



Configuring 802.1Q VLAN Interfaces on the Cisco ASR 9000 Series Router

This module describes the configuration and management of 802.1Q VLAN interfaces on the Cisco ASR 9000 Series Aggregation Services Routers.

The IEEE 802.1Q specification establishes a standard method for tagging Ethernet frames with VLAN membership information, and defines the operation of VLAN bridges that permit the definition, operation, and administration of VLAN topologies within a bridged LAN infrastructure.

The 802.1Q standard is intended to address the problem of how to divide large networks into smaller parts so broadcast and multicast traffic does not use more bandwidth than necessary. The standard also helps provide a higher level of security between segments of internal networks.

Feature History for Configuring 802.1Q VLAN Interfaces

Release	Modification
Release 3.7.2	This feature was introduced on the Cisco ASR 9000 Series Router.
Release 3.9.0	Layer 2 dot1q was updated. Encapsulation dot1q was added.

Contents

- [Prerequisites for Configuring 802.1Q VLAN Interfaces, page 595](#)
- [Information About Configuring 802.1Q VLAN Interfaces, page 596](#)
- [How to Configure 802.1Q VLAN Interfaces, page 599](#)
- [Configuration Examples for VLAN Interfaces, page 605](#)
- [Additional References, page 607](#)

Prerequisites for Configuring 802.1Q VLAN Interfaces

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.

Before configuring 802.1Q VLAN interfaces, be sure that the following conditions are met:

- You must have configured a Gigabit Ethernet interface, a 10-Gigabit Ethernet interface, or an Ethernet bundle interface.

Information About Configuring 802.1Q VLAN Interfaces

To configure 802.1Q VLAN interfaces, you must understand the following concepts:

- [802.1Q VLAN Overview, page 596](#)
- [802.1Q Tagged Frames, page 596](#)
- [CFM on 802.1Q VLAN Interfaces, page 597](#)
- [Subinterfaces, page 597](#)
- [Subinterface MTU, page 597](#)
- [Native VLAN, page 597](#)
- [EFPs, page 597](#)
- [Layer 2 VPN on VLANs, page 598](#)
- [Other Layer 2 VPN Features, page 599](#)

802.1Q VLAN Overview

A VLAN is a group of devices on one or more LANs that are configured so that they can communicate as if they were attached to the same wire, when in fact they are located on a number of different LAN segments. Because VLANs are based on logical instead of physical connections, they are very flexible for user and host management, bandwidth allocation, and resource optimization.

The IEEE 802.1Q protocol standard addresses the problem of dividing large networks into smaller parts so broadcast and multicast traffic does not consume more bandwidth than necessary. The standard also helps provide a higher level of security between segments of internal networks.

The 802.1Q specification establishes a standard method for inserting VLAN membership information into Ethernet frames.

Cisco IOS XR software supports VLAN subinterface configuration on Gigabit Ethernet and 10-Gigabit Ethernet interfaces.

802.1Q Tagged Frames

The IEEE 802.1Q tag-based VLAN uses an extra tag in the MAC header to identify the VLAN membership of a frame across bridges. This tag is used for VLAN and quality of service (QoS) priority identification. The VLANs can be created statically by manual entry or dynamically through Generic Attribute Registration Protocol (GARP) VLAN Registration Protocol (GVRP). The VLAN ID associates a frame with a specific VLAN and provides the information that switches must process the frame across the network. A tagged frame is four bytes longer than an untagged frame and contains two bytes of Tag Protocol Identifier (TPID) residing within the type and length field of the Ethernet frame and two bytes of Tag Control Information (TCI) which starts after the source address field of the Ethernet frame.

CFM on 802.1Q VLAN Interfaces

Configuring Connectivity Fault Management (CFM) for monitoring 802.1Q VLAN interfaces is identical to configuring CFM for monitoring Ethernet interfaces.

For information on configuring CFM for Ethernet interfaces, refer to the following sections in the “Configuring Ethernet OAM on the Cisco ASR 9000 Series Router” module:

- [Ethernet CFM, page 64](#)
- [Configuring Ethernet CFM, page 99](#)
- [Ethernet CFM Domain Configuration: Example, page 153](#)
- [Ethernet CFM Show Command: Examples, page 155](#)

Subinterfaces

Subinterfaces are logical interfaces created on a hardware interface. These software-defined interfaces allow for segregation of traffic into separate logical channels on a single hardware interface as well as allowing for better utilization of the available bandwidth on the physical interface.

Subinterfaces are distinguished from one another by adding an extension on the end of the interface name and designation. For instance, the Ethernet subinterface 23 on the physical interface designated TenGigE 0/1/0/0 would be indicated by TenGigE 0/1/0/0.23.

Before a subinterface is allowed to pass traffic it must have a valid tagging protocol encapsulation and VLAN identifier assigned. All Ethernet subinterfaces always default to the 802.1Q VLAN encapsulation. However, the VLAN identifier must be explicitly defined.

Subinterface MTU

The subinterface maximum transmission unit (MTU) is inherited from the physical interface with an additional four bytes allowed for the 802.1Q VLAN tag.

Native VLAN

The Cisco ASR 9000 Series Router does not support a native VLAN. However, the equivalent functionality is accomplished using an **encapsulation** command as follows:

```
encapsulation dot1q TAG-ID, untagged
```

EFPs

An Ethernet Flow Point (EFP) is a Metro Ethernet Forum (MEF) term describing abstract router architecture. On the Cisco ASR 9000 Series Router, an EFP is implemented by an L2 subinterface with a VLAN encapsulation. The term EFP is used synonymously with an VLAN tagged L2 subinterface.

Layer 2 VPN on VLANs

The Layer 2 Virtual Private Network (L2VPN) feature enables Service Providers (SPs) to provide Layer 2 services to geographically disparate customer sites.

The configuration model for configuring VLAN attachment circuits (ACs) is similar to the model used for configuring basic VLANs, where the user first creates a VLAN subinterface, and then configures that VLAN in subinterface configuration mode. To create an AC, you need to include the **l2transport** keyword in the **interface** command string to specify that the interface is a Layer 2 interface.

VLAN ACs support three modes of L2VPN operation:

- Basic Dot1Q AC—The AC covers all frames that are received and sent with a specific VLAN tag.
- QinQ AC—The AC covers all frames received and sent with a specific outer VLAN tag and a specific inner VLAN tag. QinQ is an extension to Dot1Q that uses a stack of two tags.
- Q-in-Any AC—The AC covers all frames received and sent with a specific outer VLAN tag and any inner VLAN tag, as long as that inner VLAN tag is not L3 terminated. Q-in-Any is an extension to QinQ that uses wildcarding to match any second tag.



Note The Q-in-Any mode is a variation of the basic Dot1Q mode. In Q-in-Any mode, the frames have a basic QinQ encapsulation; however, in Q-in-Any mode the inner tag is not relevant, except for the fact that a few specific inner VLAN tags are siphoned for specific services. For example, a tag may be used to provide L3 services for general internet access.

Each VLAN on a CE-to-PE link can be configured as a separate L2VPN connection (using either VC type 4 or VC type 5). To configure L2VPN on VLANs, see the [“Configuring an Attachment Circuit on a VLAN”](#) section on page 601.

Keep the following in mind when configuring L2VPN on a VLAN:

- Cisco IOS XR software supports 4k ACs per LC.
- In a point-to-point connection, the two ACs do not have to be of the same type. For example, a port mode Ethernet AC can be connected to a Dot1Q Ethernet AC.
- Pseudowires can run in VLAN mode or in port mode. A pseudowire running in VLAN mode has a single Dot1Q tag, while a pseudo-wire running in port mode has no tags. Some interworking is required to connect these different types of circuits together. This interworking takes the form of popping, pushing, and rewriting tags. The advantage of Layer 2 VPN is that it simplifies the interworking required to connect completely different media types together.
- The ACs on either side of an MPLS pseudowire can be different types. In this case, the appropriate conversion is carried out at one or both ends of the AC to pseudowire connection.

Use the **show interfaces** command to display AC and pseudowire information.



Note

For detailed information about configuring an L2VPN network, see the *“Implementing MPLS Layer 2 VPNs”* module of the *Cisco ASR 9000 Series Router Multiprotocol Label Switching Configuration Guide*.

Other Layer 2 VPN Features

For information on the following Layer 2 VPN features, refer to the *Cisco ASR 9000 Series Aggregation Services Router L2VPN and Ethernet Services Configuration Guide* and the *Cisco ASR 9000 Series Aggregation Services Router L2VPN and Ethernet Services Command Reference*:

- Provider Backbone Bridge (PBB) 802.1ah
- Policy-Based Forwarding (PBF)
- MVRP 802.1 (MVRP-lite)

How to Configure 802.1Q VLAN Interfaces

This section contains the following procedures:

- [Configuring 802.1Q VLAN Subinterfaces, page 599](#)
- [Configuring an Attachment Circuit on a VLAN, page 601](#)
- [Removing an 802.1Q VLAN Subinterface, page 603](#)

Configuring 802.1Q VLAN Subinterfaces

This task explains how to configure 802.1Q VLAN subinterfaces. To remove these subinterfaces, see the “[Removing an 802.1Q VLAN Subinterface](#)” section of this module.

SUMMARY STEPS

1. **configure**
2. **interface** { **GigabitEthernet** | **TenGigE** | **Bundle-Ether** } *interface-path-id.subinterface*
3. **encapsulation dot1q**
4. **ipv4 address** *ip-address mask*
5. **exit**
6. Repeat Step 2 through Step 5 to define the rest of the VLAN subinterfaces.
7. **end**
or
commit
8. **show ethernet trunk bundle-ether** *instance* (Optional)

DETAILED STEPS

	Command or Action	Purpose
Step 1	<p>configure</p> <p>Example: RP/0/RSP0/CPU0:router# configure</p>	Enters global configuration mode.
Step 2	<p>interface {GigabitEthernet TenGigE Bundle-Ether} <i>interface-path-id.subinterface</i></p> <p>Example: RP/0/RSP0/CPU0:router(config)# interface TenGigE 0/2/0/4.10</p>	<p>Enters subinterface configuration mode and specifies the interface type, location, and subinterface number.</p> <ul style="list-style-type: none"> Replace the <i>interface-path-id</i> argument with one of the following instances: <ul style="list-style-type: none"> Physical Ethernet interface instance, or with an Ethernet bundle instance. Naming notation is <i>rack/slot/module/port</i>, and a slash between values is required as part of the notation. Ethernet bundle instance. Range is from 1 through 65535. Replace the <i>subinterface</i> argument with the subinterface value. Range is from 0 through 4095. Naming notation is <i>interface-path-id.subinterface</i>, and a period between arguments is required as part of the notation.
Step 3	<p>encapsulation dot1q</p> <p>Example: RP/0/RSP0/CPU0:router(config-subif)# encapsulation dot1q 100, untagged</p>	<p>Sets the Layer 2 encapsulation of an interface.</p> <p>Note The dot1q vlan command is replaced by the encapsulation dot1q command on the Cisco ASR 9000 Series Router. It is still available for backward-compatibility, but only for Layer 3 interfaces.</p>
Step 4	<p>ipv4 address <i>ip-address mask</i></p> <p>Example: RP/0/RSP0/CPU0:router(config-subif)# ipv4 address 178.18.169.23/24</p>	<p>Assigns an IP address and subnet mask to the subinterface.</p> <ul style="list-style-type: none"> Replace <i>ip-address</i> with the primary IPv4 address for an interface. Replace <i>mask</i> with the mask for the associated IP subnet. The network mask can be specified in either of two ways: <ul style="list-style-type: none"> The network mask can be a four-part dotted decimal address. For example, 255.0.0.0 indicates that each bit equal to 1 means that the corresponding address bit belongs to the network address. The network mask can be indicated as a slash (/) and number. For example, /8 indicates that the first 8 bits of the mask are ones, and the corresponding bits of the address are network address.

	Command or Action	Purpose
Step 5	exit Example: RP/0/RSP0/CPU0:router(config-subif)# exit	(Optional) Exits the subinterface configuration mode. <ul style="list-style-type: none"> The exit command is not explicitly required.
Step 6	Repeat Step 2 through Step 5 to define the rest of the VLAN subinterfaces.	—
Step 7	end or commit Example: RP/0/RSP0/CPU0:router(config)# end OR RP/0/RSP0/CPU0:router(config)# commit	Saves configuration changes. <ul style="list-style-type: none"> When you issue the end command, the system prompts you to commit changes: <pre>Uncommitted changes found, commit them before exiting (yes/no/cancel)? [cancel]:</pre> <ul style="list-style-type: none"> 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 command to save the configuration changes to the running configuration file and remain within the configuration session.
Step 8	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.

Configuring an Attachment Circuit on a VLAN

Use the following procedure to configure an attachment circuit on a VLAN.

SUMMARY STEPS

- configure**
- interface** { GigabitEthernet | TenGigE | Bundle-Ether } *interface-path-id.subinterface* l2transport
- encapsulation dot1q**
- l2protocol cpsv** { tunnel | reverse-tunnel }
- end**
or
commit

6. show interfaces [GigabitEthernet | TenGigE]

DETAILED STEPS

	Command or Action	Purpose
Step 1	<p>configure</p> <p>Example: RP/0/RSP0/CPU0:router# configure terminal</p>	Enters global configuration mode.
Step 2	<p>interface [GigabitEthernet TenGigE Bundle-Ether TenGigE] <i>interface-path</i> <i>id.subinterface</i> l2transport</p> <p>Example: RP/0/RSP0/CPU0:router(config)# interface TenGigE 0/1/0/0.1 l2transport</p>	<p>Enters subinterface configuration and specifies the interface type, location, and subinterface number.</p> <ul style="list-style-type: none"> Replace the argument with one of the following instances: <ul style="list-style-type: none"> Physical Ethernet interface instance, or with an Ethernet bundle instance. Naming notation is <i>rack/slot/module/port</i>, and a slash between values is required as part of the notation. Ethernet bundle instance. Range is from 1 through 65535. Replace the <i>subinterface</i> argument with the subinterface value. Range is from 0 through 4095. Naming notation is <i>instance.subinterface</i>, and a period between arguments is required as part of the notation. <p>Note You must include the l2transport keyword in the command string; otherwise, the configuration creates a Layer 3 subinterface rather than an AC.</p>
Step 3	<p>encapsulation dot1q</p> <p>Example: RP/0/RSP0/CPU0:router(config-subif)# encapsulation dot1q 100, untagged</p>	<p>Sets the Layer 2 encapsulation of an interface.</p> <p>Note The dot1q vlan command is replaced by the encapsulation dot1q command on the Cisco ASR 9000 Series Router. It is still available for backward-compatibility, but only for Layer 3 interfaces.</p>
Step 4	<p>l2protocol cpsv {tunnel reverse-tunnel}</p> <p>Example: RP/0/RSP0/CPU0:router(config-if-l2)# l2protocol cpsv tunnel</p>	<p>Configures Layer 2 protocol tunneling and protocol data unit (PDU) filtering on an Ethernet interface for the following protocols: CDP, PVST+, STP, VTP, where:</p> <ul style="list-style-type: none"> tunnel—Specifies L2PT encapsulation on frames as they enter the interface, and de-encapsulation on frames as they exit they interface. reverse-tunnel—Specifies L2PT encapsulation on frames as they exit the interface, and de-encapsulation on frames as they enter the interface.

	Command or Action	Purpose
Step 5	<pre>end or commit</pre> <p>Example: RP/0/RSP0/CPU0:router(config-if-12)# end OR RP/0/RSP0/CPU0:router(config-if-12)# commit </p>	<p>Saves configuration changes.</p> <ul style="list-style-type: none"> When you issue the end command, the system prompts you to commit changes: <pre>Uncommitted changes found, commit them before exiting (yes/no/cancel)? [cancel]:</pre> <ul style="list-style-type: none"> 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 command to save the configuration changes to the running configuration file and remain within the configuration session.
Step 6	<pre>show interfaces [GigabitEthernet TenGigE] interface-path-id.subinterface</pre> <p>Example: RP/0/RSP0/CPU0:router# show interfaces TenGigE 0/3/0/0.1 </p>	<p>(Optional) Displays statistics for interfaces on the router.</p>

What to Do Next

- To configure a point-to-point pseudowire cross connect on the AC, see the “*Implementing MPLS Layer 2 VPNs*” module of the *Cisco ASR 9000 Series Router Multiprotocol Label Switching Configuration Guide*.
- To attach Layer 3 service policies, such as Multiprotocol Label Switching (MPLS) or QoS, to the VLAN, refer to the appropriate Cisco ASR 9000 Series Router configuration guide.

Removing an 802.1Q VLAN Subinterface

This task explains how to remove 802.1Q VLAN subinterfaces that have been previously configured using the “[Configuring 802.1Q VLAN Subinterfaces](#)” task in this module.

SUMMARY STEPS

- configure**
- no interface** { GigabitEthernet | TenGigE | Bundle-Ether } interface-path-id.subinterface
- Repeat Step 2 to remove other VLAN subinterfaces.

4. **end**
or
commit
5. **show ethernet trunk bundle-ether** *instance* (Optional)

DETAILED STEPS

	Command or Action	Purpose
Step 1	<p>configure</p> <p>Example: RP/0/RSP0/CPU0:router# configure</p>	Enters global configuration mode.
Step 2	<p>no interface {GigabitEthernet TenGigE Bundle-Ether} <i>interface-path-id.subinterface</i></p> <p>Example: RP/0/RSP0/CPU0:router(config)# no interface TenGigE 0/2/0/4.10</p>	<p>Removes the subinterface, which also automatically deletes all the configuration applied to the subinterface.</p> <ul style="list-style-type: none"> • Replace the <i>instance</i> argument with one of the following instances: <ul style="list-style-type: none"> – Physical Ethernet interface instance, or with an Ethernet bundle instance. Naming notation is <i>rack/slot/module/port</i>, and a slash between values is required as part of the notation. – Ethernet bundle instance. Range is from 1 through 65535. • Replace the <i>subinterface</i> argument with the subinterface value. Range is from 0 through 4095. <p>Naming notation is <i>instance.subinterface</i>, and a period between arguments is required as part of the notation.</p>

	Command or Action	Purpose
Step 3	Repeat Step 2 to remove other VLAN subinterfaces.	—
Step 4	<pre>end or commit</pre> <p>Example:</p> <pre>RP/0/RSP0/CPU0:router(config)# end or RP/0/RSP0/CPU0:router(config)# commit</pre>	<p>Saves configuration changes.</p> <ul style="list-style-type: none"> When you issue the end command, the system prompts you to commit changes: <pre>Uncommitted changes found, commit them before exiting(yes/no/cancel)? [cancel]:</pre> <ul style="list-style-type: none"> 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 command to save the configuration changes to the running configuration file and remain within the configuration session.
Step 5	<pre>show ethernet trunk bundle-ether instance</pre> <p>Example:</p> <pre>RP/0/RSP0/CPU0:router# show ethernet trunk bundle-ether 5</pre>	<p>(Optional) Displays the interface configuration.</p> <p>The Ethernet bundle instance range is from 1 through 65535.</p>

Configuration Examples for VLAN Interfaces

This section contains the following example:

[VLAN Subinterfaces: Example, page 605](#)

VLAN Subinterfaces: Example

The following example shows how to create three VLAN subinterfaces at one time:

```
RP/0/RSP0/CPU0:router# configure
RP/0/RSP0/CPU0:router(config)# interface TenGigE 0/2/0/4.1
RP/0/RSP0/CPU0:router(config-subif)# encapsulation dot1q 100
RP/0/RSP0/CPU0:router(config-subif)# ipv4 address 10.0.10.1/24
RP/0/RSP0/CPU0:router(config-subif)# interface TenGigE0/2/0/4.2
RP/0/RSP0/CPU0:router(config-subif)# encapsulation dot1q 101
RP/0/RSP0/CPU0:router(config-subif)# ipv4 address 10.0.20.1/24
RP/0/RSP0/CPU0:router(config-subif)# interface TenGigE0/2/0/4.3
RP/0/RSP0/CPU0:router(config-subif)# encapsulation dot1q 102
RP/0/RSP0/CPU0:router(config-subif)# ipv4 address 10.0.30.1/24
RP/0/RSP0/CPU0:router(config-subif)# commit
RP/0/RSP0/CPU0:router(config-subif)# exit
RP/0/RSP0/CPU0:router(config)# exit
```

```
RP/0/RSP0/CPU0:router# show ethernet trunk bundle-Ether 1
Trunk
Interface      St Ly   MTU   Subs   Sub types          Sub states
                Up L3   1514  1000   L2    L3    Up    Down  Ad-Down
BE1            Up L3   1514  1000   0    1000  1000  0    0

Summary
                1000   0    1000  1000  0    0
```

The following example shows how to create two VLAN subinterfaces on an Ethernet bundle:

```
RP/0/RSP0/CPU0:router# configure
RP/0/RSP0/CPU0:router(config)# interface bundle-ether 2
RP/0/RSP0/CPU0:router(config-if)# ipv4 address 192.168.2.1/24
RP/0/RSP0/CPU0:router(config-if)# exit
RP/0/RSP0/CPU0:router(config)# interface bundle-ether 2.1
RP/0/RSP0/CPU0:router(config-subif)# encapsulation dot1q 100
RP/0/RSP0/CPU0:router(config-subif)# ipv4 address 192.168.100.1/24
RP/0/RSP0/CPU0:router(config-subif)# exit
RP/0/RSP0/CPU0:router(config)# interface bundle-ether 2.2
RP/0/RSP0/CPU0:router(config-subif)# encapsulation dot1q 200
RP/0/RSP0/CPU0:router(config-subif)# ipv4 address 192.168.200.1/24
RP/0/RSP0/CPU0:router(config-subif)# exit
RP/0/RSP0/CPU0:router(config)# commit
```

The following example shows how to create a basic dot1Q AC:

```
RP/0/RSP0/CPU0:router# configure
RP/0/RSP0/CPU0:router(config)# interface GigabitEthernet 0/0/0/0.1
RP/0/RSP0/CPU0:router(config-subif)# l2transport
RP/0/RSP0/CPU0:router(config-subif)# encapsulation dot1q 100
RP/0/RSP0/CPU0:router(config-subif)# commit
RP/0/RSP0/CPU0:router(config-subif)# exit
RP/0/RSP0/CPU0:router(config)# exit
```

The following example shows how to create a Q-in-Q AC:

```
RP/0/RSP0/CPU0:router# configure
RP/0/RSP0/CPU0:router(config)# interface GigabitEthernet 0/0/0/0.2
RP/0/RSP0/CPU0:router(config-subif)# l2transport
RP/0/RSP0/CPU0:router(config-subif)# encapsulation dot1q 200 second-dot1q 201
RP/0/RSP0/CPU0:router(config-subif)# commit
RP/0/RSP0/CPU0:router(config-subif)# exit
RP/0/RSP0/CPU0:router(config)# exit
```

The following example shows how to create a Q-in-Any AC:

```
RP/0/RSP0/CPU0:router# configure
RP/0/RSP0/CPU0:router(config)# interface GigabitEthernet 0/0/0/0.3
RP/0/RSP0/CPU0:router(config-subif)# l2transport
RP/0/RSP0/CPU0:router(config-subif)# encapsulation dot1q 300 second-dot1q any
RP/0/RSP0/CPU0:router(config-subif)# commit
RP/0/RSP0/CPU0:router(config-subif)# exit
RP/0/RSP0/CPU0:router(config)# exit
```

Additional References

The following sections provide references related to VLAN interface configuration.

Related Documents

Related Topic	Document Title
Cisco ASR 9000 Series Router master command reference	<i>Cisco ASR 9000 Series Router Master Commands List</i>
Cisco ASR 9000 Series Router interface configuration commands	<i>Cisco ASR 9000 Series Router Interface and Hardware Component Command Reference</i>
Initial system bootup and configuration information for a Cisco ASR 9000 Series Router using the Cisco IOS XR software.	<i>Cisco ASR 9000 Series Router Getting Started Guide</i>
Information about user groups and task IDs	<i>Cisco ASR 9000 Series Router Interface and Hardware Component Command Reference</i>
Information about configuring interfaces and other components on the Cisco ASR 9000 Series Router from a remote Craft Works Interface (CWI) client management application	<i>Cisco ASR 9000 Series Router Craft Works Interface Configuration Guide</i>

Standards

Standards	Title
No new or modified standards are supported by this feature, and support for existing standards has not been modified by this feature	—

MIBs

MIBs	MIBs Link
There are no applicable MIBs for this module.	To locate and download MIBs for selected platforms using Cisco IOS XR software, use the Cisco MIB Locator found at the following URL: http://cisco.com/public/sw-center/netmgmt/cmtk/mibs.shtml

RFCs

RFCs	Title
No new or modified RFCs are supported by this feature, and support for existing RFCs has not been modified by this feature.	—

Technical Assistance

Description	Link
The Cisco Technical Support website contains thousands of pages of searchable technical content, including links to products, technologies, solutions, technical tips, and tools. Registered Cisco.com users can log in from this page to access even more content.	http://www.cisco.com/techsupport