

# Configuring Link Bundles

A bundle is a group of one or more ports that are aggregated together and treated as a single link. The different links within a single bundle can have varying speeds, where the fastest link can be a maximum of four times greater than the slowest link. Each bundle has a single MAC, a single IP address, and a single configuration set (such as ACLs or QoS).

The router supports bundling for these types of interfaces:

- Ethernet interfaces
- VLAN subinterfaces

**Note** Bundles do not have a one-to-one modular services card association.


- Feature History for Configuring Link Bundles, on page 1
- Prerequisites for Configuring Link Bundles, on page 1
- Information About Configuring Link Bundles, on page 2
- How to Configure Link Bundling, on page 7
- Configuration Examples for Link Bundles, on page 16


# Feature History for Configuring Link Bundles

| Release | Modification |
|---|---|
| Release 3.7.2 | This feature was introduced on the Cisco ASR 9000 Series Routers. |

# Prerequisites for Configuring Link Bundles

Before configuring Link Bundling, be sure that these tasks and conditions are met:

- You must be in a user group associated with a task group that includes the proper task IDs. The command reference guides include the task IDs required for each command.

If you suspect user group assignment is preventing you from using a command, contact your AAA administrator for assistance.

- You know the interface IP address.
- You know which links should be included in the bundle you are configuring.
- If you are configuring an Ethernet link bundle, you have at least one of these Ethernet line cards installed in the router:
  - 2-port 10-Gigabit Ethernet line card
  - 4-port 10-Gigabit Ethernet line card
  - 8-port 10-Gigabit Ethernet line card
  - 16-port 10-Gigabit Ethernet line card
  - 20-port Gigabit Ethernet line card
  - 40-port Gigabit Ethernet line card



**Note** For more information about physical interfaces, PLIMs, and modular services cards, refer to the *Cisco ASR 9000 Series Aggregation Services Router Hardware Installation Guide* .

# Information About Configuring Link Bundles

To implement the Link Bundling feature, you must understand these concepts:

## Link Bundling Overview

A link bundle is simply a group of ports that are bundled together and act as a single link. The advantages of link bundles are these:

- Multiple links can span several line cards to form a single interface. Thus, the failure of a single link does not cause a loss of connectivity.
- Bundled interfaces increase bandwidth availability, because traffic is forwarded over all available members of the bundle. Therefore, traffic can flow on the available links if one of the links within a bundle fails. Bandwidth can be added without interrupting packet flow.

Although the individual links within a single bundle can have varying speeds, all links within a bundle must be of the same type.

Cisco IOS XR software supports these methods of forming bundles of Ethernet interfaces:

- IEEE 802.3ad—Standard technology that employs a Link Aggregation Control Protocol (LACP) to ensure that all the member links in a bundle are compatible. Links that are incompatible or have failed are automatically removed from a bundle.

- EtherChannel—Cisco proprietary technology that allows the user to configure links to join a bundle, but has no mechanisms to check whether the links in a bundle are compatible.

# Characteristics of Link Bundles

This list describes the properties and limitations of link bundles:

- Any type of Ethernet interfaces can be bundled, with or without the use of LACP (Link Aggregation Control Protocol).
- Bundle membership can span across several line cards that are installed in a single router.
- A single bundle supports maximum of 64 physical links.
- Different link speeds are allowed within a single bundle, with a maximum of four times the speed difference between the members of the bundle.
- Physical layer and link layer configuration are performed on individual member links of a bundle.
- Configuration of network layer protocols and higher layer applications is performed on the bundle itself.
- A bundle can be administratively enabled or disabled.
- Each individual link within a bundle can be administratively enabled or disabled.
- Ethernet link bundles are created in the same way as Ethernet channels, where the user enters the same configuration on both end systems.
- The MAC address that is set on the bundle becomes the MAC address of the links within that bundle.
- When LACP configured, each link within a bundle can be configured to allow different keepalive periods on different members.
- Load balancing (the distribution of data between member links) is done by flow instead of by packet. Data is distributed to a link in proportion to the bandwidth of the link in relation to its bundle.
- QoS is supported and is applied proportionally on each bundle member.
- Link layer protocols, such as CDP and HDLC keepalives, work independently on each link within a bundle.
- Upper layer protocols, such as routing updates and hellos, are sent over any member link of an interface bundle.
- Bundled interfaces are point to point.
- A link must be in the up state before it can be in distributing state in a bundle.
- All links within a single bundle must be configured either to run 802.3ad (LACP) or Etherchannel (non-LACP). Mixed links within a single bundle are not supported.
- A bundle interface can contain physical links and VLAN subinterfaces only.
- Access Control List (ACL) configuration on link bundles is identical to ACL configuration on regular interfaces.
- Multicast traffic is load balanced over the members of a bundle. For a given flow, internal processes select the member link and all traffic for that flow is sent over that member.

# IEEE 802.3ad Standard

The IEEE 802.3ad standard typically defines a method of forming Ethernet link bundles.

For each link configured as bundle member, this information is exchanged between the systems that host each end of the link bundle:

- A globally unique local system identifier
- An identifier (operational key) for the bundle of which the link is a member
- An identifier (port ID) for the link
- The current aggregation status of the link

This information is used to form the link aggregation group identifier (LAG ID). Links that share a common LAG ID can be aggregated. Individual links have unique LAG IDs.

The system identifier distinguishes one router from another, and its uniqueness is guaranteed through the use of a MAC address from the system. The bundle and link identifiers have significance only to the router assigning them, which must guarantee that no two links have the same identifier, and that no two bundles have the same identifier.

The information from the peer system is combined with the information from the local system to determine the compatibility of the links configured to be members of a bundle.

Bundle MAC addresses in the routers come from a set of reserved MAC addresses in the backplane. This MAC address stays with the bundle as long as the bundle interface exists. The bundle uses this MAC address until the user configures a different MAC address. The bundle MAC address is used by all member links when passing bundle traffic. Any unicast or multicast addresses set on the bundle are also set on all the member links.



**Note** We recommend that you avoid modifying the MAC address, because changes in the MAC address can affect packet forwarding.

# Non Revertive Behavior for LACP Bundle Interface

In LACP, by default, a higher priority port would become the active port after it becomes operational again. To avoid this reversion, you can run the **lacp non-revertive** command. This configures the lower priority port to continue as the active port even after the higher priority port is capable of being operational. This avoids the traffic disruption that may happen in putting the currently active but lower priority port into standby and diverting traffic through the higher priority port that is now capable of being operational.

# QoS and Link Bundling

On the ingress direction, QoS is applied to the local instance of a bundle. Each bundle is associated with a set of queues. QoS is applied to the various network layer protocols that are configured on the bundle.

On the egress direction, QoS is applied on the bundle with a reference to the member links. QoS is applied based on the sum of the member bandwidths.

When QoS is applied on the bundle for either the ingress or egress direction, QoS is applied at each member interface.

The Link Bundling feature supports all the QoS features described in the *Cisco ASR 9000 Series Aggregation Services Router Modular Quality of Service Configuration Guide*.

The Link Bundling feature supports these QoS features:

- hi priority /lo priority—Maximum bandwidth is calculated as a percentage of the bundle interface bandwidth. This percentage is then applied to every member link on the egress, or to the local bundle instance on ingress.
- guaranteed bandwidth—Provided in percentage and applied to every member link.
- traffic shaping—Provided in percentage and applied to every member link.
- WRED—Minimum and maximum parameters are converted to the right proportion per member link or bundle instance, and then are applied to the bundle.
- marking—Process of changing the packet QoS level according to a policy.
- tail drop— Packets are dropped when the queue is full.

# VLANs on an Ethernet Link Bundle

802.1Q VLAN subinterfaces can be configured on 802.3ad Ethernet link bundles. Keep this information in mind when adding VLANs on an Ethernet link bundle:

- The maximum number of VLANs allowed per bundle is 4000.
- The maximum number of bundled VLANs allowed per router is 128000.

**Note** The memory requirement for bundle VLANs is slightly higher than standard physical interfaces.

To create a VLAN subinterface on a bundle, include the VLAN subinterface instance with the **interface Bundle-Ether** command:

**interface Bundle-Ether** `instance.subinterface`

After you create a VLAN on an Ethernet link bundle, all physical VLAN subinterface configuration is supported on that link bundle.

# Link Bundle Configuration Overview

These steps provide a general overview of the link bundle configuration process. Keep in mind that a link must be cleared of all previous network layer configuration before it can be added to a bundle:

1. In global configuration mode, create a link bundle. To create an Ethernet link bundle, enter the **interface Bundle-Ether** command.
2. Assign an IP address and subnet mask to the virtual interface using the **ipv4 address** command.

3. Add interfaces to the bundle you created in Step 1 with the **bundle id** command in the interface configuration submode. You can add up to 32 links to a single bundle.

**Note** A link is configured to be a member of a bundle from the interface configuration submode for that link.

# Nonstop Forwarding During Card Failover

The Cisco IOS XR software supports nonstop forwarding during failover between active and standby paired RSP cards. Nonstop forwarding ensures that there is no change in the state of the link bundles when a failover occurs.

For example, if an active RSP fails, the standby RSP becomes operational. The configuration, node state, and checkpoint data of the failed RSP are replicated to the standby RSP. The bundled interfaces will all be present when the standby RSP becomes the active RSP.

**Note** Failover is always onto the standby RSP.

**Note** You do not need to configure anything to guarantee that the standby interface configurations are maintained.

# Link Failover

When one member link in a bundle fails, traffic is redirected to the remaining operational member links and traffic flow remains uninterrupted.

# Bundle Interfaces: Redundancy, Load Sharing, Aggregation

A bundle is a group of one or more ports that are aggregated together and treated as a single link. The different links within a single bundle can have varying speeds, where the fastest link can be a maximum of four times greater than the slowest link. Each bundle has a single MAC, a single IP address, and a single configuration set (such as ACLs or QoS).

The router supports bundling for these types of interfaces:

- Ethernet interfaces
- VLAN subinterfaces

# How to Configure Link Bundling

## Configuring Ethernet Link Bundles

This section describes how to configure a Ethernet link bundle.



**Note** MAC accounting is not supported on Ethernet link bundles.



**Note** In order for an Ethernet bundle to be active, you must perform the same configuration on both connection endpoints of the bundle.

The creation of an Ethernet link bundle involves creating a bundle and adding member interfaces to that bundle, as shown in the steps that follow.

### SUMMARY STEPS

1. **configure**
2. **interface Bundle-Ether** *bundle-id*
3. **ipv4 address** *ipv4-address-address mask*
4. **bundle minimum-active bandwidth** *kbps* (optional)
5. **bundle minimum-active links** *links* (optional)
6. **bundle maximum-active links** *links* (optional)
7. **bundle maximum-active links** *links* **hot-standby**
8. **exit**
9. **interface {GigabitEthernet | TenGigE}** *instance*
10. **bundle id** *bundle-id* [ **mode** { **active** | **on** | **passive**} ]
11. **no shutdown**(optional)
12. **exit**
13. Repeat Step 8 through Step 11 to add more links to the bundle you created in Step 2.
14. Use the **commit** or **end** command.
15. **exit**
16. **exit**
17. Perform Step 1 through Step 15 on the remote end of the connection.
18. **show bundle Bundle-Ether** *bundle-id* [ **reasons** ] (optional)
19. **show lacp Bundle-Ether** *bundle-id* (optional)

## DETAILED STEPS

### Procedure

**Step 1** **configure**

**Example:**

```
RP/0/RSP0/CPU0:router# configure
```

Enters the Global Configuration mode.

**Step 2** **interface Bundle-Ether** *bundle-id*

**Example:**

```
RP/0/RSP0/CPU0:router(config)# interface Bundle-Ether 3
```

Creates and names a new Ethernet link bundle.

This **interface Bundle-Ether** command enters you into the interface configuration submode, where you can enter interface specific configuration commands are entered. Use the **exit** command to exit from the interface configuration submode back to the normal Global Configuration mode.

**Step 3** **ipv4 address** *ipv4-address-address mask*

**Example:**

```
RP/0/RSP0/CPU0:router(config-if)# ipv4 address 10.1.2.3 255.0.0.0
```

Assigns an IP address and subnet mask to the virtual interface using the **ipv4 address** configuration subcommand.

**Step 4** **bundle minimum-active bandwidth** *kbps* (optional)

**Example:**

```
RP/0/RSP0/CPU0:router(config-if)# bundle minimum-active bandwidth 580000
```

Sets the minimum amount of bandwidth required before a user can bring up a bundle.

**Step 5** **bundle minimum-active links** *links* (optional)

**Example:**

```
RP/0/RSP0/CPU0:router(config-if)# bundle minimum-active links 2
```

Sets the number of active links required before you can bring up a specific bundle.

**Step 6** **bundle maximum-active links** *links* (optional)

**Example:**

```
RP/0/RSP0/CPU0:router(config-if)# bundle maximum-active links 1
```

Designates one active link and one link in standby mode that can take over immediately for a bundle if the active link fails (1:1 protection).

The default number of active links allowed in a single bundle is 8.

**Note**
If the **bundle maximum-active** command is issued, then only the highest-priority link within the bundle is active. The priority is based on the value from the **bundle port-priority** command, where a lower value is a higher priority. Therefore, we recommend that you configure a higher priority on the link that you want to be the active link.

**Step 7** **bundle maximum-active links** *links* **hot-standby**

**Example:**

```
RP/0/RSP0/CPU0:router(config-if)# bundle maximum-active links 1 hot-standby
```

The **hot-standby** keyword helps to avoid bundle flaps on a switchover or switchback event during which the bundle temporarily falls below the minimum links or bandwidth threshold.

It sets default values for the wait-while timer and suppress-flaps timer to achieve this.

**Step 8** **exit**

**Example:**

```
RP/0/RSP0/CPU0:router(config-#if)exit
```

Exits interface configuration submode for the Ethernet link bundle.

**Step 9** **interface {GigabitEthernet | TenGigE}** *instance*

**Example:**

```
RP/0/RSP0/CPU0:router(config)# interface TenGigE 1/0/0/0
```

Enters the interface configuration mode for the specified interface.

Enter the **GigabitEthernet** or **TenGigE** keyword to specify the interface type. Replace the instance argument with the node-id in the *rack/slot/module* format.

Mixed bandwidth bundle member configuration is only supported when 1:1 redundancy is configured (this means that a 1 GigabitEthernet member can only be configured as the backup of the 10 GigabitEthernet interface.)

**Note**
Mixed link bundle mode is supported only when active-standby operation is configured (usually with the lower speed link in standby mode).

**Step 10** **bundle id** *bundle-id* [ **mode** { **active** | **on** | **passive**} ]

**Example:**

```
RP/0/RSP0/CPU0:router(config-if)# bundle-id 3
```

Adds the link to the specified bundle.

To enable active or passive LACP on the bundle, include the optional **mode active** or **mode passive** keywords in the command string.

To add the link to the bundle without LACP support, include the optional **mode on** keywords with the command string.

**Note**
If you do not specify the **mode** keyword, the default mode is **on** (LACP is not run over the port).

**Step 11** **no shutdown**(optional)

**Example:**

```
RP/0/RSP0/CPU0:router(config-if)# no shutdown
```

If a link is in the down state, bring it up. The **no shutdown** command returns the link to an up or down state depending on the configuration and state of the link.

**Step 12** **exit**

**Example:**

```
RP/0/RSP0/CPU0:router(config-if)# exit
```

Exits interface configuration submode for the Ethernet link bundle.

**Step 13** Repeat Step 8 through Step 11 to add more links to the bundle you created in Step 2.

**Step 14** 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 mode, without committing the configuration changes.

**Step 15** **exit**

**Example:**

```
RP/0/RSP0/CPU0:router(config-if)# exit
```

Exits interface configuration mode.

**Step 16** **exit**

**Example:**

```
RP/0/RSP0/CPU0:router(config)# exit
```

Exits the Global Configuration mode.

**Step 17** Perform Step 1 through Step 15 on the remote end of the connection.

Brings up the other end of the link bundle.

**Step 18** **show bundle Bundle-Ether** *bundle-id* [ **reasons** ] (optional)

**Example:**

```
RP/0/RSP0/CPU0:router# show bundle Bundle-Ether 3 reasons
```

Shows information about the specified Ethernet link bundle

**Step 19** **show lacp Bundle-Ether** *bundle-id* (optional)

**Example:**

```
RP/0/RSP0/CPU0:router # show lacp Bundle-Ether 3
```

Shows detailed information about LACP ports and their peers.

# Configuring VLAN Bundles

This section describes how to configure a VLAN bundle. The creation of a VLAN bundle involves three main tasks:

1. Create an Ethernet bundle.
2. Create VLAN subinterfaces and assign them to the Ethernet bundle.
3. Assign Ethernet links to the Ethernet bundle.

These tasks are describe in detail in the procedure that follows.



**Note** In order for a VLAN bundle to be active, you must perform the same configuration on both ends of the bundle connection.

The creation of a VLAN link bundle is described in the steps that follow.

### SUMMARY STEPS

1. **configure**
2. **interface Bundle-Ether** *bundle-id*
3. **ipv4 address** *ipv4-address mask*
4. **bundle minimum-active bandwidth** *kbps* (optional)
5. **bundle minimum-active links** *links* (optional)
6. **bundle maximum-active links** *links* (optional)
7. **exit**
8. **interface Bundle-Ether** *bundle-id.vlan-id*
9. **encapsulation dot1q** *vlan-id*
10. **ipv4 address** *ip-address mask*
11. **no shutdown**
12. **exit**
13. Repeat Step 7 through Step 12 to add more VLANs to the bundle you created in Step 2.
14. Use the **commit** or **end** command.
15. **exit**
16. **exit**
17. **show ethernet trunk bundle-Ether** *instance*
18. **configure**
19. **interface {GigabitEthernet | TenGigE}** *instance*
20. **bundle id** `bundle-id` **[mode {active | on | passive}]**
21. **no shutdown**
22. Repeat Step 19 through Step 21 to add more Ethernet interfaces to the bundle you created in Step 2 .
23. Use the **commit** or **end** command.

24. Perform Step 1 through Step 23 on the remote end of the connection.
25. **show bundle Bundle-Ether** *bundle-id* [ **reasons** ]
26. **show ethernet trunk bundle-Ether** *instance*

## DETAILED STEPS

### Procedure

**Step 1** **configure**

**Example:**

```
RP/0/RSP0/CPU0:router# configure
```

Enters Global Configuration mode.

**Step 2** **interface Bundle-Ether** *bundle-id*

**Example:**

```
RP/0/RSP0/CPU0:router#(config)# interface Bundle-Ether 3
```

Creates and names a new Ethernet link bundle.

This **interface Bundle-Ether** command enters you into the interface configuration submode, where you can enter interface specific configuration commands are entered. Use the **exit** command to exit from the interface configuration submode back to the normal Global Configuration mode

**Step 3** **ipv4 address** *ipv4-address mask*

**Example:**

```
RP/0/RSP0/CPU0:router(config-if)# ipv4 address 10.1.2.3 255.0.0.0
```

Assigns an IP address and subnet mask to the virtual interface using the **ipv4 address** configuration subcommand.

**Step 4** **bundle minimum-active bandwidth** *kbps* (optional)

**Example:**

```
RP/0/RSP0/CPU0:router(config-if) # bundle minimum-active bandwidth 580000
```

(Optional) Sets the minimum amount of bandwidth required before a user can bring up a bundle.

**Step 5** **bundle minimum-active links** *links* (optional)

**Example:**

```
RP/0/RSP0/CPU0:router(config-if)# bundle minimum-active links 2
```

(Optional) Sets the number of active links required before you can bring up a specific bundle.

**Step 6** **bundle maximum-active links** *links* (optional)

**Example:**

```
RP/0/RSP0/CPU0:router(config-if)# bundle maximum-active links 1
```

(Optional) Designates one active link and one link in standby mode that can take over immediately for a bundle if the active link fails (1:1 protection).

**Note**
The default number of active links allowed in a single bundle is 8.

**Note**
If the **bundle maximum-active** command is issued, then only the highest-priority link within the bundle is active. The priority is based on the value from the **bundle port-priority** command, where a lower value is a higher priority. Therefore, we recommend that you configure a higher priority on the link that you want to be the active link.

**Step 7** **exit**

**Example:**

```
RP/0/RSP0/CPU0:router(config-if)# exit
```

Exits interface configuration submode.

**Step 8** **interface Bundle-Ether** *bundle-id.vlan-id*

**Example:**

```
RP/0/RSP0/CPU0:router#(config)#interface Bundle-Ether 3.1
```

Creates a new VLAN, and assigns the VLAN to the Ethernet bundle you created in Step 2.

Replace the *bundle-id* argument with the *bundle-id* you created in Step 2.

Replace the *vlan-id* with a subinterface identifier. Range is from 1 to 4094 inclusive (0 and 4095 are reserved).

**Note**
When you include the *vlan-id* argument with the **interface Bundle-Ether** *bundle-id* command, you enter subinterface configuration mode.

**Step 9** **encapsulation dot1q** *vlan-id*

**Example:**

```
RP/0/RSP0/CPU0:router#(config-subif)# encapsulation dot1q 10
```

Assigns a VLAN to the subinterface.

Replace the *vlan-id* argument with a subinterface identifier. Range is from 1 to 4094 inclusive (0 and 4095 are reserved).

**Step 10** **ipv4 address** *ip-address mask*

**Example:**

```
RP/0/RSP0/CPU0:router#(config-subif)# ipv4 address 10.1.2.3/24
```

Assigns an IP address and subnet mask to the subinterface.

**Step 11** **no shutdown**

**Example:**

```
RP/0/RSP0/CPU0:router(config-subif) # no shutdown
```

(Optional) If a link is in the down state, bring it up. The **no shutdown** command returns the link to an up or down state depending on the configuration and state of the link.

**Step 12** **exit**

**Example:**

```
RP/0/RSP0/CPU0:router(config-subif)#exit
```

Exits subinterface configuration mode for the VLAN subinterface.

**Step 13** Repeat Step 7 through Step 12 to add more VLANs to the bundle you created in Step 2.

(Optional) Adds more subinterfaces to the bundle.

**Step 14** 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 mode, without committing the configuration changes.

**Step 15** **exit**

**Example:**

```
RP/0/RSP0/CPU0:router (config-subif)# exit
```

Exits interface configuration mode.

**Step 16** **exit**

**Example:**

```
RP/0/RSP0/CPU0:router (config)# exit
```

Exits Global Configuration mode.

**Step 17** **show ethernet trunk bundle-Ether** *instance*

**Example:**

```
RP/0/RSP0/CPU0:routerrouter# show ethernet trunk bundle-ether 5
```

(Optional) Displays the interface configuration.

The Ethernet bundle instance range is from 1 through 65535.

**Step 18** **configure**

**Example:**

```
RP/0/RSP0/CPU0:router#  configure
```

Enters Global Configuration mode.

**Step 19** **interface {GigabitEthernet | TenGigE}** *instance*

**Example:**

```
RP/0/RSP0/CPU0:router(config)# interface TenGigE 1/0/0/0
```

Enters the interface configuration mode for the specified interface.

Replace the *instance* argument with the node-id in the *rack/slot/module* format.

**Note**
A VLAN bundle is not active until you add an Ethernet interface on both ends of the link bundle.

**Step 20** **bundle id `bundle-id` [mode {active | on | passive}]**

**Example:**

```
RP/0/RSP0/CPU0:router(config-if)#  bundle-id 3
```

Adds an Ethernet interface to the bundle you configured in Step 2 through Step 13.

To enable active or passive LACP on the bundle, include the optional **mode active** or **mode passive** keywords in the command string.

To add the interface to the bundle without LACP support, include the optional **mode on** keywords with the command string.

**Note**
If you do not specify the **mode** keyword, the default mode is **on** (LACP is not run over the port).

**Step 21** **no shutdown**

**Example:**

```
RP/0/RSP0/CPU0:router(config-if)# no shutdown
```

(Optional) If a link is in the down state, bring it up. The **no shutdown** command returns the link to an up or down state depending on the configuration and state of the link.

**Step 22** Repeat Step 19 through Step 21 to add more Ethernet interfaces to the bundle you created in Step 2 .

**Step 23** 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 mode, without committing the configuration changes.

**Step 24** Perform Step 1 through Step 23 on the remote end of the connection.

Brings up the other end of the link bundle.

**Step 25** **show bundle Bundle-Ether** *bundle-id* [ **reasons** ]

**Example:**

```
RP/0/RSP0/CPU0:router#show bundle Bundle-Ether 3 reasons
```

(Optional) Shows information about the specified Ethernet link bundle.

The **show bundle Bundle-Ether** command displays information about the specified bundle. If your bundle has been configured properly and is carrying traffic, the State field in the **show bundle Bundle-Ether** command output will show the number “4,” which means the specified VLAN bundle port is “distributing.”

**Step 26** **show ethernet trunk bundle-Ether** *instance*

**Example:**

```
RP/0/RSP0/CPU0:router# show ethernet trunk bundle-ether 5
```

(Optional) Displays the interface configuration.

The Ethernet bundle instance range is from 1 through 65535.

# Configuration Examples for Link Bundles

## EtherChannel Bundle running LACP: Example

This example shows how to join two ports to form an EtherChannel bundle running LACP:

```
RP/0/RSP0/CPU0:Router# config
RP/0/RSP0/CPU0:Router(config)# interface Bundle-Ether 3
RP/0/RSP0/CPU0:Router(config-if)# ipv4 address 1.2.3.4/24
RP/0/RSP0/CPU0:Router(config-if)# bundle minimum-active bandwidth 620000
RP/0/RSP0/CPU0:Router(config-if)# bundle minimum-active links 1
RP/0/RSP0/CPU0:Router(config-if)# exit
RP/0/RSP0/CPU0:Router(config)# interface TenGigE 0/3/0/0
RP/0/RSP0/CPU0:Router(config-if)# bundle id 3 mode active
RP/0/RSP0/CPU0:Router(config-if)# no shutdown
RP/0/RSP0/CPU0:Router(config)# exit
RP/0/RSP0/CPU0:Router(config)# interface TenGigE 0/3/0/1
RP/0/RSP0/CPU0:Router(config-if)# bundle id 3 mode active
RP/0/RSP0/CPU0:Router(config-if)# no shutdown
RP/0/RSP0/CPU0:Router(config-if)# exit
```

## Creating VLANs on a Ethernet Bundle: Example

This example shows how to create and bring up two VLANs on an Ethernet bundle:

```
RP/0/RSP0/CPU0:Router# config
RP/0/RSP0/CPU0:Router(config)# interface Bundle-Ether 1
RP/0/RSP0/CPU0:Router(config-if)# ipv4 address 1.2.3.4/24
RP/0/RSP0/CPU0:Router(config-if)# bundle minimum-active bandwidth 620000
RP/0/RSP0/CPU0:Router(config-if)# bundle minimum-active links 1
RP/0/RSP0/CPU0:Router(config-if)# exit
RP/0/RSP0/CPU0:Router(config)# interface Bundle-Ether 1.1
RP/0/RSP0/CPU0:Router(config-subif)# encapsulation dot1q 10
RP/0/RSP0/CPU0:Router(config-subif)# ip addr 10.2.3.4/24
RP/0/RSP0/CPU0:Router(config-subif)# no shutdown
RP/0/RSP0/CPU0:Router(config-subif)# exit
RP/0/RSP0/CPU0:Router(config)# interface Bundle-Ether 1.2
RP/0/RSP0/CPU0:Router(config-subif)# encapsulation dot1q 20
RP/0/RSP0/CPU0:Router(config-subif)# ip addr 20.2.3.4/24
RP/0/RSP0/CPU0:Router(config-subif)# no shutdown
RP/0/RSP0/CPU0:Router(config-subif)# exit
RP/0/RSP0/CPU0:Router(config)#interface tengige 0/1/5/7
RP/0/RSP0/CPU0:Router(config-if)# bundle-id 1 mode act
RP/0/RSP0/CPU0:Router(config-if)# commit
RP/0/RSP0/CPU0:Router(config-if)# exit
```

```
RP/0/RSP0/CPU0:Router(config)# exit
RP/0/RSP0/CPU0:Router # show ethernet trunk bundle-ether 1
```

# ASR 9000 Link Bundles connected to a Cisco 7600 EtherChannel: Example

This example is an end-to-end example of a bundle between ASR 9000 Series router (ASR-9010) and a Cisco 7600 Series Router (P19_C7609-S) in the Metro Ethernet network that supports both L2 and L3 services.

On the Cisco ASR 9000 Series Routers, the bundle is configured with LACP, 1:1 link protection, two L2 subinterfaces, and two layer 3 subinterfaces.

### IOS XR side:

```
hostname PE44_IOS-XR_Router

interface Bundle-Ether16
 description Connect to P19_C7609-S Port-Ch 16
 mtu 9216
 no ipv4 address
 bundle maximum-active links 1
!
interface Bundle-Ether16.160 l2transport
 description Connect to P19_C7609-S Port-Ch 16 EFP 160
 encapsulation dot1q 160
!
interface Bundle-Ether16.161 l2transport
 description Connect to P19_C7609-S Port-Ch 16 EFP 161
 encapsulation dot1q 161
!
interface Bundle-Ether16.162
 description Connect to P19_C7609-S Port-Ch 16.162
 ipv4 address 10.194.8.44 255.255.255.0
 encapsulation dot1q 162
!
interface Bundle-Ether16.163
 description Connect to P19_C7609-S Port-Ch 16.163
 ipv4 address 10.194.12.44 255.255.255.0
 encapsulation dot1q 163
!


interface TenGigE 0/1/0/16
 description Connected to P19_C7609-S GE 8/0/16
 bundle id 16 mode active
 bundle port-priority 1
!
interface TenGigE 0/1/0/17
 description Connected to P19_C7609-S GE 8/0/17
 bundle id 16 mode active
 bundle port-priority 2
!
```

### IOS XR side - connections to CE devices:

```
hostname PE44_IOS-XR_Router

interface TenGigE 0/1/0/3.160 l2transport
 description VLAN 160 over BE 16.160
 encapsulation dot1q 100 second-dot1q 160
 rewrite ingress tag pop 1 symmetric
!
```

```
interface TenGigE 0/1/0/3.161 l2transport
 description VLAN 161 over BE 16.161
 encapsulation dot1q 161
!
l2vpn
 !
 xconnect group 160
  p2p 160
   interface Bundle-Ether16.160
   interface TenGigE 0/1/0/3.160
   description VLAN_160_over_BE_16.160
  !
!
 xconnect group 161
  p2p 161
   interface Bundle-Ether16.161
   interface TenGigE 0/1/0/3.161
   description VLAN_161_over_BE_16.161
  !
 !
```

### IOS XR side - CE devices:

```
hostname PE64_C3750-ME
!
vlan 161
!
interface TenGigE 1/0/1
 description Connected to PE65_ME-C3400 GE 0/1
 switchport access vlan 100
 switchport mode dot1q-tunnel
!
interface TenGigE 1/0/2
 description Connected to PE44_IOS-XR_Router GE 0/1/0/3
 switchport trunk encapsulation dot1q
 switchport trunk allowed vlan 100,161
 switchport mode trunk
!
interface Vlan161
 description VLAN 161 over BE 16.161 on PE44
 ip address 161.0.0.64 255.255.255.0
!
```

```
hostname PE65_ME-C3400
!
vlan 160
!
interface TenGigE 0/1
 description Connected to PE64_C3750-ME GE 1/0/1
 port-type nni
 switchport trunk allowed vlan 160
 switchport mode trunk
!
interface Vlan160
 description VLAN 160 over BE 16.160 on PE44
 ip address 160.0.0.65 255.255.255.0
!
```

### IOS side:

```
hostname P19_C7609-S

port-channel load-balance src-dst-port
!
interface Port-channel16
 description Connected to PE44_IOS-XR_Router BE 16
 mtu 9202
 no ip address
 logging event link-status
 logging event status
 speed nonegotiate
 mls qos trust dscp
 lacp fast-switchover
 lacp max-bundle 1
 service instance 160 ethernet
  description Connected to PE44_IOS-XR_Router BE 16.160
  encapsulation dot1q 160
 !
 service instance 161 ethernet
  description Connected to PE44_IOS-XR_Router BE 16.161
  encapsulation dot1q 161
 !
!
interface Port-channel16.162
 description Connected to PE44_IOS-XR_Router BE 16.162
 encapsulation dot1Q 162
 ip address 10.194.8.19 255.255.255.0
!
interface Port-channel16.163
 description Connected to PE44_IOS-XR_Router BE 16.163
 encapsulation dot1Q 163
 ip address 10.194.12.19 255.255.255.0
!

interface TenGigE 8/0/16
 no shut
 description Connected to PE44_IOS-XR_Router GE 0/1/0/16
 mtu 9202
 no ip address
 logging event link-status
 logging event status
 speed nonegotiate
 no mls qos trust dscp
 lacp port-priority 1
 channel-protocol lacp
 channel-group 16 mode active
!
interface TenGigE 8/0/17
 no shut
 description Connected to PE44_IOS-XR_Router GE 0/1/0/17
 mtu 9202
 no ip address
 logging event link-status
 logging event status
 speed nonegotiate
 no mls qos trust dscp
 lacp port-priority 2
 channel-protocol lacp
 channel-group 16 mode active
!
```

### IOS side - connections to CE devices:

```
hostname P19_C7609-S

interface TenGigE 8/0/7
 description Connected to PE62_C3750-ME GE 1/0/2
 mtu 9000
 no ip address
 speed nonegotiate
 mls qos trust dscp
 service instance 160 ethernet
  description VLAN 160 over Port-Ch 16
  encapsulation dot1q 100 second-dot1q 160
  rewrite ingress tag pop 1 symmetric
 !
 service instance 161 ethernet
  description VLAN 161 over Port-Ch 16
  encapsulation dot1q 161
 !
!
connect eline-161 Port-channel16 161 TenGigE 8/0/7 161
 !
!
connect eline-160 Port-channel16 160 TenGigE 8/0/7 160
 !
!
```

### IOS side - CE devices:

```
hostname PE62_C3750-ME
!
vlan 161
!
interface TenGigE 1/0/1
 description Connected to PE63_ME-C3400 GE 0/1
 switchport access vlan 100
 switchport mode dot1q-tunnel
!
interface TenGigE 1/0/2
 description Connected to P19_C7609-S GE 8/0/7
 switchport trunk encapsulation dot1q
 switchport trunk allowed vlan 100,161
 switchport mode trunk
!
interface Vlan161
 description VLAN 161 over Port-Chan 16 on P19
 ip address 161.0.0.62 255.255.255.0
!



hostname PE63_ME-C3400
!
vlan 160
!
interface TenGigE 0/1
 description Connected to PE62_C3750-ME GE 1/0/1
 port-type nni
 switchport trunk allowed vlan 160
 switchport mode trunk
!
interface Vlan160
 description VLAN 160 over Port-Chan 16 on P19
```

```
 ip address 160.0.0.63 255.255.255.0
!
```