back Roll Back to Sp... View Device Co...
<input type="checkbox"/> 2021-08-03 14:50:45	8c6814be-25d1-449f-b37f-59e92d31cc01b	I3vpm	--	Roll Back Roll Back to Sp... View Device Co...

NOTE

- You are advised to check whether the configurations meet expectations before performing a rollback.
- When a single commitment point is rolled back, only this specific point will be rolled back if its configuration is not associated with that of the next commitment point. Otherwise, the next commitment point is also rolled back.

For example:

Commitment point 1: An instance (instance 1) is delivered.

Commitment point 2: An attribute of instance 1 is delivered.

When commitment point 1 is rolled back, commitment point 2 is also rolled back.

- The system does not limit the service volume of a transaction. However, if the service volume is too large, the transaction may time out (timeout interval: 5 minutes). To avoid this, you need to control the size of historical configurations to be rolled back.

----End

9 CLI

This chapter describes the CLIs provided by the OPS and provides examples to facilitate user operations.

9.1 General Guidelines

9.2 applications

9.3 commit

9.4 display Commands

9.5 dry-run

9.6 nes Commands

9.7 quit

9.8 reset

9.9 return

9.10 clear Command

9.11 undo

9.12 rpc Command

9.13 recompute

9.1 General Guidelines

1. Entering a question mark (?): Display the help information. A question mark (?) can be entered following other commands, such as preset executable commands (including **clear**, **dry-run**, **reset**, and **display**) and node names (such as **applications** and **bngBngPairs**) in the current view, or entered separately.

```
[NCE]> applications ?  
bngBngPairs  
bngSystemNes  
isp  
ispPairs  
bngRoutingPairs  
bngAaaPairs  
hbng
```

```
bngVrfPairs  
[NCE]>
```

2. Completing a command: After entering a correct prefix character, you can press **Tab** to complete the character string. If there are multiple strings with the same prefix, all the strings are displayed.
3. Command output: The information enclosed in square brackets ([]) is the prompt information, and the directly displayed strings are configurable.

```
[app:applications]> hbng abc  
[app:applications/hbng:hbng[abc]]> domains  
[Valid length:[1..64] description:Name of a domain, it is not case sensitive.]  
[app:applications/hbng:hbng[abc]]>
```

NOTE

A maximum of 3,500 lines can be displayed on the CLI.

9.2 applications

Run the **applications** command to enter the service view.

```
Welcome to NCE CLI  
[NCE]> applications  
[app:applications]> ?  
bngBngPairs  
bngSystemNes  
isp  
ispPairs  
bngRoutingPairs  
bngAaaPairs  
hbng  
bngVrfPairs  
clear  
commit  
display  
dry-run  
quit  
return  
undo  
[app:applications]>
```

9.3 commit

```
Welcome to NCE CLI  
[NCE]> applications  
[app:applications]> hbng abc  
[app:applications/hbng:hbng[abc]]> commit  
[OK]  
[app:applications]>
```

Run the **commit** command to deliver configurations. This command cannot be used to set parameters.

9.4 display Commands

Run the **display** commands to query information.

9.4.1 display this

Run the **display this** command to query data in the RDB and CDB. In this example, "**domainName**:"**bbb**" is the data in the CDB, as shown in the following figure.

```
[app:applications/hbng:hbng[hbng_ins]]> display this
{
  "hbng":
  [
    {
      "bng_service_name":"hbng_ins",
      "domains":
      [
        {
          "domainName":"aaa"
        },
        {
          "domainName":"bbb"
        }
      ]
    }
  ]
}
[app:applications/hbng:hbng[hbng_ins]]>
```

9.4.2 display this-configuration

Run the **display this-configuration** command to query data in the RDB. (In this example, the data "**domainName**:"**bbb**" in the CDB is not displayed.)

```
[app:applications/hbng:hbng[hbng_ins]]> display this-configuration
{
  "hbng":
  [
    {
      "bng_service_name":"hbng_ins",
      "domains":
      [
        {
          "domainName":"aaa"
        }
      ]
    }
  ]
}
[app:applications/hbng:hbng[hbng_ins]]>
```

9.4.3 display inconsistency-discovery

Run the **display inconsistency-discovery** command to query NE configuration inconsistencies (in the NE view). Before running this command, you need to run the command for comparing NE configuration inconsistencies described in [9.6.2 nes inconsistency-discovery \[A\] \[B\]....](#)

```
[nes:nes/ne[ipi-zba900-r-bn-04]]> display inconsistency-discovery
### huawei-rsa ###
rsa
+ rsaLocalKeys
+ rsaLocalKey
+ keySize: 2048
+ rsaPeerKeys
+ rsaPeerKey
+ [keyName=*****]
+ keyName:*****
```

```
+ encodeInType:DER
+ keyCode:
3082010A028DFADF5F4A65DSFADFADCVDG055465ADSF5ASDF456A4DF5A4DF54AS65DF41FD231AD5F46
ADF5A1DF2
+ rsaGlobalCfg
+ maxKeyPair:20
### ietf-interfaces ###
.....
```

9.4.4 display dry-run preview

Run the **display dry-run preview** command to query configuration comparison information. This command provides the same function as the **Delivered Device Data** tab page in the dry-run view on the GUI. You need to run the **dry-run** command before running this command.

```
Welcome to NCE CLI
[NCE]> nes ne ipi-zba900-r-bn-04
[nes:nes/ne[ipi-zba900-r-bn-04]]> system systemInfo sysName abc
[nes:nes/ne[ipi-zba900-r-bn-04]/huawei-system/system:system/systemInfo]> dry-run
[OK]
[nes:nes/ne[ipi-zba900-r-bn-04]/huawei-system/system:system/systemInfo]> display dry-run preview
### ipi-zba900-r-bn-04 ###
<system xmlns="http://www.huawei.com/netconf/vrp/huawei-system">
  <systemInfo>
    <sysName>abc</sysName>
  </systemInfo>
</system>
[nes:nes/ne[ipi-zba900-r-bn-04]/huawei-system/system:system/systemInfo]>
```

9.4.5 display dry-run compare

Run the **display dry-run compare** command to query the comparison result. This command provides the same function as the **Inconsistent Data** tab page in the dry-run view on the GUI. You need to run the **dry-run** command before running this command.

```
[nes:nes/ne[ipi-zba900-r-bn-04]/huawei-system/system:system/systemInfo]> display dry-run compare
### ipi-zba900-r-bn-04/(http://www.huawei.com/netconf/vrp/huawei-system)system ###
system
systemInfo
+sysName: abc
[nes:nes/ne[ipi-zba900-r-bn-04]/huawei-system/system:system/systemInfo]>
```

9.5 dry-run

Run the **dry-run** command to perform comparison. The preview and comparison commands in [9.4.4 display dry-run preview](#) and [9.4.5 display dry-run compare](#) can be executed only after the **dry-run** command is executed successfully.

```
Welcome to NCE CLI
[NCE]> nes ne ipi-zba900-r-bn-04
[nes:nes/ne[ipi-zba900-r-bn-04]]> system systemInfo sysName abc
[nes:nes/ne[ipi-zba900-r-bn-04]/huawei-system/system:system/systemInfo]> dry-run
[OK]
[nes:nes/ne[ipi-zba900-r-bn-04]/huawei-system/system:system/systemInfo]>
```

9.6 nes Commands

Run the **nes** command to enter the NE management view.

9.6.1 nes ne [xxx]

Run the **nes ne** *[xxx]* command to enter the view of a single NE. *[xxx]* indicates the NE name.

```
[NCE]> nes ne ?  
[ipi-zba900-r-bn-04 ipi-zba900-r-bn-17]  
[NCE]> nes ne ipi-zba900-r-bn-04  
[nes:nes/ne[ipi-zba900-r-bn-04]]>
```

9.6.2 nes inconsistency-discovery [A] [B]...

Run the **nes inconsistency-discovery** *[A] [B]...* command to compare NE configuration inconsistencies. *[A] [B]...* indicates the NE names. One or more NE names can be entered.

```
[NCE]> nes inconsistency-discovery ipi-zba900-r-bn-04  
[OK]  
[NCE]>
```

9.6.3 nes sync-to [A] [B]...

Run the **nes sync-to** *[A] [B]...* command to synchronize data from the NMS to NEs. *[A] [B]...* indicates the NE names. One or more NE names can be entered.

9.6.4 nes sync-from [A] [B]...

Run the **nes sync-from** *[A] [B]...* command to synchronize data from NEs to the NMS. *[A] [B]...* indicates the NE names. One or more NE names can be entered.

9.7 quit

Run the **quit** command to return to the upper-level view.

```
[NCE]> applications hbng abc domains bbb  
[app:applications/hbng:hbng[abc]/domains[bbb]]> quit  
[app:applications/hbng:hbng[abc]]>
```

9.8 reset

Run the **reset** command to reset configured data. This command must be executed in the top-level view. After this command is executed, the transaction IDs are deleted on the CLI side.

```
[app:applications/hbng:hbng[abc]]> return  
[NCE]> reset  
[OK]  
[NCE]>
```

9.9 return

Run the **return** command to return to the top-level view.

```
[app:applications/hbng:hbng[abc]/environment[bbb]/vrfs[abc]]> return  
[NCE]>
```

9.10 clear Command

Run the **clear** command to clear all contents on the screen. This function is the same as the **clear** command on Linux.

9.11 undo

This command is used to perform a deletion operation, for example, deleting a list, leaf list, leaf, or presence container node.

```
[app:applications]> display this
{
  "applications":
  {
    "hbng:hbng":
    [
      {
        "bng_service_name":"hbng1"
      },
      {
        "bng_service_name":"abc"
      }
    ]
  }
}
[app:applications]> undo hbng abc
[app:applications]> display this
{
  "applications":
  {
    "hbng:hbng":
    [
      {
        "bng_service_name":"hbng1"
      }
    ]
  }
}
[app:applications]>
```

9.12 rpc Command

The **rpc** command can be executed when the YANG file of a service or NE contains an RPC structure.

For example, define RPC in the YANG file.

```
rpc change-session-board {
  description
    "Change slot id of session.";
  input {
    leaf local-discriminator {
      type uint32 {
        range "1..4294967295";
      }
      mandatory true;
      description
        "Local discriminator.";
    }
    leaf select-slot-id {
      type string {
        length "1..19";
      }
    }
  }
}
```

```
}  
  mandatory true;  
  description  
    "Target slot id.";  
}  
}
```

Commands corresponding to the RPC:

```
[nes:nes/ne[test]]> huawei-bfd:change-session-board ?  
local-discriminator  
select-slot-id  
[nes:nes/ne[test]]> huawei-bfd:change-session-board 12 2
```

9.13 recompute

Run the **recompute** command based on the **hbng** service point to perform recomputing. For details about the application scenarios of recomputing, see [Figure 6-2](#).

The recomputing function can be used only on the points under the **applications** level.

```
[app:applications]> hbng hbng1  
[app:applications/hbng:hbng[hbng1]]> recompute  
[OK]  
[app:applications/hbng:hbng[hbng1]]>
```

10 Alarm/Event


- [10.1 Alarms Reported by Devices](#)
- [10.2 Customized Events](#)
- [10.3 Alarm Synchronization](#)
- [10.4 Alarm Clearance](#)

10.1 Alarms Reported by Devices

10.1.1 Third-Party Package of SNMP Traps

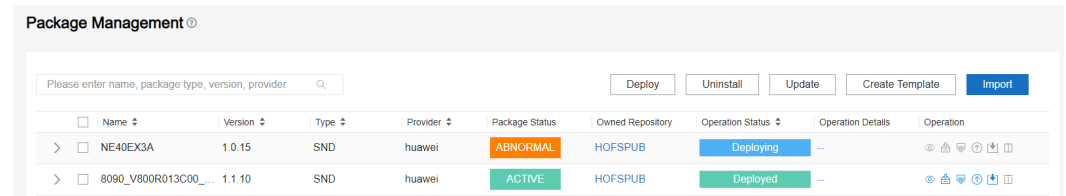
- Create a third-party package: Obtain the GND or SND framework according to the third-party package creation manual, and enable alarm interfaces as follows:
 - Java implementation

```
public class AlarmTest extends AlarmReceiver { //Inherit AlarmReceive.  
  
    //Implement trapHandle method.  
    public void trapHandle(AocContext context, AocSnmpTrapEntity traps) throws  
AocAlarmException {  
        LOGGER.info("trapHandle start !!!");  
    }  
}
```
 - Python implementation

```
class AocAlarmReciver(AlarmReceiver):  
    def trap_handle(cls,aoccontext, trapEntityList):  
        cls.logger.info("trapHandle start !!! traps = {}".format(trapEntityList))  
        AocAlarmReciver.test_sync_snmp(cls)  
        AocAlarmReciver.test_write_alarm(cls)  
        AocAlarmReciver.test_add_event(cls)
```
- Load and activate the third-party package.
 - a. Choose **Package Repo** from the main menu. Then, choose **Package Management** from the navigation pane.
 - b. On the **Package Management** page, click **Import** to import the prepared software package.
 - c. Click  in the **Operation** column to deploy the software package.

After the deployment is successful, the package status changes to **ACTIVE**.

Figure 10-1 Loading a third-party package



- Enable the trap function.

Choose **Resource > SNMP Configuration** from the main menu and set SNMP parameters.

Figure 10-2 Configuring SNMP for the AOC



- Trigger an alarm on a device, for example, an interface up/down alarm.

Figure 10-3 Interface down alarm on a device



- Search for logs to check whether the third-party package receives an alarm reported by the device.

10.1.2 NETCONF Notification

- Create a third-party package: Obtain the GND or SND framework according to the third-party package creation manual, and enable alarm interfaces as follows:
- Java implementation

```
public class AlarmTest extends AlarmReceiver { //Inherit AlarmReceive.
    //Implement netconfNotificationHandle method
    public void netconfNotificationHandle(AocContext context, AocNcfNotificationsEntity notifications)
    throws AocAlarmException {
        LOGGER.info("netconfNotificationHandle start !!!");
        try {
            List<AocAlarmEntityModel.AocAlarmEntity> inputs = new ArrayList<>();
            AocAlarmEntityModel.AocAlarmEntity alarm = getAlarm();
            inputs.add(alarm);
            if (null == AocAlarmApiFacade.getAocAlarmMgr()) {
                LOGGER.info("AocAlarmApiFacade is null !!!");
            } else {
                AocAlarmApiFacade.getAocAlarmMgr().writeAlarm(inputs);
            }
        }
    }
}
```

```

    } catch (Exception e) {
        LOGGER.info("exception", e);
    }
}

```

- **Python implementation**

```

class AocAlarmRecevier(AlarmReceiver):
    def netconf_notification_handle(cls, aoccontext, netconfNotificationList):
        cls.logger.info("netconfNotificationHandle python start input is
{}".format(netconfNotificationList))

```

- Load and activate the third-party package.
- Trigger an alarm on a device. For example, run the following command on a Huawei device to simulate a notification alarm.

```
[~DTB-diagnose]simulate alarm-name hwStorageUtilizationRisingAlarm alarm-state start
```
- Check whether the third-party package receives the notification alarm generated on the device.

10.2 Customized Events

- Create a third-party package: Obtain the GND or SND framework according to the third-party package creation manual, and enable alarm interfaces as follows:

- **Java implementation**

```

public class AocEventTest extends EventReceiver {

    public void eventHandle(AocContext context, AocEventHandlerModule.AocEventHandler message)
        throws AocEventException{
        LOGGER.info("AocEventTest start eventHandle context is {}, message is {}.", context, message);
        return;
    }
    //Implement notify method
    public void notify(AocContext context, AocEventHandlerModule.AocEventHandler message)
        throws AocEventException {
        LOGGER.info("AocEventTest start notify context is {}, message is {}.", context, message);
        return;
    }
}

```

- **Python implementation**

```

class AocEventReciever(EventReciever):
    def notify(self,aoccontext, eventHanlder):
        self.logger.info("handleAocEvent python start input is {}".format(eventHanlder))

```

- Load and activate the third-party package.
- Trigger a customized event. For example, run the following command on a device to simulate an interface down event.

```
[~DTB-Ethernet0/1/2]shutdown
[*DTB-Ethernet0/1/2]commit
```
- Check whether the third-party package receives the event generated on the device.

10.3 Alarm Synchronization

- Create a third-party package: Obtain the GND or SND framework according to the third-party package creation manual.
- Load and activate the third-party package.

- Check whether the third-party package receives alarm synchronization messages.

10.4 Alarm Clearance

- Generate an alarm according to [10.1.1 Third-Party Package of SNMP Traps](#).
- Delete the device.
- Check whether the alarm is cleared on the alarm page.

11 Service Security

The following lists the service security precautions for the AOC:

- Prerequisites for configuration delivery: Data is consistent between the controller and devices, which needs to be manually ensured. If the configurations are inconsistent, unpredictable impact may be brought to devices after configuration delivery.

The initial status of a device is unsynchronized. After a successful synchronization, the device status changes to synchronized. If the device configuration is inconsistent with that on the controller during consistency verification, the device status is changed to unsynchronized. If **No-network** is selected during single-NE configuration or service configuration delivery, the device status changes to unsynchronized.

- Configuration delivery logic:
 - Full service recomputing: The system performs full service recomputing when configurations are created or updated.

For example, interface 0/0/1 and interface attributes attr1 (value1) and attr2 (value2) are configured in the SSP package. Assume that value2 of attr2 for interface 0/0/1 is changed to value2' after configurations are synchronized from the device, and then a user changes the value of attr1 in the SSP package to value1'. In this case, the user can find two differences of the NE: the value of attr1 is changed from value1 to value1', and the value of attr2 is changed from value2' to value2.
 - Minimum delivery: The system delivers the minimum configuration based on the difference between the NE configuration decomposed from the current service configuration and the NE configuration stored in the system.

Example: A user sets the MTU to 100 and bandwidth to 1000 for a port, and the system already stores the bandwidth value of 1000. During configuration delivery, only the MTU is delivered.
 - Multi-device interaction processing: If the configuration is delivered to multiple devices, the configuration is delivered to these devices concurrently.

 **CAUTION**

When delivering configurations, you may not know the range of devices to which the configurations are delivered. Therefore, before delivering the configurations, you are advised to perform a dry run to check the range of devices to which the configurations are to be delivered.

- Deletion principle: When a service is deleted, the configurations delivered for the service are also deleted.
- Rollback specifications:
 - If both service provisioning and rollback fail, you can only view information on the NE management page and cannot detect the rollback failure.
 - Transactions are characterized by atomicity. When some configurations in the transaction fail to be delivered, the system initiates a rollback. If the rollback fails, the error message does not contain the rollback failure information. In this case, you can view the configuration difference through data consistency verification.
- ACID of configuration delivery transactions: Currently, the system provides isolation capabilities only at the read committed level. For concurrent services, issues such as non-repeatable read and phantom read may occur. You are advised to control the data volume serially in the northbound API.
- Data consistency processing logic: If the function of deleting data during data consistency verification is enabled in the SND package, device configurations will be deleted during data consistency verification.
- When you create duplicate service or NE configuration instances, the system reports an error.
- High-risk operations:
 - Data consistency: Data may be deleted during data consistency verification.
 - Configuration delivery: Currently, the system cannot determine whether a service configuration operation is a high-risk operation. Therefore, no message will be displayed on the GUI.
- Conflict detection:

When the configuration is delivered, the following conflicts are checked:

a. **Multi-terminal operation conflict**

Before delivering configurations, NCE checks whether the operation configurations of multiple terminals conflict with each other. Assume that the same service instance named service 1 ($x=1, y=2$) is configured for both terminal 1 and terminal 2. The service instance is then edited, submitted, and updated to service 1 ($x=2, y=2$) for terminal 1, while terminal 2 performs operations based on the original service 1 ($x=1, y=2$). In this case, the system displays a message indicating a conflict when terminal 2 submits data.

Current solution: Roll back to the previous configuration and perform configuration again.

b. **Device configuration conflict**

Before delivering device configurations, NCE checks whether there are risks of inconsistency between NCE data and the device data. If so, a message is displayed indicating a potential configuration conflict. A conflict message is displayed in the following scenarios:

- i. When NCE manages a device for the first time and does not synchronize the device data, data is not synchronized between NCE and the device.
- ii. NCE saves the sequence IDs of device configuration changes. (The IDs can be obtained from the device or based on the checksum for calculating the full device configuration on NCE.) Before redelivering configurations, NCE obtains the sequence IDs again. If an ID change is detected, NCE considers that the configuration changes.
- iii. When a device is disconnected from NCE during configuration delivery, NCE attempts to roll back the configurations.

Current solution:

- i. Commit the device configurations forcibly. In this case, NCE does not check for conflicts. After the configuration is forcibly committed, the system cannot automatically display the inconsistent service status and you cannot directly know which services are affected. Instead, you need to manually check the consistency of services.
- ii. Synchronize data with the device, and then edit and commit the configurations again.

c. **Service configuration conflict**

Before delivering service configurations, NCE checks whether the configurations delivered to a device conflict with each other. Assume that both service 1 and service 2 deliver the description configurations of VLAN 2 to device 1. In this case, NCE checks whether the two configurations are the same. If not, the system displays a message indicating that a configuration conflict occurs.

Current solution:

- i. Commit the service configurations forcibly. In this case, NCE does not check for conflicts.
- ii. Delete the previous service configurations and attributes related to the service configurations to avoid conflicts with the current service configurations.

Pay attention to the following point before forcibly committing the configuration:

When you forcibly commit the configuration, the system will ignore the multi-terminal operation check, device configuration conflict check, and service configuration conflict check.

- a. Multi-terminal operation check: If the check fails, you need to check the differences and check whether the old value to be overwritten by the modified content has service security risks. If not, the modification is forcibly committed.
- b. Device configuration conflict check: If the check fails, you are advised to perform synchronization before committing the configuration. If you forcibly commit the configuration, automatic rollback or post-event rollback for multiple devices may be triggered due to inconsistency of

configurations between the controller and devices. As a result, the existing configuration of the devices is affected. To prevent this, you are advised to check the device configuration differences. If data is created or modified, perform a rollback to delete the configuration or restore the original configuration.

- c. Service configuration conflict check: If the check fails, the intents of two services are inconsistent. If the configuration is forcibly committed, the existing service intents will be affected. In this case, you need to restore the existing service intents to ensure that the existing service intents are consistent. Alternatively, you need to modify the existing service to ensure that the values are the same as those of the new service.