



## Preconfiguring Physical Interfaces

This module describes the preconfiguration of physical interfaces.

Preconfiguration is supported for these types of interfaces and controllers:

- 100-Gigabit Ethernet
- Management Ethernet

Preconfiguration allows you to configure line cards before they are inserted into the router. When the cards are inserted, they are instantly configured. The preconfiguration information is created in a different system database tree, rather than with the regularly configured interfaces. That database tree is known as the *preconfiguration directory* on the route processor.

There may be some preconfiguration data that cannot be verified unless the line card is present, because the verifiers themselves run only on the line card. Such preconfiguration data is verified when the line card is inserted and the verifiers are initiated. A configuration is rejected if errors are found when the configuration is copied from the preconfiguration area to the active area.



---

**Note** One Gigabit Ethernet interface is not supported. Only physical interfaces can be preconfigured.

---



---

**Note** Eight quadrature amplitude modulation (8QAM) requires V2 (or higher) CFP2 version and 5.23 (or higher) firmware.

---



---

**Note** From Cisco IOS XR Release 6.3.2, a six-seconds delay is introduced in error propagation from the driver to DPA for the MACSec line card and Oldcastle platforms. As a result, the BER algorithm on these platforms knows the error with a delay of 6 seconds.

---

- [Physical Interface Preconfiguration Overview, on page 2](#)
- [Prerequisites for Preconfiguring Physical Interfaces, on page 2](#)
- [Benefits of Interface Preconfiguration, on page 2](#)
- [How to Preconfigure Physical Interfaces, on page 3](#)
- [Information About Preconfiguring Physical Interfaces, on page 4](#)

# Physical Interface Preconfiguration Overview

Preconfiguration is the process of configuring interfaces before they are present in the system. Preconfigured interfaces are not verified or applied until the actual interface with the matching location (rack/slot/module) is inserted into the router. When the anticipated line card is inserted and the interfaces are created, the precreated configuration information is verified and, if successful, immediately applied to the running configuration of the router.



---

**Note** When you plug the anticipated line card in, make sure to verify any preconfiguration with the appropriate **show** commands.

---

Use the **show run** command to see interfaces that are in the preconfigured state.



---

**Note** We recommend filling out preconfiguration information in your site planning guide, so that you can compare that anticipated configuration with the actual preconfigured interfaces when that line card is installed and the interfaces are up.

---



---

**Tip** Tip Use the **commit best-effort** command to save the preconfiguration to the running configuration file. The **commit best-effort** command merges the target configuration with the running configuration and commits only valid configuration (best effort). Some configuration might fail due to semantic errors, but the valid configuration still comes up.

---

## Prerequisites for Preconfiguring Physical Interfaces

Before preconfiguring physical interfaces, ensure that this condition is met:

- Preconfiguration drivers and files are installed. Although it may be possible to preconfigure physical interfaces without a preconfiguration driver installed, the preconfiguration files are required to set the interface definition file on the router that supplies the strings for valid interface names.

## Benefits of Interface Preconfiguration

Preconfigurations reduce downtime when you add new cards to the system. With preconfiguration, the new cards can be instantly configured and actively running during cards bootup.

Another advantage of performing a preconfiguration is that during a cards replacement, when the cards is removed, you can still see the previous configuration and make modifications.

# How to Preconfigure Physical Interfaces

This task describes only the most basic preconfiguration of an interface.

## SUMMARY STEPS

1. **configure**
2. **interface preconfigure** *type interface-path-id*
3. Use one of the following commands:
  - **ipv4 address** *ip-address subnet-mask*
  - **ipv4 address** *ip-address /prefix*
4. Configure additional interface parameters, as described in this manual in the configuration chapter that applies to the type of interface that you are configuring.
5. **end** or **commit** best-effort
6. **show running-config**

## DETAILED STEPS

---

### Step 1 **configure**

#### Example:

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

Enters global configuration mode.

### Step 2 **interface preconfigure** *type interface-path-id*

#### Example:

```
RP/0/RP0/CPU0:router(config)# interface preconfigure HundredGigE 0/3/0/2
```

Enters interface preconfiguration mode for an interface, where *type* specifies the supported interface type that you want to configure and *interface-path-id* specifies the location where the interface will be located in *rack/slot/module/port* notation.

### Step 3 Use one of the following commands:

- **ipv4 address** *ip-address subnet-mask*
- **ipv4 address** *ip-address /prefix*

#### Example:

```
RP/0/RP0/CPU0:router(config-if-pre)# ipv4 address 192.168.1.2/31
```

Assigns an IP address and mask to the interface.

### Step 4 Configure additional interface parameters, as described in this manual in the configuration chapter that applies to the type of interface that you are configuring.

### Step 5 **end** or **commit** best-effort

**Example:**

```
RP/0/RP0/CPU0:router(config-if-pre)# end
```

or

```
RP/0/RP0/CPU0:router(config-if-pre)# commit
```

Saves configuration changes.

- When you issue the **end** command, the system prompts you to commit changes: `Uncommitted changes found, commit them before exiting (yes/no/cancel)?`
- Entering **yes** saves configuration changes to the running configuration file, exits the configuration session, and returns the router to EXEC mode.
- Entering **no** exits the configuration session and returns the router to EXEC mode without committing the configuration changes.
- Entering **cancel** leaves the router in the current configuration session without exiting or committing the configuration changes.
- Use the **commit best-effort** command to save the configuration changes to the running configuration file and remain within the configuration session. The **commit best-effort** command merges the target configuration with the running configuration and commits only valid changes (best effort). Some configuration changes might fail due to semantic errors.

**Step 6** **show running-config****Example:**

```
RP/0/RP0/CPU0:router# show running-config
```

(Optional) Displays the configuration information currently running on the router.

**Example**

This example shows how to preconfigure a basic Ethernet interface:

```
RP/0/RP0/CPU0:router# configure
RP/0/RP0/CPU0:router(config)# interface preconfigure HundredGigE 0/3/0/24
RP/0/RP0/CPU0:router(config-if)# ipv4 address 192.168.1.2/31
RP/0/RP0/CPU0:router(config-if-pre)# commit
```

## Information About Preconfiguring Physical Interfaces

From Cisco IOS XR Release 7.0.2, the NC57-18DD-SE follows the following port mapping:

- Port number 0-17 (nine pairs) and 24-29 (three pairs): They together drive 400G mode. This means that if the top port is in 400G mode, the bottom port is unusable. These ports are retimer ports.

- Port number 18-23 (six ports): They are direct connected ports and are individually capable of 400G mode.



**Note** There's a limitation for ports 0, 1 and 14, 15. You have to insert modules of similar speed (40G or 100G) into these pairs of ports. For example, if you insert 40G module in port 0, then 40G module must be inserted in port 1.



**Note** For 400G-only mode, the ports to be used are 0, 2, 4, 6, 8, 10, 12, 14, 16, 18, 19, 20, 21, 22, 23, 24, 26, and 28.

For detailed information on port mapping and usage, see the figure *NC57-18DD-SE Line Card* in chapter *NCS 5500 Series Modular Router Overview of Hardware Installation Guide for Cisco NCS 5500 Series Modular Routers* guide.

To control the interfaces which are created, use the `hw-module port-range mode` command with the following modes:

- 40-100: This is the default port mode. Two ports are created in 100G mode by default. Online Insertion and Removal (OIR) to 40G creates the 40G port. Assume both ports to be similar to J/J+ ports.
- 400: The first port created is 400G. No port is created for the bottom port.
- 2x100: For 2x100 mode. This supports QDD-2X100-LR4 optics.

Port range can be in the form of  $n$  to  $n+1$ . Example: 0,1 or 6,7. The port range is valid for ports 0-17 and 24-29. To configure a port with 400G rate:

```
RP/0/RP0/CPU0:router(config)#hw-module port-range 0 1 location 0/3/CPU0 mode 400
RP/0/RP0/CPU0:router(config)#commit
Wed Feb  6 03:23:12.923 UTC
LC/0/3/CPU0:Feb  6 03:23:13.548 UTC: ifmgr[281]: %PKT_INFRA-LINK-3-UPDOWN : Interface
HundredGigE0/3/0/1, changed state to Down
LC/0/3/CPU0:Feb  6 03:23:13.548 UTC: ifmgr[281]: %PKT_INFRA-LINK-3-UPDOWN : Interface
HundredGigE0/3/0/0, changed state to Down
RP/0/RP0/CPU0:router(config)#end
RP/0/RP0/CPU0:router#show ipv4 int br location 0/3/CPU0
Wed Feb  6 03:26:07.935 UTC
```

Interface	IP-Address	Status	Protocol	Vrf-Name
FourHundredGigE0/3/0/0	unassigned	Shutdown	Down	default
HundredGigE0/3/0/2	unassigned	Shutdown	Down	default
HundredGigE0/3/0/3	unassigned	Shutdown	Down	default
HundredGigE0/3/0/4	unassigned	Shutdown	Down	default
HundredGigE0/3/0/5	unassigned	Shutdown	Down	default
HundredGigE0/3/0/6	unassigned	Shutdown	Down	default

To change a port mode:

```
RP/0/RP0/CPU0:router#conf
Thu Jan  9 05:13:02.853 UTC
RP/0/RP0/CPU0:router(config)#hw-module port-range 2 3 location 0/3/CPU0 mode 2x100
RP/0/RP0/CPU0:router(config)#commit
Thu Jan  9 05:13:11.411 UTC
LC/0/3/CPU0:Jan  9 05:13:11.469 UTC: optics_driver[196]: %PKT_INFRA-FM-3-FAULT_MAJOR :
ALARM_MAJOR :PORTMODE SPEED MISMATCH :CLEAR :0/3/CPU0: Optics0/3/0/3
```

```
LC/0/3/CPU0:Jan  9 05:13:13.141 UTC: ifmgr[228]: %PKT_INFRA-LINK-3-UPDOWN : Interface
HundredGigE0/3/0/3, changed state to Down
LC/0/3/CPU0:Jan  9 05:13:13.141 UTC: ifmgr[228]: %PKT_INFRA-LINK-3-UPDOWN : Interface
HundredGigE0/3/0/2, changed state to Down
RP/0/RP0/CPU0:router(config)#end
RP/0/RP0/CPU0:router#show ipv4 int br location 0/3/CPU0
Thu Jan  9 05:13:24.245 UTC
```

Interface	IP-Address	Status	Protocol	Vrf-Name
FortyGigE0/3/0/28	unassigned	Shutdown	Down	default
HundredGigE0/3/0/29	unassigned	Shutdown	Down	default
HundredGigE0/3/0/2/0	unassigned	Down	Down	default
HundredGigE0/3/0/2/1	unassigned	Down	Down	default
HundredGigE0/3/0/3/0	unassigned	Down	Down	default
HundredGigE0/3/0/3/1	unassigned	Down	Down	default

Use the following commands for the newly configured image:

```
hw-module port-range 0 1 location 0/6/CPU0 mode 400
hw-module port-range 2 3 location 0/6/CPU0 mode 400
hw-module port-range 4 5 location 0/6/CPU0 mode 400
hw-module port-range 6 7 location 0/6/CPU0 mode 400
hw-module port-range 8 9 location 0/6/CPU0 mode 400
hw-module port-range 10 11 location 0/6/CPU0 mode 400
hw-module port-range 12 13 location 0/6/CPU0 mode 400
hw-module port-range 14 15 location 0/6/CPU0 mode 400
hw-module port-range 16 17 location 0/6/CPU0 mode 400
hw-module port-range 24 25 location 0/6/CPU0 mode 400
hw-module port-range 26 27 location 0/6/CPU0 mode 400
hw-module port-range 28 29 location 0/6/CPU0 mode 400
hw-module port-range 0 1 location 0/6/CPU0 mode 2x100
hw-module port-range 2 3 location 0/6/CPU0 mode 2x100
hw-module port-range 4 5 location 0/6/CPU0 mode 2x100
hw-module port-range 6 7 location 0/6/CPU0 mode 2x100
hw-module port-range 8 9 location 0/6/CPU0 mode 2x100
hw-module port-range 10 11 location 0/6/CPU0 mode 2x100
hw-module port-range 12 13 location 0/6/CPU0 mode 2x100
hw-module port-range 14 15 location 0/6/CPU0 mode 2x100
hw-module port-range 16 17 location 0/6/CPU0 mode 2x100
hw-module port-range 24 25 location 0/6/CPU0 mode 2x100
hw-module port-range 26 27 location 0/6/CPU0 mode 2x100
hw-module port-range 28 29 location 0/6/CPU0 mode 2x100
```

To preconfigure interfaces, you must understand these concepts:

## Use of the Interface Preconfigure Command

Interfaces that are not yet present in the system can be preconfigured with the **interface preconfigure** command in global configuration mode.

The **interface preconfigure** command places the router in interface configuration mode. Users should be able to add any possible interface commands. The verifiers registered for the preconfigured interfaces verify the configuration. The preconfiguration is complete when the user enters the **end** command, or any matching exit or global configuration mode command.



**Note** It is possible that some configurations cannot be verified until the line card is inserted.

Do not enter the **no shutdown** command for new preconfigured interfaces, because the no form of this command removes the existing configuration, and there is no existing configuration.

Users are expected to provide names during preconfiguration that will match the name of the interface that will be created. If the interface names do not match, the preconfiguration cannot be applied when the interface is created. The interface names must begin with the interface type that is supported by the router and for which drivers have been installed. However, the slot, port, subinterface number, and channel interface number information cannot be validated.



---

**Note** Specifying an interface name that already exists and is configured (or an abbreviated name like Hu0/3/0/0) is not permitted.

---

