



System Setup and Software Installation Guide for Cisco NCS 1020, IOS XR Release 24.3.x

First Published: 2024-08-30

Americas Headquarters

Cisco Systems, Inc.
170 West Tasman Drive
San Jose, CA 95134-1706
USA
<http://www.cisco.com>
Tel: 408 526-4000
800 553-NETS (6387)
Fax: 408 527-0883

THE SPECIFICATIONS AND INFORMATION REGARDING THE PRODUCTS IN THIS MANUAL ARE SUBJECT TO CHANGE WITHOUT NOTICE. ALL STATEMENTS, INFORMATION, AND RECOMMENDATIONS IN THIS MANUAL ARE BELIEVED TO BE ACCURATE BUT ARE PRESENTED WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED. USERS MUST TAKE FULL RESPONSIBILITY FOR THEIR APPLICATION OF ANY PRODUCTS.

THE SOFTWARE LICENSE AND LIMITED WARRANTY FOR THE ACCOMPANYING PRODUCT ARE SET FORTH IN THE INFORMATION PACKET THAT SHIPPED WITH THE PRODUCT AND ARE INCORPORATED HEREIN BY THIS REFERENCE. IF YOU ARE UNABLE TO LOCATE THE SOFTWARE LICENSE OR LIMITED WARRANTY, CONTACT YOUR CISCO REPRESENTATIVE FOR A COPY.

The Cisco implementation of TCP header compression is an adaptation of a program developed by the University of California, Berkeley (UCB) as part of UCB's public domain version of the UNIX operating system. All rights reserved. Copyright © 1981, Regents of the University of California.

NOTWITHSTANDING ANY OTHER WARRANTY HEREIN, ALL DOCUMENT FILES AND SOFTWARE OF THESE SUPPLIERS ARE PROVIDED "AS IS" WITH ALL FAULTS. CISCO AND THE ABOVE-NAMED SUPPLIERS DISCLAIM ALL WARRANTIES, EXPRESSED OR IMPLIED, INCLUDING, WITHOUT LIMITATION, THOSE OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NON-INFRINGEMENT OR ARISING FROM A COURSE OF DEALING, USAGE, OR TRADE PRACTICE.

IN NO EVENT SHALL CISCO OR ITS SUPPLIERS BE LIABLE FOR ANY INDIRECT, SPECIAL, CONSEQUENTIAL, OR INCIDENTAL DAMAGES, INCLUDING, WITHOUT LIMITATION, LOST PROFITS OR LOSS OR DAMAGE TO DATA ARISING OUT OF THE USE OR INABILITY TO USE THIS MANUAL, EVEN IF CISCO OR ITS SUPPLIERS HAVE BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

Any Internet Protocol (IP) addresses and phone numbers used in this document are not intended to be actual addresses and phone numbers. Any examples, command display output, network topology diagrams, and other figures included in the document are shown for illustrative purposes only. Any use of actual IP addresses or phone numbers in illustrative content is unintentional and coincidental.

All printed copies and duplicate soft copies of this document are considered uncontrolled. See the current online version for the latest version.

Cisco has more than 200 offices worldwide. Addresses and phone numbers are listed on the Cisco website at www.cisco.com/go/offices.

Cisco and the Cisco logo are trademarks or registered trademarks of Cisco and/or its affiliates in the U.S. and other countries. To view a list of Cisco trademarks, go to this URL: <https://www.cisco.com/c/en/us/about/legal/trademarks.html>. Third-party trademarks mentioned are the property of their respective owners. The use of the word partner does not imply a partnership relationship between Cisco and any other company. (1721R)

© 2024 Cisco Systems, Inc. All rights reserved.



CONTENTS

CHAPTER 1

Bring-up Cisco NCS 1020 1

Boot NCS 1020 1

Boot NCS 1020 Using USB Drive 2

DHCP Configuration 3

Introduction to DHCP Relay 4

Configuring and Enabling the DHCP Relay Agent 4

Limitations for DHCP Relay Feature 5

Configuring and Enabling the DHCP Relay Agent 5

DHCP Client 5

Boot Using iPXE 6

Setup DHCP Server 6

Boot Using iPXE 8

Boot Using Zero Touch Provisioning 11

Fresh Boot Using DHCP 12

Build your Configuration File 14

Configure ZTP BootScript 15

Invoke ZTP Manually through CLI 16

Invoke ZTP Through Reload 17

ZTP Logging 19

Generate Tech Support Information for ZTP 20

Configure Management Interface 20

Link Layer Discovery Protocol Support on Management Interface 22

Configure Telnet 25

Configure SSH 26

CHAPTER 2

Perform Preliminary Checks 29

- Verify Status of Hardware Components 29
- Verify Inventory 31
- Verify Software Version 33
- Verify Firmware Version 34
- Verify Management Interface Status 37
- Verify Alarms 39
- Verify Environmental Parameters 39
- Verify Context 55
- Verify Core Files 55
- Verify Memory Information 56

CHAPTER 3

Upgrade Software and FPD 57

- Plan the Software Upgrade 57
 - Backup Current Configuration 57
 - Check System Stability 58
 - Obtain Install Files 58
 - Standard ISO and RPMs 58
- Upgrade the Software 59
 - Upgrade NCS 1020 Using CLI Commands 59
 - Upgrade NCS 1020 Using YANG Data Models 60
- Install IOS XR Image 63
 - Install ISO and RPMs 63
 - Install Golden ISO 64
- Verify the Software Upgrade 68
 - Check System Stability 68
- NCS 1020 FPD 68
 - Verify if an FPD Upgrade is Required 72
 - Upgrade FPDs Manually 75
 - Upgrade FPDs Automatically 76

CHAPTER 4

Disaster Recovery 77

- Overview 77
- SSD OIR 77
- CPU Controller Replacement Considerations 77

Chassis SSD Replacement Considerations 78

Health Check of Backup ISO Image 78

CHAPTER 5**Connection Verification 81**

Power Data Reading 81

Connection Verification 81

CCMD-16 Connection Verification with OLT 82

Verify Connection for CCMD-16 Line Card 82

Connection Verification on OTS Controller 84

CHAPTER 6**Smart Licensing 87**

Understanding Smart Licensing 87

License Entitlements of NCS 1020 89

Create an ID Token 90

Smart Licensing Transport Modes 91

Configure Callhome 91

Configure Smart 92

Configure CSLU 93

Configure Offline 94

Reserve Specific Licenses for NCS 1020 95



CHAPTER 1

Bring-up Cisco NCS 1020

After installing the hardware, boot the Cisco NCS 1020 system. You can connect to the XR console port and power on the system. NCS 1020 completes the boot process using the pre-installed operating system (OS) image. If no image is available, NCS 1020 can be booted using the iPXE boot or an external bootable USB drive.

After booting, create the root username and password, and then use it to log on to the XR console.



Note The output of the examples in the procedures is not from the latest software release. The output will change for any explicit references to the current release.

- [Boot NCS 1020, on page 1](#)
- [Configure Management Interface, on page 20](#)
- [Configure Telnet, on page 25](#)
- [Configure SSH, on page 26](#)

Boot NCS 1020

Use the console port to connect to NCS 1020. By default, the console port connects to the XR mode. If necessary, you can establish subsequent connections through the management port, after it is configured.

Step 1 Connect a terminal to the console port of the RP.

Step 2 Start the terminal emulation program on your workstation.

The console settings are 115200 bps, for NCS1010-CTR2-K9, 9600 bps for NCS1010-CTR2-B-K9, 8 data bits, 1 stop bit and no parity.

Step 3 Power on NCS 1020.

To power on the shelves, install the AC or DC power supplies and cables. As NCS 1020 boots up, you can view the boot process details at the console of the terminal emulation program.

Step 4 Press **Enter**.

The boot process is complete when the system prompts you to enter the root-system username. If the prompt does not appear, wait for a while to give NCS 1020 more time to complete the initial boot procedure; then press **Enter**.

Important If the boot process fails, it may be because the preinstalled image on the NCS 1020 is corrupt. In this case, you can boot NCS 1020 using an external bootable USB drive.

Boot NCS 1020 Using USB Drive

The bootable USB drive is used to reimage NCS 1020 for system upgrade or to boot the NCS 1020 in case of boot failure. A bootable USB drive is created by copying a compressed boot file into a USB drive. The USB drive becomes bootable after the contents of the compressed file are extracted.

You can complete this task using the Windows, Linux, or MAC operating systems available on your local machine. The exact operation to be performed for each generic step that is outlined here depends on the operating system in use.

Use this task to boot the NCS 1020 using the USB drive.

Before you begin

- You need a USB drive with a storage capacity of at least 4 GB.
- The USB drive should have a single partition.
- NCS 1020 software image can be downloaded from Software Download page on Cisco.com.



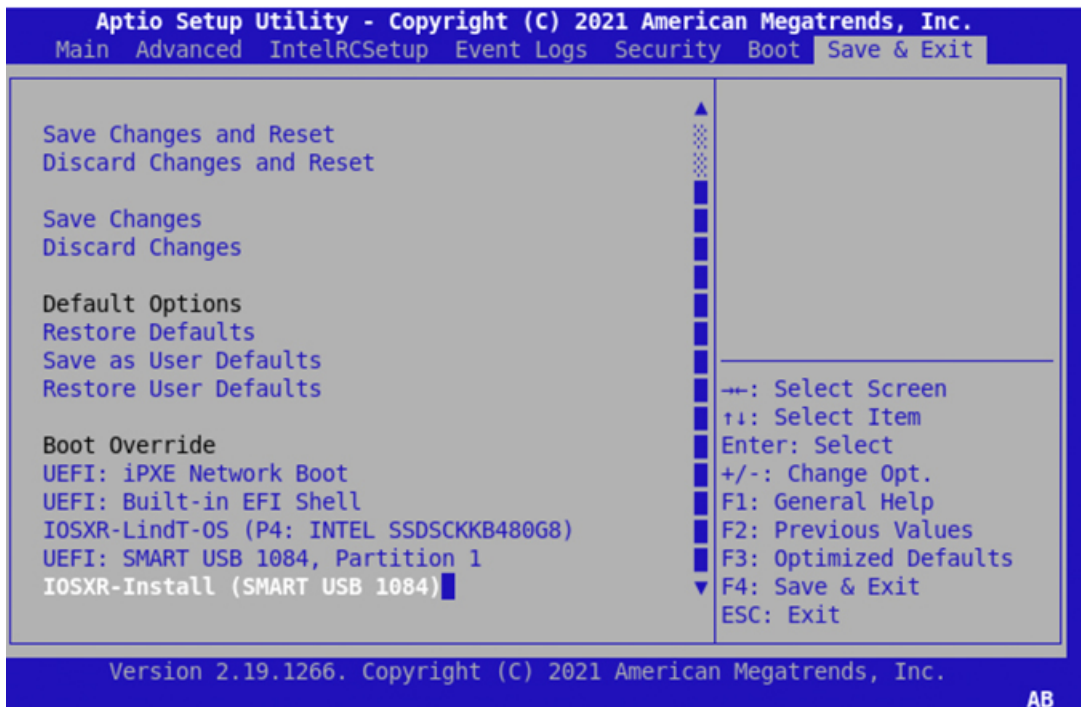
Note Since the NCS 1020 system uses the same XR image as NCS 1010, there are no changes in the USB bootable image name used. Hence the USB bootable image name is the same as NCS 1010.

- Copy the compressed boot file from the software download page at Cisco.com to your local machine. The filename for the compressed boot file is in the format *ncs1010-usb-boot-<release_number>.zip*.

-
- Step 1** Connect the USB drive to your local machine and format it with the FAT32 file system.
- Step 2** Copy the compressed boot file to the USB drive.
- Step 3** Verify that the copy operation is successful. To verify, compare the file size at source and destination. Also, verify the MD5 checksum value.
- Step 4** Extract the content of the compressed boot file by unzipping it in the USB drive. This makes the USB drive a bootable drive.
- Note** You must extract the contents of the zipped file ("EFI" and "boot" directories) directly in the root folder of the USB drive. If the unzipping application places the extracted files in a new folder, move the "EFI" and "boot" directories to the root folder of the USB drive.
- Step 5** Insert the USB drive in one of the USB ports of NCS 1020 line card/controller card.
- Step 6** Reboot NCS 1020 using power cycle or console.
- Note** Use the **reload bootmedia usb noprompt** command to boot the NCS 1020 from the USB. If you are using the **reload bootmedia usb noprompt** command, then you can skip the remaining steps.

Step 7 Press **Esc** to enter BIOS.

Step 8 Select the **Save & Exit** tab of BIOS.



Step 9 Choose **IOS -XR Install**.

The BIOS UI displays the USB drive vendor in the brackets, in this case, SMART USB 1084.

The system detects USB and boots the image from USB.

```

Booting from USB..
Loading Kernel..
Verifying (loop)/boot/bzImage...
(loop)/boot/bzImage verified using attached signature.
Loading initrd..
Verifying (loop)/boot/initrd.img...
  
```

Step 10 Remove the USB drive after the Rebooting the system after installation message is displayed. The NCS 1020 reboots automatically.

Note The USB must be removed only after the image is loaded successfully.

DHCP Configuration

DHCP configuration is required for both manual configuration and ZTP configuration. Follow the below sections to set up DHCP for booting NCS 1020 using ZTP and iPXE.

Introduction to DHCP Relay

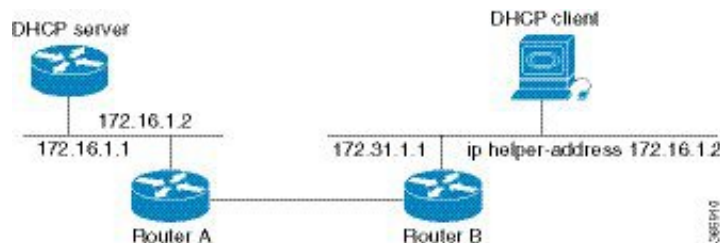
A DHCP relay agent is a host that forwards DHCP packets between clients and servers that do not reside on a shared physical subnet. Relay agent forwarding is distinct from the normal forwarding of an IP router where IP datagrams are switched between networks transparently.

DHCP clients use User Datagram Protocol (UDP) broadcasts to send DHCPDISCOVER messages when they lack information about the network to which they belong.

If a client is on a network segment that does not include a server, a relay agent is needed on that network segment to ensure that DHCP packets reach the servers on another network segment. UDP broadcast packets are not forwarded, because most routers are not configured to forward broadcast traffic. You can configure a DHCP relay agent to forward DHCP packets to a remote server by configuring a DHCP relay profile and configure one or more helper addresses in it. You can assign the profile to an interface or a VRF.

The figure below demonstrates the process. The DHCP client broadcasts a request for an IP address and additional configuration parameters on its local LAN. Acting as a DHCP relay agent, Router B picks up the broadcast, changes the destination address to the DHCP server's address and sends the message out on another interface. The relay agent inserts the IP address of the interface, on which the DHCP client's packets are received into the gateway address (giaddr) field of the DHCP packet, which enables the DHCP server to determine which subnet should receive the offer and identify the appropriate IP address range. The relay agent unicasts the messages to the server address, in this case 172.16.1.2 (which is specified by the helper address in the relay profile).

Figure 1: Forwarding UDP Broadcasts to a DHCP Server Using a Helper Address



Configuring and Enabling the DHCP Relay Agent

Configuration Example

```
RP/0/RP0/CPU0:ios# configure
RP/0/RP0/CPU0:ios(config)# dhcp ipv4
RP/0/RP0/CPU0:ios(config-dhcpv4)# profile r1 relay
RP/0/RP0/CPU0:ios(config-dhcpv4-relay-profile)# helper-address vrf default 198.51.100.1
giaddr 198.51.100.3
RP/0/RP0/CPU0:ios(config-dhcpv4-relay-profile)# !
RP/0/RP0/CPU0:ios(config-dhcpv4-relay-profile)# interface GigabitEthernet0/0/0/2 relay
profile r1
RP/0/RP0/CPU0:ios(config-dhcpv4)# commit
```

Running Configuration

```
RP/0/RP0/CPU0:ios# show running-config dhcp ipv4
Tue Aug 29 07:30:50.677 UTC
dhcp ipv4
  profile r1 relay
    helper-address vrf default 198.51.100.1 giaddr 198.51.100.3
  !
```

```
interface GigabitEthernet0/0/0/2 relay profile r1
!
```

Limitations for DHCP Relay Feature

These are the limitations for implementing DHCP relay feature:

- The multicast addresses are not supported. The **helper-address** command in DHCP relay profile submode supports valid unicast IP address as the helper address.



Note Configuring the **helper-address** command directly (not using profile) under a interface (such as BVI interface) is not supported.

- Only interface-id and remote-id DHCP option code are added by a relay agent while forwarding the packet to a DHCP server.



Note Configuring DHCP option code is not supported in DHCP relay profile submode.

Configuring and Enabling the DHCP Relay Agent

Configuration Example

```
RP/0/RP0/CPU0:ios# configure
RP/0/RP0/CPU0:ios(config)# dhcp ipv4
RP/0/RP0/CPU0:ios(config-dhcpv4)# profile r1 relay
RP/0/RP0/CPU0:ios(config-dhcpv4-relay-profile)# helper-address vrf default 198.51.100.1
giaddr 198.51.100.3
RP/0/RP0/CPU0:ios(config-dhcpv4-relay-profile)# !
RP/0/RP0/CPU0:ios(config-dhcpv4-relay-profile)# interface GigabitEthernet0/0/0/2 relay
profile r1
RP/0/RP0/CPU0:ios(config-dhcpv4)# commit
```

Running Configuration

```
RP/0/RP0/CPU0:ios# show running-config dhcp ipv4
Tue Aug 29 07:30:50.677 UTC
dhcp ipv4
  profile r1 relay
    helper-address vrf default 198.51.100.1 giaddr 198.51.100.3
  !
  interface GigabitEthernet0/0/0/2 relay profile r1
  !
```

DHCP Client

The Dynamic Host Configuration Protocol (DHCP) client functionality enables the router interfaces to dynamically acquire the IPv4 or DHCPv4 or DHCPv6 server, and forwards the responses back to the correct Layer 2 address so that the correct device gets the correct configuration information.

DHCP has the ability to allocate IP addresses only for a configurable period of time, called the lease period. If the client is required to retain this IP address for a longer period beyond the lease period, the lease period

must be renewed before the IP address expires. The client renews the lease based on configuration that was sent from the server. The client unicasts a REQUEST message using the IP address of the server. When a server receives the REQUEST message and responds with an ACK message. The lease period of the client is extended by the lease time configured in the ACK message.

Enabling DHCP Client on an Interface

You can enable both the DHCPv4 and DHCPv6 clients at an interface level. The DHCP component receives a notification when DHCPv4 or DHCPv6 is enabled or disabled on an interface.

```
RP/0/RP0/CPU0:ios# configure
Tue Aug 29 09:26:12.468 UTC
RP/0/RP0/CPU0:ios(config)# int mgmtEth 0/RP0/CPU0/0
RP/0/RP0/CPU0:ios(config-if)# ipv4 address dhcp
RP/0/RP0/CPU0:ios(config-if)# commit
Tue Aug 29 09:26:21.715 UTC
RP/0/RP0/CPU0:ios(config-if)# exit
RP/0/RP0/CPU0:ios(config)# int mgmtEth 0/RP0/CPU0/0
RP/0/RP0/CPU0:ios(config-if)# ipv6 address dhcp
dhcp dhcp-client-options
RP/0/RP0/CPU0:ios(config-if)# ipv6 address dhcp
RP/0/RP0/CPU0:ios(config-if)# commit
Tue Aug 29 09:26:50.159 UTC
```

Boot Using iPXE

iPXE is a pre-boot execution environment that is included in the network card of the management interfaces and works at the system firmware (UEFI) level of the chassis. iPXE is used to reimage the system, and boot the chassis in case of boot failure or in the absence of a valid bootable partition. iPXE downloads the ISO image, proceeds with the installation of the image, and finally bootstraps inside the new installation.



Note The time taken for iPXE to download the ISO image depends on the network speed. Ensure that the network speed is sufficient to complete the image download in less than 10 minutes. The chassis reloads if the image is not downloaded by 10 minutes.

iPXE acts as a bootloader and provides the flexibility to choose the image that the system will boot based on the Platform Identifier (PID), the Serial Number, or the management MAC-address. You must define iPXE in the DHCP server configuration file.



Note To initiate the iPXE boot process, perform one of the following methods:

- Use the **reload bootmedia network location all** command. This method is the preferred method.
- Power cycle the NCS 1020 chassis and start the iPXE boot process in the BIOS interface.

Setup DHCP Server

A DHCP server must be configured for IPv4, IPv6, or both communication protocols.



Note For DHCPv6, a routing advertisement (RA) message must be sent to all nodes in the network that indicates which method is to be used to obtain the IPv6 address. Configure Router-advertise-daemon (radvd, install using `yum install radvd`) to allow the client to send the DHCP request. For example:

```
interface eth3
{
    AdvSendAdvert on;
    MinRtrAdvInterval 60;
    MaxRtrAdvInterval 180;
    AdvManagedFlag on;
    AdvOtherConfigFlag on;
    prefix 2001:1851:c622:1::/64
    {
        AdvOnLink on;
        AdvAutonomous on;
        AdvRouterAddr off;
    };
};
```

To setup a DHCP server:

1. Create the `dhcpd.conf` file (for IPv4, IPv6 or both communication protocols), `dhcpv6.conf` file (for IPv6) or both in the `/etc/` directory. This configuration file stores the network information such as the path to the script, location of the ISO install file, location of the provisioning configuration file, serial number, MAC address of the chassis.
2. Test the server once the DHCP server is running:

For example, for ipv4:

- a. Use MAC address of the chassis:

```
host ncs1020
{
    hardware ethernet ab:cd:ef:01:23:45;
    fixed-address <ip address>;
    filename "http://<httpserver-address>/<path-to-image>/ncs1020-mini-x.iso";
}
```

Ensure that the above configuration is successful.

- b. Use serial number of the chassis:

```
host demo {
    option dhcp-client-identifier "<chassis-serial-number>";
    filename "http://<IP-address>/<hardware-platform>-mini-x.iso";
    fixed-address <IP-address>;
}
```

The serial number of the chassis is derived from the BIOS and is used as an identifier.

Example

```
host 10.89.205.202 {
    hardware ethernet 40:55:39:56:0c:e8;
    option dhcp-client-identifier "<FCB2437B066>";
    if exists user-class and option user-class = "iPXE" {
        filename "http://10.89.205.127/box1/ncs1020-x64.iso";
    } else {
        filename "http://10.89.205.127/box1/StartupConfig.cfg";
    }
}
```

```
        fixed-address 10.89.205.202;
    }
}
```

Boot Using iPXE

Before you use the iPXE boot, ensure that:

- DHCP server is set and is running.
- Management port of the NCS 1020 chassis is in *UP* state.

Use any one of the following methods to invoke the iPXE boot process:

- via CLI terminal:

Run the following command to invoke the iPXE boot process to reimage the chassis:

```
reload bootmedia network location all
```

Example:

```
RP/0/RP0/CPU0:ios# reload bootmedia network location all
Wed Jul  6 15:11:33.791 UTC
Reload hardware module ? [confirm]
```

The following example shows the output of the command:

```
Preparing system for backup. This may take a few minutes especially for large
configurations.
```

```
        Status report: node0_RP0_CPU0: BACKUP INPROGRESS
RP/0/RP0/CPU0:PlD_DT#  Status report: node0_RP0_CPU0: BACKUP HAS COMPLETED SUCCESSFULLY
```

```
[Done]
[FAILED] Failed unmounting /mnt/fuse/parser_server.
[ OK ] Unmounted /mnt/fuse/ftp.
[ OK ] Unmounted /mnt/fuse/nvgen_server.
[ OK ] Unmounted /boot/efi.
[ OK ] Unmounted /selinux.
```

```
.
.
Output Snipped
```

```
.
..          *** Sirius ***
System Initializing..
..
```

```
ERROR: Class:0; Subclass:10000; Operation: 1004
```

```
Shelf Assembly Reset
Shelf Assembly Reset for P1
```

```
..          *** Sirius ***
System Initializing..
..
```

```
ERROR: Class:0; Subclass:10000; Operation: 1004
```

```
.
.
Output Snipped
```

```
.
.
```

```

NCS1010, Initializing Devices

Booting from Primary Flash
Aldrin: Programmed MI 10
.
.
Output Snipped
.
.
Version 2.19.1266. Copyright (C) 2022 American Megatrends, Inc.
BIOS Date: 05/20/2022 10:47:39 Ver: 0ACHI0410
Press <DEL> or <ESC> to enter setup.
TAM Chipguard Validate Observed DB Error: 0x48

WARNING!!! TAM: Empty Chip DB

Software Boot OK, Validated

iPXE initialising devices...ok

iPXE 1.0.0+ (c2215) -- Open Source Network Boot Firmware -- http://ipxe.org
Features: DNS HTTP TFTP VLAN EFI ISO9660 ISO9660_grub Menu
Trying net0-2051,net0-2052 and net0-2053...
net0-2051: 68:9e:0b:b8:71:1e using NII on NII-PCI06:00.0 (open)
  [Link:down, TX:0 TXE:0 RX:0 RXE:0]
  [Link status: Unknown (http://ipxe.org/1a086194)]
Configuring (net0-2051 68:9e:0b:b8:71:1e)..... Error 0x040ee186
(http://ipxe.org/040ee186)
net0-2052: 68:9e:0b:b8:71:1f using NII on NII-PCI06:00.0 (open)
  [Link:up, TX:0 TXE:0 RX:18 RXE:14]
  [RXE: 8 x "Operation not supported (http://ipxe.org/3c086083)"]
  [RXE: 3 x "Error 0x440e6083 (http://ipxe.org/440e6083)"]
  [RXE: 3 x "The socket is not connected (http://ipxe.org/380f6093)"]
Configuring (net0-2052 68:9e:0b:b8:71:1f)..... ok
net0: fe80::6a9e:bff:feb8:711e/64
net1: fe80::6a9e:bff:feb8:7121/64 (inaccessible)
net2: fe80::6a9e:bff:feb8:7122/64 (inaccessible)
net3: fe80::6a9e:bff:feb8:7123/64 (inaccessible)
net0-2051: fe80::6a9e:bff:feb8:711e/64
net0-2051: 2001:420:5446:2014::281:0/119 gw fe80::676:b0ff:fed8:c100 (no address)
net0-2051: 2002:420:54ff:93:6a9e:bff:feb8:711e/64 gw fe80::fa4f:57ff:fe72:a640
net0-2052: 10.4.33.44/255.255.0.0 gw 10.4.33.1
net0-2052: fe80::6a9e:bff:feb8:711e/64
net0-2053: fe80::6a9e:bff:feb8:711e/64
Filename: http://10.4.33.51/P1D_DT_05/ncs1010-x64.iso
http://10.4.33.51/P1D_DT_05/ncs1010-x64.iso... ok
.
.
Output Snipped
.
.
User Access Verification

Username: cisco
Password:

```

- via BIOS interface:

1. Reboot NCS 1020 using power cycle or console.

2. Press **Esc** to enter BIOS.
3. Select the **Save & Exit** tab of BIOS.
4. Choose **UEFI: iPXE Network Boot**.

The following example shows the output of the command:

```

Preparing system for backup. This may take a few minutes especially for large
configurations.
      Status report: node0_RP0_CPU0: BACKUP INPROGRESS
RP/0/RP0/CPU0:PID_DT#  Status report: node0_RP0_CPU0: BACKUP HAS COMPLETED
SUCCESSFULLY
[Done]
[FAILED] Failed unmounting /mnt/fuse/parser_server.
[ OK ] Unmounted /mnt/fuse/ftp.
[ OK ] Unmounted /mnt/fuse/nvgen_server.
[ OK ] Unmounted /boot/efi.
[ OK ] Unmounted /selinux.
.
.
Output Snipped
.
.
..          *** Sirius ***
System Initializing..
..

ERROR: Class:0; Subclass:10000; Operation: 1004

Shelf Assembly Reset
Shelf Assembly Reset for P1

..          *** Sirius ***
System Initializing..
..

ERROR: Class:0; Subclass:10000; Operation: 1004
.
.
Output Snipped
.
.

NCS1010, Initializing Devices

Booting from Primary Flash
Aldrin: Programmed MI 10
.
.
Output Snipped
.
.
Version 2.19.1266. Copyright (C) 2022 American Megatrends, Inc.
BIOS Date: 05/20/2022 10:47:39 Ver: 0ACHI0410
Press <DEL> or <ESC> to enter setup.
TAM Chipguard Validate Observed DB Error: 0x48

WARNING!!! TAM: Empty Chip DB

Software Boot OK, Validated

```



```

iPXE initialising devices...ok

iPXE 1.0.0+ (c2215) -- Open Source Network Boot Firmware -- http://ipxe.org
Features: DNS HTTP TFTP VLAN EFI ISO9660 ISO9660_grub Menu
Trying net0-2051,net0-2052 and net0-2053...
net0-2051: 68:9e:0b:b8:71:1e using NII on NII-PCI06:00.0 (open)
  [Link:down, TX:0 TXE:0 RX:0 RXE:0]
  [Link status: Unknown (http://ipxe.org/1a086194)]
Configuring (net0-2051 68:9e:0b:b8:71:1e)..... Error 0x040ee186
(http://ipxe.org/040ee186)
net0-2052: 68:9e:0b:b8:71:1f using NII on NII-PCI06:00.0 (open)
  [Link:up, TX:0 TXE:0 RX:18 RXE:14]
  [RXE: 8 x "Operation not supported (http://ipxe.org/3c086083)"]
  [RXE: 3 x "Error 0x440e6083 (http://ipxe.org/440e6083)"]
  [RXE: 3 x "The socket is not connected (http://ipxe.org/380f6093)"]
Configuring (net0-2052 68:9e:0b:b8:71:1f)..... ok
net0: fe80::6a9e:bff:feb8:711e/64
net1: fe80::6a9e:bff:feb8:7121/64 (inaccessible)
net2: fe80::6a9e:bff:feb8:7122/64 (inaccessible)
net3: fe80::6a9e:bff:feb8:7123/64 (inaccessible)
net0-2051: fe80::6a9e:bff:feb8:711e/64
net0-2051: 2001:420:5446:2014::281:0/119 gw fe80::676:b0ff:fed8:c100 (no address)
net0-2051: 2002:420:54ff:93:6a9e:bff:feb8:711e/64 gw fe80::fa4f:57ff:fe72:a640
net0-2052: 10.4.33.44/255.255.0.0 gw 10.4.33.1
net0-2052: fe80::6a9e:bff:feb8:711e/64
net0-2053: fe80::6a9e:bff:feb8:711e/64
Filename: http://10.4.33.51/P1D_DT_05/ncs1010-x64.iso
http://10.4.33.51/P1D_DT_05/ncs1010-x64.iso... ok
.
.
Output Snipped
.
.
User Access Verification

Username: cisco
Password:

```

Boot Using Zero Touch Provisioning

ZTP allows you to provision the network device with day 0 configurations and supports both management ports and data ports.

ZTP provides multiple options, such as:

- Automatically apply specific configuration in a large-scale environment.
- Download and install specific IOS XR image.
- Install specific application package or third party applications automatically.
- Deploy containers without manual intervention.
- Upgrade or downgrade software versions effortlessly on thousands of network devices at a time

Benefits of Using ZTP

ZTP helps you manage large-scale service providers infrastructures effortlessly. Following are the added benefits of using ZTP:

- ZTP helps you to remotely provision a router anywhere in the network. Thus eliminates the need to send an expert to deploy network devices and reduces IT cost.
- Automated provisioning using ZTP can remove delay and increase accuracy and thus is cost-effective and provides better customer experience.

By automating repeated tasks, ZTP allows network administrators to concentrate on more important stuff.

- ZTP process helps you to quickly restore service. Rather than troubleshooting an issue by hand, you can reset a system to well-known working status.

Prerequisites:

ZTP does not execute, if a username is already configured in the system.

ZTP is initiated in one of the following ways:

- **Automated Fresh Boot:**

Fresh Boot: When you boot the device, the ZTP process initiates automatically if the device does not have a prior configuration. During the process, the router receives the details of the configuration file from the DHCP server. Use this method for devices that has no pre-loaded configuration. See [Fresh Boot Using DHCP, on page 12](#).

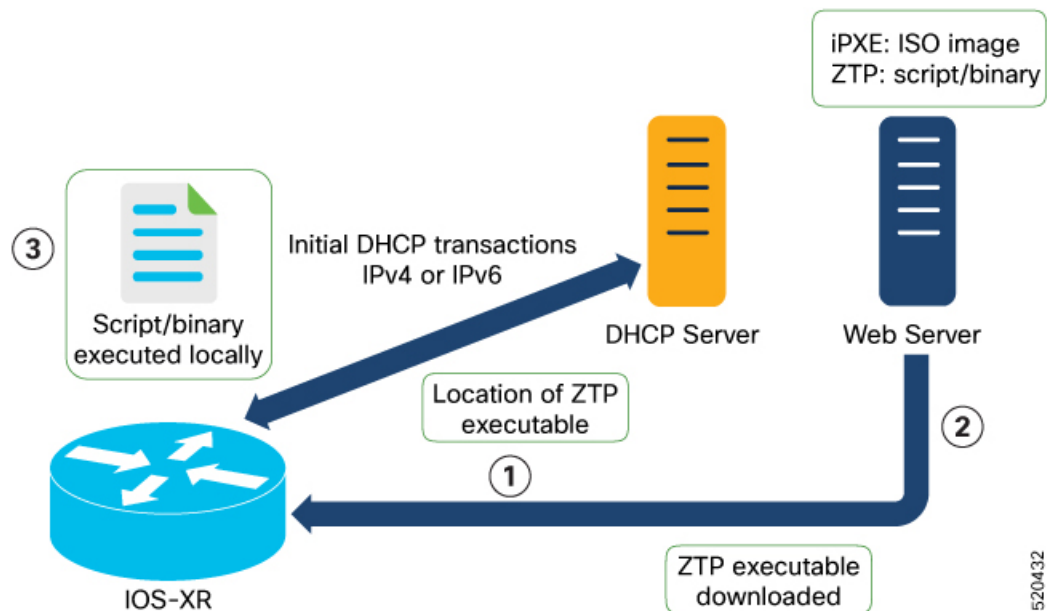
You must define the configuration file or the bootscript that is downloaded from the DHCP server:

- **Configuration File:** The first line of the file must contain **!! IOS XR configuration**", to process the file as a configuration. If you are trying to bring up ten new nodes, you have to define ten configuration files. See [Build your Configuration File, on page 14](#).
- **ZTP Bootscript:** Define the script to be executed on every boot. See [Configure ZTP BootScript, on page 15](#).
- **Manual Invocation using CLI:** Use this method when you want to forcefully initiate ZTP on a fully configured device, using CLI. See [Invoke ZTP Manually through CLI, on page 16](#).

Fresh Boot Using DHCP

The ZTP process initiates when you boot the network-device with an IOS-XR image. The process starts only on the device that doesn't have a prior configuration.

This image depicts the high-level work flow of the ZTP process:



1. ZTP sends DHCP request to fetch the ZTP configuration file or user script. To help the Bootstrap server uniquely identify the device, ZTP sends below DHCP option.
 - DHCP(v4/v6) client-id=Serial Number
 - DHCPv4 option 124: Vendor, Platform, Serial-Number
 - DHCPv6 option 16: Vendor, Platform, Serial-Number

The following is the default sequential flow of the ZTP process:

- ZTP sends IPv4 DHCP request first on all the management port. In case there is a failure, then ZTP sends IPv6 DHCP request on all the management port.
- ZTP sends IPv4 DHCP request first on all the data port. In case there is a failure, then ZTP sends IPv6 DHCP request on all the data port.

The default sequential flow is defined in configuration file and you can modify the sequence using the configuration file.

2. DHCP server identifies the device and responds with DHCP response using one of the following options: DHCP server should be configured to respond with the DHCP options.
 - DHCPv4 using BOOTP filename to supply script/config location.
 - DHCPv4 using Option 67 (bootfile-name) to supply script/config location.
 - DHCPv6 using Option 59 (OPT_BOOTFILE_URL) to supply script/config location
3. The network device downloads the file from the web server using the URL location that is provided in the DHCP response.
4. The device receives a configuration file or script file from the HTTP server.



-
- Note**
- If the downloaded file content starts with !! IOS XR it is considered as a configuration file.
 - If the downloaded file content starts with #!/bin/bash, #!/bin/sh or #!/usr/bin/python it is considered as a script file.
-

5. The device applies the configuration file or executes the script or binary in the default bash shell.
6. The Network device is now up and running.

Build your Configuration File

Based on the business need, you can use a configuration or script file to initiate the ZTP process.

The configuration file content starts with !! IOS XR.

The following is the sample configuration file. You can automate all the configurations. For more information on creating ZTP configuration file, refer [ZTP Configuration Files Creation](#).

```
Tue May 4 18:08:59.544 UTC
Building configuration...
!! IOS XR Configuration 7.7.1.22I
!! Last configuration change at Tue May 4 17:12:47 2021 by cisco
!
line console
exec-timeout 0 0
!
line default
exec-timeout 0 0
session-timeout 0
!
vty-pool default 0 20
alias alarms show alarms brief system active
interface MgmtEth0/RP0/CPU0/0
ipv4 address dhcp
no shut
!
interface MgmtEth0/RP0/CPU0/1
description noshut-interface-ztp
ipv4 address 10.127.60.160 255.255.255.0
no shut
!
interface MgmtEth0/RP0/CPU0/2
description noshut-interface-ztp
no shut
!
interface PTP0/RP0/CPU0/0
description noshut-interface-ztp
no shut
Cisco NCS 1020 System Setup and Software Installation Guide, IOS XR Release 7.7.x
19
Bring-up Cisco NCS 1020
Build your Configuration File
!
telnet vrf default ipv4 server max-servers 100a
ssh server v2
ssh server netconf vrf default
netconf-yang agent
ssh
```

```

!
netconf agent tty
grpc
router static
address-family ipv4 unicast
0.0.0.0/0 10.127.60.1
end

```

Configure ZTP BootScript

ZTP downloads and executes the script files. These script files include a programmatic approach to complete a task. For example, scripts created using IOS XR commands to perform patch upgrades. The first line of the file must contain `#!/bin/bash` or `#!/bin/sh` for ZTP to process the file as script. You can either use the ZTP bash script or the ZTP configuration file.

You can either use the ZTP bash script or the ZTP configuration file.

If you want to hardcode a script to be executed every boot, configure the following.

```

Router#configure
Router(config)#ztp bootscript /disk0:/myscript
Router(config)#commit

```

The above configuration waits for the first data-plane interface to be configured and then wait an extra minute for the management interface to be configured with an IP address, to ensure that we have connectivity in the third-party namespace for applications to use. If the delay is not desired, use:

```

Router#configure
Router(config)#ztp bootscript preip /disk0:/myscript
Router(config)#commit

```



Note When the above command is first configured, you will be prompted if you wish to invoke it now. The prompt helps with testing.

This is the example content of `/disk0:/myscript`:

```

host ncs1020_P1B_DT_08_ETH0 {
#hardware ethernet 68:9e:0b:b8:6f:5c ;
option dhcp-client-identifier "FCB2437B05N" ;
if exists user-class and option user-class = "iPXE" {
filename "http://10.33.0.51/P1B_DT_08/ncs1020-x64.iso";
} else {
filename "http://10.33.0.51/P1B_DT_08/startup.cfg";
}
}
fixed-address 10.33.0.19;
}

```

The following is the sample content of the ZTP bash script.

```

#!/bin/bash
#
# NCS1020 Demo Sample
# ZTP installation of config and day-0 SMU's
#
source ztp_helper

wget http://downloads.sourceforge.net/project/yourcode/application.tgz
#install the downloaded application.tgz

```

```
#Run XR CLI's from the script
`xrcmd "show version"`
```

The following is the sample content of the ZTP configuration file.

```
Tue May 4 18:08:59.544 UTC
Building configuration...
!! IOS XR Configuration 7.7.1.22I
!! Last configuration change at Tue May 4 17:12:47 2021 by cisco
!
line console
exec-timeout 0 0
!
line default
exec-timeout 0 0
session-timeout 0
!
vty-pool default 0 20
alias alarms show alarms brief system active
interface MgmtEth0/RP0/CPU0/0
ipv4 address dhcp
no shut
!
interface MgmtEth0/RP0/CPU0/1
description noshut-interface-ztp
ipv4 address 10.127.60.160 255.255.255.0
no shut
!
interface MgmtEth0/RP0/CPU0/2
description noshut-interface-ztp
no shut
!
interface PTP0/RP0/CPU0/0
description noshut-interface-ztp
no shut
!
interface PTP0/RP0/CPU0/1
description noshut-interface-ztp
no shut
end
```

Invoke ZTP Manually through CLI

Manual ZTP can be invoked through CLI commands. This manual way helps you to provision the router in stages. Ideal for testing out ZTP configuration without a reboot. If you want to invoke a ZTP on an interface (data ports or management port), you don't have to bring up and configure the interface first. You can execute the `ztp initiate` command, even if the interface is down, ZTP script brings it up and invoke `dhclient`. So ZTP could run over all interfaces no matter it is up or down.

Use the `ztp initiate`, `ztp terminate`, and `ztp clean` commands to force ZTP to run over more interfaces.

- `ztp initiate`—Invokes a new ZTP DHCP session. Logs can be found in `/disk0:/ztp/ztp.log`.
- `ztp terminate`—Terminates any ZTP session in progress.
- `ztp clean`—Removes only the ZTP state files.

The log file `ztp.log` is saved in `/var/log/ztp.log` folder, and a copy of log file is available at `/disk0:/ztp/ztp.log` location using a soft link. However, executing `ztp clean` clears files saved on disk and not on `/var/logztp.log` folder where current ZTP logs are saved. In order to have a log from current ZTP run, you must manually clear the ZTP log file from `/var/log/ztp.log` folder.

Step 1 (optional) **ztp clean****Example:**

```
RP/0/RP0/CPU0:ios#ztp clean
Fri Apr 29 06:49:29.760 UTC
This would remove all ZTP temporary files.
Would you like to proceed? [no]: yes
All ZTP operation files have been removed.
ZTP logs are present in /var/log/ztp*.log for logrotate.
Please remove manually if needed.
If you now wish ZTP to run again from boot, do 'conf t/commit replace' followed by reload.
```

Removes all the ZTP logs and saved settings.

Step 2 **ztp initiate****Example:**

```
RP/0/RP0/CPU0:ios#ztp initiate
Fri Jun 17 11:44:08.791 UTC
Initiating ZTP may change your configuration.
Interfaces might be brought up if they are in shutdown state
Would you like to proceed? [no]: yes
ZTP will now run in the background.
Please use "show logging" or look at /var/log/ztp.log to check progress.
RP/0/RP0/CPU0:ios#
```

Use the **show logging** command or see the /var/log/ztp.log to check progress.

Reboots the Cisco NCS 1020 system.

Step 3 (Optional) **ztp terminate****Example:**

```
RP/0/RP0/CPU0:ios#ztp terminate
Fri Apr 29 06:38:59.238 UTC
This would terminate active ZTP session if any (this may leave your system in a partially configured
state)
Would you like to proceed? [no]: yes
Terminating ZTP
No ZTP process running
```

Terminates the ZTP process.

Invoke ZTP Through Reload

The ZTP process can be automatically invoked by using the reload command.

Step 1 **configure****Example:**

```
RP/0/RP0/CPU0:P2B_DT_02#configure
```

Enters the configuration mode.

Step 2 **commit replace**

Example:

```
Fri Apr 29 06:48:46.236 UTC
RP/0/RP0/CPU0:P2B_DT_02(config)#commit replace
Fri Apr 29 06:48:53.199 UTC
```

```
This commit will replace or remove the entire running configuration. This
operation can be service affecting.
Do you wish to proceed? [no]: yes
RP/0/RP0/CPU0:ios(config)#
RP/0/RP0/CPU0:ios(config)#end
```

Removes the entire running configuration.

Step 3 ztp clean**Example:**

```
RP/0/RP0/CPU0:ios#ztp clean
Fri Apr 29 06:49:29.760 UTC
This would remove all ZTP temporary files.
Would you like to proceed? [no]: yes
All ZTP operation files have been removed.
ZTP logs are present in /var/log/ztp*.log for logrotate.
Please remove manually if needed.
If you now wish ZTP to run again from boot, do 'conf t/commit replace' followed by reload.
```

Removes all the ZTP logs and saved settings.

Step 4 reload**Example:**

```
RP/0/RP0/CPU0:ios#reload
Fri Apr 29 06:50:12.312 UTC
Proceed with reload? [confirm]

RP/0/RP0/CPU0:ios#
Preparing system for backup. This may take a few minutes especially for large configurations.
Status report: node0_RP0_CPU0: BACKUP INPROGRESS
Status report: node0_RP0_CPU0: BACKUP HAS COMPLETED SUCCESSFULLY
[Done]
```

After the node comes up, you can check that the ZTP is initiated and the configuration has been restored successfully.

```
RP/0/RP0/CPU0:Apr 29 06:55:33.242 UTC: pyztp2[377]: %INFRA-ZTP-4-CONFIG_INITIATED : ZTP has initiated
config load and commit operations
RP/0/RP0/CPU0:Apr 29 06:55:39.263 UTC: ifmgr[381]: %PKT_INFRA-LINK-3-UPDOWN : Interface
GigabitEthernet0/0/0/0, changed state to Down
RP/0/RP0/CPU0:Apr 29 06:55:39.287 UTC: osa_driver[183]: %PKT_INFRA-FM-4-FAULT_MINOR : ALARM_MINOR
:PROV-INPROGRESS :DECLARE :GigabitEthernet0/0/0/0:
RP/0/RP0/CPU0:Apr 29 06:55:39.287 UTC: osa_driver[183]: %PKT_INFRA-FM-4-FAULT_MINOR : ALARM_MINOR
:PROV-INPROGRESS :DECLARE :Osc0/0/0/0:
RP/0/RP0/CPU0:Apr 29 06:55:39.287 UTC: ifmgr[381]: %PKT_INFRA-LINK-3-UPDOWN : Interface
GigabitEthernet0/0/0/0, changed state to Up
RP/0/RP0/CPU0:Apr 29 06:55:39.716 UTC: osa_driver[183]: %PKT_INFRA-FM-4-FAULT_MINOR : ALARM_MINOR
:PROV-INPROGRESS :CLEAR :Osc0/0/0/0:
RP/0/RP0/CPU0:Apr 29 06:55:39.728 UTC: osa_driver[183]: %PKT_INFRA-FM-4-FAULT_MINOR : ALARM_MINOR
:PROV-INPROGRESS :CLEAR :GigabitEthernet0/0/0/0:
RP/0/RP0/CPU0:Apr 29 06:55:47.904 UTC: osa_driver[183]: %PKT_INFRA-FM-4-FAULT_MINOR : ALARM_MINOR
:PROV-INPROGRESS :DECLARE :Ots0/0/0/1:

User Access Verification
```



```
Username: cisco
Password:
ios con0/RP0/CPU0 is now available
```

Reboots the Cisco NCS 1020 system.

ZTP Logging

ZTP logs its operation on the flash file system in the directory `/disk0:/ztp/`. ZTP logs all the transaction with the DHCP server and all the state transition.

The following example displays the execution of a simple configuration script downloaded from a data interface using the command `ztp initiate interface Ten 0/0/0/0 verbose`, this script unshuts all the interfaces of the system and configure a load interval of 30 seconds on all of them.

```
2022-06-17 11:52:34,682 19292 [Xr          ] INF: Downloading the file to /tmp/ztp.script
2022-06-17 11:52:35,329 19292 [Report       ] INF: User script downloaded successfully.
Provisioning in progress.
2022-06-17 11:52:35,330 19292 [Engine       ] DEB: ZAdmin, current state:active. Processing
work: Config device work for ZAdmin. done = False
2022-06-17 11:52:35,330 19292 [ZAdmin       ] DEB: Proceeding to provision the router
2022-06-17 11:52:35,331 19292 [Engine       ] DEB: ZAdmin, current state:active. Processing
work: ZAdmin: Apply configuration. done = False
2022-06-17 11:52:35,331 19292 [Engine       ] INF: ZAdmin, current state:active: state tag
changed to provision
RP/0/RP0/CPU0:Jun 17 11:52:35.341 UTC: pyztp2[140]: %INFRA-ZTP-4-CONFIG_INITIATED : ZTP has
initiated config load and commit operations
2022-06-17 11:52:35,339 19292 [Env          ] DEB: No MTU configs detected
2022-06-17 11:52:35,340 19292 [Engine       ] DEB: ZAdmin, current state:active. Processing
work: ZAdmin: Apply configuration. done = False
2022-06-17 11:52:35,354 19292 [Xr          ] DEB: Will apply the following config:
/disk0:/ztp/customer/config.candidate
2022-06-17 11:52:35,354 19292 [Xr          ] INF: Applying user configurations
2022-06-17 11:52:35,355 19292 [Configuration] INF: Provisioning via config replace
2022-06-17 11:52:54,656 19292 [Configuration] INF: Configuration has been applied
2022-06-17 11:52:54,656 19292 [Engine       ] DEB: ZAdmin, current state:active. Processing
work: Sending standby sync message. done = False
2022-06-17 11:52:54,663 19292 [Engine       ] DEB: ZAdmin, current state:active. Processing
work: [privileged] getting engine status. done = False
2022-06-17 11:52:54,664 19292 [Engine       ] DEB: ZAdmin, current state:active. Processing
work: ZAdmin: Execute post-configuration script. done = False
2022-06-17 11:52:55,212 19292 [Env          ] INF: Env::cleanup, success:True, exiting:False
2022-06-17 11:52:55,213 19292 [ZtpHelpers   ] DEB: Executing: source /pkg/bin/ztp_helper.sh
&& echo -ne | xrcmd "show running-config"
2022-06-17 11:52:55,825 19292 [Env          ] INF: Executing command ip netns exec
vrf-default /sbin/dhclient -4 -cf /etc/dhcp/dhclient.conf.ztp -lf
/var/lib/dhcp/dhclient.leases.ztp -sf /etc/dhcp/dhclient-script.ztp2 -r Mg0_RP0_CPU0_0 to
release IP
2022-06-17 11:52:56,968 19292 [Xr          ] INF: Removing linux route with ip 10.33.0.63
2022-06-17 11:52:57,023 19292 [Engine       ] INF: ZAdmin, current state:active, exit
code:sucess
2022-06-17 11:52:57,023 19292 [Engine       ] INF: ZAdmin, current state:final, exit
code:sucess: state changed to final
2022-06-17 11:52:59,737 19292 [Engine       ] DEB: ZAdmin, current state:final, exit
code:sucess. Processing work: Sending standby sync message. done = False
2022-06-17 11:52:59,738 19292 [Engine       ] WAR: ZAdmin, current state:final, exit
code:sucess: work is ignored: work=<desc='Sending standby sync message' done=False
priv=False>
2022-06-17 11:52:59,738 19292 [Engine       ] DEB: ZAdmin, current state:final, exit
code:sucess. Processing work: [privileged] getting engine status. done = False
```

```

2022-06-17 11:53:04,744 19292 [main      ] DEB: Moved to final state
2022-06-17 11:53:04,745 19292 [main      ] DEB: ZTP completed successfully
2022-06-17 11:53:04,745 19292 [main      ] INF: Exiting SUCCESSFULLY
2022-06-17 11:53:04,746 19292 [main      ] DEB: Exiting. Will not retry now.
2022-06-17 11:53:04,746 19292 [main      ] DEB: Shutting down adaptor. Cleanup False. Exiting
False
2022-06-17 11:53:04,748 19292 [Engine    ] DEB: ZAdmin, current state:final, exit
code:success. Processing work: [privileged] prepare engine shutdown. done = False
2022-06-17 11:53:04,849 19292 [Engine    ] DEB: ZAdmin, current state:final, exit
code:success. Processing work: [privileged] shutting down ZAdmin engine. done = False
2022-06-17 11:53:04,849 19292 [Engine    ] INF: ZAdmin, current state:final, exit
code:shutdown
2022-06-17 11:53:04,849 19292 [Engine    ] INF: ZAdmin, exit code:shutdown: state changed
to None
2022-06-17 11:53:04,849 19292 [Engine    ] DEB: ZAdmin, exit code:shutdown: breaking
engine loop after shutdown
2022-06-17 11:53:04,850 19292 [Engine    ] DEB: ZAdmin, exit code:shutdown: end of event
loop
2022-06-17 11:53:04,850 19292 [Adaptor   ] DEB: Adaptor : Cleanup for admin context on
Terminate
2022-06-17 11:53:06,119 19292 [main      ] INF: Exiting SUCCESSFULLY
2022-06-17 11:53:06,119 19292 [main      ] INF: ZTP Exited
RP/0/RP0/CPU0:Jun 17 11:53:06.119 UTC: pyztp2[140]: %INFRA-ZTP-4-EXITED : ZTP exited

```

Generate Tech Support Information for ZTP

When you have a problem in the ztp process that you cannot resolve, the resource of last resort is your Cisco Systems technical support representative. To analyze a problem, your technical support representative needs certain information about the situation and the symptoms that you are experiencing. To speed up the problem isolation and resolution process, collect the necessary data before you contact your representative.

Use the **show tech-support ztp** command to collect all debugging information of ztp process.

Example:

```

RP/0/RP0/CPU0:R1#show tech-support ztp
Thu Jul 28 08:33:27.531 UTC
++ Show tech start time: 2022-Jul-28.083327.UTC ++
Thu Jul 28 08:33:28 UTC 2022 Waiting for gathering to complete
..
Thu Jul 28 08:33:34 UTC 2022 Compressing show tech output
Show tech output available at 0/RP0/CPU0 :
/harddisk:/showtech/showtech-R1-ZTP-2022-Jul-28.083327.UTC.tgz
++ Show tech end time: 2022-Jul-28.083334.UTC ++
RP/0/RP0/CPU0:R1#

```

In the above example, the tech support information is saved as .tgz file in the specified location. This information can be shared with the Cisco Technical Support representatives for troubleshooting the ztp process.

Configure Management Interface

The management interface can be used for system management and remote communication. To use the management interface for system management, you must configure an IP address and subnet mask. To use the management interface for remote communication, you must configure a static route. Use this procedure when NCS 1020 chassis is not booted using ZTP.

Before you begin

- Consult your network administrator to procure IP addresses and a subnet mask for the management interface.
- Ensure that the management interface is connected to the management network.

Step 1 **configure****Example:**

```
RP/0/RP0/CPU0:ios#configure
```

Enters IOS XR configuration mode.

Step 2 **interface mgmtEth** *rack/slot/instance/port***Example:**

```
RP/0/RP0/CPU0:ios(config)#interface mgmtEth 0/RP0/CPU0/0
```

Enters interface configuration mode for the management interface.

Step 3 **ipv4 address** *ipv4-address subnet-mask***Example:**

```
RP/0/RP0/CPU0:ios(config-if)#ipv4 address 10.1.1.1 255.0.0.0
```

Assigns an IP address and a subnet mask to the management interface.

Step 4 **no shutdown****Example:**

```
RP/0/RP0/CPU0:ios(config-if)#no shutdown
```

Places the management interface in an "up" state.

Step 5 **exit****Example:**

```
RP/0/RP0/CPU0:ios(config-if)#exit
```

Exits the management interface configuration mode.

Step 6 **router static address-family ipv4 unicast** *0.0.0.0/0 default-gateway***Example:**

```
RP/0/RP0/CPU0:ios(config)#router static address-family ipv4 unicast 0.0.0.0/0 198.51.100.4
```

Specifies the IP address of the default gateway to configure a static route. This IP address must be used for communication with devices on other networks.

Step 7 Use the **commit** or **end** command.

commit-Saves the configuration changes and remains within the configuration session.

end-Prompts user to take one of these actions:

- **Yes**-Saves configuration changes and exits the configuration session.
- **No**-Exits the configuration session without committing the configuration changes.

- **Cancel**-Remains in the configuration session without committing the configuration changes.

What to do next

Connect the management interface to the Ethernet network. Establish a [Configure SSH](#) or [Configure Telnet](#) connection to the management interface using its IP address.

Link Layer Discovery Protocol Support on Management Interface

The Link Layer Discovery Protocol (LLDP) support on management interface feature requires a system to form LLDP neighbor relationship over the system management interface. It advertises and learns the LLDP neighbor information through this system. This information about neighbors is then used to learn about the neighbors as well as the topology of the devices for Operations, Administration, and Maintenance (OAM) purposes.



Note Slot-0 and Slot-1 are provisioned for LLDP in NCS 1020.

Advantages of LLDP

- Provides support on both Cisco and non-Cisco devices.
- Enables neighbor discovery between Cisco and non-Cisco devices.

Limitation

- When you disable LLDP globally, the LLDP gets disabled on all the interfaces.



Note By default, LLDP is enabled for NCS 1020. But when you enable and disable LLDP in the global configuration mode, LLDP gets disabled on all the interfaces. To enable it again you must enable LLDP globally or reload NCS 1020.

Table 1: Comparison of Cisco Discovery Protocol (CDP) and LLDP

CDP	LLDP
The CDP is a device discovery protocol that runs over Layer 2. Layer 2 is also known as the data link layer that runs on all Cisco devices, such as routers, bridges, access servers, and switches. This protocol allows the network management applications to automatically discover and learn about other Cisco devices that connect to the network.	The LLDP is also a device discovery protocol that runs over Layer 2. This protocol allows the network management applications to automatically discover and learn about other non-Cisco devices that connect to the network.

Interoperability between Non-Cisco Devices Using LLDP

LLDP is also a neighbor discovery protocol that is used by network devices to advertise information about themselves to other devices on the network. This protocol runs over the data link layer, which allows two systems running different network layer protocols to learn about each other.

Neighbor Discovery

System advertises the LLDP TLV (Type Length Value) details over the management network using which other devices in the management network can learn about this device.

Considerations for LLDP Configuration

- LLDP full stack functionality is supported on all the three management interfaces that are supported in NCS 1020.
- You can selectively enable or disable LLDP on any of the management interfaces.
- You can selectively enable or disable LLDP transmit or receive functionality at the management interface level.
- LLDP operational data is available on both the CLI and NETCONF-YANG interfaces.

Global LLDP Attributes

The following table describes the global LLDP attributes that you can configure:

Table 2: Global LLDP Attributes

Attribute	Default	Range	Description
Holdtime	120 seconds	0–65535 seconds	Specifies the holdtime (in sec). Holdtime refers to the time or duration that an LLDP device maintains the neighbor information before discarding.
Reinit	2 seconds	2–5 seconds	Delay (in sec) for LLDP initialization on any interface
Timer	30 seconds	5–65534 seconds	Specifies the rate at which LLDP packets are sent (in sec)

Enabling LLDP Globally

When you enable LLDP globally, all interfaces that support LLDP are automatically enabled for both transmit and receive operations.

You can override this default operation at the interface to disable receive or transmit operations.



Note Two additional GigabitEthernet based interfaces are enabled as part of the newly provisioned Slot-1 as given in the example below.

```
GigabitEthernet0/1/0/0 unassigned Down Down default
GigabitEthernet0/1/0/2 unassigned Down Down default
```

The following example shows the commands to configure LLDP globally. The global LLDP configuration enables LLDP on all the three management interfaces.

```
RP/0/RP0/CPU0:regen#configure terminal
RP/0/RP0/CPU0:regen(config)#lldp management enable
RP/0/RP0/CPU0:regen(config)#lldp holdtime 30
RP/0/RP0/CPU0:regen(config)#lldp reinit 2
RP/0/RP0/CPU0:regen(config)#commit
```

Verification

You can verify the LLDP configuration using the **show running-config lldp** command.

The output of **show running-config lldp** command is as follows:

```
RP/0/RP0/CPU0:regen#show running-config lldp
Tue Dec 10 10:36:11.567 UTC
lldp
timer 30
reinit 2
holdtime 120
management enable
!
```

You can verify the LLDP data using the **show lldp interface** and **show lldp neighbors** commands.

The output of **show lldp interface** command is as follows:

```
RP/0/RP0/CPU0:regen#show lldp interface
Thu Nov 7 08:45:22.934 UTC

MgmtEth0/RP0/CPU0/0:
  Tx: enabled
  Rx: enabled
  Tx state: IDLE
  Rx state: WAIT FOR FRAME

MgmtEth0/RP0/CPU0/1:
  Tx: enabled
  Rx: enabled
  Tx state: IDLE
  Rx state: WAIT FOR FRAME
```

The output of **show lldp neighbors** command is as follows:

```
RP/0/RP0/CPU0:M-131#show lldp neighbors
Mon Dec 2 11:01:20.143 CET
Capability codes:
  (R) Router, (B) Bridge, (T) Telephone, (C) DOCSIS Cable Device
  (W) WLAN Access Point, (P) Repeater, (S) Station, (O) Other

Device ID           Local Intf           Hold-time  Capability  Port ID
[DISABLED]          MgmtEth0/RP0/CPU0/0  120        B           gi19
MYS-130             MgmtEth0/RP0/CPU0/1  120        R           MgmtEth0/RP0/CPU0/1
```

where [DISABLED] shows that the LLDP is disabled on the interface MgmtEth0/RP0/CPU0/0.

Enable LLDP for Each Management Interface

The following example shows the commands to configure LLDP at the management interface level.

```
RP/0/RP0/CPU0:ios(config)#interface mgmtEth 0/RP0/CPU0/X
RP/0/RP0/CPU0:ios(config-if)#lldp enable
RP/0/RP0/CPU0:ios(config-if)#commit
```

Disabling LLDP Transmit and Receive Operations

The following example shows the commands to disable the LLDP transmit operations at the specified management interface.

```
RP/0/RP0/CPU0:ios(config)#interface mgmtEth 0/RP0/CPU0/X
RP/0/RP0/CPU0:ios(config-if)#lldp transmit disable
RP/0/RP0/CPU0:ios(config-if)#commit
```

The following example shows the commands to disable the LLDP receive operations at the specified management interface.

```
RP/0/RP0/CPU0:ios(config)#interface mgmtEth 0/RP0/CPU0/X
RP/0/RP0/CPU0:ios(config-if)#lldp receive disable
RP/0/RP0/CPU0:ios(config-if)#commit
```

Debugging LLDP Issues

The following commands are used for debugging issues in the LLDP functionality.

- **show lldp traffic**
- **debug lldp all**
- **debug lldp errors**
- **debug lldp events**
- **debug lldp packets**
- **debug lldp tlvs**
- **debug lldp trace**
- **debug lldp verbose**

Configure Telnet

This procedure allows you to establish a telnet session to the management interface using its IP address. Use this procedure when NCS 1020 chassis is not booted using ZTP.

Before you begin

Ensure that two `xr-telnet-*` rpms are installed..

Step 1 configure

Example:

```
RP/0/RP0/CPU0:ios#configure
```

Enters the configuration mode.

Step 2 `telnet {ipv4 | ipv6} server max-servers limit`

Example:

```
RP/0/RP0/CPU0:ios(config)#telnet ipv4 server max-servers 10
```

Specifies the number of allowable telnet servers (up to 100). By default, telnet servers are not allowed. You must configure this command to enable the use of telnet servers.

Step 3 Use the **commit** or **end** command.

commit-Saves the configuration changes and remains within the configuration session.

end-Prompts user to take one of these actions:

- **Yes**-Saves configuration changes and exits the configuration session.
- **No**-Exits the configuration session without committing the configuration changes.
- **Cancel**-Remains in the configuration session without committing the configuration changes.

Configure SSH

This procedure allows you to establish an SSH session to the management interface using its IP address. Use this procedure when NCS 1020 chassis is not booted using ZTP.

Before you begin

- Generate the crypto key for SSH using the **crypto key generate dsa** command.

Step 1 `configure`

Example:

```
RP/0/RP0/CPU0:ios#configure
```

Enters the configuration mode.

Step 2 `ssh server v2`

Example:

```
RP/0/RP0/CPU0:ios(config)# ssh server v2
```

Enables the SSH server to accept only SSHv2 client connections.

Step 3 Use the **commit** or **end** command.

commit-Saves the configuration changes and remains within the configuration session.

end-Prompts the user to take one of these actions:

- **Yes**-Saves configuration changes and exits the configuration session.
- **No**-Exits the configuration session without committing the configuration changes.

- **Cancel**-Remains in the configuration session without committing the configuration changes.
-



CHAPTER 2

Perform Preliminary Checks

After successfully logging into the console, you must perform some preliminary checks to verify the default setup. If any setup issue is detected, take corrective action before making further configurations.



Note The output of the examples in the procedures is not from the latest software release. The output will change for any explicit references to the current release.

- [Verify Status of Hardware Components, on page 29](#)
- [Verify Inventory, on page 31](#)
- [Verify Software Version, on page 33](#)
- [Verify Firmware Version, on page 34](#)
- [Verify Management Interface Status, on page 37](#)
- [Verify Alarms, on page 39](#)
- [Verify Environmental Parameters, on page 39](#)
- [Verify Context, on page 55](#)
- [Verify Core Files, on page 55](#)
- [Verify Memory Information, on page 56](#)

Verify Status of Hardware Components

To verify the status of all the hardware components installed on NCS 1020, perform the following procedure.

Before you begin

Ensure that all the required hardware components are installed on NCS 1020. For installation details, see *Cisco Network Convergence System 1020 Hardware Installation Guide*.

show platform

When you execute this command, the status of Cisco IOS XR is displayed.

Example:

```
RP/0/RP0/CPU0:ios#show platform
```

```
Mon May 6 16:38:13.609 IST
```

Node	Type	State	Config state
0/RP0/CPU0	NCS1010-CTR2-B-K9 (Active)	IOS XR RUN	NSHUT, NMON
0/FT0	NCS1010-FAN	OPERATIONAL	NSHUT, NMON
0/FT1	NCS1010-FAN	OPERATIONAL	NSHUT, NMON
0/FT4	NCS1020-FAN	OPERATIONAL	NSHUT, NMON
0/FT5	NCS1020-FAN	OPERATIONAL	NSHUT, NMON
0/FT6	NCS1020-FAN	OPERATIONAL	NSHUT, NMON
0/FT7	NCS1020-FAN	OPERATIONAL	NSHUT, NMON
0/0/NXR0	NCS1K-E-OLT-C	OPERATIONAL	NSHUT, NMON
0/1/NXR0	NCS1010-FLR-P	PRESENT	NSHUT, NMON
0/2/NXR0	NCS1K14-CCMD-16-C	OPERATIONAL	NSHUT, NMON
0/3/NXR0	NCS1K14-CCMD-16-C	OPERATIONAL	NSHUT, NMON
0/4/NXR0	NCS1K14-BLANK	PRESENT	NSHUT, NMON
0/5/NXR0	NCS1K14-BLANK	PRESENT	NSHUT, NMON
0/6/NXR0	NCS1K14-BLANK	PRESENT	NSHUT, NMON
0/7/NXR0	NCS1K14-BLANK	PRESENT	NSHUT, NMON
0/8/NXR0	NCS1K14-BLANK	PRESENT	NSHUT, NMON
0/9/NXR0	NCS1K14-BLANK	PRESENT	NSHUT, NMON
0/PM0	NCS1K4-AC-PSU-2	OPERATIONAL	NSHUT, NMON
0/PM1	NCS1K4-AC-PSU-2	OPERATIONAL	NSHUT, NMON

Verify that all the components of NCS 1020 are displayed in output. The state must be in the OPERATIONAL state. The various states are:

- OPERATIONAL—Node is operating normally and is fully functional.
- POWERED_ON—Power is on and the node is booting up.
- FAILED—Node is powered on but has encountered an internal failure.
- PRESENT—Node is in intermediate state in the boot sequence.
- POWERED_OFF—Power is off and the node cannot be accessed.
- IOS XR RUN—Node is running IOS XR.
- OFFLINE—Input power is not connected to the power modules.

Verify Inventory

The **show inventory** command displays details of the hardware inventory of NCS 1020.

To verify the inventory information for all the physical entities, perform the following procedure.

show inventory

Displays the details of the physical entities of NCS 1020 along with the details of SFPs.

Example:

```
RP/0/RP0/CPU0:ios#show inventory
```

```
Mon May 6 16:38:33.857 IST
```

```
NAME: "Rack 0", DESCR: "NCS 1020 Shelf Assembly"
```

```
PID: NCS1020-SA , VID: V00, SN: FCB2749B0FD
```

```
NAME: "0/RP0/CPU0", DESCR: "NCS 1010, 1012, 1020 Controller with 9600bps console rate"
```

```
PID: NCS1010-CTR2-B-K9 , VID: V00, SN: FCB2748B01S
```

```
NAME: "0/RP0-PTP0", DESCR: "Cisco Pluggable Optics Module"
```

```
PID: ONS-SI-GE-LX , VID: V01, SN: AGC1703UE1M
```

```
NAME: "0/RP0-PTP1", DESCR: "Cisco Pluggable Optics Module"
```

```
PID: GLC-SX-MMD , VID: V01, SN: OPM221407E5
```

```
NAME: "0/RP0-PTP2", DESCR: "Cisco Pluggable Optics Module"
```

```
PID: SFP-GE-S , VID: V01, SN: FNS17040APG
```

```
NAME: "0/RP0-PTP3", DESCR: "Cisco Pluggable Optics Module"
```

```
PID: ONS-SI-GE-LX , VID: V02, SN: FNS19170MGZ
```

```
NAME: "0/0/NXR0", DESCR: "NCS 1010 Optical Line Terminal - C-band, enhanced"
```

```
PID: NCS1K-E-OLT-C , VID: V01, SN: FCB2721B1DP
```

```
NAME: "0/9/NXR0", DESCR: "Network Convergence System 1014 Filler"
```

PID: NCS1K14-BLANK , VID: V01, SN: N/A

NAME: "0/1/NXR0", DESCR: "Network Convergence System 1010 Passive Filler"

PID: NCS1010-FLR-P , VID: V01, SN: N/A

NAME: "0/2/NXR0", DESCR: "NCS 1014 16 port Colorless Direct attach LC with EDFA C-band"

PID: NCS1K14-CCMD-16-C , VID: V01, SN: FCB2749B06U

NAME: "0/3/NXR0", DESCR: "NCS 1014 16 port Colorless Direct attach LC with EDFA C-band"

PID: NCS1K14-CCMD-16-C , VID: V00, SN: FCB2744B0FA

NAME: "0/4/NXR0", DESCR: "Network Convergence System 1014 Filler"

PID: NCS1K14-BLANK , VID: V01, SN: N/A

NAME: "0/5/NXR0", DESCR: "Network Convergence System 1014 Filler"

PID: NCS1K14-BLANK , VID: V01, SN: N/A

NAME: "0/6/NXR0", DESCR: "Network Convergence System 1014 Filler"

PID: NCS1K14-BLANK , VID: V01, SN: N/A

NAME: "0/7/NXR0", DESCR: "Network Convergence System 1014 Filler"

PID: NCS1K14-BLANK , VID: V01, SN: N/A

NAME: "0/8/NXR0", DESCR: "Network Convergence System 1014 Filler"

PID: NCS1K14-BLANK , VID: V01, SN: N/A

NAME: "0/FT0", DESCR: "NCS1010 - Shelf Fan"

PID: NCS1010-FAN , VID: V01, SN: FCB2719B0WP

NAME: "0/FT1", DESCR: "NCS1010 - Shelf Fan"

PID: NCS1010-FAN , VID: V01, SN: FCB2720B2H2

NAME: "0/FT4", DESCR: "NCS 1020 FAN Tray"

PID: NCS1020-FAN , VID: V00, SN: FCB2752B05V

NAME: "0/FT5", DESCR: "NCS 1020 FAN Tray"

PID: NCS1020-FAN , VID: V00, SN: FCB2752B05W

NAME: "0/FT6", DESCR: "NCS 1020 FAN Tray"

PID: NCS1020-FAN , VID: V00, SN: FCB2752B08C

NAME: "0/FT7", DESCR: "NCS 1020 FAN Tray"

PID: NCS1020-FAN , VID: V00, SN: FCB2752B072

NAME: "0/PM0", DESCR: "Network Convergence System 1004 AC Power Supply Unit 2.5KW"

PID: NCS1K4-AC-PSU-2 , VID: V01, SN: POG27430N29

NAME: "0/PM1", DESCR: "Network Convergence System 1004 AC Power Supply Unit 2.5KW"

PID: NCS1K4-AC-PSU-2 , VID: V01, SN: POG27430N1P

Verify Software Version

NCS 1020 is shipped with the Cisco IOS XR software preinstalled. Verify that the latest version of the software is installed.

To verify the version of Cisco IOS XR Software running on NCS 1020, perform the following procedure.

show version

Displays the software version and details such as system uptime.

Example:

```
RP/0/RP0/CPU0:ios#show version
```

```
Mon May 6 16:38:47.424 IST
```

```
Cisco IOS XR Software, Version 24.2.1.32I LNT
```

```
Copyright (c) 2013-2024 by Cisco Systems, Inc.
```

Build Information:

```

Built By      : cisco
Built On     : Thu Apr 11 02:06:44 UTC 2024
Build Host   : iox-ucs-033
Workspace    : /auto/iox-ucs-033-san1/prod/24.2.1.32I.SIT_IMAGE/ncs1010/ws/
Version     : 24.2.1.32I
Label       : 24.2.1.32I-MSFT_PILOT

```

```

cisco NCS1010 (C3758R @ 2.40GHz)
cisco NCS1020-SA (C3758R @ 2.40GHz) processor with 32GB of memory
NCS1020_P1B_DT_10 uptime is 4 days, 5 hours, 9 minutes
NCS 1020 Chassis

```

Verify Firmware Version

The firmware version on various hardware components of NCS 1020 must be compatible with the installed Cisco IOS XR release. Incompatibility may cause NCS 1020 to malfunction.

To verify the firmware version, perform the following procedure.

Step 1 show hw-module fpd

Displays the firmware information of various hardware components of NCS 1020.

Example:

```
RP/0/RP0/CPU0:ios#show hw-module fpd
```

```
Mon May 6 16:39:04.587 IST
```

```
Auto-upgrade:Enabled
```

```
Attribute codes: B golden, P protect, S secure, A Anti Theft aware
```

```

                                     FPD Versions
                                     =====
Location   Card type                HWver FPD device      ATR Status  Running Programd  Reload Loc
-----
0/RP0/CPU0 NCS1010-CTR2-B-K9      0.1   ADMCONFIG           CURRENT     1.00    1.00          NOT REQ

```


0/RP0/CPU0	NCS1010-CTR2-B-K9	0.1	BIOS	S	CURRENT	5.20	5.20	0/RP0
0/RP0/CPU0	NCS1010-CTR2-B-K9	0.1	BIOS-Golden	BS	CURRENT		1.90	0/RP0
0/RP0/CPU0	NCS1010-CTR2-B-K9	0.1	CpuFpga	S	CURRENT	1.06	1.06	0/RP0
0/RP0/CPU0	NCS1010-CTR2-B-K9	0.1	CpuFpgaGolden	BS	CURRENT		1.02	0/RP0
0/RP0/CPU0	NCS1010-CTR2-B-K9	0.1	SsdMicron5300	S	CURRENT	0.01	0.01	0/RP0
0/RP0/CPU0	NCS1010-CTR2-B-K9	0.1	TamFw	S	CURRENT	9.07	9.07	0/RP0
0/RP0/CPU0	NCS1010-CTR2-B-K9	0.1	TamFwGolden	BS	CURRENT		9.06	0/RP0
0/PM0	NCS1K4-AC-PSU-2	1.0	PO-PrimMCU		CURRENT	1.03	1.03	NOT REQ
0/PM0	NCS1K4-AC-PSU-2	1.0	PO-SecMCU		CURRENT	1.05	1.05	NOT REQ
0/PM1	NCS1K4-AC-PSU-2	1.0	PO-PrimMCU		CURRENT	1.03	1.03	NOT REQ
0/PM1	NCS1K4-AC-PSU-2	1.0	PO-SecMCU		CURRENT	1.05	1.05	NOT REQ
0/0/NXR0	NCS1K-E-OLT-C	1.0	OLT	S	CURRENT	3.14	3.14	NOT REQ
0/2/NXR0	NCS1K14-CCMD-16-C	1.0	CpuModFw	S	CURRENT	42.15	42.15	NOT REQ
0/2/NXR0	NCS1K14-CCMD-16-C	1.0	OptModFw	S	CURRENT	20.02	20.02	NOT REQ
0/3/NXR0	NCS1K14-CCMD-16-C	1.0	CpuModFw	S	CURRENT	42.15	42.15	NOT REQ
0/3/NXR0	NCS1K14-CCMD-16-C	1.0	OptModFw	S	CURRENT	20.02	20.02	NOT REQ
0/Rack	NCS1020-SA	0.1	ADMCONFIG		CURRENT	1.00	1.00	NOT REQ
0/Rack	NCS1020-SA	0.1	IoFpgaLow	S	CURRENT	1.08	1.08	NOT REQ
0/Rack	NCS1020-SA	0.1	IoFpgaLowGolden	BS	CURRENT		0.07	NOT REQ
0/Rack	NCS1020-SA	0.1	IoFpgaUp	S	CURRENT	1.08	1.08	NOT REQ
0/Rack	NCS1020-SA	0.1	IoFpgaUpGolden	BS	CURRENT		0.06	NOT REQ
0/Rack	NCS1020-SA	0.1	SsdMicron5400	S	CURRENT	0.02	0.02	0/Rack

Step 2 show fpd package

Displays the FPD image version available with this software release for each hardware component.

Example:

```
RP/0/RP0/CPU0:ios#show fpd package
Wed Apr 24 15:59:13.897 IST
```

```

=====
                                Field Programmable Device Package
                                =====
Card Type          FPD Description          Req   SW   Min Req  Min Req
                    =====          Reload Ver   SW Ver  Board Ver
=====
NCS1010-CTR2-B-K9  ADMCONFIG                  NO    1.00   1.00    0.1
                    BIOS                      YES    5.20   5.20    0.0
                    BIOS-Golden                 YES    5.10   0.01    0.0
                    CpuFpga                      YES    1.06   1.06    0.0
=====

```

	CpuFpgaGolden	YES	1.02	0.01	0.0
	SsdIntelS4510	YES	11.32	11.32	0.0
	SsdMicron5300	YES	0.01	0.01	0.0
	SsdMicron5400	YES	0.02	0.02	0.0
	TamFw	YES	9.07	9.07	0.0
	TamFwGolden	YES	9.06	0.01	0.0

NCS1010-CTR2-K9	ADMCONFIG	NO	1.00	1.00	0.1
	BIOS	YES	5.20	5.20	0.0
	BIOS-Golden	YES	5.10	0.01	0.0
	CpuFpga	YES	1.06	1.06	0.0
	CpuFpgaGolden	YES	1.02	0.01	0.0
	SsdIntelS4510	YES	11.32	11.32	0.0
	SsdMicron5300	YES	0.01	0.01	0.0
	SsdMicron5400	YES	0.02	0.02	0.0
	TamFw	YES	9.07	9.07	0.0
	TamFwGolden	YES	9.06	0.01	0.0

NCS1020-SA	ADMCONFIG	NO	1.00	1.00	0.0
	IoFpgaLow	NO	1.08	1.08	0.0
	IoFpgaLowGolden	NO	0.07	0.01	0.0
	IoFpgaUp	NO	1.08	1.08	0.0
	IoFpgaUpGolden	NO	0.06	0.01	0.0
	SsdIntelSC2KB	YES	1.20	1.20	0.0
	SsdMicron5400	YES	0.02	0.02	0.0

NCS1K-E-ILA-2R-C	ILA	NO	3.14	3.14	0.1
	ILA	NO	0.28	0.28	99.1
	Raman-1	NO	3.14	3.14	0.1
	Raman-1	NO	0.28	0.28	99.1
	Raman-2	NO	3.14	3.14	0.1
	Raman-2	NO	0.28	0.28	99.1

NCS1K-E-ILA-R-C	ILA	NO	3.14	3.14	0.1
	ILA	NO	0.28	0.28	99.1
	Raman-1	NO	3.14	3.14	0.1
	Raman-1	NO	0.28	0.28	99.1

NCS1K-E-ILA-R-C-2	ILA	NO	3.14	3.14	0.1
	ILA	NO	0.28	0.28	99.1
	Raman-2	NO	3.14	3.14	0.1
	Raman-2	NO	0.28	0.28	99.1

NCS1K-E-OLT-C	OLT	NO	3.14	3.14	0.1
	OLT	NO	0.28	0.28	99.1

NCS1K-E-OLT-L	OLT	NO	3.12	3.12	0.1

NCS1K-E-OLT-R-C	OLT	NO	3.14	3.14	0.1
	OLT	NO	0.28	0.28	99.1
	Raman-1	NO	3.14	3.14	0.1
	Raman-1	NO	0.28	0.28	99.1

NCS1K-ILA-2R-C	ILA	NO	3.14	3.14	0.1
	ILA	NO	0.28	0.28	99.1
	Raman-1	NO	3.14	3.14	0.1
	Raman-1	NO	0.28	0.28	99.1
	Raman-2	NO	3.14	3.14	0.1
	Raman-2	NO	0.28	0.28	99.1

NCS1K-ILA-C	ILA	NO	3.14	3.14	0.1
	ILA	NO	0.28	0.28	99.1

NCS1K-ILA-L	ILA	NO	3.12	3.12	0.1

NCS1K-ILA-R-C	ILA	NO	3.14	3.14	0.1
	ILA	NO	0.28	0.28	99.1
	Raman-1	NO	3.14	3.14	0.1
	Raman-1	NO	0.28	0.28	99.1
NCS1K-OLT-C	OLT	NO	3.14	3.14	0.1
	OLT	NO	0.28	0.28	99.1
NCS1K-OLT-L	OLT	NO	3.12	3.12	0.1
NCS1K-OLT-R-C	OLT	NO	3.14	3.14	0.1
	OLT	NO	0.28	0.28	99.1
	Raman-1	NO	3.14	3.14	0.1
	Raman-1	NO	0.28	0.28	99.1
NCS1K14-CCMD-16-C	CpuModFw	NO	42.14	42.14	0.0
	OptModFw	NO	20.02	20.02	0.0
NCS1K14-CCMD-16-L	CpuModFw	NO	42.14	42.14	0.0
	OptModFw	NO	20.02	20.02	0.0
NCS1K4-AC-PSU-2	PO-PrimMCU	NO	1.03	1.03	0.1
	PO-SecMCU	NO	1.05	1.05	0.1

Verify Management Interface Status

To verify the management interface status, perform the following procedure.

Step 1 show interfaces MgmtEth 0/RP0/CPU0/0

Displays the management interface configuration.

Example:

```
RP/0/RP0/CPU0:ios#show interfaces MgmtEth 0/RP0/CPU0/0
Wed May 25 11:49:18.118 UTC
MgmtEth0/RP0/CPU0/0 is up, line protocol is up
Interface state transitions: 1
Hardware is Management Ethernet, address is 38fd.f866.0964 (bia 38fd.f866.0964)
Internet address is 10.33.0.61/16
MTU 1514 bytes, BW 1000000 Kbit (Max: 1000000 Kbit)
  reliability 255/255, txload 0/255, rxload 0/255
Encapsulation ARPA,
Full-duplex, 1000Mb/s, CX, link type is autonegotiation
loopback not set,
Last link flapped 15:05:21
ARP type ARPA, ARP timeout 04:00:00
Last input never, output 00:00:00
Last clearing of "show interface" counters never
5 minute input rate 0 bits/sec, 0 packets/sec
5 minute output rate 0 bits/sec, 0 packets/sec
  53138 packets input, 6636701 bytes, 0 total input drops
  0 drops for unrecognized upper-level protocol
  Received 12145 broadcast packets, 40082 multicast packets
    0 runts, 0 giants, 0 throttles, 0 parity
  0 input errors, 0 CRC, 0 frame, 0 overrun, 0 ignored, 0 abort
  217288 packets output, 60964220 bytes, 0 total output drops
```

Verify Management Interface Status

```

Output 1 broadcast packets, 15 multicast packets
0 output errors, 0 underruns, 0 applique, 0 resets
0 output buffer failures, 0 output buffers swapped out
1 carrier transitions

```

Step 2 show interfaces summary and show interfaces brief

Verifies the management interface status.

Example:

```

RP/0/RP0/CPU0:ios#show interfaces summary
Wed May 25 11:50:02.558 UTC
Interface Type          Total    UP      Down    Admin Down
-----
ALL TYPES                9        5       0       4
-----
IFT_ETHERNET             1        1       0       0
IFT_LOOPBACK             2        2       0       0
IFT_ETHERNET             3        1       0       2
IFT_NULL                 1        1       0       0
IFT_PTP_ETHERNET        2        0       0       2

```

Example:

```

RP/0/RP0/CPU0:ios#show interfaces brief
Wed May 25 11:50:28.438 UTC

```

Intf Name	Intf State	LineP State	Encap Type	MTU (byte)	BW (Kbps)
Lo0	up	up	Loopback	1500	0
Lo3	up	up	Loopback	1500	0
Nu0	up	up	Null	1500	0
Gi0/0/0/0	up	up	ARPA	1514	1000000
Mg0/RP0/CPU0/0	up	up	ARPA	1514	1000000
Mg0/RP0/CPU0/1	admin-down	admin-down	ARPA	1514	1000000
Mg0/RP0/CPU0/2	admin-down	admin-down	ARPA	1514	1000000
PT0/RP0/CPU0/0	admin-down	admin-down	ARPA	1514	1000000
PT0/RP0/CPU0/1	admin-down	admin-down	ARPA	1514	1000000

Example:

```

RP/0/RP0/CPU0:ios#show ipv4 interfaces brief
Tue Jul 12 07:32:42.390 UTC

```

Interface	IP-Address	Status	Protocol	Vrf-Name
Loopback0	10.3.3.21	Up	Up	default
Loopback3	10.1.1.2	Up	Up	default
GigabitEthernet0/0/0/0	10.7.1.20	Up	Up	default
MgmtEth0/RP0/CPU0/0	10.4.33.63	Up	Up	default
PTP0/RP0/CPU0/0	unassigned	Shutdown	Down	default
MgmtEth0/RP0/CPU0/1	unassigned	Down	Down	default
PTP0/RP0/CPU0/1	unassigned	Shutdown	Down	default
MgmtEth0/RP0/CPU0/2	unassigned	Down	Down	default

Verify Alarms

You can view the alarm information using the **show alarms** command.

```
show alarms [ brief [ card | rack | system ] [ location location ] [ active | history ] | detail [ card
| rack | system ] [ location location ] [ active | clients | history | stats ] ]
```

Displays alarms in brief or detail.

Example:

```
RP/0/RP0/CPU0:ios#show alarms brief system active
```

```
Thu Apr 28 06:16:50.524 UTC
```

```
-----
Active Alarms
-----
Location          Severity    Group      Set Time          Description
-----
0/RP0/CPU0        Major      Ethernet   04/28/2022 06:03:39 UTC  RP-SW: SPI flash config is incorrect
0/PM0             Major      Environ    04/28/2022 06:03:50 UTC  Power Module Error (PM_VIN_VOLT_OOR)
0/PM0             Major      Environ    04/28/2022 06:03:50 UTC  Power Module Output Disabled
(PM_OUTPUT_DISABLED)
0                 Major      Environ    04/28/2022 06:03:50 UTC  Power Group redundancy lost
0/PM0             Major      FPD_Infra  04/28/2022 06:04:08 UTC  One Or More FPDs Need Upgrade Or Not In
Current State
0/PM1             Major      FPD_Infra  04/28/2022 06:04:09 UTC  One Or More FPDs Need Upgrade Or Not In
Current State
0/0               Minor      Environ    04/28/2022 06:04:10 UTC  ILAC_CT_1: Invalid sensor read error.
0/0               Minor      Environ    04/28/2022 06:04:10 UTC  ILAC_CT_2: Invalid sensor read error.
0/0               Minor      Environ    04/28/2022 06:04:10 UTC  ILAC_CT_3: Invalid sensor read error.
0/0               Minor      Environ    04/28/2022 06:04:10 UTC  ILAC_CT_4: Invalid sensor read error.
0/0               Major      Controller 04/28/2022 06:05:12 UTC  Osc0/0/0/0 - Provisioning Failed
0/0               Major      Controller 04/28/2022 06:05:12 UTC  Osc0/0/0/2 - Provisioning Failed
0/0               Major      Controller 04/28/2022 06:05:12 UTC  Ots0/0/0/0 - Provisioning Failed
0/0               Major      Controller 04/28/2022 06:05:12 UTC  Ots0/0/0/2 - Provisioning Failed
```

Note In the maintenance mode, all the alarms are moved from active to suppressed and the **show alarms** command does not display the alarms details.

Verify Environmental Parameters

The **show environment** command displays the environmental parameters of NCS 1020.

To verify that the environmental parameters are as expected, perform the following procedure.

```
show environment [ all | current | fan | power | voltages [ location | location ] | temperature [
location | location ] ]
```

Displays the environmental parameters of NCS 1020.

Example:

The following example shows a sample output of the **show environment** command with the **fan** keyword.

```
RP/0/RP0/CPU0:ios#show environment fan
```

```
Mon Apr 29 11:34:01.781 IST
```

```
=====
```

Location	FRU Type	Fan speed (rpm)		
		FAN_0	FAN_1	FAN_2
0/FT0	NCS1010-FAN	7860	7860	7860
0/FT1	NCS1010-FAN	7800	7740	7740
0/FT4	NCS1020-FAN	7740	7740	
0/FT5	NCS1020-FAN	7740	7740	
0/FT6	NCS1020-FAN	3960	3960	
0/FT7	NCS1020-FAN	4020	3960	
0/PM0	NCS1K4-AC-PSU-2	5632	5504	
0/PM1	NCS1K4-AC-PSU-2	5536	5568	

The following example shows a sample output of the **show environment** command with the **temperatures** keyword for *0/RP0* location.

```
RP/0/RP0/CPU0:ios#show environment temperature location 0/RP0
```

```
Mon Apr 29 11:41:39.134 IST
```

```
=====
```

Location	TEMPERATURE	Value	Crit	Major	Minor	Minor	Major
Crit							
(Hi)	Sensor	(deg C)	(Lo)	(Lo)	(Lo)	(Hi)	(Hi)

0/RP0/CPU0							
90	RP_TEMP_PCB	22	-10	-5	0	80	85
90	RP_TEMP_HOT_SPOT	26	-10	-5	0	80	85
95	RP_TEMP_LTM4638_0	41	-10	-5	0	85	90
95	RP_TEMP_LTM4644_0	29	-10	-5	0	85	90
	RP_TEMP_LTM4644_1	32	-10	-5	0	85	90

95							
90	TEMP_CPU_DIE	29	-10	-5	0	80	85
90	TEMP_DDR_DIMM	25	-10	-5	0	80	85
80	TEMP_CPU_SSD	28	-10	-5	0	70	75
80	TEMP_CHASSIS_SSD	24	-10	-5	0	70	75

The following example shows a sample output of the **show environment** command with the **temperatures** keyword for *0/0/NXR0* location.

RP/0/RP0/CPU0:ios#**show environment temperature location 0/0/NXR0**

Mon Apr 29 11:41:39.134 IST

```

=====
Location  TEMPERATURE                Value  Crit  Major  Minor  Minor  Major
  Crit
(Hi)      Sensor                      (deg C)  (Lo)  (Lo)   (Lo)   (Hi)   (Hi)
-----
0/0/NXR0
33      OLTC_LT_P0_iEDFA0                25     17    18     19     31     32
33      OLTC_LT_P0_iEDFA1                25     17    18     19     31     32
33      OLTC_LT_P0_iEDFA2                25     17    18     19     31     32
33      OLTC_LT_P2_iEDFA0                25     17    18     19     31     32
33      OLTC_LT_P3_iEDFA0                25     17    18     19     31     32
33      OLTC_LT_P0_eEDFA0                24     17    18     19     31     32
81      OLTC_CT_1                        23    -11    -8     -6     76     78
33      OLTC_LT_P0_eEDFA1                24     17    18     19     31     32
76      OLTC_CT_2                        20    -11    -8     -6     71     74
76      OLTC_CT_3                        22    -11    -8     -6     71     74
      OLTC_CT_4                        23    -11    -8     -6     71     74
    
```

```

76
66      OLTC_FT_P0_iEDFA0      59      54      56      57      63      65
66      OLTC_FT_P2_iEDFA0      59      54      56      57      63      65
66      OLTC_FT_P3_iEDFA0      59      54      56      57      63      65
66      OLTC_FT_P0_eEDFA0      59      54      56      57      63      65

```

The following example shows a sample output of the **show environment** command with the **power** keyword.

```
RP/0/RP0/CPU0:ios#show environment power
```

```
Mon Apr 29 11:36:36.442 IST
```

```

=====
CHASSIS LEVEL POWER INFO: 0
=====

Total output power capacity (Group 0 + Group 1) :    2500W +    2500W
Total output power required                      :    1189W
Total power input                               :    278W
Total power output                              :    204W

```

```
Power Group 0:
```

```

=====
Power      Supply      -----Input-----  -----Output---  Status
Module     Type                Volts    Amps    Volts    Amps
=====
0/PM0      NCS1K4-AC-PSU-2  232.0    0.6    12.1    9.5    OK

```

```
Total of Group 0:          139W/0.6A          114W/9.5A
```

```
Power Group 1:
```

```

=====
Power      Supply      -----Input-----  -----Output---  Status
Module     Type                Volts    Amps    Volts    Amps
=====
0/PM1      NCS1K4-AC-PSU-2  231.8    0.6    12.1    7.5    OK

```


Total of Group 1: 139W/0.6A 90W/7.5A

```

=====
Location      Card Type                Power      Power      Status
                  Allocated   Used
                  Watts       Watts
=====
0/RP0/CPU0    NCS1010-CTR2-B-K9       70         16         ON
0/FT0         NCS1010-FAN             110        9          ON
0/FT1         NCS1010-FAN             110        9          ON
0/FT2         -                        110        0         RESERVED
0/FT3         -                        110        0         RESERVED
0/FT4         NCS1020-FAN             73         6          ON
0/FT5         NCS1020-FAN             73         6          ON
0/FT6         NCS1020-FAN             73         3          ON
0/FT7         NCS1020-FAN             73         1          ON
0/0/NXR0      NCS1K-E-OLT-C           142        73         ON
0/2/NXR0      NCS1K14-CCMD-16-C       100        17         ON
0/3/NXR0      NCS1K14-CCMD-16-C       100        21         ON
0/Rack        NCS1020-SA               45         18         ON
    
```

The following example shows a sample output of the **show environment** command with the **voltages** keyword.

RP/0/RP0/CPU0:ios#**show environment voltage location 0/RP0**

Mon Apr 29 11:40:53.949 IST

```

=====
Location  VOLTAGE                Value      Crit      Minor      Minor      Crit
          Sensor              (mV)      (Lo)     (Lo)     (Hi)     (Hi)
-----
0/RP0/CPU0
          RP_ADM1266_12V0      12035     10800     11280     12720     13200
          RP_ADM1266_3V3_STANDBY      3314      3070      3200      3400      3530
          RP_ADM1266_5V0      4993      4650      4850      5150      5350
    
```

RP_ADM1266_3V3	3325	3070	3200	3400	3530
RP_ADM1266_2V5_PLL	2522	2330	2430	2580	2680
RP_ADM1266_2V5_FPGA	2494	2330	2430	2580	2680
RP_ADM1266_1V2_FPGA	1204	1120	1160	1240	1280
RP_ADM1266_3V3_CPU	3336	3070	3200	3400	3530
RP_ADM1266_2V5_CPU	2509	2330	2430	2580	2680
RP_ADM1266_1V8_CPU	1803	1670	1750	1850	1930
RP_ADM1266_1V24_VCCREF	1238	1150	1200	1280	1330
RP_ADM1266_1V05_CPU	1050	980	1020	1080	1120
RP_ADM1266_1V2_DDR_VDDQ	1199	1120	1160	1240	1280
RP_ADM1266_1V0_VCC_RAM	1106	650	700	1250	1300
RP_ADM1266_1V0_VNN	918	550	600	1250	1300
RP_ADM1266_1V0_VCCP	1153	450	500	1250	1300
RP_ADM1266_0V6_DDR_VTT	602	560	580	620	640

The following example shows a sample output of the **show environment current** command with the **current** keyword.

```
RP/0/RP0/CPU0:ios#show environment current
```

```
Mon Apr 29 11:33:28.410 IST
```

```
=====
Location  CURRENT                               Value
          Sensor                           (mA)
-----
```

```
0/RP0/CPU0
```

```
RP_CURRMON_LTM4638_0          387
RP_CURRMON_LTM4644_0          171
RP_CURRMON_LTM4644_1          320
RP_JMAC_1V0_VCCP_IMON         125
RP_JMAC_1V0_VNN_IMON          125
RP_JMAC_1V0_VCC_RAM_IMON      0
RP_JMAC_1V2_DDR_VDDQ_IMON     156
```

```
0/2/NXR0
```

```
IMON_OPTM                     1167
IMON_CTLPL                     521
```

0/3/NXR0

IMON_OPTM	1242
IMON_CTLPL	525

0/Rack

SA_U_INA230_3V3_IMON	2740
SA_U_INA230_1V0_XGE_CORE_IMON	3295
SA_U_INA230_1V0_POLLUX10_CORE	979
SA_U_ADM1275_12V_EITU_IMON	1528
SA_U_ADM1275_12V_CPU0_IMON	1398
SA_U_ADM1275_12V_MOD0_IMON	5886
SA_U_ADM1275_12V_MOD1_IMON	18
SA_U_ADM1275_12V_MOD2_IMON	18
SA_U_ADM1275_12V_LC2_IMON	1629
SA_U_ADM1275_12V_LC3_IMON	1716
SA_U_ADM1275_12V_LC4_IMON	6
SA_U_ADM1275_12V_LC5_IMON	18
SA_L_INA230_1V0_PROPUS10_CORE	942
SA_L_INA230_5V_USB_IMON_A	99
SA_L_INA230_5V_USB_IMON_B	97
SA_L_ADM1275_12V_CPU1_IMON	13
SA_L_ADM1275_12V_MOD3_IMON	51
SA_L_ADM1275_12V_MOD4_IMON	18
SA_L_ADM1275_12V_MOD5_IMON	18
SA_L_ADM1275_12V_FAN0_IMON	799
SA_L_ADM1275_12V_FAN1_IMON	762
SA_L_ADM1275_12V_FAN2_IMON	43
SA_L_ADM1275_12V_FAN3_IMON	18
SA_L_ADM1275_12V_LC6_IMON	6
SA_L_ADM1275_12V_LC7_IMON	30
SA_L_ADM1275_12V_LC8_IMON	6
SA_L_ADM1275_12V_LC9_IMON	30
SA_L_ADM1275_12V_FAN4_0_IMON	294

```

SA_L_ADM1275_12V_FAN4_1_IMON      288
SA_L_ADM1275_12V_FAN5_0_IMON      319
SA_L_ADM1275_12V_FAN5_1_IMON      362
SA_L_ADM1275_12V_FAN6_0_IMON       96
SA_L_ADM1275_12V_FAN6_1_IMON      114
SA_L_ADM1275_12V_FAN7_0_IMON      145
SA_L_ADM1275_12V_FAN7_1_IMON      127

```

The following example shows a sample output of the **show environment** command with the **all** keyword.

```
RP/0/RP0/CPU0:ios#show environment all
```

```
Mon Apr 29 11:32:46.987 IST
```

```

=====
Location  TEMPERATURE          Value   Crit   Major   Minor   Minor   Major
  Crit
(Hi)      Sensor                (deg C) (Lo)   (Lo)   (Lo)   (Hi)   (Hi)
-----
0/RP0/CPU0
  90      RP_TEMP_PCB           22     -10    -5      0      80     85
  90      RP_TEMP_HOT_SPOT     27     -10    -5      0      80     85
  95      RP_TEMP_LTM4638_0    42     -10    -5      0      85     90
  95      RP_TEMP_LTM4644_0    30     -10    -5      0      85     90
  95      RP_TEMP_LTM4644_1    32     -10    -5      0      85     90
  90      TEMP_CPU_DIE         29     -10    -5      0      80     85
  90      TEMP_DDR_DIMM        25     -10    -5      0      80     85
  80      TEMP_CPU_SSD         29     -10    -5      0      70     75
  80      TEMP_CHASSIS_SSD     24     -10    -5      0      70     75
0/0/NXR0
  33      OLTC_LT_P0_iEDFA0    25      17     18     19     31     32
        OLTC_LT_P0_iEDFA1    25      17     18     19     31     32

```

33							
33	OLTC_LT_P0_iEDFA2	25	17	18	19	31	32
33	OLTC_LT_P2_iEDFA0	25	17	18	19	31	32
33	OLTC_LT_P3_iEDFA0	25	17	18	19	31	32
33	OLTC_LT_P0_eEDFA0	24	17	18	19	31	32
81	OLTC_CT_1	23	-11	-8	-6	76	78
33	OLTC_LT_P0_eEDFA1	24	17	18	19	31	32
76	OLTC_CT_2	20	-11	-8	-6	71	74
76	OLTC_CT_3	23	-11	-8	-6	71	74
76	OLTC_CT_4	23	-11	-8	-6	71	74
66	OLTC_FT_P0_iEDFA0	59	54	56	57	63	65
66	OLTC_FT_P2_iEDFA0	60	54	56	57	63	65
66	OLTC_FT_P3_iEDFA0	60	54	56	57	63	65
66	OLTC_FT_P0_eEDFA0	59	54	56	57	63	65
0/2/NXR0							
105	TEMP_DX_PCB	24	-10	-5	0	95	100
105	TEMP_DX_ZYNQ	27	-10	-5	0	95	100
0/3/NXR0							
105	TEMP_DX_PCB	23	-10	-5	0	95	100
105	TEMP_DX_ZYNQ	27	-10	-5	0	95	100
0/PM0							
72	Air Inlet Temperature	22	-10	-5	0	62	67
92	Air Outlet Temperature	24	-10	-5	0	82	87

Verify Environmental Parameters

87	Oring MOSFET	29	-10	-5	0	77	82
0/PM1							
72	Air Inlet Temperature	24	-10	-5	0	62	67
92	Air Outlet Temperature	25	-10	-5	0	82	87
87	Oring MOSFET	30	-10	-5	0	77	82
0/Rack							
90	SA_U_TMP421_EITU_PCB_HOTSPOT0	28	-10	-5	0	80	85
90	SA_U_TMP421_EITU_PCB_HOTSPOT1	37	-10	-5	0	80	85
90	SA_U_TMP421_EITU_PCB_HOTSPOT2	30	-10	-5	0	80	85
90	SA_U_TMP421_EITU_PCB_HOTSPOT3	34	-10	-5	0	80	85
60	SA_L_TMP421_CHASSIS_INLET0	20	-10	-5	0	45	55
90	SA_L_TMP421_CHASSIS_EXHAUST0	22	-10	-5	0	75	85
90	SA_L_TMP421_CHASSIS_EXHAUST1	24	-10	-5	0	75	85
90	SA_L_TMP421_CHASSIS_EXHAUST2	22	-10	-5	0	75	85
90	SA_L_TMP421_CHASSIS_EXHAUST3	21	-10	-5	0	75	85
90	SA_L_TMP421_CHASSIS_EXHAUST4	23	-10	-5	0	75	85
90	SA_L_TMP421_CHASSIS_EXHAUST5	22	-10	-5	0	75	85

Location	VOLTAGE	Value	Crit	Minor	Minor	Crit
	Sensor	(mV)	(Lo)	(Lo)	(Hi)	(Hi)

0/RP0/CPU0

RP_ADM1266_12V0	12035	10800	11280	12720	13200
RP_ADM1266_3V3_STANDBY	3314	3070	3200	3400	3530

RP_ADM1266_5V0	4993	4650	4850	5150	5350
RP_ADM1266_3V3	3325	3070	3200	3400	3530
RP_ADM1266_2V5_PLL	2522	2330	2430	2580	2680
RP_ADM1266_2V5_FPGA	2494	2330	2430	2580	2680
RP_ADM1266_1V2_FPGA	1204	1120	1160	1240	1280
RP_ADM1266_3V3_CPU	3336	3070	3200	3400	3530
RP_ADM1266_2V5_CPU	2509	2330	2430	2580	2680
RP_ADM1266_1V8_CPU	1803	1670	1750	1850	1930
RP_ADM1266_1V24_VCCREF	1238	1150	1200	1280	1330
RP_ADM1266_1V05_CPU	1050	980	1020	1080	1120
RP_ADM1266_1V2_DDR_VDDQ	1199	1120	1160	1240	1280
RP_ADM1266_1V0_VCC_RAM	1103	650	700	1250	1300
RP_ADM1266_1V0_VNN	918	550	600	1250	1300
RP_ADM1266_1V0_VCCP	1154	450	500	1250	1300
RP_ADM1266_0V6_DDR_VTT	598	560	580	620	640
0/2/NXR0					
VIN_5_0V	5023	4500	4750	5250	5500
VAUX_3_3V_ST	3317	3053	3135	3465	3548
VIN_12_0V	12066	10800	11400	12600	13200
VMON_3_3V	3298	3135	3201	3399	3465
VMON_1_8V_MGT	1796	1710	1746	1854	1890
VMON_1_8V	1799	1710	1746	1854	1890
VMON_1_2V	1200	1140	1164	1236	1260
VMON_2_5V	2489	2357	2425	2575	2625
VMON_0_85V_MGT	852	808	825	876	893
VMON_0_85V	849	808	825	876	893
0/3/NXR0					
VIN_5_0V	5003	4500	4750	5250	5500
VAUX_3_3V_ST	3313	3053	3135	3465	3548
VIN_12_0V	12051	10800	11400	12600	13200
VMON_3_3V	3292	3135	3201	3399	3465
VMON_1_8V_MGT	1799	1710	1746	1854	1890

Verify Environmental Parameters

VMON_1_8V	1796	1710	1746	1854	1890
VMON_1_2V	1202	1140	1164	1236	1260
VMON_2_5V	2500	2357	2425	2575	2625
VMON_0_85V_MGT	853	808	825	876	893
VMON_0_85V	849	808	825	876	893
0/Rack					
SA_U_ADM1266_12V_BUS_EITU	12042	10800	11280	12720	13200
SA_U_ADM1266_2V5	2498	2325	2400	2600	2675
SA_U_ADM1266_3V3	3308	3069	3168	3432	3531
SA_U_ADM1266_3V3_STANDBY	3297	3069	3168	3432	3531
SA_U_ADM1266_1V2_PROPUS10_FPGA	1203	1116	1152	1248	1284
SA_U_ADM1266_1V0_PROPUS10_FPGA	1008	930	960	1040	1070
SA_U_ADM1266_5V0_USB_A	5085	4650	4800	5200	5350
SA_U_ADM1266_5V0_USB_B	5070	4650	4800	5200	5350
SA_U_ADM1275_12V_EITU	12048	10800	11280	12720	13200
SA_U_ADM1275_12V_CPU0	12027	10800	11280	12720	13200
SA_U_ADM1275_12V_MOD0	12048	10800	11280	12720	13200
SA_U_ADM1266_5V0	4998	4650	4800	5200	5350
SA_U_ADM1275_12V_MOD1	-	10800	11280	12720	13200
SA_U_ADM1275_12V_MOD2	-	10800	11280	12720	13200
SA_U_ADM1275_12V_LC2	12042	10800	11280	12720	13200
SA_U_ADM1275_12V_LC3	12037	10800	11280	12720	13200
SA_U_ADM1275_12V_LC4	-	10800	11280	12720	13200
SA_U_ADM1275_12V_LC5	-	10800	11280	12720	13200
SA_U_ADM1266_1V8_ZARLINK_DPLL	1799	1674	1728	1872	1926
SA_U_ADM1266_1V0_PHY	1009	930	960	1040	1070
SA_L_ADM1275_12V_CPU1	-	10800	11280	12720	13200
SA_L_ADM1275_12V_MOD3	-	10800	11280	12720	13200
SA_L_ADM1275_12V_MOD4	-	10800	11280	12720	13200
SA_L_ADM1275_12V_MOD5	-	10800	11280	12720	13200
SA_L_ADM1275_12V_FAN0	12027	10800	11280	12720	13200
SA_L_ADM1275_12V_FAN1	12032	10800	11280	12720	13200

SA_L_ADM1275_12V_FAN2	-	10800	11280	12720	13200
SA_L_ADM1275_12V_FAN3	-	10800	11280	12720	13200
SA_U_ADM1266_1V0_ALDRIN_CORE	983	910	930	1070	1090
SA_L_ADM1275_12V_LC6	-	10800	11280	12720	13200
SA_L_ADM1275_12V_LC7	-	10800	11280	12720	13200
SA_L_ADM1275_12V_LC8	-	10800	11280	12720	13200
SA_L_ADM1275_12V_LC9	-	10800	11280	12720	13200
SA_L_ADM1275_12V_FAN4_0	12027	10800	11280	12720	13200
SA_L_ADM1275_12V_FAN4_1	12037	10800	11280	12720	13200
SA_L_ADM1275_12V_FAN5_0	12058	10800	11280	12720	13200
SA_L_ADM1275_12V_FAN5_1	12017	10800	11280	12720	13200
SA_L_ADM1275_12V_FAN6_0	12058	10800	11280	12720	13200
SA_L_ADM1275_12V_FAN6_1	12037	10800	11280	12720	13200
SA_U_ADM1266_1V0_ALDRIN_SERDES	1008	930	960	1040	1070
SA_L_ADM1275_12V_FAN7_0	12032	10800	11280	12720	13200
SA_L_ADM1275_12V_FAN7_1	12058	10800	11280	12720	13200
SA_U_ADM1266_1V0_POLLUX10_FPGA	1006	930	960	1040	1070
SA_U_ADM1266_1V2_POLLUX10_FPGA	1203	1116	1152	1248	1284
SA_U_ADM1266_1V8	1806	1674	1728	1872	1926

=====
 Location CURRENT Value
 Sensor (mA)

0/RP0/CPU0

RP_CURRMON_LTM4638_0	387
RP_CURRMON_LTM4644_0	171
RP_CURRMON_LTM4644_1	320
RP_JMAC_1V0_VCCP_IMON	187
RP_JMAC_1V0_VNN_IMON	125
RP_JMAC_1V0_VCC_RAM_IMON	0
RP_JMAC_1V2_DDR_VDDQ_IMON	156

0/2/NXR0

Verify Environmental Parameters

	IMON_OPTM	1209
	IMON_CTLPL	521
0/3/NXR0		
	IMON_OPTM	1234
	IMON_CTLPL	525
0/Rack		
	SA_U_INA230_3V3_IMON	2680
	SA_U_INA230_1V0_XGE_CORE_IMON	3295
	SA_U_INA230_1V0_POLLUX10_CORE	979
	SA_U_ADM1275_12V_EITU_IMON	1528
	SA_U_ADM1275_12V_CPU0_IMON	1417
	SA_U_ADM1275_12V_MOD0_IMON	6773
	SA_U_ADM1275_12V_MOD1_IMON	43
	SA_U_ADM1275_12V_MOD2_IMON	55
	SA_U_ADM1275_12V_LC2_IMON	1666
	SA_U_ADM1275_12V_LC3_IMON	1679
	SA_U_ADM1275_12V_LC4_IMON	43
	SA_U_ADM1275_12V_LC5_IMON	43
	SA_L_INA230_1V0_PROPUS10_CORE	942
	SA_L_INA230_5V_USB_IMON_A	99
	SA_L_INA230_5V_USB_IMON_B	97
	SA_L_ADM1275_12V_CPU1_IMON	13
	SA_L_ADM1275_12V_MOD3_IMON	113
	SA_L_ADM1275_12V_MOD4_IMON	30
	SA_L_ADM1275_12V_MOD5_IMON	18
	SA_L_ADM1275_12V_FAN0_IMON	836
	SA_L_ADM1275_12V_FAN1_IMON	836
	SA_L_ADM1275_12V_FAN2_IMON	43
	SA_L_ADM1275_12V_FAN3_IMON	55
	SA_L_ADM1275_12V_LC6_IMON	43
	SA_L_ADM1275_12V_LC7_IMON	18
	SA_L_ADM1275_12V_LC8_IMON	6

```

SA_L_ADM1275_12V_LC9_IMON          80
SA_L_ADM1275_12V_FAN4_0_IMON      281
SA_L_ADM1275_12V_FAN4_1_IMON      281
SA_L_ADM1275_12V_FAN5_0_IMON      263
SA_L_ADM1275_12V_FAN5_1_IMON      331
SA_L_ADM1275_12V_FAN6_0_IMON      114
SA_L_ADM1275_12V_FAN6_1_IMON      114
SA_L_ADM1275_12V_FAN7_0_IMON      120
SA_L_ADM1275_12V_FAN7_1_IMON      52
    
```

=====

```

                                Fan speed (rpm)
Location      FRU Type          FAN_0    FAN_1    FAN_2
-----
0/FT0         NCS1010-FAN          7860     7920     7860
0/FT1         NCS1010-FAN          7800     7740     7800
0/FT4         NCS1020-FAN          7800     7740
0/FT5         NCS1020-FAN          7800     7740
0/FT6         NCS1020-FAN          3960     3960
0/FT7         NCS1020-FAN          4020     3960
0/PM0         NCS1K4-AC-PSU-2     5504     5504
0/PM1         NCS1K4-AC-PSU-2     5536     5600
    
```

=====

```

Location      Altitude Value (Meters)    Source
-----
0              0                          config
    
```

=====

CHASSIS LEVEL POWER INFO: 0

=====

```

Total output power capacity (Group 0 + Group 1) :    2500W +    2500W
Total output power required                      :    1189W
    
```

Verify Environmental Parameters

```

Total power input           :    278W
Total power output          :    206W

```

Power Group 0:

```

=====
Power      Supply      -----Input-----  -----Output----  Status
Module    Type              Volts    Amps    Volts    Amps
=====
0/PM0     NCS1K4-AC-PSU-2  232.0    0.6    12.1    9.6    OK

Total of Group 0:          139W/0.6A          116W/9.6A

```

Power Group 1:

```

=====
Power      Supply      -----Input-----  -----Output----  Status
Module    Type              Volts    Amps    Volts    Amps
=====
0/PM1     NCS1K4-AC-PSU-2  231.8    0.6    12.1    7.5    OK

Total of Group 1:          139W/0.6A          90W/7.5A

```

```

=====
Location   Card Type              Power      Power      Status
              Allocated  Used
              Watts      Watts
=====
0/RP0/CPU0 NCS1010-CTR2-B-K9      70         17         ON
0/FT0       NCS1010-FAN            110        10         ON
0/FT1       NCS1010-FAN            110        10         ON
0/FT2       -                       110         0         RESERVED
0/FT3       -                       110         0         RESERVED
0/FT4       NCS1020-FAN            73         6          ON

```

0/FT5	NCS1020-FAN	73	7	ON
0/FT6	NCS1020-FAN	73	2	ON
0/FT7	NCS1020-FAN	73	2	ON
0/0/NXR0	NCS1K-E-OLT-C	142	81	ON
0/2/NXR0	NCS1K14-CCMD-16-C	100	20	ON
0/3/NXR0	NCS1K14-CCMD-16-C	100	20	ON
0/Rack	NCS1020-SA	45	18	ON

Environment parameter anomalies are logged in the syslog. As a result, if an environment parameter that is displayed in the **show environment** command output is not as expected, check the syslog using the **show logging** and **show alarms brief system active** command. The syslog provides details on any logged problems.

Verify Context

The **show context** command displays core dump context information of NCS 1020. Core dump is a result of abnormal exit of any process running in the system.

show context

Displays the core dump context information of NCS 1020.

Example:

```
RP/0/RP0/CPU0:ios# show context
Mon Sep 27 17:21:59.219 UTC
```

```
node: node0_RP0_CPU0
```

```
-----
No context
```

The command output is empty during system upgrade.

Verify Core Files

Use the **run** command to go to the hard disk location and check for the core dumps of NCS 1020.

run

Example:

```
RP/0/RP0/CPU0:ios# run
Mon Sep 27 17:29:11.163 UTC
```

```
[xr-vm_node0_RP0_CPU0:~]$cd /misc/disk1/  
[xr-vm_node0_RP0_CPU0:/misc/disk1]$ls -ltr *.gz
```

Verify Memory Information

You can view the memory information using the show watchdog memory-state command.

show watchdog memory-state location all

Displays memory snapshot in brief.

Example:

```
RP/0/RP0/CPU0:ios#show watchdog memory-state location all  
Thu Jun 16 08:36:44.436 UTC  
---- node0_RP0_CPU0 ----  
Memory information:  
  Physical Memory      : 31935.167 MB  
  Free Memory          : 29236.0 MB  
  Memory State         : Normal
```



CHAPTER 3

Upgrade Software and FPD

- [Plan the Software Upgrade](#) , on page 57
- [Upgrade the Software](#), on page 59
- [Verify the Software Upgrade](#), on page 68
- [NCS 1020 FPD](#), on page 68

Plan the Software Upgrade

Before you upgrade the software version, prepare the NCS 1020 to ensure that the upgrade process is seamless.

This section describes the following processes to prepare your NCS 1020 for an upgrade:

Backup Current Configuration

The ability to recover from a disaster is an essential part of any system maintenance plan. We recommend you backup the configurations in a secure remote location and verify that the transfer is a success, both before and after upgrade.

Step 1 Create a backup of the running configuration to one of the following locations based on your requirement:

- Copy the configuration to the `harddisk:` location on the NCS 1020.

```
RP/0/RP0/CPU0:ios#copy running-config harddisk:/running_config-<mmddyyyy>
Destination filename [running_config-<mmddyyyy>]?
Building configuration...
[OK]
Verifying checksum... OK (0xDCF1)
```

- Copy the configuration to a remote server. Ensure the NCS 1020 has root access to the server.

```
RP/0/RP0/CPU0:ios#scp harddisk:/ running_config-<mmddyyyy>
user:password@<ip-address>:<location>
```

Step 2 Verify that the configuration is backed up.

Check System Stability

System stability checks are essential to measure the efficiency and ability of an upgrade to function over an extended period.

At the EXEC prompt, execute the following commands to assess basic system stability checks before and after the software upgrade.

Command	Reason	Workaround
<code>show platform</code>	Verify that all nodes are in IOS XR RUN/OPERATIONAL state	NA
<code>show install active summary</code>	Verify that the proper set of packages are active	NA
<code>show install committed summary</code>	Verify that the proper set of committed packages are same as active	Execute 'install commit' command
<code>clear configuration inconsistency</code>	Verify/fix configuration file system	NA
<code>show hw-module fpd</code>	Ensure all the FPD versions status are CURRENT	Execute <code>upgrade hw-module fpd</code> command
<code>show media</code>	Display the current state of the disk storage media	To free up space, remove older .iso image files and bug fix .tar files.
<code>show inventory</code>	Show chassis inventory information	NA

Obtain Install Files

You can obtain the install files based on one of the following options that is best suited to your network:

- **Base ISO and Optional RPMs:** You can upgrade the software through the standard method where you install the ISO followed by the required RPMs.
- **Golden ISO:** You can build a customized golden ISO (GISO) image with the base ISO and the required RPMs to automatically upgrade the software.

Standard ISO and RPMs



Note Contact your Cisco Systems technical support representative for NCS1020 ISO image with required IOS-XR version and optional RPM's.

Golden ISO



Note Contact your Cisco Systems technical support representative for NCS1020 Golden GIOS image creation with required IOS-XR version and optional RPM's information.

Create Repository to Access Install Files

A **Repository** is a directory where the ISO, RPMs, and their metadata are downloaded. The package manager uses this repository to query the packages.

The repository can either be created locally on the NCS 1020, or on a remote location that can be accessed through FTP, HTTP, or HTTPS. In a repository, you can create directories based on different Cisco IOS XR platforms, releases or both. You can create and use multiple repositories. The files to be installed can saved in the local repository, remote repository or a combination of both.



Note The Golden ISO (GISO) method does not require you to create a repository. However, you can still install the GISO from a remote repository.



Important Each package is named based on its name, version, software release, and architecture. Hence, any packages that have these attributes in common and differ only by platform are indistinguishable. We recommend that you create different repositories for different platforms and releases.

Upgrade the Software

This section provides information about the processes involved in upgrading the IOS XR software on your Cisco NCS 1020.

The Cisco IOS XR software can be upgraded using one of these methods:

Upgrade NCS 1020 Using CLI Commands

There are two options to upgrade your Cisco IOS XR software using the Command Line Interface (CLI):

- Base ISO and optional RPMs
- Golden ISO (GISO)

Before you begin

Note Ensure that you have adequate disk space. Run the **fsck** command to check the status of the file system, for a successful IOS XR upgrade. You must run the **fsck** command in the System Admin EXEC mode to install a System Admin package, and in the XR EXEC mode to install the XR package. All install commands are applicable in both the System Admin EXEC mode and in XR EXEC mode. System Admin install operations are done from XR EXEC mode.



Note If an interface on a NCS 1020 does not have a configuration and is brought up by performing no-shut operation, then upon NCS 1020 reload, the interface state changes to **admin-shutdown** automatically.

Perform a system upgrade by installing a base package—Cisco IOS XR Unicast Routing Core Bundle.

Upgrade NCS 1020 Using YANG Data Models

Data models are a programmatic way of configuring and collecting operational data of a network device. They replace the process of manual configuration and can be used to automate configuration tasks across heterogeneous devices in a network.

Access Install-related Data Models

You can use YANG data models to install and upgrade the NCS 1020. The data models are packaged with the release image in the `/pkg/yang` directory.

Step 1 Navigate to the directory in the release image where the YANG data models are available.

Example:

```
RP/0/RP0/CPU0:ios#run
[node_RP0_CPU0:~]$cd /pkg/yang
```

Step 2 View the list of install-related data models on your NCS 1020.

Example:

```
node0_RP0_CPU0:/pkg/yang]$ls -ltr *install*
-rw-r--r--. 1 root root 8646 Jul 2 01:59 Cisco-IOS-XR-install-act.yang
-rw-r--r--. 1 root root 7267 Jul 2 01:59 Cisco-IOS-XR-install-search-act.yang
-rw-r--r--. 1 root root 10664 Jul 2 01:59 Cisco-IOS-XR-install-augmented-act.yang
-rw-r--r--. 1 root root 2511 Jul 2 02:00 Cisco-IOS-XR-um-install-cfg.yang
-rw-r--r--. 1 root root 2270 Jul 2 02:04 Cisco-IOS-XR-install-cfg.yang
-rw-r--r--. 1 root root 6222 Jul 2 02:04 Cisco-IOS-XR-install-oper.yang
-rw-r--r--. 1 root root 14009 Jul 2 02:04 Cisco-IOS-XR-install-augmented-oper.yang
```

The following table describes the function of the install-related data models:

Date Model	Description
Cisco-IOS-XR-um-install-cfg	Unified data model that contains a collection of YANG definitions for Cisco IOS XR install package configuration, and augments the modules with configuration data.

Date Model	Description
Cisco-IOS-XR-install-oper	Operational data model to view details that are related to basic package information, active and committed packages, and fixes.
Cisco-IOS-XR-install-cfg	Configuration data model to specify the location of the install source.
Cisco-IOS-XR-install-act	Action model to perform basic install operations and software upgrade.
Cisco-IOS-XR-install-search-act	Action model that contains a collection of YANG definitions for install actions related to searching for package information.
Cisco-IOS-XR-install-augmented-oper	Augmented operational model that displays information about packaging, atomic changes, and history of the install operation on the NCS 1020.
Cisco-IOS-XR-install-augmented-act	Action model to perform flexible install operations, including controlling the exact timing of system reloads and rolling back to a previous commit.
Cisco-IOS-XR-shellutil-copy-act	Action model to copy files on the NCS 1020 from a source location.

You can also access the supported data models to install Cisco IOS XR software from the [Github](#) repository.

Use Manageability Agent to Connect to NCS 1020

Use a manageability agent like NETCONF or gRPC to connect and communicate with the NCS 1020. You can send Remote Procedure Calls (RPC) requests to configure or retrieve operational data from the NCS 1020. The NCS 1020 processes the request and responds to the request through an RPC response. You use the RPCs to send requests to install the software by populating the relevant parameters of a container and leaf in the data model. For more information about understanding the data model structure and using data models, see the .

Generate RPC Messages to Install IOS XR Image

Before you begin

Not all software versions are supported as the target upgrade software version. You must review the supported upgrade and downgrade paths, hardware or software limitations, and bridging SMUs required for the version.

- Step 1** Use the `install-replace` RPC on the `Cisco-IOS-XR-install-act.yang` data model to upgrade the NCS 1020(s).
- Step 2** Configure the values of the `source-type`, `source`, and `file` parameters.
- Step 3** Send `edit-config` NETCONF RPC request using the data model to configure the repository. Edit the values in the `repositories` parameters and send this request to the NCS 1020 from the client.

Example:

In this example, the request is to install the `ncs1010-x64-24.2.1.iso` image from the local repository.

```
<rpc xmlns="urn:ietf:params:xml:ns:netconf:base:1.0" message-id="101">
  <edit-config>
    <target>
      <candidate/>
    </target>
    <config>
      <install xmlns="http://cisco.com/ns/yang/Cisco-IOS-XR-install-cfg">
        <repositories>
          <repository>
            <id>repo_local</id>
            <url>file:///harddisk:/repo/</url>
            <description>local repository</description>
          </repository>
        </repositories>
      </install>
    </config>
  </edit-config>
</rpc>
```

View the RPC response received from the NCS 1020.

```
<?xml version="1.0"?>
<rpc-reply message-id="101" xmlns="urn:ietf:params:xml:ns:netconf:base:1.0">
  <ok/>
</rpc-reply>
```

In the response, the NCS 1020 acknowledges the configuration and sends a reply to the client with an `ok` message.

Step 4 Apply the changes to activate the ISO on the NCS 1020 using RPCs by using the `install-apply` RPC on the `Cisco-IOS-XR-install-augmented-act.yang` data model and send the RPC from the client to the NCS 1020.

Example:

```
<rpc xmlns="urn:ietf:params:xml:ns:netconf:base:1.0" message-id="101">
  <install-apply xmlns="http://cisco.com/ns/yang/Cisco-IOS-XR-install-augmented-act">
    <apply-method>least-impactful</apply-method>
  </install-apply>
</rpc>
```

View the RPC response received from the NCS 1020.

```
<?xml version="1.0"?>
<rpc-reply message-id="101" xmlns="urn:ietf:params:xml:ns:netconf:base:1.0">
  <op-id xmlns="http://cisco.com/ns/yang/Cisco-IOS-XR-install-augmented-act">2.1</op-id>
</rpc-reply>
```

In the response, the NCS 1020 sends an ID indicating that the changes are applied successfully.

Step 5 Verify that the software upgrade is successful. Use the `getRPC` on `Cisco-IOS-XR-install-oper.yang` data model. Edit the `install` parameter and send an RPC request from the client to the NCS 1020.

Example:

```
<rpc xmlns="urn:ietf:params:xml:ns:netconf:base:1.0" message-id="101">
  <get>
    <filter>
      <install xmlns="http://cisco.com/ns/yang/Cisco-IOS-XR-install-oper">
        <request/>
      </install>
    </filter>
  </get>
</rpc>
```

View the RPC response received from the NCS 1020.

```
<?xml version="1.0"?>
<rpc-reply message-id="101" xmlns="urn:ietf:params:xml:ns:netconf:base:1.0">
  <data>
    <install xmlns="http://cisco.com/ns/yang/Cisco-IOS-XR-install-oper">
      <request>
        <request>install commit</request>
        <state>success</state>
        <timestamp>2022-06-27 T02:52:07Z</timestamp>
        <operation-id>26</operation-id>
      </request>
    </install>
```

The state of the install operation in the RPC response indicates that the software and the RPMs are upgraded successfully.

What to do next

Perform preliminary checks to verify that the NCS 1020 is upgraded successfully.

Install IOS XR Image

Install ISO and RPMs

Use this procedure to install the base ISO and optional RPMs.

Before you begin

Ensure you have created a repository locally on the NCS 1020 or on a remote server which is reachable over HTTP, HTTPS or FTP. This repository will be used to copy the required RPMs. Ensure the NCS 1020 can reach the repository server over the Management Ethernet interface. For information about creating the repository to host the RPMs, see [Create Repository to Access Install Files, on page 59](#).

Step 1 You can either install from the remote repository or copy the ISO image file to the /harddisk: of the NCS 1020.

Example:

```
RP/0/RP0/CPU0:ios#scp root@<ip-address>:<dir>/ncs1010-x64-24.2.1.iso harddisk:
```

Step 2 To verify data integrity, verify the md5 checksum of the copied file with the original MD5 values on CCO.

Example:

```
RP/0/RP0/CPU0:ios#show md5 file /harddisk:/ncs1010-x64-24.2.1.iso
```

Step 3 Install the base image to upgrade the system.

- **Option 1:** Install ISO without control over reload timing.

```
RP/0/RP0/CPU0:ios#install replace /harddisk:/ncs1010-x64-24.2.1.iso
```

The image is installed, the changes are applied through a reload or a restart of the system, and commits the changes. However, you do not have control over the timing of the reload or restart—these occur as soon as the package operation completes and the system is ready.

If you want to control when your system reloads (management of a network outage), we recommend that you schedule an upgrade window and perform an **install replace**, letting the system reload without intervention.

- **Option 2:** Install ISO with control over reload timing.

- a. Install the image.

```
RP/0/RP0/CPU0:ios#install package replace /harddisk:/ncs1010-x64-24.2.1.iso
```

- b. Apply the changes.

```
RP/0/RP0/CPU0:ios#install apply [reload | restart]
```

You can use either the `reload` or `restart` options based on the file that is installed. To determine whether a `reload` or `restart` is required, check the output of **show install request** command. The output indicates the required actions.

Step 4 After the base image is upgraded, install the additional packages.

If a system fails to boot successfully, or reboots unexpectedly when the package is undergoing a version change, the system is automatically recovered to its old software state.

Note If you perform a manual or automatic system reload without completing the transaction with the **install commit** command, the action will revert the system to the point before the install transaction commenced, including any configuration changes. Only the log is preserved for debugging.

Install Golden ISO

Use this procedure to install the Golden ISO (GISO) that contains the base ISO and a customized list of optional RPMs that you built using the *gisobuild.py* tool. For details, see Build Customized Golden ISO Image.

Golden ISO (GISO) upgrades the NCS 1020 to a version that has a predefined list of bug fixes (sometimes also called software maintenance updates) with a single operation.

To update the system to the same release version with a different set of bug fixes:

- Create a GISO with the base version and all the bug fixes you require
- Use the **install replace** or **install package replace** commands to install the GISO.

The GISO can include bridging bug fixes for multiple source releases, and installs only the specific bridging bug fixes required for the target release.

The bridging bug fix RPMs can be used in the following scenarios:

- To resolve a bug that might stop upgrade.
- To meet the prerequisite requirements of a new release version that were not met by the earlier version.



Note The **install replace** command is supported only with GISO, but not with `.rpm` packages directly.

Step 1 Copy the GISO image file to either the `/harddisk:` of the NCS 1020 or a repository based on your requirement.

Example:

In this example, the image is copied to the /harddisk: of the NCS 1020.

```
RP/0/RP0/CPU0:ios#scp root@<ip-address>:/auto/tftp-test/ncs1010-x64-24.2.1.iso harddisk:
```

Step 2 Install the GISO.

- **Option 1:** Install GISO without control over reload timing.

- a. Install GISO to upgrade to a new release, add or remove bugfixes or optional packages.

```
RP/0/RP0/CPU0:ios#install replace source-location/giso-name.iso
```

The *source-location* can be one of the following locations based on step 1.

- Local path to the GISO—files located in or under /var/xr/disk1/, /harddisk:/ or /misc/disk1/
- Remote repository—ftp://<server>[;<vrf>]/<remote_path> or
http://<server>[;<vrf>]/<remote_path>

This command runs the replace operation and applies the new version via NCS 1020 restart or reload, whichever is least impactful, given the change. For example, if you have a GISO that is the same as your base image except one bugfix, and that bugfix can be applied by process restart, the command will install the bugfix and apply by restart, no NCS 1020 reload occurs. However, you do not have control over the timing of the reload or restart—these operations occur as soon as the packaging is complete and the system is ready. If you want to control the timing of system reloads, we recommend that you schedule an upgrade window and run the **install replace** command, allowing the system to reload without manual intervention or network impact.

- b. [Optional] Specify **reload** keyword to force reload for all operations. This may be useful if you want a reliable flow.
- c. [Optional] Specify **commit** keyword for the install, apply and commit operations to be performed without user intervention.

- **Option 2:** Install GISO with control over reload timing.

- a. Install GISO to upgrade to a new release, add or remove bugfixes or optional packages. The functionality is similar to **install replace** command, except that the staging of packaging changes is performed using this command.

```
RP/0/RP0/CPU0:ios#install package replace source-location/giso-name.iso
```

The **install package replace** command does not apply the changes.

- b. Apply the changes.

```
RP/0/RP0/CPU0:ios#install apply [reload | restart]
```

You can use either the `reload` or `restart` options based on the change that is installed. You can only apply the changes by restarting the software if the difference between the GISO being installed and the running image is minimal such as bugfixes or package updates.

To determine whether a `reload` or `restart` is required, check the output of **show install request** command. The output indicates the required actions.

Note

A GISO label is a string that identifies a GISO. Any install operation, such as adding or removing a package or modifying the software image (replace or package replace) will change the custom label to a system-generated default label. For example:

```
RP/0/RP0/CPU0:ios#show install active summary
Thu May 30 19:28:28.720 IST
Active Packages:   XR: 153   All: 1321
Label:             24.2.1.40I
XR Software Hash:  2cda979eb34cdbbbf204a5d4e080c4939d2e6311c044fcf4f634f8fe26fa3f38
```

Optional Packages	Version
xr-bgp	24.2.1.40Iv1.0.0-1
xr-healthcheck	24.2.1.40Iv1.0.0-1
xr-ipsla	24.2.1.40Iv1.0.0-1
xr-is-is	24.2.1.40Iv1.0.0-1
xr-lldp	24.2.1.40Iv1.0.0-1
xr-mps-oam	24.2.1.40Iv1.0.0-1
xr-netsim	24.2.1.40Iv1.0.0-1
xr-olc	24.2.1.40Iv1.0.0-1
xr-ospf	24.2.1.40Iv1.0.0-1
xr-perfmgmt	24.2.1.40Iv1.0.0-1
xr-track	24.2.1.40Iv1.0.0-1

In this example, the software image is modified to remove the CDP package.

```
RP/0/RP0/CPU0:ios#install package remove xr-cdp

Install remove operation 39.1.1 has started
Install operation will continue in the background
...
Packaging operation 39.1.1: 'install package remove xr-cdp' completed without error
```

Apply the changes.

```
RP/0/RP0/CPU0:ios#install apply
Thu Feb 02 11:13:09.015
Once the packaging dependencies have been determined, the install operation may have to reload
the system.
If you want more control of the operation, then explicitly use 'install apply restart' or
'install apply reload' as
reported by 'show install request'.
Continue? [yes/no]:[yes] yes
RP/0/RP0/CPU0:Feb 02 11:13:12.771 : instorch[404]: %INSTALL-6-ACTION_BEGIN : Apply by restart
39.1 started
Install apply operation 39.1 has started
Install operation will continue in the background
```

View the software version.

```
RP/0/RP0/CPU0:ios#show version
Wed Jun 12 14:43:12.934 IST
Cisco IOS XR Software, Version 24.2.1 LNT
Copyright (c) 2013-2024 by Cisco Systems, Inc.
```

```
Build Information:
  Built By       : cisco
  Built On       : Tue Jun 11 13:58:26 UTC 2024
  Build Host     : iox-ucs-033
  Workspace      : /auto/srcarchive11/prod/24.2.1/ncs1010/ws/
  Version        : 24.2.1
  Label          : 24.2.1
  cisco NCS1010 (C3758R @ 2.40GHz)
  cisco NCS1020-SA (C3758R @ 2.40GHz) processor with 32GB of memory
```

NCS 1020 Chassis

The `GIS01` custom label is replaced with the label `24.2.1` generated by the system.

Verify the Software Upgrade

This section provides information about the processes involved in verifying the upgraded software on your Cisco NCS 1020.

This section contains the following topics:

Check System Stability

System stability checks are essential to measure the efficiency and ability of an upgrade to function over an extended period.

At the EXEC prompt, execute the following commands to assess basic system stability checks before and after the software upgrade.

Command	Reason	Workaround
<code>show platform</code>	Verify that all nodes are in <code>IOS XR RUN/OPERATIONAL</code> state	NA
<code>show install active summary</code>	Verify that the proper set of packages are active	NA
<code>show install committed summary</code>	Verify that the proper set of committed packages are same as active	Execute 'install commit' command
<code>clear configuration inconsistency</code>	Verify/fix configuration file system	NA
<code>show hw-module fpd</code>	Ensure all the FPD versions status are <code>CURRENT</code>	Execute <code>upgrade hw-module fpd</code> command
<code>show media</code>	Display the current state of the disk storage media	To free up space, remove older .iso image files and bug fix .tar files.
<code>show inventory</code>	Show chassis inventory information	NA

NCS 1020 FPD

A Field Programmable Device (FPD) refers to any programmable hardware device on a chassis, which includes a Field Programmable Gate Array (FPGA). NCS 1020 uses several FPDs that are necessary for chassis, route processor, line cards, and power modules to function properly.



Note If the FPD in a given SSD is not supported by the current IOS XR software release, the status is not displayed.

The following table lists the NCS 1020 FPDs that are distributed across route processor (RP), power modules (PM), line cards (LC), and Rack.

Table 3: NCS 1020 FPDs

Location	FPDs
RP	<ul style="list-style-type: none"> • ADMConfig • CpuFpga • CpuFpgaGolden • BIOS • BIOS-Golden • SsdIntelS4510 • SsdMicron5300 • SsdMicron5400 • TamFw • TamFwGolden
PM0 and PM1	<ul style="list-style-type: none"> • PO-PrimMCU • PO-SecMCU
LC	<ul style="list-style-type: none"> • ILA • OLT • Raman-1 • Raman-2 • CpuModFw • OptModFw

Location	FPDs
Rack	<ul style="list-style-type: none"> • IoFpgaLow • IoFpgaUp • IoFpgaLowGolden • IoFpgaUpGolden • ADMCONFIG • SsdIntelISC2KB • SsdMicron5400

Golden FPDs serve as backup FPDs for the primary FPDs. For example, **BIOS-Golden** is the backup Golden FPD for the **BIOS** primary FPD. If a primary FPD is corrupted, NCS 1020 boots with the corresponding Golden FPD. The Golden FPDs cannot be upgraded.

Retrieve FPD Information

There are multiple types of FPDs for each type of module. The **show hw-module fpd** command provides information about each FPD.

```
RP/0/RP0/CPU0:ios#sh hw-module fpd
Wed Apr 24 15:54:04.551 IST
```

```
Auto-upgrade:Enabled
Attribute codes: B golden, P protect, S secure, A Anti Theft aware
```

Location Reload Loc	Card type	HWver	FPD device	ATR	Status	FPD Versions	
						Running	Programd
0/RP0/CPU0 NOT REQ	NCS1010-CTR2-B-K9	0.1	ADMCONFIG		CURRENT	1.00	1.00
0/RP0/CPU0 0/RP0	NCS1010-CTR2-B-K9	0.1	BIOS	S	CURRENT	5.20	5.20
0/RP0/CPU0 0/RP0	NCS1010-CTR2-B-K9	0.1	BIOS-Golden	BS	CURRENT		1.90
0/RP0/CPU0 0/RP0	NCS1010-CTR2-B-K9	0.1	CpuFpga	S	CURRENT	1.06	1.06
0/RP0/CPU0 0/RP0	NCS1010-CTR2-B-K9	0.1	CpuFpgaGolden	BS	CURRENT		1.02
0/RP0/CPU0 0/RP0	NCS1010-CTR2-B-K9	0.1	SsdMicron5300	S	CURRENT	0.01	0.01
0/RP0/CPU0 0/RP0	NCS1010-CTR2-B-K9	0.1	TamFw	S	CURRENT	9.07	9.07
0/RP0/CPU0 0/RP0	NCS1010-CTR2-B-K9	0.1	TamFwGolden	BS	CURRENT		9.06
0/PM0 NOT REQ	NCS1K4-AC-PSU-2	1.0	PO-PrimMCU		CURRENT	1.03	1.03
0/PM0 NOT REQ	NCS1K4-AC-PSU-2	1.0	PO-SecMCU		CURRENT	1.05	1.05
0/PM1 NOT REQ	NCS1K4-AC-PSU-2	1.0	PO-PrimMCU		CURRENT	1.03	1.03
0/PM1 NOT REQ	NCS1K4-AC-PSU-2	1.0	PO-SecMCU		CURRENT	1.05	1.05
0/0/NXR0 NOT REQ	NCS1K-E-OLT-C	1.0	OLT	S	CURRENT	3.14	3.14
0/2/NXR0	NCS1K14-CCMD-16-C	1.0	CpuModFw	S	CURRENT	42.14	42.14

```

NOT REQ
0/2/NXR0 NCS1K14-CCMD-16-C 1.0 OptModFw S CURRENT 20.02 20.02
NOT REQ
0/3/NXR0 NCS1K14-CCMD-16-C 1.0 CpuModFw S CURRENT 42.14 42.14
NOT REQ
0/3/NXR0 NCS1K14-CCMD-16-C 1.0 OptModFw S CURRENT 20.02 20.02
NOT REQ
0/Rack NCS1020-SA 0.1 ADMCONFIG CURRENT 1.00 1.00
NOT REQ
0/Rack NCS1020-SA 0.1 IoFpgaLow S CURRENT 1.08 1.08
NOT REQ
0/Rack NCS1020-SA 0.1 IoFpgaLowGolden BS CURRENT 0.07
NOT REQ
0/Rack NCS1020-SA 0.1 IoFpgaUp S CURRENT 1.08 1.08
NOT REQ
0/Rack NCS1020-SA 0.1 IoFpgaUpGolden BS CURRENT 0.06
NOT REQ
0/Rack NCS1020-SA 0.1 SsdMicron5400 S CURRENT 0.02 0.02
0/Rack

```

The following table describes the significant fields in the output of the **show hw-module fpd** command.

Table 4: Description of Fields in show hw-module fpd Command

Field	Description
Location	Location of the FPD.
Card type	PID of the modules such as chassis, card, CPU, and PSU.
HWver	Hardware version where the FPD resides.
FPD device	Name of the FPD.
ATR	Attribute codes. The possible values are: <ul style="list-style-type: none"> • B - Golden Image • S - Secure Image • P - Protect Image The attribute code of the primary FPDs is S and the Golden FPDs is BS.
Status	Status of the FPD. See Table 5: Description of FPD Status Values in show hw-module fpd Command, on page 72.
Running	FPD image version that has been activated and currently running in the FPD device.
Programd	FPD image version that has been programmed into the FPD device, but might not be activated.
Reload Loc	Indicates whether reload of the location is required or not.

The following table describes the possible values of the Status field in the output of the **show hw-module fpd** command.

Table 5: Description of FPD Status Values in show hw-module fpd Command

FPD Status	Description
NOT READY	The driver that owns the FPD device has not initialized the FPD client to handle this device.
CURRENT	FPD version is up to date and upgrade is not required.
NEED UPGD	Upgrade is required for this FPD. Check the output of the show fpd package command to determine the recommended FPD version.
UPGD PREP	FPD is preparing for upgrade.
IN QUEUE	Upgrade of this FPD is in queue.
UPGD SKIP	FPD upgrade is not required. For example, <ul style="list-style-type: none"> • FPD version is up to date and compatible. • FPD image is protected.
UPGRADING	FPD upgrade started and the driver did not report the upgrade progress information yet.
%UPGD	Percentage of FPD upgrade completion.
RLOAD REQ	FPD upgrade is successfully completed and the FPD must be reloaded for the new version to take effect.
UPGD FAIL	FPD upgrade has failed. Check the syslog for failure reason. It could be a timeout or a failure that is reported by the driver.
UPGD DONE	FPD upgrade is successfully completed.

Verify if an FPD Upgrade is Required

Step 1 Use the **show hw-module fpd** command to check whether all the FPDs are in the Current state.

If the status of any FPD is **NEED UPGD**, then the upgrade is required for that FPD.

Step 2 Use the **show fpd package** command to determine the FPDs that are supported with the current software release and the minimum hardware requirements for each FPD.

```
RP/0/RP0/CPU0:ios#show fpd package
Wed Apr 24 15:59:13.897 IST
```

```
=====
Field Programmable Device Package
=====
```

Card Type	FPD Description	Req Reload	SW Ver	Min Req SW Ver	Min Req Board Ver
NCS1010-CTR2-B-K9	ADMCONFIG	NO	1.00	1.00	0.1
	BIOS	YES	5.20	5.20	0.0
	BIOS-Golden	YES	5.10	0.01	0.0
	CpuFpga	YES	1.06	1.06	0.0
	CpuFpgaGolden	YES	1.02	0.01	0.0
	SsdIntelS4510	YES	11.32	11.32	0.0
	SsdMicron5300	YES	0.01	0.01	0.0
	SsdMicron5400	YES	0.02	0.02	0.0
	TamFw	YES	9.07	9.07	0.0
TamFwGolden	YES	9.06	0.01	0.0	
NCS1010-CTR2-K9	ADMCONFIG	NO	1.00	1.00	0.1
	BIOS	YES	5.20	5.20	0.0
	BIOS-Golden	YES	5.10	0.01	0.0
	CpuFpga	YES	1.06	1.06	0.0
	CpuFpgaGolden	YES	1.02	0.01	0.0
	SsdIntelS4510	YES	11.32	11.32	0.0
	SsdMicron5300	YES	0.01	0.01	0.0
	SsdMicron5400	YES	0.02	0.02	0.0
	TamFw	YES	9.07	9.07	0.0
TamFwGolden	YES	9.06	0.01	0.0	
NCS1020-SA	ADMCONFIG	NO	1.00	1.00	0.0
	IoFpgaLow	NO	1.08	1.08	0.0
	IoFpgaLowGolden	NO	0.07	0.01	0.0
	IoFpgaUp	NO	1.08	1.08	0.0
	IoFpgaUpGolden	NO	0.06	0.01	0.0
	SsdIntelSC2KB	YES	1.20	1.20	0.0
	SsdMicron5400	YES	0.02	0.02	0.0
NCS1K-E-ILA-2R-C	ILA	NO	3.14	3.14	0.1
	ILA	NO	0.28	0.28	99.1
	Raman-1	NO	3.14	3.14	0.1
	Raman-1	NO	0.28	0.28	99.1
	Raman-2	NO	3.14	3.14	0.1
	Raman-2	NO	0.28	0.28	99.1
NCS1K-E-ILA-R-C	ILA	NO	3.14	3.14	0.1
	ILA	NO	0.28	0.28	99.1
	Raman-1	NO	3.14	3.14	0.1
	Raman-1	NO	0.28	0.28	99.1
NCS1K-E-ILA-R-C-2	ILA	NO	3.14	3.14	0.1
	ILA	NO	0.28	0.28	99.1
	Raman-2	NO	3.14	3.14	0.1
	Raman-2	NO	0.28	0.28	99.1
NCS1K-E-OLT-C	OLT	NO	3.14	3.14	0.1
	OLT	NO	0.28	0.28	99.1
NCS1K-E-OLT-L	OLT	NO	3.12	3.12	0.1
NCS1K-E-OLT-R-C	OLT	NO	3.14	3.14	0.1
	OLT	NO	0.28	0.28	99.1
	Raman-1	NO	3.14	3.14	0.1
	Raman-1	NO	0.28	0.28	99.1
NCS1K-ILA-2R-C	ILA	NO	3.14	3.14	0.1
	ILA	NO	0.28	0.28	99.1
	Raman-1	NO	3.14	3.14	0.1

	Raman-1	NO	0.28	0.28	99.1
	Raman-2	NO	3.14	3.14	0.1
	Raman-2	NO	0.28	0.28	99.1

NCS1K-ILA-C	ILA	NO	3.14	3.14	0.1
	ILA	NO	0.28	0.28	99.1

NCS1K-ILA-L	ILA	NO	3.12	3.12	0.1

NCS1K-ILA-R-C	ILA	NO	3.14	3.14	0.1
	ILA	NO	0.28	0.28	99.1
	Raman-1	NO	3.14	3.14	0.1
	Raman-1	NO	0.28	0.28	99.1

NCS1K-OLT-C	OLT	NO	3.14	3.14	0.1
	OLT	NO	0.28	0.28	99.1

NCS1K-OLT-L	OLT	NO	3.12	3.12	0.1

NCS1K-OLT-R-C	OLT	NO	3.14	3.14	0.1
	OLT	NO	0.28	0.28	99.1
	Raman-1	NO	3.14	3.14	0.1
	Raman-1	NO	0.28	0.28	99.1

NCS1K14-CCMD-16-C	CpuModFw	NO	42.14	42.14	0.0
	OptModFw	NO	20.02	20.02	0.0

NCS1K14-CCMD-16-L	CpuModFw	NO	42.14	42.14	0.0
	OptModFw	NO	20.02	20.02	0.0

NCS1K4-AC-PSU-2	PO-PrimMCU	NO	1.03	1.03	0.1
	PO-SecMCU	NO	1.05	1.05	0.1

The following table describes the fields in the output of the **show fpd package** command.

Table 6: Description of Fields in show fpd package Command

Field	Description
Card Type	PID of the modules such as chassis, card, CPU, and PSU.
FPD Description	Description of the FPD.
Req Reload	Determines whether reload is required to activate the FPD image.
SW Ver	Recommended FPD software version for the associated module running the current Cisco IOS XR Software.
Min Req SW Ver	Minimum required FPD software version to operate the module.
Min Req Board Ver	Minimum required hardware version for the associated FPD. A minimum hardware requirement of version 0.0 indicates that all the hardware can support this FPD version.

FPD can be upgraded using two methods:

- [Upgrade FPDs Manually](#)

- [Upgrade FPDs Automatically](#)

Upgrade FPDs Manually

Use the following procedure to upgrade the FPDs manually.



Note The Golden FPDs cannot be upgraded using the CLI.

-
- Step 1** Use the **show hw-module fpd** command to display information about the current FPD version.
You can use this command to determine if you must upgrade the FPD.
- Step 2** Use the **show alarms brief system active** command to display the active alarms.
You must upgrade the FPD when the **One Or More FPDs Need Upgrade Or Not In Current State** alarm is present.
- Step 3** Use the **upgrade hw-module location [location-id] fpd [fpd name]** command to upgrade a specific FPD.
After upgrading the FPD, the user must wait for upgrade completion. The progress of the FPD upgrade can be monitored using the **show hw-module fpd** command.
- Example:**
- ```
RP/0/RP0/CPU0:ios#upgrade hw-module location 0/Rack fpd IoFpgaLow
```
- Note** The FPDs of power modules belong to 0/PM0 and 0/PM1 locations. The FPDs belonging to both the PM locations cannot be simultaneously upgraded.
- Step 4** Use the **reload location location-id** to reload the FPDs belonging to a specific location with the new version.  
The **Reload Loc** field in the output of **show hw-module fpd** command indicates whether the reload is required or not.
- Example:**
- ```
RP/0/RP0/CPU0:ios#reload location 0/RP0/CPU0
```
- Step 5** (Optional) Use the **upgrade hw-module location all fpd all** command to upgrade all the FPDs at once.
- Step 6** (Optional) Use the **upgrade hw-module [location [location-id | all]] fpd [fpd name] | all** command to upgrade a specific FPD, all the FPDs, or the FPDs belonging to a specific location.
- Example:**
- ```
RP/0/RP0/CPU0:ios#upgrade hw-module location all fpd all
```
- Note** The FPDs of power modules and SSDs cannot be forcefully upgraded.
-

## Upgrade FPDs Automatically

The automatic FPD upgrade upgrades the FPD version of all the modules to the latest version. When automatic FPD upgrade is enabled, all the FPDs (except the Golden FPDs) that are in NEED UPGD status are upgraded to CURRENT status during the software upgrade.

In NCS 1020, automatic FPD upgrade is enabled by default.

---

Use the following commands to disable automatic FPD upgrade.

**Example:**

```
RP/0/RP0/CPU0:ios#configure
RP/0/RP0/CPU0:ios(config)#fpd auto-upgrade disable
RP/0/RP0/CPU0:ios(config)#commit
RP/0/RP0/CPU0:ios(config)#end
```

---



## CHAPTER 4

# Disaster Recovery

---

This chapter describes the disaster recovery process and the health check feature.

- [Overview, on page 77](#)
- [SSD OIR, on page 77](#)
- [CPU Controller Replacement Considerations, on page 77](#)
- [Chassis SSD Replacement Considerations, on page 78](#)
- [Health Check of Backup ISO Image, on page 78](#)

## Overview

There are two partitions in NCS 1020: The controller SSD (CPU partition) and chassis SSD (Disaster Recovery partition). The Disaster Recovery partition contains all the backup configurations such as ISO images, RPMs, and system configuration files. When the node is corrupted, the Disaster Recovery feature allows the CPU to be replaced with the existing configuration. After replacing the CPU, the node reboots and comes up by restoring the software and configuration files from the chassis SSD without traffic loss.



---

**Note** When Chassis SSD is corrupted and replaced, the new chassis SSD takes backup of the running software and configuration files from the controller SSD without traffic loss.

---

## SSD OIR

The chassis Solid State Drive (SSD) is a semiconductor-based storage device. It is used to store a backup of the software and configuration which will be used by the CPU to re-boot in case of any node failure. Online Insertion and Removal (OIR) is an operation that allows you to replace any faulty module in a system. In NCS 1020, the SSD OIR feature allows the chassis SSD to be removed and replaced anytime.

## CPU Controller Replacement Considerations

You must consider the following points for CPU replacement.

Table 7: CPU Controller Replacement

| Type of Replacement                                                                                 | Result                                                                                                                                                                 |
|-----------------------------------------------------------------------------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| When the CPU is removed from the chassis.                                                           | NCS 1020 chassis runs in a headless mode which is non-traffic impacting.                                                                                               |
| When the CPU is replaced with another CPU having the same software and RPMs as in the chassis SSD.  | The configuration is restored from the chassis SSD.                                                                                                                    |
| When the CPU is replaced with another CPU having different software and RPMs as in the chassis SSD. | The Disaster recovery process starts. In this case, the node boots with the software from the chassis SSD and the configuration is also restored from the chassis SSD. |

## Chassis SSD Replacement Considerations

The chassis SSD can be removed in NCS 1020.

You must consider the following points for chassis SSD replacement.

Table 8: Chassis SSD Replacement

| Type of Replacement                                                                                             | Result                                                                |
|-----------------------------------------------------------------------------------------------------------------|-----------------------------------------------------------------------|
| When the chassis SSD is removed while the RP is running.                                                        | Disaster Recovery boot is disabled and an alarm is raised.            |
| When the chassis SSD is removed.                                                                                | The Disaster Recovery Unavailable alarm is raised.                    |
| When the chassis SSD is replaced with another chassis SSD having the same software and RPMs as in the CPU SSD.  | The configuration is backed up from the CPU SSD.                      |
| When the Chassis SSD is replaced with another chassis SSD having different software and RPMs as in the CPU SSD. | Software, RPMs, and the configuration are backed up from the CPU SSD. |
| When the Chassis SSD is replaced with a spare received from Cisco manufacturing or RMA process.                 | Software, RPMs, and the configuration are backed up from the CPU SSD. |

## Health Check of Backup ISO Image

The Health Check feature ensures error-free booting of NCS 1020 chassis during disaster recovery operations. NCS 1020 has a partition for disaster recovery where the backup ISO image is stored. The backup ISO image is stored in the chassis SSD.

The chassis SSD content is audited against the running software by the install process in the background every 12 hours to detect corruption. If the ISO image is corrupted, the software will recover it by copying from the

backup location. If the software fails to synchronize with the chassis SSD, then the **Disaster Recovery ISO Image Corruption** alarm is raised. See the *Troubleshooting Guide for Cisco NCS 1020* to clear the alarm.





## CHAPTER 5

# Connection Verification

This chapter describes the tasks to verify connection between the OLT Line Card of NCS 1020 and NCS1K14-CCMD-16-C line card.

- [Power Data Reading, on page 81](#)
- [Connection Verification, on page 81](#)

## Power Data Reading

Photodiodes (PDs) are optical power monitors available on all input and aggregated output ports to monitor power levels. Tone detection is enabled on some PD monitors.

*Table 9: NCS1K-CCMD-16 Calibrated Port References*

| Port Calibrated | Port Label (Direction) | Minimum Power (dBm) | Maximum Power (dBm) | Dynamic Range (dBm) |
|-----------------|------------------------|---------------------|---------------------|---------------------|
| LC input ports  | (TX)                   | -50                 | 30                  | 80                  |
| LC output ports | (RX)                   | -50                 | 30                  | 80                  |

## Connection Verification

Connection verification checks the connection between the OLT line card and the CCMD-16 line cards to avoid miscabling during the node installation. The dedicated Connection Verification Tunable Laser (CV-TL) available at the OLT card generates a specific probe signal at a given frequency and power. This signal is detected by the CCMD-16 line card that is connected to the OLT line card.

- The same OLT-C line card
- The CCMD-16 line card that is connected to the OLT line card.
- A different unit (OLT-C line card or passive module) belonging to the same NE
- An optical interface (Router ports or Transponder) connected to the OLT-C line card

Connection verification uses a probe signal or adds a low frequency ON/OFF modulation tone transmitting a given tone pattern at 5 Hz (200 ms bit time). The tone pattern length ranges 4–32 bytes (including an

alignment byte) and it includes the Cable-IDs of the cables in the connection and in case also the optical frequency of the specific connection.

The Cable-ID is generated by the Optical Node Controller supervising the complete NE.

The connection verification process uses the out-of-band (OOB) and in-band (IB) WSS frequencies to reach the CCMD-16 line card.

## CCMD-16 Connection Verification with OLT

The OLT line card generates the tone and connection verification is performed using the OOB channel with CV-TL tuned at 191.175 THz. To univocally identify the optical path under test, the CV-TL is modulated with a low-frequency pattern including the Cable ID of the connection.

For connection verification toward the CCMD-16 card, the CV-TL is routed to the PD inside the CCMD-16 card. The out-of-band (OOB) and the in-band (IB) connections are verified at two different PDs on the CCMD-16 line cards.

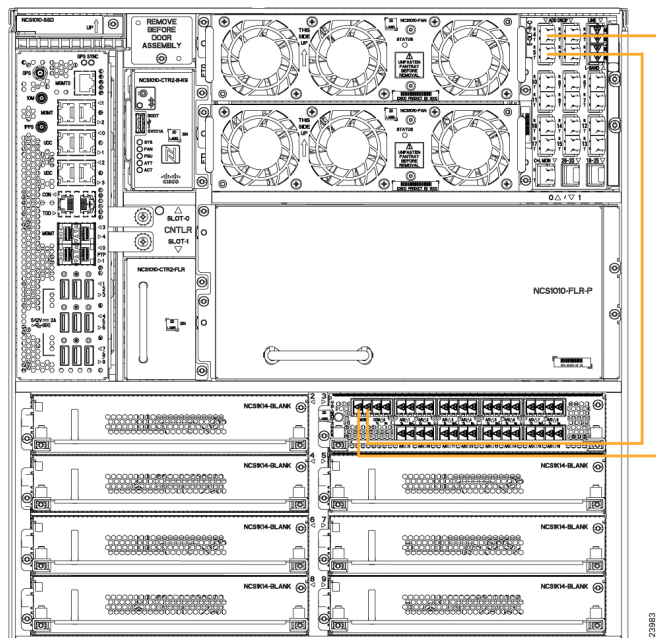
The PD monitors receiving a connection verification signal detect and buffer the Cable-ID pattern encoded in the tone to allow the connection verification process by the node controller.

### Verify Connection for CCMD-16 Line Card

The connection verification procedure checks the connection between the OLT line card and CCMD-16 line cards to match the different instances regarding the OLT LC connectors.

The OLT-C line card and the NCS1K-CCMD-16 line card are connected as shown in the following image:

**Figure 2: NCS 10120 and NCS1K-CCMD-16 Connection**



The OLT-C line card performs connection verification between the OLT-C line card and the NCS1KCCMD-16 line card panels as described in [CCMD-16 Connection Verification with OLT](#), on page 82.



The identification/verification of the NCS1K-CCMD-16 line card is performed by checking the connection verification signal at the monitor present on the OOB loop and IB PD of the NCS1K-CCMD-16 line card respectively.

This task describes on how to verify the connection between the NCS 1010/ NCS 1020 OLT-C line card and NCS1K-CCMD-16 line card.

Start tone-pattern on OTS controller.

### Before you begin

Configure the OTS controller in NCS 1020 to generate the tone for connection verification. See [Connection Verification on OTS Controller](#).

**Step 1** Configure the OTS controller to generate the tone for connection verification.

#### Example:

```
RP/0/RP0/CPU0:ios(config)#controller ots 0/0/0/4
RP/0/RP0/CPU0:ios(config-Ots)#tone-rate 25
RP/0/RP0/CPU0:ios(config-Ots)#tone-frequency 191.175 (OOB frequency)
RP/0/RP0/CPU0:ios(config-Ots)#tone-pattern abcd1234
RP/0/RP0/CPU0:ios(config-Ots)#commit
```

**Step 2** Configure the OMS controller to detect the tone for connection verification.

#### Example:

```
RP/0/RP0/CPU0:ios(config)#controller oms 0/2/0/0
RP/0/RP0/CPU0:ios(config-Oms)#tone-rate 25
RP/0/RP0/CPU0:ios(config-Oms)#tone-pattern-expected aabbccdd
RP/0/RP0/CPU0:ios(config-Oms)#tone-detect-oob
RP/0/RP0/CPU0:ios(config-Oms)#commit
```

**tone-detect-oob** must be configured on the OMS x/x/x/0 for NCS1K-CCMD-16.

**Step 3** Start the **tone-pattern** on the OTS controller.

#### Example:

```
RP/0/RP0/CPU0:ios#tone-pattern controller ots 0/0/0/4 start
Tue May 10 11:37:51.597 UTC
Tone pattern started
```

When tone generation is in progress on the OTS interface, the tone generation on other OTS interfaces is not allowed until the current tone generation is stopped.

**Step 4** Use the **tone-pattern-detect** command to start the detection of tone pattern.

#### Example:

The following is a sample on starting the tone pattern detection on the OMS controller.

```
RP/0/RP0/CPU0:ios#tone-pattern-detect controller oms 0/2/0/0 start
Tue May 10 11:38:03.775 UTC
Tone pattern detect started
```

**Step 5** Use the **tone-info** command to check for successful connection verification.

#### Example:

The following is a sample to view the Tone Info for successful connection verification on the OMS controller.

```
RP/0/RP0/CPU0:ios#show controllers oms 0/2/0/0 tone-info
Fri Sep 22 06:04:03.787 UTC
```

```
Tone Info:
Tone Rate : 25 bits/second
Tone Pattern Expected(Hex value) : aabbccdd
Tone Pattern Received(Hex value) : aabbccdd
Tone Detected OOB : Enabled
Detection State: Success
```

The following is a sample to view the Tone Info for failed connection verification on the OMS controller.

```
RP/0/RP0/CPU0:ios#show controllers oms 0/2/0/0 tone-info
Fri Sep 22 11:10:22.425 UTC
Tone Info:
Tone Frequency : 191.1750000 THz
Tone Rate : 25 bits/second
Tone Pattern Expected(Hex value) : aabbccdd
Tone Pattern Received(Hex value) : 12b36bd3e1
Tone Detected OOB : Enabled
Detection State: Failed
```

**Step 6** After successful connection verification, stop **tone-pattern-detect** on the OMS controller.

**Example:**

```
RP/0/RP0/CPU0:ios#tone-pattern-detect controller oms 0/2/0/0 stop
Fri Sep 22 06:23:15.165 UTC
Tone pattern detect stopped
```

**Step 7** Stop the tone-pattern generation on the OTS controller.

**Example:**

```
RP/0/RP0/CPU0:ios#tone-pattern controller ots 0/0/0/4 stop
Wed Sep 22 06:25:25.187 UTC
Tone pattern stopped
```

---

## Connection Verification on OTS Controller

This task describes how to check OTS interface connectivity on OLT nodes.

---

**Step 1** Start tone-pattern on OTS controller.

**Example:**

```
RP/0/RP0/CPU0:ios#tone-pattern controller ots 0/0/0/4 start
Wed May 25 11:59:51.040 UTC
Tone pattern started
```

**Step 2** Start tone-pattern-detect on OTS controller on one side.

**Example:**

```
RP/0/RP0/CPU0:ios#tone-pattern-detect controller ots 0/0/0/4 start
Wed May 25 12:00:03.271 UTC
Tone pattern detect started
```

**Step 3** Configure the OTS controller to generate the tone for connection verification.

**Example:**

```
RP/0/RP0/CPU0:ios(config)#controller ots 0/0/0/4
RP/0/RP0/CPU0:ios(config-Ots)#tone-rate 25
RP/0/RP0/CPU0:ios(config-Ots)#tone-frequency 191.175 (OOB frequency)
```

```
RP/0/RP0/CPU0:ios(config-Ots)#tone-pattern abcd1234
RP/0/RP0/CPU0:ios(config-Ots)#tone-detect-oob
RP/0/RP0/CPU0:ios(config-Ots)#tone-pattern-expected abcd1234
RP/0/RP0/CPU0:ios(config-Ots)#commit
```

**Step 4** Check for successful connection verification on the Line 2 OTS controller.

**Example:**

```
RP/0/RP0/CPU0:ios#show controllers ots 0/0/0/4 tone-info
Wed May 25 12:00:11.393 UTC
Tone Info:
Tone Frequency : 191.1750000 THz
Tone Rate : 20 bits/second
Tone Pattern(Hex value) : abcd1234
Tone Pattern Expected(Hex value) : abcd1234
Tone Pattern Received(Hex value) : abcd1234
Tone Detected OOB : Enabled
Detection State: Success
```

**Step 5** Stop the tone-pattern-detect on the OTS controller.

**Example:**

```
RP/0/RP0/CPU0:ios#tone-pattern-detect controller ots 0/0/0/4 stop
Wed May 25 12:00:56.540 UTC
Tone pattern detect stoped
```

**Step 6** Stop the tone-pattern generation on the OTS controller.

**Example:**

```
RP/0/RP0/CPU0:ios#tone-pattern controller ots 0/0/0/4 stop
Wed May 25 12:01:04.226 UTC
Tone pattern stopped
```

---





## CHAPTER 6

# Smart Licensing

This chapter describes the smart licensing configuration on Cisco NCS 1020.

- [Understanding Smart Licensing, on page 87](#)
- [License Entitlements of NCS 1020, on page 89](#)
- [Create an ID Token, on page 90](#)
- [Smart Licensing Transport Modes, on page 91](#)
- [Reserve Specific Licenses for NCS 1020, on page 95](#)

## Understanding Smart Licensing

Smart Licensing is a cloud-based, software license management solution that enables you to automate time-consuming, manual licensing tasks. The solution allows you to easily track the status of your license and software usage trends.

Smart Licensing helps you simplify three core functions:

- **Purchasing:** The software that you have installed in your network can be registered, without Product Activation Keys (PAKs).
- **Management:** You can automatically track activations against your license entitlements. Also, there is no need to install the license file on every node. You can create license pools (logical grouping of licenses) to reflect your organization structure. Smart Licensing offers you Cisco Smart Software Manager, a centralized portal that enables you to manage all your Cisco software licenses from one centralized website.
- **Reporting:** Through the portal, Smart Licensing offers an integrated view of the licenses you have purchased and what has been deployed in your network. You can use this data to make better purchasing decisions, based on your consumption.

### Smart Licensing Features

- Your device initiates a call home and requests the licenses it needs.
- Pooled licenses - Licenses are company account-specific, and can be used with any compatible device in your company. You can activate or deactivate different types of licenses on the device without actually installing a license file on the device.
- Licenses are stored securely on Cisco servers.

- Licenses can be moved between product instances without license transfer. This greatly simplifies the reassignment of a software license as part of the Return Material Authorization (RMA) process.
- It provides a complete view of all the Smart Software Licenses used in the network using a consolidated usage report of software licenses and devices in one easy-to-use portal.

### Cisco Smart Account

Cisco Smart Account is an account where all products enabled for Smart Licensing are deposited. Cisco Smart Account allows you to manage and activate your licenses to devices, monitor license use, and track Cisco license purchases. Through transparent access, you have a real-time view into your Smart Licensing products. IT administrators can manage licenses and account users within your organization's Smart Account through the Smart Software Manager.

When creating a Smart Account, you must have the authority to represent the requesting organization. After you submit the request, it goes through a brief approval process. Access <http://software.cisco.com> to learn about, set up, or manage Smart Accounts.

Cisco Smart Software Manager enables you to manage all your Cisco Smart software licenses from one centralized website. With Cisco Smart Software Manager, you organize and view your licenses in groups called virtual accounts (collections of licenses and product instances). Use the Cisco Smart Software Manager to do the following tasks:

- Create, manage, or view virtual accounts.
- Create and manage Product Instance ID Tokens.
- Transfer licenses between virtual accounts or view licenses.
- Transfer, remove, or view product instances.
- Run reports against your virtual accounts.
- Modify your email notification settings.
- View overall account information.

### Virtual Accounts

A Virtual Account exists as a subaccount within the Smart Account. Virtual Accounts are a customer-defined structure based on organizational layout, business function, geography, or any defined hierarchy. They are created and maintained by the Smart Account administrator. Smart Licensing allows you to create multiple license pools or virtual accounts within the Smart Software Manager portal. Using the Virtual Accounts option that you can aggregate licenses into discrete bundles that are associated with a cost center so that one section of an organization cannot use the licenses of another section of the organization. For example, if you segregate your company into different geographic regions, you can create a virtual account for each region to hold the licenses and product instances for that region.

All new licenses and product instances are placed in the default virtual account in the Smart Software Manager, unless you specify a different one during the order process. After you access the default account, you may choose to transfer them to any other account, provided you have the required access permissions.

Use the Smart Software Manager portal to create license pools or transfer licenses.

### Product Instance ID Tokens

ID tokens are stored in the Product Instance ID Token Table that is associated with your enterprise account. ID tokens can be valid 1–365 days.

## Product Instances

A product instance is an individual device with a unique device identifier (UDI) that is registered using a product instance ID token (or ID token). You can register any number of instances of a product with a single ID token. Each product instance can have one or more licenses residing in the same virtual account. Product instances must periodically connect to the Cisco Smart Software Manager servers during a specific renewal period. If you remove the product instance, its licenses are released and made available within the virtual account.

## Smart Licensing Work Flow

The following figure depicts a working model of smart licensing that involves a three-step procedure.

**Figure 3: Smart Licensing Work Flow**



# License Entitlements of NCS 1020

Cisco NCS 1020 supports the Vortex model for smart licensing software, and has the following Essential and Advantage licenses:

- ILA Essentials—One per ILA card (Maximum one license)
- OLT Essentials—One per WSS port per OLT card (Maximum 32 licenses).
- ILA Advantage—One per ILA card for Timing access and OTDR access features (Maximum 1 license).
- OLT Advantage—One per WSS port per OLT card for Timing access, OTDR access, Connection Verification features (Maximum 32 licenses)
- CCMD Essentials—One per CCMD card (Maximum one license)

With Nyquist channels enabled, for each cross connection that is created on a port, one Essentials license and one Advantage license are consumed.

The following table lists the various licenses that can be enabled on Cisco NCS 1020.

Table 10: NCS 1020 License Entitlements

| Display Name in CSSM Server | Description                                             |
|-----------------------------|---------------------------------------------------------|
| NCS1010_ADV_ILA_RTU         | NCS 1010 ILA Advantage Right-to-Use (RTU)               |
| NCS1010_ADV_ILA_SIA         | NCS 1010 ILA Advantage Software Innovation Access (SIA) |
| NCS1010_ADV_OLT_RTU         | NCS 1010 OLT Advantage RTU (per port)                   |
| NCS1010_ADV_OLT_SIA         | NCS 1010 OLT Advantage SIA (per port)                   |
| NCS1010_ESS_ILA_RTU         | NCS 1010 ILA Essentials RTU                             |
| NCS1010_ESS_ILA_SIA         | NCS 1010 ILA Essentials SIA                             |
| NCS1010_ESS_OLT_RTU         | NCS 1010 OLT Essentials RTU (per port)                  |
| NCS1010_ESS_OLT_SIA         | NCS 1010 OLT Essentials SIA (per port)                  |
| NCS1010_CCMD_CDMS_SIA       | NCS 1010 CCMD Essentials SIA                            |

## Create an ID Token

To create a new token using Cisco Smart Software Manager, perform the following tasks:

### Before you begin

- 
- Step 1** Log in to the Cisco Smart Software Manager.  
<https://software.cisco.com/software/cswws/platform/home#SmartLicensing-Inventory>
- Step 2** Click the **Inventory** tab, and select your virtual account from the **Virtual Account** drop-down list.
- Step 3** Click the **General** tab, and click **New Token**.  
 The **Create ID Token** window is displayed.
- Step 4** Enter the token description. Specify the number of days the token must be active.
- Step 5** Check the **Allow export-controlled functionality on the products registered with this token** check box.
- Step 6** Click **Create ID Token**.
- Step 7** Copy the token and register NCS1020 with the same token ID.

An example of the token ID: YzY2ZjYyNjktY2NlOS00NTc4LWlxNTAtMjZkNmNiNzMxMTY1LTE2NjAzNjQ3%0ANzY4NjI8ZVJSckxKN2pFV2tIeHV0MUkxbGxTazFDVm9kc1B5MGhQmIFWUJi%0Ac3VNRT0%3D%0A

---



# Smart Licensing Transport Modes

Smart Licensing software management solution enables you to choose from one of the three transport modes, Cisco Smart Licensing Utility(CSLU), Smart Transport or Offline modes. This is in addition to the existing Call-Home mode. The default transport mode is CSLU, but you can change the mode to Call-Home, Smart Transport or Offline mode.

The following transport modes are available for you to choose now:

- Call-Home
- Smart
- CSLU
- Offline

## Configure Callhome

You can use the Call Home to connect to the CSSM. To configure callhome in Cisco NCS 1020, perform the following steps:

**Step 1** Use this sample configuration to enable call home mode settings.

**Example:**

```
RP/0/RP0/CPU0:ios#call-home
RP/0/RP0/CPU0:ios (config-call-home)#service active
RP/0/RP0/CPU0:ios (config-call-home)#contact smart-licensing
RP/0/RP0/CPU0:ios (config-call-home)#profile CiscoTAC-1
RP/0/RP0/CPU0:ios (config-call-home-profile)#active
RP/0/RP0/CPU0:ios (config-call-home-profile)#destination address http
https://tools.cisco.com/its/service/oddce/services/DDCEService
RP/0/RP0/CPU0:ios (config-call-home-profile)#reporting smart-call-home-data
RP/0/RP0/CPU0:ios (config-call-home-profile)#reporting smart-licensing-data
RP/0/RP0/CPU0:ios (config-call-home-profile)#destination transport-method email disable
RP/0/RP0/CPU0:ios (config-call-home-profile)#destination transport-method http
RP/0/RP0/CPU0:ios (config-call-home-profile)#commit
RP/0/RP0/CPU0:ios (config-call-home-profile)#end
```

**Step 2** Use this sample configuration to enable a domine name server.

**Example:**

```
RP/0/RP0/CPU0:ios#config
RP/0/RP0/CPU0:ios (config)#domain name cisco.com
RP/0/RP0/CPU0:ios (config)#domain name-server 64.102.6.247
RP/0/RP0/CPU0:ios (config)#commit
RP/0/RP0/CPU0:ios (config)#end
```

**Step 3** Use this sample configuration to enable CRL Configuration.

**Example:**

```
RP/0/RP0/CPU0:ios#config
RP/0/RP0/CPU0:ios (config)#crypto ca trustpoint Trustpool crl optional
```

```
RP/0/RP0/CPU0:ios(config)#commit
RP/0/RP0/CPU0:ios(config)#end
```

**Step 4** Use this sample configuration to enable Call Home as transport mode.

**Example:**

```
RP/0/RP0/CPU0:ios#config
RP/0/RP0/CPU0:ios(config)#License smart transport callhome
RP/0/RP0/CPU0:ios(config)#commit
RP/0/RP0/CPU0:ios(config)#end
```

Verify whether the Callhome is Configured.

```
RP/0/RP0/CPU0:ios#show license all
Transport: Type: Callhome
```

**Step 5** Use this sample configuration to establish trust using id-token.

**Example:**

```
license smart trust idtoken Zesdf3243u48329fdhfsfhskjs1233j4h1j1j4j41n
```

**Step 6** Use this sample configuration to sync the token with the licenses.

**Example:**

```
license smart sync all
```

## Configure Smart

You can use the smart transport as an alternative option to Call Home, to connect to the CSSM. To configure smart transport in Cisco NCS 1020, perform the following steps:

**Step 1** Use this sample configure "Smart" proxy and "hostname"

**Example:**

```
RP/0/RP0/CPU0:ios#config
RP/0/RP0/CPU0:ios(config)#license smart proxy port 80
RP/0/RP0/CPU0:ios(config)#license smart proxy hostname proxy.esl.cisco.com
RP/0/RP0/CPU0:ios(config)#commit
RP/0/RP0/CPU0:ios(config)#end
```

**Step 2** Use this sample configuration to enable CRL Configuration.

**Example:**

```
RP/0/RP0/CPU0:ios#config
RP/0/RP0/CPU0:ios(config)#crypto ca trustpoint Trustpool crl optional
RP/0/RP0/CPU0:ios(config)#commit
RP/0/RP0/CPU0:ios(config)#end
```

**Step 3** Use this sample configuration to enable Call Home.

**Example:**

```
RP/0/RP0/CPU0:ios#config
RP/0/RP0/CPU0:ios(config)#License smart transport smart
```

```
RP/0/RP0/CPU0:ios(config)#commit
RP/0/RP0/CPU0:ios(config)#end
```

Verify whether Smart Transport is Configured.

```
RP/0/RP0/CPU0:ios#show license all
Transport:
Type: Smart
URL: https://smartreceiver.cisco.com/licservice/license
Proxy:
Address: proxy.esl.cisco.com
Port: 80
Username: <empty>
Password: <empty>
VRF:
Not Supported
```

**Step 4** Use this sample configuration to establish trust using id-token.

**Example:**

```
license smart trust idtoken Zesdf3243u48329fdhfsfhskjs1233j4h1j1j4j41n
```

**Step 5** Use this sample configuration to sync the token with the licenses.

**Example:**

```
license smart sync all
```

## Configure CSLU

You can configure CSLU as one of the transport modes, CSLU is the default mode for software licensing policy. To configure CSLU in Cisco NCS 1020, perform the following steps:

**Step 1** Use this sample configuration to configure the CSLU URL.

**Example:**

```
RP/0/RP0/CPU0:ios#config
RP/0/RP0/CPU0:ios(config)#license smart url cslu http://10.127.60.58:8182/cslu/v1/pi
RP/0/RP0/CPU0:ios(config)#commit
RP/0/RP0/CPU0:ios(config)#end
```

**Step 2** Use this sample configuration to enable CRL Configuration.

**Example:**

```
RP/0/RP0/CPU0:ios#config
RP/0/RP0/CPU0:ios(config)#crypto ca trustpoint Trustpool crl optional
RP/0/RP0/CPU0:ios(config)#commit
RP/0/RP0/CPU0:ios(config)#end
```

**Step 3** Use this sample configuration to enable CSLU.

**Example:**

```
RP/0/RP0/CPU0:ios#config
RP/0/RP0/CPU0:ios(config)#License smart transport cslu
```

```
RP/0/RP0/CPU0:ios(config)#commit
RP/0/RP0/CPU0:ios(config)#end
```

Verify whether CSLU is Configured.

```
RP/0/RP0/CPU0:ios#show license all
Transport:
 Type: cslu
 Cslu address: http://10.127.60.58:8182/cslu/v1/pi
 Proxy:
 Not Configured
 VRF:
 Not Supported
```

**Step 4** Use this sample configuration to establish trust using id-token.

**Example:**

```
license smart trust idtoken Zesdf3243u48329fdfhsfhsfkjs1233j4hlj1j4j41n
```

**Step 5** Use this sample configuration to sync the token with the licenses.

**Example:**

```
license smart sync all
```

## Configure Offline

You can configure Offline as one of the options. To configure Offline in Cisco NCS 1020, perform the following steps:

**Step 1** Use this sample configuration to disable transport.

**Example:**

```
RP/0/RP0/CPU0:ios#config
RP/0/RP0/CPU0:ios(config)#License smart transport off
RP/0/RP0/CPU0:ios(config)#commit
RP/0/RP0/CPU0:ios(config)#end
```

**Step 2** Use this sample configuration to save the report.

**Example:**

```
RP/0/RP0/CPU0:ios#license smart save usage unreported /misc1/disk1/usage.txt
```

**Step 3** Use this sample configuration to import the acknowledgment report.

**Example:**

```
RP/0/RP0/CPU0:ios#license smart import /misc/disk1/ACK_usage.txt
```

# Reserve Specific Licenses for NCS 1020

Specific License Reservation (SLR) lets you reserve a license for your product instance from the CSSM. To reserve specific licenses for NCS 1020, perform the following steps:

**Step 1** Generate the request code using the `license smart mfg reservation request local` command.

**Example:**

```
RP/0/RP0/CPU0:iso#license smart mfg reservation request local
Thu Jul 19 13:33:47.241 UTC

Enter this request code in the Cisco Smart Software Manager portal:
DB-ZNCS1020:FCB2717B142-BBTQDthRu-1A
```

**Step 2** Use the generated code and generate the authorization code through Cisco Smart Software Manager.

**Step 3** Enter the `run` command to launch the iso XR Linux bash shell.

**Example:**

```
RP/0/RP0/CPU0:iso#run

RP/0/RP0/CPU0:Jul 19 13:35:20.236: run_cmd[67213]: %INFRA-INFRA_MSG.5-RUN_LOGIN : User Cisco logged
into shell from con0/RP0/CP0
```

**Step 4** Create a file using the `vim file name` command.

**Example:**

```
[node0_RP0_CPU0:~]$vim smart1
```

**Step 5** Copy the authorization code in the file and type `:wq` to save and exit the file.

**Step 6** Use the `exit` command to exit the shell.

**Example:**

```
[node0_RP0_CPU0:~]$exit
logout
RP/0/RP0/CPU0:Jul 19 13:45:21.146 UTC run-cmd[67213] %INFRA_MSG-5-LOGOUT : User cisco logged out of
shell from con0/RP0/CPU0
```

**Step 7** Install the authorization code using the `license smart mfg reservation install file` command.

**Example:**

```
RP/0/RP0/CPU0:iso#license smart mfg reservation install file /disk0:/smart1
Thu Jul 19 13:46:22.877 UTC
Last Confirmation code 8572aa81
```

**Note** You can verify the number of reservations in the Cisco smart software manger portal and can view the product instance name changed to a UDI.

**Step 8** Verify the udi using the `show license udi` command.

**Example:**

```
RP/0/RP0/CPU0:iso#show license udi
Thu Jul 19 13:43:19.731 UTC
UDI: PID:NCS1020-SA,SN:FCB2546B08T
```

**Step 9** Verify the license reservation using the command **show license status**.

**Example:**

```
RP/0/RP0/CPU0:P2A_DT_08#show license status
Thu Jul 19 15:45:27.137 UTC

Smart Licensing is ENABLED

Utility:
 Status: DISABLED
License Reservation is ENABLED

Data Privacy:
 Sending Hostname: yes
 Callhome hostname privacy: DISABLED
 Smart Licensing hostname privacy: DISABLED
 Version privacy: DISABLED

Transport:
 Type: Transport Off

License Authorization:
 Status: AUTHORIZED - RESERVED on Jul 19 2022 15:21:24 UTC

Export Authorization Key:
 Features Authorized:
 <none>

Miscellaneous:
 Custom Id: <empty>
```

---