

# Preconfiguring Physical Interfaces

This module describes the preconfiguration of physical interfaces.

Preconfiguration is supported for these types of interfaces and controllers:

- 1-Gigabit Ethernet
- 10-Gigabit Ethernet
- 25-Gigabit Ethernet
- 40-Gigabit Ethernet
- 100-Gigabit Ethernet
- Management Ethernet

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

There may be some preconfiguration data that cannot be verified unless the interface modules are present, because the verifiers themselves run only on the interface modules. Such preconfiguration data is verified when the interface modules 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.


- Physical Interface Preconfiguration Overview, on page 1
- Prerequisites for Preconfiguring Physical Interfaces, on page 2
- Benefits of Interface Preconfiguration, on page 2
- How to Preconfigure Physical Interfaces, on page 2
- Use of the Interface Preconfigure Command, 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 interface module 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 interface module 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 interface module 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 interface modulesto the system. With preconfiguration, the new interface modules can be instantly configured and actively running during interface modules bootup.

Another advantage of performing a preconfiguration is that during a interface modules replacement, when the interface modules 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.

**Procedure**

**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 0/0/1/0
```

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 0/0/1/0
RP/0/RP0/CPU0:router(config-if)# ipv4 address 192.168.1.2/31
RP/0/RP0/CPU0:router(config-if-pre)# commit
```

# 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 interface module 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.