
Contents

1 Configuring Software Upgrade.....	1
1.1 Introduction.....	1
1.2 Principles of System Version Upgrade.....	1
1.3 Principles of Patch Version Upgrade.....	3
1.3.1 Basic Concepts.....	3
1.3.2 Patch Package Management.....	3
1.3.3 Patch Auto-Sync.....	3
1.4 Configuration Task Summary.....	4
1.5 Configuring System Version Upgrade.....	4
1.5.1 Configuration Tasks.....	4
1.5.2 Configuring Subsystem Upgrade/Degrade.....	4
1.5.3 Configuring Version Auto-sync.....	5
1.6 Configuring Patch Version Upgrade.....	6
1.6.1 Overview.....	6
1.6.2 Configuration Tasks.....	6
1.6.3 Installing a Patch Package.....	6
1.6.4 Uninstalling a Patch Package.....	7
1.7 Monitoring.....	8
1.8 Configuration Examples.....	9
1.8.1 Configuring Subsystem Upgrade.....	9
1.8.2 Configuring Auto-sync Upgrade.....	11
1.8.3 Installing a Patch Package.....	12

1.8.4 Uninstalling a Patch Package.....15

1 Configuring Software Upgrade

1.1 Introduction

Adopting a modular structure, Orion B26Q Networks Operating System (NOS) supports overall system upgrade and subsystem upgrade, as well as the upgrade through patches. The package management module of NOS is used to install, query, and maintain components of the device. By upgrading the software of the device, users can install software that is more stable or contains more features in the system.

1.2 Principles of System Version Upgrade

1. Basic Concepts

- Subsystem

A subsystem exists on a device in the form of images. The NOS subsystems include:

- Uboot: After being powered on, the device loads and runs the Uboot subsystem. This subsystem initializes the device, and loads and runs system images.
- Rboot: This subsystem is a set of dedicated programs used to install or upgrade the main program.
- Basic Input Output System (BIOS): This subsystem is dedicated for devices running the X86 architecture.
- Main program: This subsystem is a collection of Operating Systems (OSs) and applications in the system.

- Main package

The main package is often used to upgrade/degrade subsystems of box-type devices. The main package is a combination package of the Uboot, Rboot, and main program subsystems. The main package is used for overall system upgrade/degradation.

- BIOS package

The BIOS package is used to upgrade the BIOS of the device. It is used on the devices running the X86 architecture only.

⚠ Caution

An "installation package" in this document refers to installation files of all the subsystems of the device.

2. Upgrading/Degrading and Managing Subsystems

Subsystem upgrade/degradation aims to update the software functions by replacing the subsystem components in the device with the ones in an installation package. Redundancy design is adopted for subsystems, so subsystems of the device are often not directly replaced with the subsystems in the package during upgrade/degradation. Instead, subsystems are added to the device and then activated.

- Upgrade/Degradation

Subsystems exist on the device in different forms. Therefore, they are upgraded or degraded differently.

- Uboot: Generally, this subsystem exists in the NOR flash memory of the device in the form of images. It is upgraded/degraded by writing images into the NOR flash memory of the device.
- Rboot: Generally, this subsystem exists in the NOR flash memory of the device in the form of images. It is upgraded/degraded by writing images into the NOR flash memory of the device.
- BIOS: Generally, this subsystem exists in the flash memory of the device in the form of images. It is upgraded/degraded by writing images into the flash memory of the device.
- Main program: Generally, this subsystem exists in the NAND flash memory of the device in the form of images. It is upgraded/degraded by writing images into the NAND flash memory of the device.

- Management

During upgrade of the subsystem component of the main program, the upgrade/degradation module always records the subsystem component in use, the redundant subsystem components, and management information of various versions in the configuration file. The module queries the available subsystem components and then loads subsystem components as needed.

The redundancy design is adopted for each subsystem component. During the upgrade/degradation:

- Uboot: This subsystem contains a master boot subsystem and a slave boot subsystem. Only the master boot subsystem is upgraded, and the slave boot subsystem is always redundant.
- Rboot: This subsystem contains one program. More redundancies are allowed if there is enough space.
- BIOS: This subsystem contains a master BIOS and a slave BIOS. The two BIOSs can be upgraded.
- Main program: One redundancy is allowed if there is enough space.

- Upgrade

Put upgrade files in the local device, and then run the **upgrade** command.

3. Auto-Sync Upgrade

Auto-sync upgrade aims to coordinate multiple subsystems (boards, chassis, and virtual switching unit (VSU) system) within one system. During software upgrade, the system automatically sends an upgrade package to all matched members, or upgrades the software version of new members to the specified version according to the specified upgrade policy to ensure that the software version of the entire system meets expectations.

- Select one from the following auto-sync upgrade policies:
 - Check the software version of a new member, and upgrade the member if its version is earlier than the version of the system installation package. Otherwise, no upgrade is performed.
 - Do not auto-sync new members.
 - Upgrade new members to the version of the system upgrade package regardless of the current version of the new members.
- Auto-sync upgrade check is performed in the following scenarios:
 - During upgrade, if no upgrade target is specified, the upgrade function module sends the upgrade package to all matched members (including boards, chassis, and VSU system) for auto-sync upgrade.
 - When the device is started, the version of every member is checked, and auto-sync upgrade is performed according to the version check result and auto-sync upgrade policy.
 - When a new member is added, the version of the member is checked and auto-sync upgrade is performed

according to the version check result and auto-sync upgrade policy.

- Management

To perform auto-sync upgrade, configure an auto-sync upgrade policy and install an upgrade package.

- Configuring an auto-sync upgrade policy

To perform the expected upgrade when a new member is added to the system, check the auto-sync upgrade configuration, such as preparation of the installation package.

If a board is not checked because the system is not configured with an auto-sync upgrade policy, you can upgrade the specific members.

1.3 Principles of Patch Version Upgrade

1.3.1 Basic Concepts

- Patch package

A patch package contains several patches. You can upgrade the package to fix various problems for the main program. If a patch package contains hot patches, they take effect before you restart the device restart. If a patch package contains cold patches, they take effect only after the device is restarted.

1.3.2 Patch Package Management

Upgrade of cold patches is to expand the component files in a package into the device. The upgrade takes effect only after the device is restarted.

Such process is similar to the upgrade of hot patches. But in the latter upgrade, only the files to be revised are replaced and then the new files take effect automatically.

Patch management is to use the database to record the patch information. The query, installation, and uninstallation correspond to the insertion, query, and deletion of the database.

Patch management includes three states: installed, activated, and confirmed:

- A patch in the installed state only indicates that this patch exists on the device, but it has not taken effect.
- Only a patch in the activated state is effective, but it will automatically roll back to the previous state after the device is restarted.
- A confirmed patch is also an effective patch, and the patch will not roll back even if the device is restarted.

1.3.3 Patch Auto-Sync

Patch auto-sync aims to coordinate multiple subsystems (boards and chassis) within a system. During software upgrade, the system automatically sends the patch installed on the current supervisor module to all the members meeting component installation rules. When a new member is added, the system automatically synchronizes the patch installed on the supervisor module to the new member. Thus, no resolved patch problem occurs even if synchronization is omitted.

Patch auto-sync rules:

- The subsystems comply with all the normal patch installation rules.
 - The versions of all the subsystems must come from the same rack package before patch auto-sync is performed.
 - The patch installed on the master supervisor module is used as the benchmark. If the same patch is
-

installed on a newly connected member but the patch status is higher than that of the current supervisor module, the patch on the new member is not be processed. The status order of patches is: add, active, and running.

1.4 Configuration Task Summary

Software upgrade configuration includes the following tasks:

- [Configuring System Version Upgrade](#)

a [Configuring Subsystem Upgrade/Degrade](#)

b (Optional) [Configuring Version Auto-sync](#)

- [Configuring Patch Version Upgrade](#)

c [Installing a Patch Package](#)

d [Uninstalling a Patch Package](#)

1.5 Configuring System Version Upgrade

1.5.1 Configuration Tasks

System version upgrade configuration includes the following tasks:

(1) [Configuring Subsystem Upgrade/Degrade](#)

(2) (Optional) [Configuring Version Auto-sync](#)

1.5.2 Configuring Subsystem Upgrade/Degrade

Warning

Boot upgrade is risky. If the device is powered off during upgrade, the device may be damaged and thus does not work. If the upgrade fails, do not power off the device, but attempt to upgrade the boot again.

1. Overview

Subsystem installation packages include the following types:

- Main package for device upgrade: After the upgrade using the main package is completed, all system software in the device is updated, and the overall software is enhanced.
- Rack package for upgrade: After the upgrade using the rack package is completed, all system software in the rack-type device is updated, and the overall software is enhanced.

Specification

Generally a main package is released to upgrade box-type devices.

2. Restrictions and Guidelines

- To upgrade a main package of the device, download the installation package to the local device and run the **upgrade** command.
 - To upgrade a rack package, store the rack package in a USB flash drive or SD card of the device because
-

the rack package is large, and then run the **upgrade** command.

3. Procedure

(1) Enter the privileged EXEC mode.

```
enable
```

(2) Upgrade the installation package corresponding to the device.

```
upgrade [ boot ] { url | download { oob_ftp://path [ via mgmt { interface-number } ] | oob_tftp://path [ via mgmt { interface-number } ] | ftp://path [ vrf vrf-name ] | tftp://path [ vrf vrf-name ] } [ slot slot-number ] } [ force ]
```

1.5.3 Configuring Version Auto-sync

1. Overview

This section describes how to configure a version auto-sync policy for a distributed system.

2. Restrictions and Guidelines

- To upgrade a new member in the system, check the version auto-sync upgrade configuration, such as preparation of the installation package.
- If a device member is not checked because the system is not configured with a version auto-sync policy, you can upgrade a specific member manually.
- After the path of an upgrade package used for version auto-sync is configured, the upgrade function module will determine the location of the installation package according to this path when new members (boards, chassis, or new VSU members) join the system. Each time the system is upgraded, the upgrade function module records the path of the installation package for this upgrade and uses it for version auto-sync. You can also use the **upgrade auto-sync package** command to set an upgrade package path.

3. Procedure

(1) Enter the privileged EXEC mode.

```
enable
```

(2) Configure a version auto-sync policy.

```
upgrade auto-sync policy [ none | compatible | coordinate ]
```

✔ Specification

All the parameters apply to only the VSUs.

(3) Configure the range of version auto-sync.

```
upgrade auto-sync range [ vsu ]
```

(4) Configure the path of the upgrade package for version auto-sync.

```
upgrade auto-sync package url
```

✔ Specification

All the parameters apply to only the VSUs.

(5) Configure the status of the version auto-sync service.

```
upgrade sync-server [ close | open ]
```

1.6 Configuring Patch Version Upgrade

1.6.1 Overview

Besides the main program version, NOS provides a patch mechanism to quickly fix the problems occurring on the main program version. A hot patch package can fix software bugs before the device is restarted; but an installed cold patch package can take effect only after the device is restarted. This patch package fixes a specific software version only. After the main program version is upgraded, the corresponding patch will become invalid automatically.

1.6.2 Configuration Tasks

Patch version configuration includes the following tasks:

(1) [Installing a Patch Package](#)

(2) [Uninstalling a Patch Package](#)

1.6.3 Installing a Patch Package

1. Overview

This section describes how to install a patch package to fix software bugs. A patch package is used to upgrade a specific version of software only.

2. Restrictions and Guidelines

- Before installing a patch package, download it to a local directory of the device.

Caution

The downloaded patch package cannot be renamed. The name of the patch package is used as the operation keyword for the subsequent patch activation, deactivation, and uninstallation.

- The installed patch must be activated before it is used. You must activate the patch first temporarily and then permanently. If a patch is activated temporarily, the patch will roll back to the previous state after the device is restarted. You can activate a patch temporarily only when verifying the patch function temporarily.

3. Procedure

(1) Enter the privileged EXEC mode.

```
enable
```

(2) Download a patch package.

```
install add url [ slot { all } ]
```

(3) Install the patch package.

- Run the general patch command to install the patch package.
 - a Activate the installed patch temporarily.
-

install activate *package_name* [slot { 0 | all }]

b Activate the patch permanently.

install commit [slot { 0 | all }]

- Run the **patch** command to install the patch package.
-

c Activate the installed patch temporarily.

patch active [slot { 0 | all }]

d Activate the patch permanently.

patch running [slot { 0 | all }]

- (4) Activate the patch permanently by one click.

patch auto-running [slot { 0 | all }]

1.6.4 Uninstalling a Patch Package

1. Overview

This section describes how to deactivate the activated patches and remove the existing patches from the device, that is, the database information.

2. Restrictions and Guidelines

For users, only one patch is effective on the device. Therefore, only one patch to be activated is allowed on the device. To install another patch package of a function component, you need to uninstall the previous patch.

3. Procedure

- (1) Enter the privileged EXEC mode.

enable

- (2) Uninstall a patch.

- Run the general patch command to uninstall a patch package.

a Deactivate an activated patch.

install deactivate *package_name* [slot { 0 | all }]

b Uninstall a patch.

install remove *package_name* [slot { 0 | all }]

- Run the **patch** command to uninstall a patch package.

c Deactivate an activated patch.

patch deactivate *package_name* [slot { 0 | all }]

d Uninstall a patch.

patch delete [slot { 0 | all }]

1.7 Monitoring

Run the **show** commands to check the running status of a configured function to verify the configuration effect.

Run the **check** commands to check information.

Run the **clear** commands to clear information.

⚠ Caution

Running the **clear** commands may lose vital information and thus interrupt services.

Table 1-1 System Version Upgrade Monitoring

Command	Purpose
show upgrade status	Displays the upgrade status of the device and the installation status of a patch.
show upgrade history	Displays the upgrade history.
check version	Checks the version matching.

Table 1-2 Patch Version Upgrade Monitoring

Command	Purpose
install remove <i>package_name</i> [slot { 0 all }]	Removes an unactivated patch package from a multi-patch solution.
patch delete [slot { 0 all }]	Removes an unactivated patch package from a single-patch solution.
clear install storage [slot { 0 all }]	Clears all the patch packages not running currently and corresponding database information.
show install	Displays information about the hot patch package already installed on the device in a multi-patch solution.
show patch	Displays information about the hot patch package already installed on the device in a single-patch solution.
show install auto-sync	Displays whether the device enables patch auto-sync.

1.8 Configuration Examples

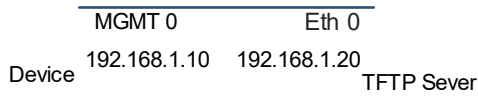
1.8.1 Configuring Subsystem Upgrade

1. Requirements

Upgrade a subsystem installation package to update all software in the device so that the overall software is enhanced and the known software bugs are fixed. Use [Figure 1-1](#) as an example. The **upgrade download oob_tftp://path** command is used to upgrade the subsystems of the device by using the installation package file stored on the TFTP server.

2. Topology

Figure 1-1 Subsystem Upgrade Topology



3. Notes

- Configure an IP address for the network management interface on the device so that the L3 route is reachable between the device and the TFTP server.
- Run the **upgrade download oob_tftp://path** command to upgrade the installation package file stored on the TFTP server.

4. Procedure

- (1) Configure an IP address for the network management interface on the device.

```

Hostname> enable
Hostname# configure terminal
Hostname(config)# interface mgmt 0
Hostname(config-if-Mgmt 0)# ip address 192.168.1.10 255.255.255.0
Hostname(config-if-Mgmt 0)# gateway 192.168.1.1
Hostname(config-if-Mgmt 0)# end
  
```

- (2) Run the **upgrade download oob_tftp://path** command to upgrade the device.

```

Hostname# upgrade download oob_tftp://192.168.1.20/main.bin via mgmt 0 force
*Nov 23 13:21:39: %UPGRADE-6-INFO: Copy to /tmp/vsd/0/upgrade_rep/
*Nov 23 13:21:39: %UPGRADE-6-INFO: Please wait for a moment.....
Press Ctrl+C to quit
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
!!!!!!
< The terminal is locked by upgrade module >
Upgrade start
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!10%
< you can press Ctrl+C to unlock terminal >
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!20%
< you can press Ctrl+C to unlock terminal >
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!30%
< you can press Ctrl+C to unlock terminal >
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!100%
Upgrade success
< The terminal is unlocked by upgrade module >
[Slot 0]
  
```

```
Device type      : B26Q
Status           : success
```

(3) After upgrading the subsystems, run the **reload** command to restart the device so that the configuration takes effect.

```
Hostname# reload
Reload system?(Y/N)y
*Dec 25 06:48:02: %SYSMON-5-RELOAD: Reset system from [CONSOLE] with reason
[Management port/Reset system by reload command]
Hostname#*Dec 25 06:48:02: %DP-3-RESET_DEV: Reset device 1 due to
SYSMON/3766/CONSOLE/Management port/Reset system by reload command.
*Dec 25 06:48:04: %DP-3-MACHINE_RESTART: MACHINE_RESTART.
```

5. Verification

(1) Run the **ping** command to check whether the L3 route between the device and the TFTP server is reachable.

```
Hostname# ping oob 192.168.1.10
Sending 5, 100-byte ICMP Echoes to 192.168.1.20, timeout is 2 seconds:
 < press Ctrl+C to break >
!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 1/1/3 ms.
```

(2) Run the **show upgrade status** command to display the upgrade result.

```
Hostname# show upgrade status
upgrade global status: INIT
[Slot 0]
Device type      : Orion_B26Q
Status           : success
```

(3) Check the running version on the device. If the version information changes, the upgrade is successful.

```
Hostname# show version
System description      : Orion_B26Q 10G Ethernet Switch(B26Q-20XS4VS2QXS) By
Orion Networks
System start time      : 2020-11-23 13:13:59
System uptime          : 0:00:03:36
System hardware version : 1.0B
System software version : B26Q_NOS 12.1(PL1)
System patch number    : NA
System serial number    : 1234942570025
System boot version     : 1.4.2(Master) 1.4.2(Slave)
Module information:
Slot 1/0 : Orion_B26Q
Hardware version      : 1.0B
Boot version          : 1.4.2(Master) 1.4.2(Slave)
Software version      : B26Q_NOS 12.1(PL1)
Serial number         : 1234942570025
Slot 2/0 : Orion_B26Q
Hardware version      : 1.00
Boot version          : 1.4.2(Master) 1.4.2(Slave)
```

```
Software version   : B26Q_NOS 12.1 (PL1)
Serial number      : 1234942570022
```

6. Common Errors

If an error occurs during the upgrade, the system displays the cause of the upgrade failure. You can also run the **show upgrade status** command to display the result of the last upgrade.

The following describes several types of common error messages:

- Invalid installation package: The possible cause is that the package is damaged or is not an installation package. You are advised to obtain the correct installation package and perform the upgrade again.
- Installation package not supported by the device: The possible cause is that you use the installation package of other devices. You are advised to obtain the correct installation package, verify the package information, and perform the upgrade again.
- Insufficient device space: Generally, this error occurs on a rack-type device. You are advised to check whether the device is equipped with a USB flash drive or SD card. Rack-type devices often have USB flash drives.

1.8.2 Configuring Auto-sync Upgrade

1. Requirements

On devices using a distributed structure such as rack-type devices and VSUs, auto-sync upgrade can upgrade the software version of matched members according to a configured auto-sync upgrade policy when software is upgraded or new members are added. This function ensures the coordination of the entire system. In the example of the VSU system using rack-type devices, we auto-sync an installation package.

2. Procedure

Configure an auto-sync upgrade policy based on the software version of the controlling device.

```
Hostname> enable
Hostname# upgrade auto-sync policy coordinate
```

Set the auto-sync upgrade range to the VSU system.

```
Hostname# upgrade auto-sync range vsu
```

3. Verification

Display the current auto-sync upgrade policy.

```
Hostname# show upgrade auto-sync
auto-sync range       : vsu
auto-sync policy      : coordinate
auto-sync package     : flash:install_file/B26Q_install.bin
```

Check version matching of device nodes.

```
Hostname# check version
Dev Slot State
--- ----
1 4 Compatible
1 FE6 Compatible
1 M1 Compatible
```

4. Common Errors

- The url parameter in the command used for configuring auto-sync upgrade does not point to a valid upgrade package.

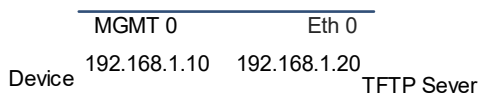
1.8.3 Installing a Patch Package

1. Requirements

Install a patch package to fix software bugs without upgrading the main program version.

2. Topology

Figure 1-1 Patch Package Installation Topology



3. Notes

- (1) Configure the L3 route reachable to the device.
- (2) Copy the patch package to the device.
- (3) Run the patch installation command.
- (4) Activate a hot patch.
 - a Activate a hot patch temporarily.
 - b Activate a hot patch permanently.
- (5) Configure patch auto-sync.

4. Procedure

- (1) Configure an IP address for the network management interface on the device.

```

Hostname> enable
Hostname# configure terminal
Hostname(config)#interface mgmt 0
Hostname(config-if-Mgmt 0)# ip address 192.168.1.10 255.255.255.0
Hostname(config-if-Mgmt 0)# gateway 192.168.1.1
  
```

- (2) Install a patch package.

```

Hostname# install add tftp://192.1.1.20/smu_rf_hot1002_0118.bin
< The terminal is locked by patch moduleed by patch module >
Press Ctrl+C to quit
!
Operating, please wait for a moment.....
Operate finish!
Operate result information:
-----
Slot          Result          Comment
  
```

```

1/0          Success      None
2/0          Success      None

< The terminal is unlocked by patch module >

```

(3) Activate a patch.

- Run the **install** command to activate the patch.

a Run the **install active** command to activate the hot patch temporarily.

```

Hostname# install activate smu_rf_hot1002_0118.bin
< The terminal is locked by patch module >
Operating, please wait for a moment.....
Operate finish!
Operate result information:
-----
1/0          Success      None
2/0          Success      None

< The terminal is unlocked by patch module >

```

b Run the **install commit** command to activate the hot patch permanently.

```

Hostname# install commit
< The terminal is locked by patch module >
Operating, please wait for a moment...
Operate finish!
Operate result information:
-----
Slot          Result      Comment
1/0           Success      None
2/0           Success      None

< The terminal is unlocked by patch module >
Hostname#

```

- Run the **patch** command to activate the patch.

c Run the **patch active** command to activate the hot patch temporarily.

```

Hostname# patch active
< The terminal is locked by patch module >
Operating, please wait for a moment.....
Operate finish!
Operate result information:
-----
1/0          Success      None
2/0          Success      None

< The terminal is unlocked by patch module >

```

d Run the **patch running** command to activate the hot patch permanently.

```

Hostname# patch running
< The terminal is locked by patch module >
Operating, please wait for a moment...
Operate finish!
Operate result information:
-----
Slot          Result      Comment
1/0           Success    None
2/0           Success    None

< The terminal is unlocked by patch module >

```

(4) Configure patch auto-sync.

```

Hostname# install sync enable

```

5. Verification

(1) Display the patch installation information.

Run the **show install** command to display information about the hot patch installed on the current device.

```

Hostname# show install
Install information:
 [Slot 1/0]
      Name                Package                State                Flag
Effective time
      smu_rf_hot1002_0118.bin                Running             Hot                2019-
09-17 19:00:02

 [Slot 2/0]
      Name                Package                State                Flag
Effective time
      smu_rf_hot1002_0118.bin                Running             Hot                2019-
09-17 19:00:02

```

Run the **show patch** command to display information about the hot patch installed on the current device.

```

Hostname# show patch
 [Slot 1/0]
      Patch name          : smu_rf_hot1002_0118
      File name           : smu_rf_hot1002_0118.bin
      Patch state         : Install
      Patch flag          : Hot
      Effective time      :
      Last patch          :

 [Slot 2/0]
      Patch name          : smu_rf_hot1002_0118
      File name           : smu_rf_hot1002_0118.bin
      Patch state         : Install
      Patch flag          : Hot

```



```
Effective time      :
Last patch         :
```

6. Common Errors

- If you run the **install commit** command when a patch is not activated, an error is prompted. The **install commit** command takes effect only when the patch is in the active state.
- If you run the **patch running** command when a patch is not activated, an error is prompted. The **patch running** command takes effect only when the patch is in the active state.

1.8.4 Uninstalling a Patch Package

1. Requirements

One function component in one device can run one patch only. To install a new patch for this function component, you need to uninstall the previous patch before installation.

2. Notes

- (1) Run the patch deactivation command.
- (2) Run the patch uninstall command.

3. Procedure

- (1) Uninstall a patch.

- Run the **install** command to uninstall a patch.

a Run the **install deactivate** command to deactivate the patch.

```
Hostname# install deactivate smu_rf_hot1002_0118.bin
< The terminal is locked by patch module >
Operating, please wait for a moment.....
Operate finish!
Operate result information:
-----
1/0          Success      None
2/0          Success      None
< The terminal is unlocked by patch module >
```

b Run the **install remove** command to uninstall the patch.

```
Hostname#install remove smu_rf_hot1002_0118.bin
< The terminal is locked by patch module >
Operating, please wait for a moment...
Operate finish!
Operate result information:
-----
Slot         Result      Comment
1/0          Success     None
2/0          Success     None
< The terminal is unlocked by patch module >
```

- Run the **patch** command to uninstall a patch.

c Run the **patch deactivate** command to deactivate the patch.

```
Hostname# patch deactivate
< The terminal is locked by patch module >
Operating, please wait for a moment.....
Operate finish!
Operate result information:
-----
1/0          Success      None
2/0          Success      None

< The terminal is unlocked by patch module >
```

d Run the **patch delete** command to uninstall the patch.

```
Hostname# patch delete
< The terminal is locked by patch module >
Operating, please wait for a moment...
Operate finish!
Operate result information:
-----
Slot          Result      Comment
1/0           Success    None
2/0           Success    None

< The terminal is unlocked by patch module >
```

4. Verification

(1) Display the patch status.

Run the **show install** command to display the patch uninstallation result.

```
Hostname# show install
[Slot 1/0]
    [No Install information]

[Slot 2/0]
    [No Install information]
```

Run the **show patch** command to display the patch uninstallation result.

```
Hostname# show patch
[Slot 1/0]
    [No Install information]

[Slot 2/0]
    [No Install information]
```