Cisco IOS Network Management Configuration Guide

Release 12.4

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.

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 NONINFRINGEMENT 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 used in this document are not intended to be actual addresses. Any examples, command display output, and figures included in the document are shown for illustrative purposes only. Any use of actual IP addresses in illustrative content is unintentional and coincidental.

Cisco IOS Network Management Configuration Guide

About Cisco IOS and Cisco IOS XE Software Documentation

Last updated: August 6, 2008

This document describes the objectives, audience, conventions, and organization used in Cisco IOS and Cisco IOS XE software documentation, collectively referred to in this document as Cisco IOS documentation. Also included are resources for obtaining technical assistance, additional documentation, and other information from Cisco. This document is organized into the following sections:

- Documentation Objectives, page i
- Audience, page i
- Documentation Conventions, page ii
- Documentation Organization, page iii
- Additional Resources and Documentation Feedback, page xi

Documentation Objectives

Cisco IOS documentation describes the tasks and commands available to configure and maintain Cisco networking devices.

Audience

The Cisco IOS documentation set is i ntended for users who configure and maintain Cisco networking devices (such as routers and switches) but who may not be familiar with the configuration and maintenance tasks, the relationship among tasks, or the Cisco IOS commands necessary to perform particular tasks. The Cisco IOS documentation set is also intended for those users experienced with Cisco IOS who need to know about new features, new configuration options, and new software characteristics in the current Cisco IOS release.

Documentation Conventions

In Cisco IOS documentation, the term *router* may be used to refer to various Cisco products; for example, routers, access servers, and switches. These and other networking devices that support Cisco IOS software are shown interchangeably in examples and are used only for illustrative purposes. An example that shows one product does not necessarily mean that other products are not supported.

This section includes the following topics:

- Typographic Conventions, page ii
- Command Syntax Conventions, page ii
- Software Conventions, page iii
- Reader Alert Conventions, page iii

Typographic Conventions

Cisco IOS documentation uses the following typographic conventions:

Convention	Description
^ or Ctrl	Both the **^** symbol and Ctrl represent the Control (Ctrl) key on a keyboard. For example, the key combination **^D** or **Ctrl-D** means that you hold down the Control key while you press the D key. (Keys are indicated in capital letters but are not case sensitive.)
string	A string is a nonquoted set of characters shown in italics. For example, when setting a Simple Network Management Protocol (SNMP) community string to *public*, do not use quotation marks around the string; otherwise, the string will include the quotation marks.

Command Syntax Conventions

Cisco IOS documentation uses the following command syntax conventions:

Convention	Description
bold	Bold text indicates commands and keywords that you enter as shown.
italic	Italic text indicates arguments for which you supply values.
[x]	Square brackets enclose an optional keyword or argument.
|	A vertical line, called a pipe, indicates a choice within a set of keywords or arguments.
[x | y]	Square brackets enclosing keywords or arguments separated by a pipe indicate an optional choice.
{x | y}	Braces enclosing keywords or arguments separated by a pipe indicate a required choice.
[x {y | z}]	Braces and a pipe within square brackets indicate a required choice within an optional element.

Software Conventions

Cisco IOS uses the following program code conventions:

Convention	Description
`Courier font`	Courier font is used for information that is displayed on a PC or terminal screen.
`Bold Courier font`	Bold Courier font indicates text that the user must enter.
< >	Angle brackets enclose text that is not displayed, such as a password. Angle brackets also are used in contexts in which the italic font style is not supported; for example, ASCII text.
!	An exclamation point at the beginning of a line indicates that the text that follows is a comment, not a line of code. An exclamation point is also displayed by Cisco IOS software for certain processes.
[ ]	Square brackets enclose default responses to system prompts.

Reader Alert Conventions

The Cisco IOS documentation set uses the following conventions for reader alerts:

Caution Means *reader be careful*. In this situation, you might do something that could result in equipment damage or loss of data.

Note Means *reader take note*. Notes contain helpful suggestions or references to material not covered in the manual.

Timesaver Means *the described action saves time*. You can save time by performing the action described in the paragraph.

Documentation Organization

This section describes the Cisco IOS documentation set, how it is organized, and how to access it on Cisco.com. Included are lists of configuration guides, command references, and supplementary references and resources that make up the documentation set. The following topics are included:

- Cisco IOS Documentation Set, page iv
- Cisco IOS Documentation on Cisco.com, page iv
- Configuration Guides, Command References, and Supplementary Resources, page v

Cisco IOS Documentation Set

Cisco IOS documentation consists of the following:

- Release notes and caveats provide information about platform, technology, and feature support for a release and describe severity 1 (catastrophic), severity 2 (severe), and severity 3 (moderate) defects in released Cisco IOS code. Review release notes before other documents to learn whether or not updates have been made to a feature.
- Sets of configuration guides and command references organized by technology and published for each standard Cisco IOS release.
  - Configuration guides—Compilations of documents that provide informational and task-oriented descriptions of Cisco IOS features.
  - Command references—Compilations of command pages that provide detailed information about the commands used in the Cisco IOS features and processes that make up the related configuration guides. For each technology, there is a single command reference that covers all Cisco IOS releases and that is updated at each standard release.
- Lists of all the commands in a specific release and all commands that are new, modified, removed, or replaced in the release.
- Command reference book for **debug** commands. Command pages are listed in alphabetical order.
- Reference book for system messages for all Cisco IOS releases.

Cisco IOS Documentation on Cisco.com

The following sections describe the documentation organization and how to access various document types.

Use Cisco Feature Navigator to find information about platform support and Cisco IOS and Catalyst OS software image support. To access Cisco Feature Navigator, go to http://www.cisco.com/go/cfn. An account on Cisco.com is not required.

New Features List

The New Features List for each release provides a list of all features in the release with hyperlinks to the feature guides in which they are documented.

Feature Guides

Cisco IOS features are documented in feature guides. Feature guides describe one feature or a group of related features that are supported on many different software releases and platforms. Your Cisco IOS software release or platform may not support all the features documented in a feature guide. See the Feature Information table at the end of the feature guide for information about which features in that guide are supported in your software release.

Configuration Guides

Configuration guides are provided by technology and release and comprise a set of individual feature guides relevant to the release and technology.

Command References

Command reference books describe Cisco IOS commands that are supported in many different software releases and on many different platforms. The books are provided by technology. For information about all Cisco IOS commands, use the Command Lookup Tool at http://tools.cisco.com/Support/CLILookup or the *Cisco IOS Master Command List, All Releases*, at http://www.cisco.com/en/US/docs/ios/mcl/all_release/all_mcl.html.

Cisco IOS Supplementary Documents and Resources

Supplementary documents and resources are listed in Table 2 on page xi.

Configuration Guides, Command References, and Supplementary Resources

Table 1 lists, in alphabetical order, Cisco IOS and Cisco IOS XE software configuration guides and command references, including brief descriptions of the contents of the documents. The Cisco IOS command references are comprehensive, meaning that they include commands for both Cisco IOS software and Cisco IOS XE software, for all releases. The configuration guides and command references support many different software releases and platforms. Your Cisco IOS software release or platform may not support all these technologies.

For additional information about configuring and operating specific networking devices, go to the Product Support area of Cisco.com at http://www.cisco.com/web/psa/products/index.html.

Table 2 lists documents and resources that supplement the Cisco IOS software configuration guides and command references. These supplementary resources include release notes and caveats; master command lists; new, modified, removed, and replaced command lists; system messages; and the debug command reference.

Table 1 Cisco IOS and Cisco IOS XE Configuration Guides and Command References

Configuration Guide and Command Reference Titles	Features/Protocols/Technologies
Cisco IOS AppleTalk Configuration Guide *Cisco IOS XE AppleTalk Configuration Guide* *Cisco IOS AppleTalk Command Reference*	AppleTalk protocol.
Cisco IOS Asynchronous Transfer Mode Configuration Guide *Cisco IOS Asynchronous Transfer Mode Command Reference*	LAN ATM, multiprotocol over ATM (MPoA), and WAN ATM.

Table 1 Cisco IOS and Cisco IOS XE Configuration Guides and Command References (continued)

Configuration Guide and Command Reference Titles	Features/Protocols/Technologies
Cisco IOS Bridging and IBM Networking Configuration Guide *Cisco IOS Bridging Command Reference* *Cisco IOS IBM Networking Command Reference*	• Transparent and source-route transparent (SRT) bridging, source-route bridging (SRB), Token Ring Inter-Switch Link (TRISL), and token ring route switch module (TRRSM). • Data-link switching plus (DLSw+), serial tunnel (STUN), block serial tunnel (BSTUN); logical link control, type 2 (LLC2), synchronous data link control (SDLC); IBM Network Media Translation, including Synchronous Data Logical Link Control (SDLLC) and qualified LLC (QLLC); downstream physical unit (DSPU), Systems Network Architecture (SNA) service point, SNA frame relay access, advanced peer-to-peer networking (APPN), native client interface architecture (NCIA) client/server topologies, and IBM Channel Attach.
Cisco IOS Broadband and DSL Configuration Guide *Cisco IOS XE Broadband and DSL Configuration Guide* *Cisco IOS Broadband and DSL Command Reference*	Point-to-Point Protocol (PPP) over ATM (PPPoA) and PPP over Ethernet (PPPoE).
Cisco IOS Carrier Ethernet Configuration Guide *Cisco IOS Carrier Ethernet Command Reference*	Connectivity fault management (CFM), Ethernet Local Management Interface (ELMI), IEEE 802.3ad link bundling, Link Layer Discovery Protocol (LLDP), media endpoint discovery (MED), and operations, administration, and maintenance (OAM).
Cisco IOS Configuration Fundamentals Configuration Guide *Cisco IOS XE Configuration Fundamentals Configuration Guide* *Cisco IOS Configuration Fundamentals Command Reference*	Autoinstall, Setup, Cisco IOS command-line interface (CLI), Cisco IOS file system (IFS), Cisco IOS web browser user interface (UI), basic file transfer services, and file management.
Cisco IOS DECnet Configuration Guide *Cisco IOS XE DECnet Configuration Guide* *Cisco IOS DECnet Command Reference*	DECnet protocol.
Cisco IOS Dial Technologies Configuration Guide *Cisco IOS XE Dial Technologies Configuration Guide* *Cisco IOS Dial Technologies Command Reference*	Asynchronous communications, dial backup, dialer technology, dial-in terminal services and AppleTalk remote access (ARA), large scale dialout, dial-on-demand routing, dialout, modem and resource pooling, ISDN, multilink PPP (MLP), PPP, virtual private dialup network (VPDN).
Cisco IOS Flexible NetFlow Configuration Guide *Cisco IOS Flexible NetFlow Command Reference*	Flexible NetFlow.

Table 1 Cisco IOS and Cisco IOS XE Configuration Guides and Command References (continued)

Configuration Guide and Command Reference Titles	Features/Protocols/Technologies
Cisco IOS H.323 Configuration Guide	Gatekeeper enhancements for managed voice services, Gatekeeper Transaction Message Protocol, gateway codec order preservation and shutdown control, H.323 dual tone multifrequency relay, H.323 version 2 enhancements, Network Address Translation (NAT) support of H.323 v2 Registration, Admission, and Status (RAS) protocol, tokenless call authorization, and VoIP gateway trunk and carrier-based routing.
Cisco IOS High Availability Configuration Guide *Cisco IOS XE High Availability Configuration Guide* *Cisco IOS High Availability Command Reference*	A variety of High Availability (HA) features and technologies that are available for different network segments (from enterprise access to service provider core) to facilitate creation of end-to-end highly available networks. Cisco IOS HA features and technologies can be categorized in three key areas: system-level resiliency, network-level resiliency, and embedded management for resiliency.
Cisco IOS Integrated Session Border Controller Command Reference	A VoIP-enabled device that is deployed at the edge of networks. An SBC is a toolkit of functions, such as signaling interworking, network hiding, security, and quality of service (QoS).
Cisco IOS Intelligent Service Gateway Configuration Guide *Cisco IOS Intelligent Service Gateway Command Reference*	Subscriber identification, service and policy determination, session creation, session policy enforcement, session life-cycle management, accounting for access and service usage, session state monitoring.
Cisco IOS Interface and Hardware Component Configuration Guide *Cisco IOS XE Interface and Hardware Component Configuration Guide* *Cisco IOS Interface and Hardware Component Command Reference*	LAN interfaces, logical interfaces, serial interfaces, virtual interfaces, and interface configuration.
Cisco IOS IP Addressing Services Configuration Guide *Cisco IOS XE Addressing Services Configuration Guide* *Cisco IOS IP Addressing Services Command Reference*	Address Resolution Protocol (ARP), Network Address Translation (NAT), Domain Name System (DNS), Dynamic Host Configuration Protocol (DHCP), and Next Hop Address Resolution Protocol (NHRP).
Cisco IOS IP Application Services Configuration Guide *Cisco IOS XE IP Application Services Configuration Guide* *Cisco IOS IP Application Services Command Reference*	Enhanced Object Tracking (EOT), Gateway Load Balancing Protocol (GLBP), Hot Standby Router Protocol (HSRP), IP Services, Server Load Balancing (SLB), Stream Control Transmission Protocol (SCTP), TCP, Web Cache Communication Protocol (WCCP), User Datagram Protocol (UDP), and Virtual Router Redundancy Protocol (VRRP).
Cisco IOS IP Mobility Configuration Guide *Cisco IOS IP Mobility Command Reference*	Mobile ad hoc networks (MANet) and Cisco mobile networks.
Cisco IOS IP Multicast Configuration Guide *Cisco IOS XE IP Multicast Configuration Guide* *Cisco IOS IP Multicast Command Reference*	Protocol Independent Multicast (PIM) sparse mode (PIM-SM), bidirectional PIM (bidir-PIM), Source Specific Multicast (SSM), Multicast Source Discovery Protocol (MSDP), Internet Group Management Protocol (IGMP), and Multicast VPN (MVPN).

Table 1 Cisco IOS and Cisco IOS XE Configuration Guides and Command References (continued)

Configuration Guide and Command Reference Titles	Features/Protocols/Technologies
Cisco IOS IP Routing Protocols Configuration Guide *Cisco IOS XE IP Routing Protocols Configuration Guide* *Cisco IOS IP Routing Protocols Command Reference*	Border Gateway Protocol (BGP), multiprotocol BGP, multiprotocol BGP extensions for IP multicast, bidirectional forwarding detection (BFD), Enhanced Interior Gateway Routing Protocol (EIGRP), Interior Gateway Routing Protocol (IGRP), Intermediate System-to-Intermediate System (IS-IS), on-demand routing (ODR), Open Shortest Path First (OSPF), and Routing Information Protocol (RIP).
Cisco IOS IP SLAs Configuration Guide *Cisco IOS XE IP SLAs Configuration Guide* *Cisco IOS IP SLAs Command Reference*	Cisco IOS IP Service Level Agreements (IP SLAs).
Cisco IOS IP Switching Configuration Guide *Cisco IOS XE IP Switching Configuration Guide* *Cisco IOS IP Switching Command Reference*	Cisco Express Forwarding, fast switching, and Multicast Distributed Switching (MDS).
Cisco IOS IPv6 Configuration Guide *Cisco IOS XE IPv6 Configuration Guide* *Cisco IOS IPv6 Command Reference*	For IPv6 features, protocols, and technologies, go to the IPv6 "Start Here" document at the following URL: http://www.cisco.com/en/US/docs/ios/ipv6/configuration/guide/ip6-roadmap.html
Cisco IOS ISO CLNS Configuration Guide *Cisco IOS XE ISO CLNS Configuration Guide* *Cisco IOS ISO CLNS Command Reference*	ISO connectionless network service (CLNS).
Cisco IOS LAN Switching Configuration Guide *Cisco IOS XE LAN Switching Configuration Guide* *Cisco IOS LAN Switching Command Reference*	VLANs, Inter-Switch Link (ISL) encapsulation, IEEE 802.10 encapsulation, IEEE 802.1Q encapsulation, and multilayer switching (MLS).
Cisco IOS Mobile Wireless Gateway GPRS Support Node Configuration Guide *Cisco IOS Mobile Wireless Gateway GPRS Support Node Command Reference*	Cisco IOS Gateway GPRS Support Node (GGSN) in a 2.5-generation general packet radio service (GPRS) and 3-generation universal mobile telecommunication system (UMTS) network.
Cisco IOS Mobile Wireless Home Agent Configuration Guide *Cisco IOS Mobile Wireless Home Agent Command Reference*	Cisco Mobile Wireless Home Agent, an anchor point for mobile terminals for which mobile IP or proxy mobile IP services are provided.
Cisco IOS Mobile Wireless Packet Data Serving Node Configuration Guide *Cisco IOS Mobile Wireless Packet Data Serving Node Command Reference*	Cisco Packet Data Serving Node (PDSN), a wireless gateway that is between the mobile infrastructure and standard IP networks and that enables packet data services in a code division multiple access (CDMA) environment.
Cisco IOS Mobile Wireless Radio Access Networking Configuration Guide *Cisco IOS Mobile Wireless Radio Access Networking Command Reference*	Cisco IOS radio access network products.

Table 1 Cisco IOS and Cisco IOS XE Configuration Guides and Command References (continued)

Configuration Guide and Command Reference Titles	Features/Protocols/Technologies
Cisco IOS Multiprotocol Label Switching Configuration Guide *Cisco IOS XE Multiprotocol Label Switching Configuration Guide* *Cisco IOS Multiprotocol Label Switching Command Reference*	MPLS Label Distribution Protocol (LDP), MPLS Layer 2 VPNs, MPLS Layer 3 VPNs, MPLS Traffic Engineering (TE), and MPLS Embedded Management (EM) and MIBs.
Cisco IOS Multi-Topology Routing Configuration Guide *Cisco IOS Multi-Topology Routing Command Reference*	Unicast and multicast topology configurations, traffic classification, routing protocol support, and network management support.
Cisco IOS NetFlow Configuration Guide *Cisco IOS XE NetFlow Configuration Guide* *Cisco IOS NetFlow Command Reference*	Network traffic data analysis, aggregation caches, export features.
Cisco IOS Network Management Configuration Guide *Cisco IOS XE Network Management Configuration Guide* *Cisco IOS Network Management Command Reference*	Basic system management; system monitoring and logging; troubleshooting, logging, and fault management; Cisco Discovery Protocol; Cisco IOS Scripting with Tool Control Language (Tcl); Cisco networking services (CNS); DistributedDirector; Embedded Event Manager (EEM); Embedded Resource Manager (ERM); Embedded Syslog Manager (ESM); HTTP; Remote Monitoring (RMON); SNMP; and VPN Device Manager Client for Cisco IOS Software (XSM Configuration).
Cisco IOS Novell IPX Configuration Guide *Cisco IOS XE Novell IPX Configuration Guide* *Cisco IOS Novell IPX Command Reference*	Novell Internetwork Packet Exchange (IPX) protocol.
Cisco IOS Optimized Edge Routing Configuration Guide *Cisco IOS Optimized Edge Routing Command Reference*	Optimized edge routing (OER) monitoring, policy configuration, routing control, logging and reporting, and VPN IPsec/generic routing encapsulation (GRE) tunnel interface optimization.
Cisco IOS Quality of Service Solutions Configuration Guide *Cisco IOS XE Quality of Service Solutions Configuration Guide* *Cisco IOS Quality of Service Solutions Command Reference*	Class-based weighted fair queuing (CBWFQ), custom queuing, distributed traffic shaping (DTS), generic traffic shaping (GTS), IP- to-ATM class of service (CoS), low latency queuing (LLQ), modular QoS CLI (MQC), Network-Based Application Recognition (NBAR), priority queuing, Security Device Manager (SDM), Multilink PPP (MLPPP) for QoS, header compression, AutoQoS, QoS features for voice, Resource Reservation Protocol (RSVP), weighted fair queuing (WFQ), and weighted random early detection (WRED).
Cisco IOS Security Configuration Guide *Cisco IOS XE Security Configuration Guide* *Cisco IOS Security Command Reference*	Access control lists (ACLs), authentication, authorization, and accounting (AAA), firewalls, IP security and encryption, neighbor router authentication, network access security, network data encryption with router authentication, public key infrastructure (PKI), RADIUS, TACACS+, terminal access security, and traffic filters.

Table 1 Cisco IOS and Cisco IOS XE Configuration Guides and Command References (continued)

Configuration Guide and Command Reference Titles	Features/Protocols/Technologies
Cisco IOS Service Selection Gateway Configuration Guide *Cisco IOS Service Selection Gateway Command Reference*	Subscriber authentication, service access, and accounting.
Cisco IOS Software Activation Configuration Guide *Cisco IOS Software Activation Command Reference*	An orchestrated collection of processes and components to activate Cisco IOS software feature sets by obtaining and validating Cisco software licenses.
Cisco IOS Software Modularity Installation and Configuration Guide *Cisco IOS Software Modularity Command Reference*	Installation and basic configuration of software modularity images, including installations on single and dual route processors, installation rollbacks, software modularity binding, software modularity processes and patches.
Cisco IOS Terminal Services Configuration Guide *Cisco IOS Terminal Services Command Reference* *Cisco IOS XE Terminal Services Command Reference*	DEC, local-area transport (LAT), and X.25 packet assembler/disassembler (PAD).
Cisco IOS Virtual Switch Command Reference	Virtual switch redundancy, high availability, and packet handling; converting between standalone and virtual switch modes; virtual switch link (VSL); Virtual Switch Link Protocol (VSLP). **Note** For information about virtual switch configuration, refer to the product-specific software configuration information for the Cisco Catalyst 6500 series switch or for the Metro Ethernet 6500 series switch.
Cisco IOS Voice Configuration Library *Cisco IOS Voice Command Reference*	Cisco IOS support for voice call control protocols, interoperability, physical and virtual interface management, and troubleshooting. The library includes documentation for IP telephony applications.
Cisco IOS VPDN Configuration Guide *Cisco IOS XE VPDN Configuration Guide* *Cisco IOS VPDN Command Reference*	Layer 2 Tunneling Protocol (L2TP) dial-out load balancing and redundancy, L2TP extended failover, L2TP security VPDN, multihop by Dialed Number Identification Service (DNIS), timer and retry enhancements for L2TP and Layer 2 Forwarding (L2F), RADIUS Attribute 82: tunnel assignment ID, shell-based authentication of VPDN users, tunnel authentication via RADIUS on tunnel terminator.
Cisco IOS Wide-Area Networking Configuration Guide *Cisco IOS XE Wide-Area Networking Configuration Guide* *Cisco IOS Wide-Area Networking Command Reference*	Frame Relay, Layer 2 Tunneling Protocol Version 3 (L2TPv3), Link Access Procedure, Balanced (LAPB), Switched Multimegabit Data Service (SMDS), and X.25.
Cisco IOS Wireless LAN Configuration Guide *Cisco IOS Wireless LAN Command Reference*	Broadcast key rotation, IEEE 802.11x support, IEEE 802.1x authenticator, IEEE 802.1x local authentication service for Extensible Authentication Protocol-Flexible Authentication via Secure Tunneling (EAP-FAST), Multiple Basic Service Set ID (BSSID), Wi-Fi Multimedia (WMM) required elements, and Wi-Fi Protected Access (WPA).

Table 2 Cisco IOS Supplementary Documents and Resources

Document Title	Description
Cisco IOS Master Command List, All Releases	Alphabetical list of all the commands documented in all Cisco IOS releases.
Cisco IOS New, Modified, Removed, and Replaced Commands	List of all the new, modified, removed, and replaced commands for a Cisco IOS release.
Cisco IOS Software System Messages	List of Cisco IOS system messages and descriptions. System messages may indicate problems with your system; be informational only; or may help diagnose problems with communications lines, internal hardware, or the system software.
Cisco IOS Debug Command Reference	Alphabetical list of **debug** commands including brief descriptions of use, command syntax, and usage guidelines.
Release Notes and Caveats	Information about new and changed features, system requirements, and other useful information about specific software releases; information about defects in specific Cisco IOS software releases.
MIBs	Files used for network monitoring. To locate and download MIBs for selected platforms, Cisco IOS releases, and feature sets, use Cisco MIB Locator at the following URL: http://www.cisco.com/go/mibs
RFCs	Standards documents maintained by the Internet Engineering Task Force (IETF) that Cisco IOS documentation references where applicable. The full text of referenced RFCs may be obtained at the following URL: http://www.rfc-editor.org/

Additional Resources and Documentation Feedback

What's New in Cisco Product Documentation is published monthly and describes all new and revised Cisco technical documentation. The *What's New in Cisco Product Documentation* publication also provides information about obtaining the following resources:

- Technical documentation
- Cisco product security overview
- Product alerts and field notices
- Technical assistance

Cisco IOS technical documentation includes embedded feedback forms where you can rate documents and provide suggestions for improvement. Your feedback helps us improve our documentation.

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

Using the Command-Line Interface in Cisco IOS and Cisco IOS XE Software

Last updated: August 6, 2008

This document provides basic information about the command-line interface (CLI) in Cisco IOS and Cisco IOS XE software and how you can use some of the CLI features. This document contains the following sections:

- Initially Configuring a Device, page i
- Using the CLI, page ii
- Saving Changes to a Configuration, page xii
- Additional Information, page xii

For more information about using the CLI, see the "Using the Cisco IOS Command-Line Interface" section of the *Cisco IOS Configuration Fundamentals Configuration Guide*.

For information about the software documentation set, see the "About Cisco IOS and Cisco IOS XE Software Documentation" document.

Initially Configuring a Device

Initially configuring a device varies by platform. For information about performing an initial configuration, see the hardware installation documentation that is provided with the original packaging of the product or go to the Product Support area of Cisco.com at http://www.cisco.com/web/psa/products/index.html.

After you have performed the initial configuration and connected the device to your network, you can configure the device by using the console port or a remote access method, such as Telnet or Secure Shell (SSH), to access the CLI or by using the configuration method provided on the device, such as Security Device Manager.

Changing the Default Settings for a Console or AUX Port

There are only two changes that you can make to a console port and an AUX port:

- Change the port speed with the **config-register 0x** command. Changing the port speed is not recommended. The well-known default speed is 9600.
- Change the behavior of the port; for example, by adding a password or changing the timeout value.

Note The AUX port on the Route Processor (RP) installed in a Cisco ASR1000 series router does not serve any useful customer purpose and should be accessed only under the advisement of a customer support representative.

Using the CLI

This section describes the following topics:

- Understanding Command Modes, page ii
- Using the Interactive Help Feature, page v
- Understanding Command Syntax, page vi
- Understanding Enable and Enable Secret Passwords, page viii
- Using the Command History Feature, page viii
- Abbreviating Commands, page ix
- Using Aliases for CLI Commands, page ix
- Using the no and default Forms of Commands, page x
- Using the debug Command, page x
- Filtering Output Using Output Modifiers, page x
- Understanding CLI Error Messages, page xi

Understanding Command Modes

The CLI command mode structure is hierarchical, and each mode supports a set of specific commands. This section describes the most common of the many modes that exist.

Table 1 lists common command modes with associated CLI prompts, access and exit methods, and a brief description of how each mode is used.

Table 1 CLI Command Modes

Command Mode	Access Method	Prompt	Exit Method	Mode Usage
User EXEC	Log in.	`Router>`	Issue the **logout** or **exit** command.	• Change terminal settings. • Perform basic tests. • Display device status.
Privileged EXEC	From user EXEC mode, issue the **enable** command.	`Router#`	Issue the **disable** command or the **exit** command to return to user EXEC mode.	• Issue **show** and **debug** commands. • Copy images to the device. • Reload the device. • Manage device configuration files. • Manage device file systems.
Global configuration	From privileged EXEC mode, issue the **configure terminal** command.	`Router(config)#`	Issue the **exit** command or the **end** command to return to privileged EXEC mode.	Configure the device.
Interface configuration	From global configuration mode, issue the **interface** command.	`Router(config-if)#`	Issue the **exit** command to return to global configuration mode or the **end** command to return to privileged EXEC mode.	Configure individual interfaces.
Line configuration	From global configuration mode, issue the **line vty** or **line console** command.	`Router(config-line)#`	Issue the **exit** command to return to global configuration mode or the **end** command to return to privileged EXEC mode.	Configure individual terminal lines.

Table 1 CLI Command Modes (continued)

Command Mode	Access Method	Prompt	Exit Method	Mode Usage
ROM monitor	From privileged EXEC mode, issue the **reload** command. Press the **Break** key during the first 60 seconds while the system is booting.	`rommon # >` The # symbol represents the line number and increments at each prompt.	Issue the **continue** command.	• Run as the default operating mode when a valid image cannot be loaded. • Access the fall-back procedure for loading an image when the device lacks a valid image and cannot be booted. • Perform password recovery when a CTRL-Break sequence is issued within 60 seconds of a power-on or reload event.
Diagnostic (available only on the Cisco ASR1000 series router)	The router boots or enters diagnostic mode in the following scenarios. When a Cisco IOS process or processes fail, in most scenarios the router will reload. • A user-configured access policy was configured using the **transport-map** command, which directed the user into diagnostic mode. • The router was accessed using an RP auxiliary port. • A break signal (**Ctrl-C**, **Ctrl-Shift-6**, or the **send break** command) was entered, and the router was configured to enter diagnostic mode when the break signal was received.	`Router(diag)#`	If a Cisco IOS process failure is the reason for entering diagnostic mode, the failure must be resolved and the router must be rebooted to exit diagnostic mode. If the router is in diagnostic mode because of a transport-map configuration, access the router through another port or using a method that is configured to connect to the Cisco IOS CLI. If the RP auxiliary port was used to access the router, use another port for access. Accessing the router through the auxiliary port is not useful for customer purposes.	• Inspect various states on the router, including the Cisco IOS state. • Replace or roll back the configuration. • Provide methods of restarting the Cisco IOS software or other processes. • Reboot hardware, such as the entire router, an RP, an ESP, a SIP, a SPA, or possibly other hardware components. • Transfer files into or off of the router using remote access methods such as FTP, TFTP, and SCP.

EXEC commands are not saved when the software reboots. Commands that you issue in a configuration mode can be saved to the startup configuration. If you save the running configuration to the startup configuration, these commands will execute when the software is rebooted. Global configuration mode is the highest level of configuration mode. From global configuration mode, you can enter a variety of other configuration modes, including protocol-specific modes.

ROM monitor mode is a separate mode that is used when the software cannot load properly. If a valid software image is not found when the software boots or if the configuration file is corrupted at startup, the software might enter ROM monitor mode. Use the question symbol (?) to view the commands that you can use while the device is in ROM monitor mode.

```
rommon 1 > ?
alias                set and display aliases command
boot                 boot up an external process
confreg              configuration register utility
cont                 continue executing a downloaded image
context              display the context of a loaded image
cookie               display contents of cookie PROM in hex
.
.
.
rommon 2 >
```

The following example shows how the command prompt changes to indicate a different command mode:

```
Router> enable
Router# configure terminal
Router(config)# interface ethernet 1/1
Router(config-if)# ethernet
Router(config-line)# exit
Router(config)# end
Router#
```

Note A keyboard alternative to the **end** command is Ctrl-Z.

Using the Interactive Help Feature

The CLI includes an interactive Help feature. Table 2 describes how to use the Help feature.

Table 2 CLI Interactive Help Commands

Command	Purpose
help	Provides a brief description of the help feature in any command mode.
?	Lists all commands available for a particular command mode.
partial command?	Provides a list of commands that begin with the character string (no space between the command and the question mark).
partial command<**Tab**>	Completes a partial command name (no space between the command and <Tab>).
command **?**	Lists the keywords, arguments, or both associated with the command (space between the command and the question mark).
command keyword **?**	Lists the arguments that are associated with the keyword (space between the keyword and the question mark).

The following examples show how to use the help commands:

help

```
Router> help

Help may be requested at any point in a command by entering a question mark '?'.  If
nothing matches, the help list will be empty and you must backup until entering a '?'
shows the available options.

Two styles of help are provided:

1. Full help is available when you are ready to enter a command argument (e.g. 'show ?')
and describes each possible argument.

2. Partial help is provided when an abbreviated argument is entered and you want to know
what arguments match the input (e.g. 'show pr?'.)
```

?

```
Router# ?
Exec commands:
  access-enable        Create a temporary access-List entry
  access-profile       Apply user-profile to interface
  access-template      Create a temporary access-List entry
  alps                 ALPS exec commands
  archive              manage archive files
<snip>
```

partial command?

```
Router(config)# zo?
zone  zone-pair
```

partial command<Tab>

```
Router(config)# we<Tab> webvpn
```

command ?

```
Router(config-if)# pppoe ?
  enable        Enable pppoe
  max-sessions  Maximum PPPOE sessions
```

command keyword ?

```
Router(config-if)# pppoe enable ?
  group  attach a BBA group
  <cr>
```

Understanding Command Syntax

Command syntax is the format in which a command should be entered in the CLI. Commands include the name of the command, keywords, and arguments. Keywords are alphanumeric strings that are used literally. Arguments are placeholders for values that a user must supply. Keywords and arguments may be required or optional.

Specific conventions convey information about syntax and command elements. Table 3 describes these conventions.

Table 3 CLI Syntax Conventions

Symbol/Text	Function	Notes
< > (angle brackets)	Indicate that the option is an argument.	Sometimes arguments are displayed without angle brackets.
A.B.C.D.	Indicates that you must enter a dotted decimal IP address.	Angle brackets (< >) are not always used to indicate that an IP address is an argument.
WORD (all capital letters)	Indicates that you must enter one word.	Angle brackets (< >) are not always used to indicate that a WORD is an argument.
LINE (all capital letters)	Indicates that you must enter more than one word.	Angle brackets (< >) are not always used to indicate that a LINE is an argument.
<cr> (carriage return)	Indicates the end of the list of available keywords and arguments, and also indicates when keywords and arguments are optional. When <cr> is the only option, you have reached the end of the branch or the end of the command if the command has only one branch.	—

The following examples show syntax conventions:

```
Router(config)# ethernet cfm domain ?
  WORD  domain name
Router(config)# ethernet cfm domain dname ?
  level
Router(config)# ethernet cfm domain dname level ?
  <0-7>  maintenance level number
Router(config)# ethernet cfm domain dname level 7 ?
  <cr>
Router(config)# snmp-server file-transfer access-group 10 ?
  protocol  protocol options
  <cr>
Router(config)# logging host ?
  Hostname or A.B.C.D  IP address of the syslog server
  ipv6                 Configure IPv6 syslog server
Router(config)# snmp-server file-transfer access-group 10 ?
  protocol  protocol options
  <cr>
```

Understanding Enable and Enable Secret Passwords

Some privileged EXEC commands are used for actions that impact the system, and it is recommended that you set a password for these commands to prevent unauthorized use. Two types of passwords, enable (not encrypted) and enable secret (encrypted), can be set. The following commands set these passwords and are issued in global configuration mode:

- **enable** *password*
- **enable secret** *password*

Using an enable secret password is recommended because it is encrypted and more secure than the enable password. When you use an enable secret password, text is encrypted (unreadable) before it is written to the config.text file. When you use an enable password, the text is written as entered (readable) to the config.text file.

Each type of password is case sensitive, can contain from 1 to 25 uppercase and lowercase alphanumeric characters, and can start with a number. Spaces are also valid password characters; for example, “two words” is a valid password. Leading spaces are ignored, but trailing spaces are recognized.

Note Both password commands have numeric keywords that are single integer values. If you choose a number for the first character of your password followed by a space, the system will read the number as if it were the numeric keyword and not as part of your password.

When both passwords are set, the enable secret password takes precedence over the enable password.

To remove a password, use the **no** form of the commands: **no enable** *password* or **no enable secret** *password*.

For more information about password recovery procedures for Cisco products, see http://www.cisco.com/en/US/products/sw/iosswrel/ps1831/products_tech_note09186a00801746e6.shtml.

Using the Command History Feature

The CLI command history feature saves the commands you enter during a session in a command history buffer. The default number of commands saved is 10, but the number is configurable within the range of 0 to 256. This command history feature is particularly useful for recalling long or complex commands.

To change the number of commands saved in the history buffer for a terminal session, issue the **terminal history size** command:

```
Router# terminal history size num
```

A command history buffer is also available in line configuration mode with the same default and configuration options. To set the command history buffer size for a terminal session in line configuration mode, issue the **history** command:

```
Router(config-line)# history [size num]
```

To recall commands from the history buffer, use the following methods:

- Press Ctrl-P or the up arrow key—Recalls commands beginning with the most recent command. Repeat the key sequence to recall successively older commands.

- Press Ctrl-N or the down arrow key—Recalls the most recent commands in the history buffer after they have been recalled using Ctrl-P or the up arrow key. Repeat the key sequence to recall successively more recent commands.

Note The arrow keys function only on ANSI-compatible terminals such as the VT100.

- Issue the **show history** command in user EXEC or privileged EXEC mode—Lists the most recent commands that you entered. The number of commands that are displayed is determined by the setting of the **terminal history size** and **history** commands.

  The CLI command history feature is enabled by default. To disable this feature for a terminal session, issue the **terminal no history** command in user EXEC or privileged EXEC mode or the **no history** command in line configuration mode.

Abbreviating Commands

Typing a complete command name is not always required for the command to execute. The CLI recognizes an abbreviated command when the abbreviation contains enough characters to uniquely identify the command. For example, the **show version** command can be abbreviated as **sh ver**. It cannot be abbreviated as **s ver** because **s** could mean **show**, **set**, or **systat**. The **sh v** abbreviation also is not valid because the **show** command has **vrrp** as a keyword in addition to **version**. (Command and keyword examples from Cisco IOS Release 12.4(13)T.)

Using Aliases for CLI Commands

To save time and the repetition of entering the same command multiple times, you can use a command alias. An alias can be configured to do anything that can be done at the command line, but an alias cannot move between modes, type in passwords, or perform any interactive functions.

Table 4 shows the default command aliases.

Table 4 Default Command Aliases

Command Alias	Original Command
h	help
lo	logout
p	ping
s	show
u or **un**	undebug
w	where

To create a command alias, issue the **alias** command in global configuration mode. The syntax of the command is **alias** *mode command-alias original-command*. Following are some examples:

- Router(config)# **alias exec prt partition**—privileged EXEC mode
- Router(config)# **alias configure sb source-bridge**—global configuration mode
- Router(config)# **alias interface rl rate-limit**—interface configuration mode

To view both default and user-created aliases, issue the **show alias** command.

For more information about the **alias** command, see http://www.cisco.com/en/US/docs/ios/fundamentals/command/reference/cf_book.html.

Using the no and default Forms of Commands

Most configuration commands have a **no** form that is used to reset a command to its default value or disable a feature or function. For example, the **ip routing** command is enabled by default. To disable this command, you would issue the **no ip routing** command. To re-enable IP routing, you would issue the **ip routing** command.

Configuration commands may also have a **default** form, which returns the command settings to their default values. For commands that are disabled by default, using the **default** form has the same effect as using the **no** form of the command. For commands that are enabled by default and have default settings, the **default** form enables the command and returns the settings to their default values.

The **no** and **default** forms of commands are described in the command pages of command references.

Using the debug Command

A **debug** command produces extensive output that helps you troubleshoot problems in your network. These commands are available for many features and functions within Cisco IOS and Cisco IOS XE software. Some **debug** commands are **debug all**, **debug aaa accounting**, and **debug mpls packets**. To use **debug** commands during a Telnet session with a device, you must first enter the **terminal monitor** command. To turn off debugging completely, you must enter the **undebug all** command.

For more information about **debug** commands, see the *Cisco IOS Debug Command Reference* at http://www.cisco.com/en/US/docs/ios/debug/command/reference/db_book.html.

Caution Debugging is a high priority and high CPU utilization process that can render your device unusable. Use **debug** commands only to troubleshoot specific problems. The best times to run debugging are during periods of low network traffic and when few users are interacting with the network. Debugging during these periods decreases the likelihood that the **debug** command processing overhead will affect network performance or user access or response times.

Filtering Output Using Output Modifiers

Many commands produce lengthy output that may use several screens to display. Using output modifiers, you can filter this output to show only the information that you want to see.

Three output modifiers are available and are described as follows:

- **begin** *regular expression*—Displays the first line in which a match of the regular expression is found and all lines that follow.
- **include** *regular expression*—Displays all lines in which a match of the regular expression is found.
- **exclude** *regular expression*—Displays all lines except those in which a match of the regular expression is found.

To use one of these output modifiers, type the command followed by the pipe symbol (|), the modifier, and the regular expression that you want to search for or filter. A regular expression is a case-sensitive alphanumeric pattern. It can be a single character or number, a phrase, or a more complex string.

The following example illustrates how to filter output of the **show interface** command to display only lines that include the expression "protocol."

```
Router# show interface | include protocol

FastEthernet0/0 is up, line protocol is up
Serial4/0 is up, line protocol is up
Serial4/1 is up, line protocol is up
Serial4/2 is administratively down, line protocol is down
Serial4/3 is administratively down, line protocol is down
```

Understanding CLI Error Messages

You may encounter some error messages while using the CLI. Table 5 shows the common CLI error messages.

Table 5 Common CLI Error Messages

Error Message	Meaning	How to Get Help
% Ambiguous command: "show con"	You did not enter enough characters for the command to be recognized.	Reenter the command followed by a space and a question mark (?). The keywords that you are allowed to enter for the command appear.
% Incomplete command.	You did not enter all the keywords or values required by the command.	Reenter the command followed by a space and a question mark (?). The keywords that you are allowed to enter for the command appear.
% Invalid input detected at "^" marker.	You entered the command incorrectly. The caret (^) marks the point of the error.	Enter a question mark (?) to display all the commands that are available in this command mode. The keywords that you are allowed to enter for the command appear.

For more system error messages, see the following documents:

- *Cisco IOS Release 12.2SR System Message Guide*
- *Cisco IOS System Messages, Volume 1 of 2* (Cisco IOS Release 12.4)
- *Cisco IOS System Messages, Volume 2 of 2* (Cisco IOS Release 12.4)

Saving Changes to a Configuration

To save changes that you made to the configuration of a device, you must issue the **copy running-config startup-config** command or the **copy system:running-config nvram:startup-config** command. When you issue these commands, the configuration changes that you made are saved to the startup configuration and saved when the software reloads or power to the device is turned off or interrupted. The following example shows the syntax of the **copy running-config startup-config** command:

```
Router# copy running-config startup-config
Destination filename [startup-config]?
```

You press Enter to accept the startup-config filename (the default), or type a new filename and then press Enter to accept that name. The following output is displayed indicating that the configuration was saved:

```
Building configuration...
[OK]
Router#
```

On most platforms, the configuration is saved to NVRAM. On platforms with a Class A flash file system, the configuration is saved to the location specified by the CONFIG_FILE environment variable. The CONFIG_FILE variable defaults to NVRAM.

Additional Information

- "Using the Cisco IOS Command-Line Interface" section of the *Cisco IOS Configuration Fundamentals Configuration Guide*:

  http://www.cisco.com/en/US/docs/ios/fundamentals/configuration/guide/cf_cli-basics.html

  or

  "Using Cisco IOS XE Software" chapter of the *Cisco ASR1000 Series Aggregation Services Routers Software Configuration Guide*:

  http://www.cisco.com/en/US/docs/routers/asr1000/configuration/guide/chassis/using_cli.html

- Cisco Product Support Resources

  http://www.cisco.com/web/psa/products/index.html

- Support area on Cisco.com (also search for documentation by task or product)

  http://www.cisco.com/en/US/support/index.html

- *White Paper: Cisco IOS Reference Guide*

  http://www.cisco.com/en/US/products/sw/iosswrel/ps1828/products_white_paper09186a008018305e.shtml

- Software Download Center (downloads; tools; licensing, registration, advisory, and general information) (requires Cisco.com User ID and password)

  http://www.cisco.com/kobayashi/sw-center/

- Error Message Decoder, a tool to help you research and resolve error messages for Cisco IOS software

  http://www.cisco.com/pcgi-bin/Support/Errordecoder/index.cgi

- Command Lookup Tool, a tool to help you find detailed descriptions of Cisco IOS commands (requires Cisco.com user ID and password)

  http://tools.cisco.com/Support/CLILookup

- Output Interpreter, a troubleshooting tool that analyzes command output of supported **show** commands

  https://www.cisco.com/pcgi-bin/Support/OutputInterpreter/home.pl\

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

Performing Basic System Management

This chapter describes the basic tasks that you can perform to manage the general system features of the Cisco IOS software—those features that are generally not specific to a particular protocol.

This document applies to Cisco IOS Release 12.2.

For a complete description of the basic system management commands in this chapter, refer to the "Basic System Management Commands" chapter in the "Cisco IOS System Management Commands" part of the Release 12.2 *Cisco IOS Configuration Fundamentals Command Reference*. To locate documentation of other commands that appear in this chapter, use the *Cisco IOS Command Reference Master Index* or search online.

To identify hardware or software image support for a specific feature, use Feature Navigator on Cisco.com or refer to the software release notes for a specific release. For more information, see the "Identifying Platform Support for Cisco IOS Software Features" section in the "About Cisco IOS Software Documentation" chapter.

Basic System Management Task List

To customize the general functionality of your system, perform any of the tasks in the following sections. All tasks in this chapter are optional, though some, such as setting time and calendar services, are highly recommended.

- Configuring the System Name (Recommended)
- Customizing the CLI Prompt
- Creating and Displaying Command Aliases
- Controlling Minor Services (Recommended)
- Hiding Telnet Addresses
- Setting Time and Calendar Services (Recommended)
- Delaying EXEC Startup
- Handling an Idle Telnet Connection
- Setting the Interval for Load Data
- Limiting the Number of TCP Transactions

Americas Headquarters:
Cisco Systems, Inc., 170 West Tasman Drive, San Jose, CA 95134-1706 USA

- Configuring Switching and Scheduling Priorities
- Modifying the System Buffer Size

See the end of this chapter for the "Basic System Management Examples" section.

Configuring the System Name

The most basic system management task is to assign a name to your system (router, access server, switch, and so on). The system name, also called the host name, is used to uniquely identify the system in your network. The system name is displayed at the CLI prompt. If no name is configured, the system default name is `Router`. To configure a name for your device, use the following command in global configuration mode:

Command	Purpose
`Router(config)#` **`hostname`** *`name`*	Sets the host name.

For an example of configuring a system name, see the section "System Configuration File Example" at the end of this chapter.

Customizing the CLI Prompt

By default, the CLI prompt consists of the system name followed by an angle bracket (>) for EXEC mode or a pound sign (#) for privileged EXEC mode. To customize the CLI prompt for your system, use either of the following commands in global configuration mode, as needed:

Command	Purpose
`Router(config)#` **`prompt`** *`string`*	Customizes the CLI prompt.
`Router(config)#` **`no service prompt config`**	Disables the display of the CLI prompt.

Creating and Displaying Command Aliases

Command aliases allow you to configure alternative syntax for commands. You may want to create aliases for commonly used or complex commands. For example, you could assign the alias **save config** to the **copy running-config startup-config** command to reduce the amount of typing you have to perform, or if your users might find a **save config** command easier to remember. Use word substitutions or abbreviations to tailor command syntax for you and your user community.

To create a command alias, use the following command in global configuration mode:

Command	Purpose
`Router(config)#` **`alias`** *`mode alias-name alias-command-line`*	Configures a command alias.

To display a list of command aliases currently configured on your system, and the original command syntax for those aliases, use the following command in EXEC mode:

Command	Purpose
`Router# show aliases [mode]`	Displays all command aliases and original command syntax, or displays the aliases for only a specified command mode.

Keep in mind that any aliases you configure will only be effective on your system, and that the original command syntax will appear in the configuration file.

Controlling Minor Services

The minor services are “small servers” that run on your routing device and are useful for basic system testing and for providing basic network functions. Minor services are useful for testing connections from another host on the network.

Cisco small servers are conceptually equivalent to daemons.

Small servers provided by Cisco IOS software-based devices include TCP, UDP, HTTP, BOOTP, and Finger. For information about the HTTP server, see the“Using the Cisco Web Browser User Interface” chapter in this book.

The TCP small server provides the following minor services:

- Echo—Echoes back whatever you type. To test this service, issue the **telnet** *a.b.c.d* **echo** command from a remote host.
- Chargen—Generates a stream of ASCII data. To test this service, issue the **telnet** *a.b.c.d* **chargen** command from a remote host.
- Discard—Discards whatever you type. To test this service, issue the **telnet** *a.b.c.d* **discard** command from a remote host.
- Daytime—Returns system date and time if you have configured NTP or have set the date and time manually. To test this service, issue the **telnet** *a.b.c.d* **daytime** command from a remote host.

The User Datagram Protocol (UDP) small server provides the following minor services:

- Echo—Echoes the payload of the datagram you send.
- Chargen—Discards the datagram you send and responds with a 72 character string of ASCII characters terminated with a CR+LF (carriage return and line feed).
- Discard—Silently discards the datagram you send.

To enable TCP or UDP services, use the following commands in global configuration mode, as needed:

Command	Purpose
`Router(config)# service tcp-small-servers`	Enables the minor TCP services echo, chargen, discard, and daytime.
`Router(config)# service udp-small-servers`	Enables the minor UDP services echo, chargen, and discard.

Because the minor services can be misused, these commands are disabled by default.

Caution Enabling minor services creates the potential for certain types of denial-of-service attacks, such as the UDP diagnostic port attack. Therefore, any network device that has UDP, TCP, BOOTP, or Finger services should be protected by a firewall or have the services disabled. For information on preventing UDP diagnostic port attacks, see the white paper titled *Defining Strategies to Protect Against UDP Diagnostic Port Denial of Service Attacks*, available on Cisco.com.

Note that the **no** form of the **service tcp-small-servers** and **service udp-small-servers** commands will appear in the configuration file to inform you when these basic services are disabled.

Controlling the BOOTP Server

You can enable or disable an async line Bootstrap Protocol (BOOTP) service on your routing device. This small server is enabled by default. Due to security considerations, this service should be disabled if you are not using it. To disable the BOOTP server on your platform, use the following command in global configuration mode:

Command	Purpose
`Router(config)# no ip bootp server`	Disables the BOOTP server.

Because Dynamic Host Configuration Protocol (DHCP) is based on the Bootstrap Protocol, both of these service share the "well-known" UDP server port of 67 (per the internet standards and RFCs). For more information about DHCP configuration in Cisco IOS software, see the *Cisco IOS IP Configuration Guide*. For more information about BOOTP, see RFC 951. Interoperation between BOOTP and DHCP is defined in RFC 1534. DHCP is defined in RFC 2131.

Controlling the Finger Protocol

The Finger protocol allows users throughout the network to get a list of the users currently using a particular routing device. The information displayed includes the processes running on the system, the line number, connection name, idle time, and terminal location. This information is provided through the Cisco IOS software **show users** EXEC command.

To enable a Cisco device to respond to Finger (port 79) requests, use the following command in global configuration mode:

Command	Purpose
`Router(config)# ip finger`	Enables the Finger protocol service, which allows the system to respond to finger requests.

To configure the finger protocol to be compliant with RFC 1288, use the following command in global configuration mode:

Command	Purpose
`Router(config)# ip finger rfc-compliant`	Configures the device to wait for “Return” or “/W” input when processing Finger requests.

The **rfc-compliant** form of this command should not be configured for devices with more than 20 simultaneous users (see caveat CSCds92731 on Cisco.com for details). The difference between the two forms of this command is as follows: when the **ip finger** command is configured, the router will respond to a **telnet** *a.b.c.d* **finger** command from a remote host by immediately displaying the output of the **show users** command and then closing the connection. When the **ip finger rfc-compliant** command is configured, the router will wait for input before displaying anything. The remote user can then press the Return key to display the output of the **show users** command, or enter **/W** to display the output of the **show users wide** command. After this information is displayed, the connection is closed.

Hiding Telnet Addresses

You can hide addresses while attempting to establish a Telnet session. To configure the router to suppress Telnet addresses, use the following command in global configuration mode:

Command	Purpose
`Router(config)# service hide-telnet-address`	Hides addresses while establishing a Telnet session.

The hide feature suppresses the display of the address and continues to display all other messages that normally would be displayed during a connection attempt, such as detailed error messages if the connection failed.

Use the **busy-message** line configuration command with the **service hide-telnet-address** command to customize the information displayed during Telnet connection attempts. If the connection attempt fails, the router suppresses the address and displays the message specified with the **busy-message** command.

Setting Time and Calendar Services

All Cisco routers provide an array of time-of-day services. These services allow the products to accurately keep track of the current time and date, to synchronize multiple devices to the same time, and to provide time services to other systems. The following sections describe the concepts and task associated with time and calendar services:

- Understanding Time Sources
- Configuring NTP
- Configuring SNTP
- Configuring VINES Time Service
- Configuring Time and Date Manually
- Using the Hardware Clock
- Monitoring Time and Calendar Services
- Configuring Time Ranges

Understanding Time Sources

Most Cisco routers have two clocks: a battery-powered hardware clock (referenced in CLI commands as the “calendar”) and a software clock (referenced in CLI commands as the “clock”). These two clocks are managed separately.

The primary source for time data on your system is the software clock. This clock runs from the moment the system starts up and keeps track of the current date and time. The software clock can be set from a number of sources and in turn can be used to distribute the current time through various mechanisms to other systems. When a router with a hardware clock is initialized or rebooted, the software clock is initially set based on the time in the hardware clock. The software clock can then be updated from the following sources:

- Network Time Protocol (NTP)
- Simple Network Time Protocol (SNTP)
- VINES Time Service
- Manual configuration (using the hardware clock)

Because the software clock can be dynamically updated it has the potential to be more accurate than the hardware clock.

The software clock can provide time to the following services:

- Access lists
- NTP
- VINES time service
- User **show** commands
- Logging and debugging messages
- The hardware clock

Note The software clock cannot provide time to the NTP or VINES Time Service if it was set using SNTP.

The software clock keeps track of time internally based on Coordinated Universal Time (UTC), also known as Greenwich Mean Time (GMT). You can configure information about the local time zone and summer time (daylight savings time) so that the time is displayed correctly relative to the local time zone.

The software clock keeps track of whether the time is “authoritative” (that is, whether it has been set by a time source considered to be authoritative). If it is not authoritative, the time will be available only for display purposes and will not be redistributed.

Network Time Protocol

The Network Time Protocol (NTP) is a protocol designed to time-synchronize a network of machines. NTP runs over UDP, which in turn runs over IP. NTP Version 3 is documented in RFC 1305.

An NTP network usually gets its time from an authoritative time source, such as a radio clock or an atomic clock attached to a time server. NTP then distributes this time across the network. NTP is extremely efficient; no more than one packet per minute is necessary to synchronize two machines to the accuracy of within a millisecond of one another.

NTP uses the concept of a “stratum” to describe how many NTP “hops” away a machine is from an authoritative time source. A “stratum 1” time server typically has an authoritative time source (such as a radio or atomic clock, or a GPS time source) directly attached, a “stratum 2” time server receives its time via NTP from a “stratum 1” time server, and so on.

NTP avoids synchronizing to a machine whose time may not be accurate in two ways. First, NTP will never synchronize to a machine that is not in turn synchronized itself. Second, NTP will compare the time reported by several machines, and will not synchronize to a machine whose time is significantly different than the others, even if its stratum is lower. This strategy effectively builds a self-organizing tree of NTP servers.

The Cisco implementation of NTP does not support stratum 1 service; in other words, it is not possible to connect to a radio or atomic clock (for some specific platforms, however, you can connect a GPS time-source device). We recommend that time service for your network be derived from the public NTP servers available in the IP internet.

If the network is isolated from the internet, the Cisco implementation of NTP allows a machine to be configured so that it acts as though it is synchronized via NTP, when in fact it has determined the time using other means. Other machines can then synchronize to that machine via NTP.

A number of manufacturers include NTP software for their host systems, and a publicly available version for systems running UNIX and its various derivatives is also available. This software also allows UNIX-derivative servers to acquire the time directly from an atomic clock which would subsequently propagate time information along to Cisco routers.

The communications between machines running NTP (known as “associations”) are usually statically configured; each machine is given the IP address of all machines with which it should form associations. Accurate timekeeping is made possible by exchanging NTP messages between each pair of machines with an association.

However, in a LAN environment, NTP can be configured to use IP broadcast messages instead. This alternative reduces configuration complexity because each machine can simply be configured to send or receive broadcast messages. However, the accuracy of timekeeping is marginally reduced because the information flow is one-way only.

The time kept on a machine is a critical resource, so we strongly recommend that you use the security features of NTP to avoid the accidental or malicious setting of incorrect time. Two mechanisms are available: an access list-based restriction scheme and an encrypted authentication mechanism.

When multiple sources of time (VINES, hardware clock, manual configuration) are available, NTP is always considered to be more authoritative. NTP time overrides the time set by any other method.

Simple Network Time Protocol

Simple Network Time Protocol (SNTP) is a simplified, client-only version of NTP for use on Cisco 1003, Cisco 1004, Cisco 1005, Cisco 1600, Cisco 1720, and Cisco 1750 routers. SNTP can receive only the time from NTP servers; it cannot be used to provide time services to other systems.

SNTP typically provides time within 100 milliseconds of the accurate time, but it does not provide the complex filtering and statistical mechanisms of NTP. In addition, SNTP does not authenticate traffic, although you can configure extended access lists to provide some protection. An SNTP client is more vulnerable to misbehaving servers than an NTP client and should be used only in situations where strong authentication is not required.

You can configure SNTP to request and accept packets from configured servers or to accept NTP broadcast packets from any source. When multiple sources are sending NTP packets, the server with the best stratum is selected. (See the “Network Time Protocol” section for a description of strata.) If multiple servers are at the same stratum, a configured server is preferred over a broadcast server. If multiple servers pass both tests, the first one to send a time packet is selected. SNTP will choose a new server only if it stops receiving packets from the currently selected server, or if a better server (according to the above criteria) is discovered.

VINES Time Service

Time service is available when Banyan VINES is configured. This protocol is a standard part of VINES. The Cisco implementation allows the VINES time service to be used in two ways. First, if the system has learned the time from some other source, it can act as a VINES time server and provide time to other machines running VINES. Second, it can use the VINES time service to set the software clock if no other form of time service is available.

Note Support for Banyan VINES and XNS is removed from Cisco IOS software in Cisco IOS Release 12.2(13)T and later.

Hardware Clock

Some routers contain a battery-powered hardware clock that tracks the date and time across system restarts and power outages. The hardware clock is always used to initialize the software clock when the system is restarted.

Note Within the CLI command syntax, the hardware clock is referred to as the “system calendar.”

If no other source is available, the hardware clock can be considered to be an authoritative source of time and be redistributed via NTP or VINES time service. If NTP is running, the hardware clock can be updated periodically from NTP, compensating for the inherent drift in the hardware clock.

Configuring NTP

NTP services are disabled on all interfaces by default. The following sections contain optional tasks that you can perform on your networking device:

- Configuring Poll-Based NTP Associations
- Configuring Broadcast-Based NTP Associations
- Configuring an NTP Access Group
- Configuring NTP Authentication
- Disabling NTP Services on a Specific Interface
- Configuring the Source IP Address for NTP Packets
- Configuring the System as an Authoritative NTP Server
- Updating the Hardware Clock
- Configuring an External Reference Clock

Configuring Poll-Based NTP Associations

Networking devices running NTP can be configured to operate in variety of association modes when synchronizing time with reference time sources. There are two ways that a networking device can obtain time information on a network: by polling host servers and by listening to NTP broadcasts. In this section, we will focus on the poll-based association modes. Broadcast-based NTP associations will be discussed in the next section.

The following are two most commonly used, poll-based association modes:

- Client mode
- Symmetric active mode

The *client* and the *symmetric active* modes should be used when NTP is required to provide a high level of time accuracy and reliability.

When a networking device is operating in the *client mode*, it polls its assigned time serving hosts for the current time. The networking device will then pick a host from all the polled time servers to synchronize with. Since the relationship that is established in this case is a client-host relationship, the host will not capture or use any time information sent by the local client device. This mode is most suited for file-server and workstation clients that are not required to provide any form of time synchronization to other local clients. Use the **ntp server** command to individually specify the time serving hosts that you want your networking device to consider synchronizing with and to set your networking device to operate in the *client mode*.

When a networking device is operating in the *symmetric active mode*, it polls its assigned time serving hosts for the current time and it responds to polls by its hosts. Since this is a peer-to-peer relationship, the host will also retain time-related information about the local networking device that it is communicating with. This mode should be used when there is a number of mutually redundant servers that are interconnected via diverse network paths. Most Stratum 1 and stratum 2 servers on the Internet today adopt this form of network setup. Use the **ntp peer** command to individually specify the time serving hosts that you want your networking device to consider synchronizing with and to set your networking device to operate in the *symmetric active mode*.

The specific mode that you should set each of your networking devices to depends primarily on the role that you want it to assume as a timekeeping device (server or client) and its proximity to a stratum 1 timekeeping server.

A networking device engages in polling when it is operating as a client or a host in the *client mode* or when it is acting as a peer in the *symmetric active mode*. Although polling does not usually exact a toll on memory and CPU resources such as bandwidth, an exceedingly large number of ongoing and simultaneous polls on a system can seriously impact the performance of a system or slow the performance of a given network. To avoid having an excessive number of ongoing polls on a network, you should limit the number of direct, peer-to-peer or client-to-server associations. Instead, you should consider using NTP broadcasts to propagate time information within a localized network.

Command	Purpose
`Router(config)#` **`ntp peer`** *`ip-address`* `[`**`normal-sync`**`] [`**`version`** *`number`*`] [`**`key`** *`keyid`*`] [`**`source`** *`interface`*`] [`**`prefer`**`]`	Forms a peer association with another system.
`Router(config)#` **`ntp server`** *`ip-address`* `[`**`version`** *`number`*`] [`**`key`** *`keyid`*`] [`**`source`** *`interface`*`] [`**`prefer`**`]`	Forms a server association with another system.

Note that only one end of an association needs to be configured; the other system will automatically establish the association.

Caution The **ntp clock-period** command is automatically generated to reflect the constantly changing *correction factor* when the **copy running-configuration startup-configuration** command is entered to save the configuration to NVRAM. Do not attempt to manually use the **ntp clock-period** command. Ensure that you remove this command line when copying configuration files to other devices.

For an example of configuring an NTP server-peer relationship, see the "Clock, Calendar, and NTP Configuration Examples" section at the end of this chapter.

Configuring Broadcast-Based NTP Associations

Broadcast-based NTP associations should be used when time accuracy and reliability requirements are modest and if your network is localized and has a large number of clients (more than 20). Broadcast-based NTP associations is also recommended for use on networks that have limited bandwidth, system memory, or CPU resources.

When a networking device is operating in the *broadcastclient mode*, it does not engage in any polling. Instead, it listens for NTP broadcast packets transmitted by broadcast time servers. Consequently, time accuracy can be marginally reduced since time information flows only one way.

Use the **ntp broadcast client** command to set your networking device to listen for NTP broadcast packets propagated through a network. In order for *broadcastclient mode* to work, the broadcast server and its clients must be located on the same subnet. The time server that is transmitting NTP broadcast packets will also have to be enabled on the interface of the given device using the **ntp broadcast** command.

To configure an interface to send NTP broadcasts, use the following command in interface configuration mode:

Command	Purpose
`Router(config-if)# ntp broadcast [version number]`	Configures the specified interface to send NTP broadcast packets.

To configure an interface to receive NTP broadcasts, use the following command in interface configuration mode:

Command	Purpose
`Router(config-if)# ntp broadcast client`	Configures the specified interface to receive NTP broadcast packets.

To manually set the estimated round-trip delay between the device and the NTP broadcast server, use the following command in global configuration mode:

Command	Purpose
`Router(config)# ntp broadcastdelay microseconds`	Adjusts the estimated round-trip delay for NTP broadcasts.

Caution The **ntp clock-period** command is automatically generated to reflect the constantly changing *correction factor* when the **copy running-configuration startup-configuration** command is entered to save the configuration to NVRAM. Do not attempt to manually use the **ntp clock-period** command. Ensure that you remove this command line when copying configuration files to other devices.

For an example of configuring broadcast-based NTP associations, see the "Clock, Calendar, and NTP Configuration Examples" section at the end of this chapter.

Configuring an NTP Access Group

The access list-based restriction scheme allows you to grant or deny certain access privileges to an entire network, a subnet within a network, or a host within a subnet. To define an NTP access group, use the following command in global configuration mode:

Command	Purpose
`Router(config)# ntp access-group {query-only	serve-only	serve	peer} access-list-number`	Creates an access group and applies a basic IP access list to it.

The access group options are scanned in the following order, from least restrictive to most restrictive:

1. **peer**—Allows time requests and NTP control queries and allows the system to synchronize itself to a system whose address passes the access list criteria.
2. **serve**—Allows time requests and NTP control queries, but does not allow the system to synchronize itself to a system whose address passes the access list criteria.
3. **serve-only**—Allows only time requests from a system whose address passes the access list criteria.
4. **query-only**—Allows only NTP control queries from a system whose address passes the access list criteria.

If the source IP address matches the access lists for more than one access type, the first type is granted. If no access groups are specified, all access types are granted to all systems. If any access groups are specified, only the specified access types will be granted.

For details on NTP control queries, see RFC 1305 (NTP version 3).

Configuring NTP Authentication

The encrypted NTP authentication scheme should be used when a reliable form of access control is required. Unlike the access list-based restriction scheme which is based on IP addresses, the encrypted authentication scheme uses authentication keys and an authentication process to determine if NTP synchronization packets sent by designated peers or servers on a local network are deemed as trusted before the time information that it carries along with it, is accepted.

The authentication process begins from the moment an NTP packet is created. Cryptographic checksum keys are generated using the MD5 Message Digest Algorithm and are embedded into the NTP synchronization packet that is sent to a receiving client. Once a packet is received by a client, its cryptographic checksum key is decrypted and checked against a list of trusted keys. If the packet contains a matching authenticator key, the timestamp information that is contained within it is accepted by the receiving client. NTP synchronization packets that do not contain a matching authenticator key will be ignored.

It is important to note that the encryption and decryption processes used in NTP authentication can be very CPU-intensive and can seriously degrade the accuracy of the time that is propagated within a network. If your network setup permits a more comprehensive model of access control, you should consider the use of the access list-based form of control instead.

After NTP authentication is properly configured, your networking device will only synchronize with and provide synchronization to trusted time sources. To enable your networking device to send and receive encrypted synchronization packets, use the following commands in global configuration mode:

	Command	Purpose
Step 1	`Router(config)# ntp authenticate`	Enables the NTP authentication feature.
Step 2	`Router(config)# ntp authentication-key number md5 value`	Defines the authentication keys. Each key has a key number, a type, and a value. Currently the only key type supported is **md5**.
Step 3	`Router(config)# ntp trusted-key key-number`	Defines trusted authentication keys. If a key is trusted, this system will be ready to synchronize to a system that uses this key in its NTP packets.

Note In Cisco IOS software versions previous to release 12.0, the cryptotype value is displayed along with the ntp authentication key md5 value when the **show running-configuration** command is entered. Avoid copying and pasting the string cryptotype value that is displayed with the authentication-key as it will result in authentication failure.

Disabling NTP Services on a Specific Interface

NTP services are disabled on all interfaces by default.

NTP is enabled globally when any NTP commands are entered. you can selectively prevent NTP packets from being received through a specific interface by using the following command in interface configuration mode to turn off NTP on a given interface:

Command	Purpose
`Router(config-if)# ntp disable`	Disables NTP services on a specific interface.

Configuring the Source IP Address for NTP Packets

When the system sends an NTP packet, the source IP address is normally set to the address of the interface through which the NTP packet is sent. Use the following command in global configuration mode if you want to configure a specific interface from which the IP source address will be taken:

Command	Purpose
`Router(config)# ntp source interface`	Configures an interface from which the IP source address will be taken.

This interface will be used for the source address for all packets sent to all destinations. If a source address is to be used for a specific association, use the **source** parameter on the **ntp peer** or **ntp server** command shown earlier in this chapter.

Configuring the System as an Authoritative NTP Server

Use the following command in global configuration mode if you want the system to be an authoritative NTP server, even if the system is not synchronized to an outside time source:

Command	Purpose
`Router(config)# ntp master [stratum]`	Makes the system an authoritative NTP server.

Note Use the **ntp master** command with caution. It is very easy to override valid time sources using this command, especially if a low stratum number is configured. Configuring multiple machines in the same network with the **ntp master** command can cause instability in timekeeping if the machines do not agree on the time.

For an example of configuring an authoritative NTP server, see the "Clock, Calendar, and NTP Configuration Examples" section at the end of this chapter.

Updating the Hardware Clock

On devices that have hardware clocks (system calendars), you can configure the hardware clock to be periodically updated from the software clock. This is advisable for any device using NTP, because the time and date on the software clock (set using NTP) will be more accurate than the hardware clock, because the time setting on the hardware clock has the potential to drift slightly over time.

Use the following command in global configuration mode if a routing device is synchronized to an outside time source via NTP and you want the hardware clock to be synchronized to NTP time:

Command	Purpose
`Router(config)# ntp update-calendar`	Configures the system to update its hardware clock from the software clock at periodic intervals.

For an example of configuring NTP to update the calendar, see the section "Clock, Calendar, and NTP Configuration Examples" at the end of this chapter.

Configuring an External Reference Clock

Because Cisco's implementation of NTP does not support stratum 1 service, it is not possible to connect to a radio or atomic clock (for some specific platforms however, you can connect a GPS timesource device). However, certain Cisco devices allow you to connect a external GPS-based time-source device for the purposes of distributing a time signal to your network using NTP.

For example, the Trimble Palisade NTP Synchronization Kit can be connected to the auxiliary port of a Cisco 7200 Series router. Also, selected platforms support the use of GPS clocks from Symmetricom (formerly Telecom-Solutions). The refclock (reference clock) drivers provided on these platforms provides the ability to receive an RTS time-stamp signal on the auxiliary port of your routing device.

To configure a Trimble Palisade GPS product connected to the auxiliary port of a Cisco 7200 series router as the NTP reference clock, use the following commands, beginning in global configuration mode:

	Command	Purpose
Step 1	Router(config)# **line aux 0**	Enters line configuration mode for the auxiliary port 0.
Step 2	Router(config-line)# **ntp refclock trimble pps none stratum 1**	Enables the driver that allows the Trimble Palisade NTP Synchronization Kit to be used as the NTP reference clock source (Cisco 7200 series routers only).

To configure a Symmetricom GPS product connected to the auxiliary port of a supported router or switch as the NTP reference clock, use the following commands beginning in global configuration mode:

	Command	Purpose
Step 1	Router(config)# **line aux 0**	Enters line configuration mode for the auxiliary port zero.
Step 2	Router(config-line)# **ntp refclock telecom-solutions pps cts stratum 1**	Enables the driver that allows the Symmetricom GPS product to be used as the NTP reference clock source.

To configure a PPS signal as the source for NTP synchronization, use the following form of the **ntp refclock** command in line configuration mode:

Command	Purpose
Router(config-line)# **ntp refclock pps** {**cts** | **ri**} [**inverted**] [**pps-offset** *number*] [**stratum** *number*] [**timestamp-offset** *number*]	Configures a PPS signal as the source for NTP synchronization.

Verifying the Status of the External Reference Clock

To verify the status of NTP components, use the following commands in privileged EXEC mode:

Command	Purpose
Router# **show ntp associations**	Displays the status of NTP associations, including the status of the GPS reference clock.
Router# **show ntp status**	Displays the status of NTP.
Router# **debug ntp refclock**	Allows advanced monitoring of reference clock activities for the purposes of debugging.

Configuring SNTP

SNTP generally is supported on those platforms that do not provide support for NTP, such as the Cisco 1000 series, 1600 series, and 1700 series platforms. SNTP is disabled by default. In order to enable SNTP, use one or both of the following commands in global configuration mode:

Command	Purpose
`Router(config)# sntp server {address | hostname} [version number]`	Configures SNTP to request NTP packets from an NTP server.
`Router(config)# sntp broadcast client`	Configures SNTP to accept NTP packets from any NTP broadcast server.

Enter the **sntp server** command once for each NTP server. The NTP servers must be configured to respond to the SNTP messages from the router.

If you enter both the **sntp server** command and the **sntp broadcast client** command, the router will accept time from a broadcast server but prefer time from a configured server, assuming that the strata are equal. To display information about SNTP, use the **show sntp** EXEC command.

Configuring VINES Time Service

Note Support for Banyan VINES and XNS has been removed from Cisco IOS software, beginning in Cisco IOS Release 12.2(13)T. The following VINES commands are not available in releases derived from 12.2(13)T, such as the 12.3 mainline release.

To distribute the system time and date to other devices on the network using VINES time services, use the following command in global configuration mode:

Command	Purpose
`Router(config)# vines time use-system`	Distributes the system software clock time to other VINES systems.

To set the system time and date from received VINES time services, use the following command in global configuration mode:

Command	Purpose
`Router(config)# vines time set-system`	Sets the software clock system time from received VINES time services.

Configuring Time and Date Manually

If no other source of time is available, you can manually configure the current time and date after the system is restarted. The time will remain accurate until the next system restart. We recommend that you use manual configuration only as a last resort.

To set up time services, complete the tasks in the following sections as needed. If you have an outside source to which the router can synchronize, you do not need to manually set the software clock.

- Configuring the Time Zone
- Configuring Summer Time (Daylight Savings Time)
- Manually Setting the Software Clock
- Using the Hardware Clock

Configuring the Time Zone

To manually configure the time zone used by the Cisco IOS software, use the following command in global configuration mode :

Command	Purpose
`Router(config)# clock timezone zone hours-offset [minutes-offset]`	Sets the time zone. The *zone* argument is the name of the time zone (typically a standard acronym). The *hours-offset* argument is the number of hours the time zone is different from UTC. The *minutes-offset* argument is the number of minutes the time zone is different from UTC.

Tip The *minutes-offset* argument of the **clock timezone** command is available for those cases where a local time zone is a percentage of an hour different from UTC/GMT. For example, the time zone for some sections of Atlantic Canada (AST) is UTC -3.5. In this case, the necessary command would be **clock timezone AST -3 30**.

For an example of configuring the time zone, see the section "Clock, Calendar, and NTP Configuration Examples" at the end of this chapter.

Configuring Summer Time (Daylight Savings Time)

To configure summer time (daylight savings time) in areas where it starts and ends on a particular day of the week each year, use the following command in global configuration mode:

Command	Purpose
`Router(config)# clock summer-time zone recurring [week day month hh:mm week day month hh:mm [offset]]`	Configures a recurring summer time start and end date. The *offset* argument is used to indicate the number of minutes to add to the clock during summer time.

If summer time in your area does not follow this pattern, you can configure the exact date and time of the next summer time event by using one of the following commands in global configuration mode:

Command	Purpose

`Router(config)# **clock summer-time** *zone* **date** *month date year hh***:***mm month date year hh***:***mm* [*offset*]`
or
`Router(config)# **clock summer-time** *zone* **date** *date month year hh***:***mm date month year hh***:***mm* [*offset*]`
	Configures a specific summer time start and end date. The *offset* argument is used to indicate the number of minutes to add to the clock during summer time.

For an example of configuring summer time, see the section "Clock, Calendar, and NTP Configuration Examples" at the end of this chapter.

Manually Setting the Software Clock

Generally, if the system is synchronized by a valid outside timing mechanism, such as an NTP or VINES clock source, or if you have a router with a hardware clock, you need not set the software clock. Use this command if no other time sources are available. The time specified in this command is relative to the configured time zone. To set the software clock manually, use the following command in privileged EXEC mode:

Command	Purpose
Router# **clock set** *hh***:***mm***:***ss date month year* or Router# **clock set** *hh***:***mm***:***ss month date year*	Sets the software clock.

Using the Hardware Clock

Most Cisco devices have a separate hardware-based clock in addition to the software-based clock. The hardware clock is a chip with a rechargeable backup battery that can retain the time and date information across reboots of the device.

To maintain the most accurate time update from an authoritative time source on the network, the software clock should receive time updates from an authoritative time on the network. The hardware clock should in turn be updated at regular intervals from the software clock while the system is running.

To customize the use of the hardware clock on your system, perform any of the following optional tasks:

- Setting the Hardware Clock
- Configuring the Router as a Network Time Source
- Setting the Software Clock from the Hardware Clock
- Setting the Hardware Clock from the Software Clock

Setting the Hardware Clock

The hardware clock (system calendar) maintains time separately from the software clock. The hardware clock continues to run when the system is restarted or when the power is turned off. Typically, the hardware clock needs to be manually set only once, when the system is first installed.

You should avoid setting the hardware clock manually if you have access to a reliable external time source. Time synchronization should instead be established using NTP.

If you do not have access to an external time source, use one of the forms of the following command in EXEC mode to set the hardware clock:

Command	Purpose

`Router> calendar set hh:mm:ss day month year`
or
`Router> calendar set hh:mm:ss month day year`
	Sets the hardware clock manually.

Configuring the Router as a Network Time Source

By default, the time maintained on the software clock is not considered to be authoritative and will not be redistributed with NTP or VINES Time Service. To classify the hardware clock as authoritative, use the following command in global configuration mode:

Command	Purpose
`Router(config)# clock calendar-valid`	Enables the router to act as a valid time source to which network peers can synchronize.

For an example of making the hardware clock authoritative, see the "Clock, Calendar, and NTP Configuration Examples" section at the end of this chapter.

Setting the Software Clock from the Hardware Clock

To set the software clock to the new hardware clock setting, use the following command in EXEC mode:

Command	Purpose
`Router# clock read-calendar`	Sets the software clock from the hardware clock.

Setting the Hardware Clock from the Software Clock

To update the hardware clock with a new software clock setting, use the following command in EXEC mode:

Command	Purpose
`Router# clock update-calendar`	Sets the hardware clock from the software clock.

Monitoring Time and Calendar Services

To monitor clock, calendar, and NTP EXEC services, use the following commands in EXEC mode, as needed:

Command	Purpose
`Router# show calendar`	Displays the current hardware clock time.
`Router# show clock [detail]`	Displays the current software clock time.
`Router# show ntp associations [detail]`	Displays the status of NTP associations.
`Router# show ntp status`	Displays the status of NTP.
`Router# show sntp`	Displays information about SNTP (Cisco 1003, Cisco 1004, Cisco 1005, Cisco 1600, Cisco 1720, or Cisco 1750 routers only).

Configuring Time Ranges

Cisco IOS allows implementation of features based on the time of day. The **time-range** global configuration command defines specific times of the day and week, which then can be referenced by a function, so that those time restrictions are imposed on the function itself.

In Cisco IOS Release 12.2, IP and IPX extended access lists are the only functions that can use time ranges. The time range allows the network administrator to define when the **permit** or **deny** statements in the access list are in effect. Prior to the introduction of this feature, access list statements were always in effect once they were applied. Both named or numbered access lists can reference a time range.

Benefits of time ranges include the following:

- The network administrator has more control over permitting or denying a user access to resources. These resources could be an application (identified by an IP address/mask pair and a port number), policy routing, or an on-demand link (identified as interesting traffic to the dialer).
- Network administrators can set a time-based security policy, including the following:
  - Perimeter security using the Cisco IOS Firewall feature set or access lists
  - Data confidentiality with Cisco Encryption Technology or IPSec
- Policy-based routing and queueing functions are enhanced.
- When provider access rates vary by time of day, it is possible to automatically reroute traffic cost effectively.
- Service providers can dynamically change a committed access rate (CAR) configuration to support the quality of service (QoS) service level agreements (SLAs) that are negotiated for certain times of day.
- Network administrators can control logging messages. Access list entries can log traffic at certain times of the day, but not constantly. Therefore, administrators can simply deny access without needing to analyze many logs generated during peak hours.

Defining a Time Range

Note The time range relies on the system’s software clock. For the time range feature to work the way you intend, you need a reliable clock source. We recommend that you use NTP to synchronize the system’s software clock.

To define a time range, use the following commands beginning in global configuration mode.

	Command	Purpose
Step 1	`Router(config)# time-range time-range-name`	Assigns a name to the time range to be configured and enters time-range configuration mode.

Step 2	`Router(config-time-range)# absolute [start time date] [end time date]`
	or
	`Router(config-time-range)# periodic days-of-the-week hh:mm to [days-of-the-week] hh:mm`
		Specifies when the time range will be in effect. Use some combination of these commands; multiple **periodic** statements are allowed; only one **absolute** statement is allowed.

Repeat these tasks if you have multiple items you want in effect at different times. For example, repeat the steps to include multiple **permit** or **deny** statements in an access list in effect at different times. For more information about these commands, refer to the "Basic System Management Commands" chapter in the "Cisco IOS System Management Commands" part of the Release 12.2 *Cisco IOS Configuration Fundamentals Command Reference*.

Referencing the Time Range

In order for a time range to be applied, you must reference it by name in a feature that can implement time ranges. You can reference the time range in the following Cisco IOS software features:

- IP Extended Access Lists
  - Refer to the "Configuring IP Services" chapter of the Release 12.2 *Cisco IOS IP Configuration Guide* for instructions on creating an IP Extended Access List and referencing a time range.
- IPX Extended Access Lists
  - Refer to the "Configuring Novell IPX" chapter of the Release 12.2 *Cisco IOS AppleTalk and Novell IPX Configuration Guide* for instructions on creating an IPX Extended Access List and referencing a time range.

Delaying EXEC Startup

To delay the startup of the EXEC process on noisy lines until the line has been idle for 3 seconds, use the following command in global configuration mode:

Command	Purpose
`Router(config)# service exec-wait`	Delays startup of the EXEC.

This command is useful on noisy modem lines or when a modem attached to the line is configured to ignore MNP or V.42 negotiations, and MNP or V.42 modems may be dialing in. In these cases, noise or MNP/V.42 packets might be interpreted as usernames and passwords, causing authentication failure before the user can type a username or password. The command is not useful on nonmodem lines or lines without some kind of login configured.

Handling an Idle Telnet Connection

To configure the Cisco IOS software to set the TCP window to zero (0) when the Telnet connection is idle, use the following command in global configuration mode:

Command	Purpose
`Router(config)# `**`service telnet-zero-idle`**	Sets the TCP window to zero when the Telnet connection is idle.

Normally, data sent to noncurrent Telnet connections is accepted and discarded. When **service telnet-zero-idle** is enabled, if a session is suspended (that is, some other connection is made active), the TCP window is set to zero. This action prevents the remote host from sending any more data until the connection is resumed. Use this command when it is important that all messages sent by the host be seen by the users and the users are likely to use multiple sessions. Do not use this command if your host will eventually time out and log out a TCP user whose window is zero.

Setting the Interval for Load Data

You can change the period of time over which a set of data is used for computing load statistics. Decisions, such as for dial backup, depend on these statistics. If you decrease the load interval, the average statistics are computed over a shorter period of time and are more responsive to bursts of traffic.

To change the length of time for which a set of data is used to compute load statistics, use the following command in interface configuration mode:

Command	Purpose
`Router(config-if)# `**`load-interval`** *`seconds`*	Sets the length of time for which data is used for load calculations.

Limiting the Number of TCP Transactions

When using a standard TCP implementation to send keystrokes between machines, TCP tends to send one packet for each keystroke typed, which can use up bandwidth and contribute to congestion on larger networks.

John Nagle’s algorithm (RFC 896) helps alleviate the small-packet problem in TCP. The first character typed after connection establishment is sent in a single packet, but TCP holds any additional characters typed until the receiver acknowledges the previous packet. Then the second, larger packet is sent, and additional typed characters are saved until the acknowledgment comes back. The effect is to accumulate characters into larger chunks, and pace their transmission to the network at a rate matching the round-trip time of the given connection. This method is usually preferable for all TCP-based traffic.

By default, the Nagle algorithm is not enabled. To enable the Nagle algorithm and thereby reduce the number of TCP transactions, use the following command in global configuration mode:

Command	Purpose
`Router(config)# service nagle`	Enables the Nagle slow packet avoidance algorithm.

Configuring Switching and Scheduling Priorities

The normal operation of the network server allows the switching operations to use as much of the central processor as is required. If the network is running unusually heavy loads that do not allow the processor the time to handle the routing protocols, you may need to give priority to the system process scheduler. To do so, use the following command in global configuration mode:

Command	Purpose
Router(config)# **scheduler interval** *milliseconds*	Defines the maximum amount of time that can elapse without running the lowest-priority system processes.

To change the amount of time that the CPU spends on fast-switching and process-level operations on the Cisco 7200 series and Cisco 7500 series routers, use the following command in global configuration mode:

Command	Purpose
Router(config)# **scheduler allocate** *network-microseconds process-microseconds*	For the Cisco 7200 series and Cisco 7500 series routers, changes the default time the CPU spends on process tasks and fast switching.

Caution We recommend that you do not change the default values of the **scheduler allocate** command.

To configure the characteristics for a looping process, use the following command in global configuration mode:

Command	Purpose
Router(config)# **scheduler process-watchdog** {**hang** | **normal** | **reload** | **terminate**}	Configures an action for a looping process.

Modifying the System Buffer Size

You can adjust initial buffer pool settings and the limits at which temporary buffers are created and destroyed. To do so, use the following commands in global configuration mode, as needed:

Command	Purpose
`Router(config)# buffers {small	middle	big	verybig	large	huge	type number} {permanent	max-free	min-free	initial} number`	Adjusts the system buffer sizes.
`Router(config)# buffers huge size number`	Dynamically resizes all huge buffers to the value that you supply.

Caution Normally you need not adjust these parameters; do so only after consulting with technical support personnel. Improper settings can adversely impact system performance.

During normal system operation, there are two sets of buffer pools: public and interface. They behave as follows:

- The buffers in the public pools grow and shrink based upon demand. Some public pools are temporary and are created and destroyed as needed. Other public pools are permanently allocated and cannot be destroyed. Public buffer pools are labeled as small, middle, big, large, very big, and huge.
- Interface pools are static—that is, they are all permanent. One interface pool exists for each interface. For example, a Cisco 4000 1E 4T configuration has one Ethernet buffer pool and four serial buffer pools. In the **buffers** EXEC command, the *type* and *number* arguments allow the user to tune the interface pools.

See the section "Buffer Modification Examples" at the end of this chapter for more information.

The server has one pool of queueing elements and six public pools of packet buffers of different sizes. For each pool, the server keeps count of the number of buffers outstanding, the number of buffers in the free list, and the maximum number of buffers allowed in the free list. To display statistics about the buffer pool on the system, use the following commands in EXEC mode, as needed:

Command	Purpose
`Router> show buffers`	Displays all public pool information.
`Router> show buffers address hex-addr`	Displays buffer information for an address.
`Router> show buffers all [dump	header	packet]`	Displays all public and interface pool information.
`Router> show buffers assigned [dump	header	packet]`	Displays a listing of all buffers in use.
`Router> show buffers failures [dump	header	packet]`	Displays buffer allocation failures.
`Router> show buffers free [dump	header	packet]`	Displays buffers available for use.
`Router> show buffers old [dump	header	packet]`	Displays buffers older than one minute.
`Router> show buffers input-interface interface-type identifier`	Displays buffer information for an input interface.
`Router> show buffers pool pool name`	Displays all interface pool information.

Basic System Management Examples

This section provides the following system management examples:

- System Configuration File Example
- Clock, Calendar, and NTP Configuration Examples
- Buffer Modification Examples

System Configuration File Example

The following is an example of a typical system configuration file:

```
! Define line password
line 0 4
 password secret
 login
!
! Define privileged-level password
enable-password Secret Word
!
! Define a system hostname
hostname TIP
! Specify a configuration file to load at system startup
boot host host1-confg 192.168.1.111
boot host host2-confg 192.168.1.111
! Specify the system image to boot at startup
boot system sys1-system 192.168.13.111
boot system sys2-system 192.168.1.111
boot system rom
!
! Enable SNMP
snmp-server community red
snmp-server enable traps snmp authentication
snmp-server host 192.168.1.27 public
snmp-server host 192.168.1.111 public
snmp-server host 192.168.2.63 public
!
! Define TACACS server hosts
tacacs-server host 192.168.1.27
tacacs-server host 192.168.13.33
tacacs-server host 192.168.1.33
!
! Define a message-of-the-day banner
banner motd ^C
The Information Place welcomes you

Please call 1-800-555-2222 for a login account, or enter
your password at the prompt.
^C
```

Clock, Calendar, and NTP Configuration Examples

In the following example, a router with a hardware clock has server associations with two other systems, sends broadcast NTP packets, periodically updates the hardware clock, and redistributes time into VINES:

```
clock timezone PST -8
clock summer-time PDT recurring
ntp update-calendar
ntp server 192.168.13.57
ntp server 192.168.11.58
interface Ethernet 0/0
```

```
 ntp broadcast
vines time use-system
```

In the following example, a router with a hardware clock has no outside time source, so it uses the hardware clock as an authoritative time source and distributes the time via NTP broadcast packets:

```
clock timezone MET 2
clock calendar-valid
ntp master
interface fddi 0/0
 ntp broadcast
```

Buffer Modification Examples

The following example instructs the system to keep at least 50 small buffers free:

```
Router> buffers small min-free 50
```

The following example instructs the system to keep no more than 200 middle buffers free:

```
Router> buffers middle max-free 200
```

The following example instructs the system to create one large temporary extra buffer, just after a reload:

```
Router> buffers large initial 1
```

The following example instructs the system to create one permanent huge buffer:

```
Router> buffers huge permanent 1
```

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

Troubleshooting and Fault Management

This chapter describes basic tasks that you can perform to troubleshoot your system and the network. For detailed troubleshooting procedures and scenarios, refer to the *Internetwork Troubleshooting Guide*. For complete details on all **debug** commands, refer to the *Cisco IOS Debug Command Reference*.

For a complete description of the troubleshooting commands in this chapter, refer to the "Troubleshooting and Fault Management Commands" chapter in "Cisco IOS System Management Commands" part of the Release 12.2 *Cisco IOS Configuration Fundamentals Command Reference*. To locate documentation of other commands that appear in this chapter, use the *Cisco IOS Command Reference Master Index* or search online.

To identify hardware or software image support for a specific feature, use Feature Navigator on Cisco.com to search for information about the feature or refer to the software release notes for a specific release. For more information, see the "Identifying Platform Support for Cisco IOS Software Features" section in the "About Cisco IOS Software Documentation" chapter.

Troubleshooting and Fault Management Task List

To manage network faults, you need to discover, isolate, and correct problems. You can discover problems with the system monitoring commands, isolate problems with the system test commands, and resolve problems with other commands, including **debug** commands.

To perform general fault management, perform the tasks described in the following sections:

- Displaying System Information Using show Commands
- Testing Network Connectivity
- Logging System Messages
- Using Field Diagnostics on Line Cards
- Troubleshooting Specific Line Cards
- Storing Line Card Crash Information
- Creating Core Dumps for System Exceptions
- Enabling Debug Operations
- Enabling Conditionally Triggered Debugging

Americas Headquarters:
Cisco Systems, Inc., 170 West Tasman Drive, San Jose, CA 95134-1706 USA

- Using the Environmental Monitor
- Packet Capture Infrastructure

In addition to the material presented in this chapter, many chapters in the Cisco IOS software configuration guides include fault management tasks specific to certain technologies and features. You can find these tasks in the "Monitoring and Maintaining" sections.

Displaying System Information Using show Commands

To provide information about system processes, the Cisco IOS software includes an extensive list of EXEC commands that begin with the word **show**, which, when executed, display detailed tables of system information. Following is a partial list of system management **show** commands. To display the information described, use the following commands in EXEC mode, as needed:

Command	Purpose
`Router#` **`show c2600`**	Displays information about the Cisco 2600 platform, including interrupts, IOS Priority Masks, and IDMA status, for troubleshooting.
`Router#` **`show c7200`**	Displays information about the CPU and midplane for the Cisco 7200 series routers.
`Router#` **`show context`**	Displays information stored in NVRAM when the router crashes. This command is only useful to your technical support representative. This command is supported on the Cisco 2600 and 7000 series routers.
`Router#` **`show controllers`**	Displays information specific to the hardware on a line card.
`Router#` **`show controllers logging`**	Displays logging information about a line card.
`Router#` **`show controllers tech-support`**	Displays general information about a line for use when reporting a problem.
`Router#` **`show controllers vip`** *`slot-number`* **`tech-support`**	Displays information about the Versatile Interface Processor (VIP) card for use when reporting a problem
`Router#` **`show diag`**	Displays hardware information (including DRAM and static RAM details) for line cards.
`Router#` **`show environment`** `[`**`all`** `|` **`last`** `|` **`table`**`]`	Displays a message indicating whether an environmental warning condition currently exists, the temperature and voltage information, the last measured value from each of the six test points stored in nonvolatile memory, or environmental specifications. Examples of systems that support this command include the Cisco 7000 and the Cisco 12000 series routers.
`Router#` **`show gsr`**	Displays hardware information on the Cisco 12000 series Gigabit Switch Router (GSR).
`Router#` **`show gt64010`**	Displays all GT64010 internal registers and interrupt status on the Cisco 7200 series routers.
`Router#` **`show memory`** `[`*`memory-type`*`] [`**`free`**`] [`**`summary`**`]`	Displays memory pool statistics including summary information about the activities of the system memory allocator and a block-by-block listing of memory use.

Command	Purpose
`Router# show pci {hardware | bridge [register]}`	Displays information about the peripheral component interconnect (PCI) hardware registers or bridge registers for the Cisco 2600 and 7000 series routers.
`Router# show processes [cpu]`	Displays information about all active processes.
`Router# show processes memory`	Displays information about memory usage.
`Router# show protocols`	Displays the configured protocols.
`Router# show stacks`	Displays stack usage of processes and interrupt routines, including the reason for the last system reboot. This command is only useful to your technical support representative.
`Router# show subsys [class class | name name]`	Displays subsystem information.
`Router# show tcp [line-number]`	Displays the status of TCP connections.
`Router# show tcp brief [all]`	Displays a concise description of TCP connection endpoints.
`Router# show tdm connections [motherboard | slot number]`	Displays a snapshot of the time-division multiplexing (TDM) bus connection or data memory in a Cisco AS5200 access server.
`Router# show tech-support [page] [password]`	Displays information about the system for use when reporting a problem.

Refer to specific **show** commands in the tables of configuration commands found throughout the chapters in Cisco IOS software configuration guides. Refer to the Cisco IOS software command reference publications for detailed descriptions of the commands.

Testing Network Connectivity

To test basic network connectivity, perform the tasks described in the following sections:

- Configuring the TCP Keepalive Packet Service
- Testing Connections with the ping Command
- Tracing Packet Routes

Configuring the TCP Keepalive Packet Service

The TCP keepalive capability allows a router to detect when the host with which it is communicating experiences a system failure, even if data stops being sent (in either direction). This capability is most useful on incoming connections. For example, if a host failure occurs while the router is communicating with a printer, the router might never notice, because the printer does not generate any traffic in the opposite direction. If keepalives are enabled, they are sent once every minute on otherwise idle connections. If 5 minutes pass and no keepalives are detected, the connection is closed. The connection is also closed if the host replies to a keepalive packet with a reset packet. This will happen if the host crashes and comes back up again.

To generate the TCP keepalive packet service, use the following command in global configuration mode:

Command	Purposes
`Router(config)# service {tcp-keepalives-in | tcp-keepalives-out}`	Generates TCP keepalive packets on idle network connections, either incoming connections initiated by a remote host, or outgoing connections initiated by a user.

Testing Connections with the ping Command

As an aid to diagnosing basic network connectivity, many network protocols support an echo protocol. The protocol involves sending a special datagram to the destination host, then waiting for a reply datagram from that host. Results from this echo protocol can help in evaluating the path-to-host reliability, delays over the path, and whether the host can be reached or is functioning.

To invoke the echo protocol, use the following command in either user or privileged EXEC mode:

Command	Purposes
`Router# ping [protocol] {host | address}`	Invokes a diagnostic tool for testing connectivity.

Refer to specific **ping** commands in the tables of configuration commands found throughout the chapters in Cisco IOS software configuration guides. Refer to the Cisco IOS software command reference publications for detailed descriptions of the command.

Tracing Packet Routes

To trace the routes that packets will actually take when traveling to their destinations, use the following command in either user or privileged EXEC mode:

Command	Purposes
`Router# trace [protocol] [destination]`	Traces packet routes through the network (privileged level).

Logging System Messages

By default, routers send logging messages (including debug command output) a logging process. The logging process controls the distribution of logging messages to various destinations, such as the logging buffer, terminal lines, or a UNIX syslog server, depending on your configuration. The process also sends messages to the console. When the logging process is on, the messages are displayed on the console after the process that generated them has finished.

When the logging process is disabled, messages are sent only to the console. The messages are sent as they are generated, so error and debug output will be interspersed with prompts or output from the command.

You can set the severity level of the messages to control the type of messages displayed for the console and each destination. You can time-stamp log messages or set the syslog source address to enhance real-time debugging and management.

System logging messages are traditionally referred to as System Error Messages. Refer to the *Cisco IOS Software System Error Messages* publication for detailed information on specific system logging messages.

Enabling System Message Logging

System message logging is enabled by default. It must be enabled in order to send messages to any destination other than the console.

To disable message logging, use the **no logging on** command. Note that disabling the logging process can slow down the router because a process cannot continue until the messages are written to the console.

To reenable message logging after it has been disabled, use the following command in global configuration mode:

Command	Purposes
`Router(config)# logging on`	Enables message logging.

Enabling Message Logging for a Slave Card

To enable slave VIP cards to log status messages to the console (print the messages to the screen), use the following command in global configuration mode:

Command	Purposes
`Router(config)# service slave-log`	Enables slave message logging.

Setting the Syslog Destination

If message logging is enabled, you can send messages to specified locations, in addition to the console.

To set the locations that receive messages, use the following commands in global configuration mode, as needed:

Command	Purposes
`Router(config)# logging buffered [size]`	Logs messages to an internal buffer.
`Router(config)# terminal monitor`	Logs messages to a nonconsole terminal.
`Router(config)# logging host`	Logs messages to a syslog server host.

The **logging buffered** command copies logging messages to an internal buffer. The buffer is circular, so newer messages overwrite older messages after the buffer is full. To display the messages that are logged in the buffer, use the **show logging** EXEC command. The first message displayed is the oldest message in the buffer. To clear the current contents of the buffer, use the **clear logging** privileged EXEC command.

The **terminal monitor** EXEC command locally accomplishes the task of displaying the system logging messages to a terminal.

The **logging** command identifies a syslog server host to receive logging messages. The *host* argument is the name or IP address of the host. By issuing this command more than once, you build a list of syslog servers that receive logging messages. The **no logging** command deletes the syslog server with the specified address from the list of syslogs.

Configuring Synchronization of Logging Messages

You can configure the system to synchronize unsolicited messages and **debug** command output with solicited device output and prompts for a specific line. You can identify the types of messages to be output asynchronously based on the level of severity. You can also determine the maximum number of buffers for storing asynchronous messages for the terminal after which messages are dropped.

When synchronous logging of unsolicited messages and **debug** command output is turned on, unsolicited device output is displayed on the console or printed after solicited device output is displayed or printed. Unsolicited messages and **debug** command output is displayed on the console after the prompt for user input is returned. Therefore, unsolicited messages and **debug** command output are not interspersed with solicited device output and prompts. After the unsolicited messages are displayed, the console displays the user prompt again.

To configure for synchronous logging of unsolicited messages and **debug** command output with solicited device output and prompts, use the following commands beginning in global configuration mode:

	Command	Purposes

Step 1	`Router(config)#` **`line`** `[`**`aux`** `|` **`console`** `|` **`vty`**`]`
	`beginning-line-number` `[`*`ending-line-number`*`]`
		Specifies the line to be configured for synchronous logging of messages.

Step 2	`Router(config-line)#` **`logging synchronous`**
	`[`**`level`** *`severity-level`* `|` **`all`**`]`
	`[`**`limit`** *`number-of-buffers`*`]`
		Enables synchronous logging of messages.

Enabling Time-Stamps on Log Messages

By default, log messages are not time-stamped. To enable time-stamping of log messages, use either of the following commands in global configuration mode:

Command	Purposes

`Router(config)#` **`service timestamps log uptime`**
or
`Router(config)#` **`service timestamps log datetime`** `[`**`msec`**`]`
`[`**`localtime`**`] [`**`show-timezone`**`]`
	Enables log time stamps.

Limiting the Error Message Severity Level and Facilities

You can limit the number of messages displayed to the selected device by specifying the severity level of the error message (see Table 1 for level descriptions). To do so, use the following commands in global configuration mode, as needed:

Command	Purposes
Router(config)# **logging console** *level*	Limits the number of messages logged to the console.
Router(config)# **logging monitor** *level*	Limits the number of messages logged to the terminal lines.
Router(config)# **logging trap** *level*	Limits the number of messages logged to the syslog servers.

If you have enabled syslog messages traps to be sent to a Simple Network Management Protocol (SNMP) network management station with the **snmp-server enable trap** command, you can change the level of messages sent and stored in a history table on the router. You can also change the number of messages that get stored in the history table.

Messages are stored in the history table because SNMP traps are not guaranteed to reach their destination. By default, one message of the level warning and above (see Table 1) is stored in the history table even if syslog traps are not enabled.

To change level and table size defaults, use the following commands in global configuration mode:

	Command	Purposes
Step 1	Router(config)# **logging history** *level*	Changes the default level of syslog messages stored in the history file and sent to the SNMP server.
Step 2	Router(config)# **logging history size** *number*	Changes the number of syslog messages that can be stored in the history table.

Note Table 1 lists the level keywords and severity level. For SNMP usage, the severity level values use +1. For example, **emergency** equals 1 not 0 and **critical** equals 3 not 2.

The **logging console** command limits the logging messages displayed on the console terminal to messages with a level number at or below the specified severity level, which is specified by the *level* argument. Table 1 lists the error message *level* keywords and corresponding UNIX syslog definitions in order from the most severe level to the least severe level.

Table 1 System Logging Message Severity Levels

Level Keyword	Level	Description	Syslog Definition
emergencies	0	System unusable	LOG_EMERG
alerts	1	Immediate action needed	LOG_ALERT
critical	2	Critical conditions	LOG_CRIT
errors	3	Error conditions	LOG_ERR
warnings	4	Warning conditions	LOG_WARNING
notifications	5	Normal but significant condition	LOG_NOTICE
informational	6	Informational messages only	LOG_INFO
debugging	7	Debugging messages	LOG_DEBUG

The **no logging console** command disables logging to the console terminal.

The default is to log messages to the console at the **debugging** level and those level numbers that are lower, which means all levels. The **logging monitor** command defaults to **debugging** also. The **logging trap** command defaults to the **informational** level.

To display logging messages on a terminal, use the **terminal monitor** EXEC command.

Current software generates the following four categories of error messages:

- Error messages about software or hardware malfunctions, displayed at levels **warnings** through **emergencies**
- Output from the **debug** commands, displayed at the **debugging** level
- Interface up/down transitions and system restart messages, displayed at the **notifications** level
- Reload requests and low-process stack messages, displayed at the **informational** level

Defining the UNIX System Logging Facility

You can log messages produced by UNIX system utilities. To do this, enable this type logging and define the UNIX system facility from which you want to log messages. Table 2 lists the UNIX system facilities supported by the Cisco IOS software. Consult the operator manual for your UNIX operating system for more information about these UNIX system facilities. The syslog format is compatible with Berkeley Standard Distribution (BSD) UNIX version 4.3.

To define UNIX system facility message logging, use the following command in global configuration mode:

Command	Purposes
`Router(config)# logging facility facility-type`	Configures system log facilities.

Table 2 Logging Facility Type Keywords

Facility Type Keyword	Description
auth	Indicates the authorization system.
cron	Indicates the cron facility.
daemon	Indicates the system daemon.
kern	Indicates the Kernel.
local0–7	Reserved for locally defined messages.
lpr	Indicates line printer system.
mail	Indicates mail system.
news	Indicates USENET news.
sys9	Indicates system use.
sys10	Indicates system use.
sys11	Indicates system use.
sys12	Indicates system use.
sys13	Indicates system use.
sys14	Indicates system use.

Table 2 Logging Facility Type Keywords (continued)

Facility Type Keyword	Description
syslog	Indicates the system log.
user	Indicates user process.
uucp	Indicates UNIX-to-UNIX copy system.

Displaying Logging Information

To display logging information, use the following commands in EXEC mode, as needed:

Command	Purposes
`Router# show logging`	Displays the state of syslog error and event logging, including host addresses, whether console logging is enabled, and other logging statistics.
`Router# show controllers vip slot-number logging`	Displays the state of syslog error and event logging of a VIP card, including host addresses, whether console logging is enabled, and other logging statistics.
`Router# show logging history`	Displays information in the syslog history table such as the table size, the status of messages, and the text of the messages stored in the table.

Logging Errors to a UNIX Syslog Daemon

To configure the syslog daemon on a 4.3 BSD UNIX system, include a line such as the following in the /etc/syslog.conf file:

```
local7.debugging /usr/adm/logs/cisco.log
```

The **debugging** keyword specifies the syslog level; see Table 1 for a general description of other keywords. The **local7** keyword specifies the logging facility to be used; see Table 2 for a general description of other keywords.

The syslog daemon sends messages at this level or at a more severe level to the file specified in the next field. The file must already exist, and the syslog daemon must have permission to write to it.

Setting the Syslog Source Address

By default, a syslog message contains the IP address of the interface it uses to leave the router. To set all syslog messages to contain the same IP address, regardless of which interface they use, use the following command in global configuration mode:

Command	Purposes
`Router(config)# logging source-interface type number`	Sets the syslog source address.

Using Field Diagnostics on Line Cards

Each line card on the Cisco 12000 series routers can perform field diagnostic testing to isolate faulty hardware without disrupting normal operation of the system. However, performing field diagnostic testing on a line card does halt all activity on the line card for the duration of the testing. After successful completion of the field diagnostic testing, the Cisco IOS software is automatically reloaded on the line card.

Note The field diagnostic **diag** command must be executed from the Gigabit Route Processor (GRP) main console port.

To perform field diagnostic testing on a line card, use the following command in privileged EXEC mode:

Command	Purposes
`Router# diag slot-number [previous | post | verbose | wait]`	Specifies the line card on which you want to perform diagnostic testing. Optionally, specifies that previous test results are displayed, that only extended power-on self-tests (POST) be performed, that the maximum messages are displayed, or that the Cisco IOS software not be reloaded on the line card after successful completion of the tests. The following prompt is displayed: `Running Diags will halt ALL activity on the requested slot. [confirm]` At the prompt, press **Return** to confirm that you want to perform field diagnostic testing on the specified line card, or type **no** to stop the testing.

To stop field diagnostic testing on a line card, use either of the following commands in privileged EXEC mode:

Command	Purpose

`Router# diag slot-number halt`
or
`Router# no diag slot-number`
	Specifies the line card on which you want to stop diagnostic testing.

Note When you stop the field diagnostic test, the line card remains down (that is, in an unbooted state). In most cases, you stopped the testing because you need to remove the line card or replace the line card. If that is not the case and you want to bring the line card back up (that is, online), you must use the **microcode reload** global configuration command or power cycle the line card.

Troubleshooting Specific Line Cards

Cisco IOS provides the **execute-on** command to allow you to issue Cisco IOS commands (such as **show** commands) to a specific line card for monitoring and maintenance. For example, you could show which Cisco IOS image is loaded on the card in slot 3 of a Cisco 12012 Gigabit Switch Router (GSR) by issuing the **execute-on slot 3 show version** command. You can also use this command for troubleshooting cards in the dial shelf of Cisco access servers. For complete documentation of this command, refer to the "Troubleshooting" chapter of the Release 12.2 *Cisco IOS Configuration Fundamentals Command Reference*.

Storing Line Card Crash Information

This section explains how to enable storing of crash information for a line card and optionally specify the type and amount of information stored. Technical support representatives need to be able to look at the crash information from the line card to troubleshoot serious problems on the line card. The crash information contains all the line card memory information, including the main memory and transmit and receive buffer information.

Caution Use the **exception linecard** global configuration command only when directed by a technical support representative, and only enable options that the technical support representative requests you to enable.

To enable and configure the crash information options for a line card, use the following command in global configuration mode:

Command	Purpose
`Router(config)# exception linecard {all	slot slot-number} [corefile filename	main-memory size [k	m]	queue-ram size [k	m]	rx-buffer size [k	m]	sqe-register-rx	sqe-register-tx	tx-buffer size [k	m]]`	Specifies the line card for which you want crash information when a line card resets. Optionally, specify the type and amount of memory to be stored.

Creating Core Dumps for System Exceptions

"System exceptions" are any unexpected system shutdowns or reboots (most frequently caused by a system failure, commonly referred to as a "system crash"). When an exception occurs, it is sometimes useful to obtain a full copy of the memory image (called a core dump) to identify the cause of the unexpected shutdown. Not all exception types will produce a core dump.

Core dumps are generally useful only to your technical support representative. The core dump file, which is a very large binary file, can be transferred to a Trivial File Transfer Protocol (TFTP), File Transfer Protocol (FTP), or Remote Copy Protocol (RCP) server, or (on limited platforms) saved to the flash disk, and subsequently interpreted by technical personnel who have access to source code and detailed memory maps.

Caution Use the **exception** commands only under the direction of a technical support representative. Creating a core dump while the router is functioning in a network can disrupt network operation.

Specifying the Destination for the Core Dump File

To configure the router to generate a core dump, you must enable exception dumps and configure a destination for the core dump file, as described in the following sections:

- Using TFTP for Core Dumps
- Using FTP for Core Dumps
- Using rcp for Core Dumps
- Using a Flash Disk for Core Dumps

Using TFTP for Core Dumps

Due to a limitation of most TFTP applications, the router will dump only the first 16 MB of the core file. Therefore, if your router's main memory is larger than 16 MB, do not use TFTP.

To configure a router for a core dump using TFTP, use the following commands in global configuration mode:

	Command or Action	Purpose
Step 1	`exception protocol tftp`	(Optional) Explicitly specifies TFTP as the protocol to be used for router exceptions (core dumps for unexpected system shutdowns). **Note** Because TFTP is the default exception protocol, the **exception protocol tftp** command does not need to be used unless the protocol has been previously changed to ftp or rcp in your system's configuration. To determine if the exception protocol has been changed, use the **show running-config** command in EXEC mode.
Step 2	`exception dump` *`ip-address`*	Configures the router to dump a core file to the specified server if the router crashes.
Step 3	`exception core-file` [*`filepath/`*] *`filename`*	(Optional) Specifies the name to be used for the core dump file. The file usually must pre-exist on the TFTP server, and be writable.

For example, the following command configures a router to send a core file to the server at the IP address 172.17.92.2. As the exception protocol is not specified, the default protocol of TFTP will be used.

```
Router(config)# exception dump 172.17.92.2
```

The core dump is written to a file named "*hostname*-core" on the TFTP server, where *hostname* is the name of the route (in the example above, the file would be named Router-core). You can change the name of the core file by adding the **exception core-file** *filename* configuration command.

Depending on the TFTP server application used, it may be necessary to create, on the TFTP server, the empty target file to which the router can write the core. Also, make sure there is enough memory on your TFTP server to hold the complete core dump.

Using FTP for Core Dumps

To configure the router for a core dump using FTP, use the following commands in global configuration mode:

	Command	Purposes
Step 1	`Router(config)# ip ftp username username`	(Optional) Configures the user name for FTP connections.
Step 2	`Router(config)# ip ftp password [type] password`	(Optional) Specifies the password to be used for FTP connections.
Step 3	`Router(config)# exception protocol ftp`	Specifies that FTP should be used for core dump file transfers.
Step 4	`Router(config)# exception dump ip-address`	Configures the router to dump a core file to a particular server if the router crashes.
Step 5	`Router(config)# exception core-file filename`	(Optional) Specifies the name to be used for the core dump file.

The following example configures a router to use FTP to dump a core file named "dumpfile" to the FTP server at 172.17.92.2 when it crashes.

```
ip ftp username red
ip ftp password blue
exception protocol ftp
exception dump 172.17.92.2
exception core-file dumpfile
```

Using rcp for Core Dumps

The remote copy protocol can also be used to send a core dump file. To configure the router to send core dump files using rcp, use the following commands:

	Command or Action	Purpose
Step 1	`ip rcmd remote-username username`	(Optional) Specifies the username sent by the router to the remote server with an rcp copy/write request. The remote rcp server must configured to grant write access to the specified username (in other words, an account must be defined on the network server for the username).
Step 2	`exception protocol rcp`	Configures the rcp as the protocol to use for sending core dump files.
Step 3	`exception dump ip-address`	Configures the router to dump a core file to the specified server if the router crashes.
Step 4	`exception core-file filename`	(Optional) Specifies the name to be used for the core dump file.

When an rcp username is not configured through the **ip rcmd remote-username** command, the rcp username defaults to the username associated with the current terminal (tty) connection. For example, if the user is connected to the router through Telnet and was authenticated through the username command, the router software sends the Telnet username as the rcp username. If the terminal username is not available, the router hostname will be used as the rcp username.

Using a Flash Disk for Core Dumps

Some router platforms support the Flash disk as an alternative to the linear Flash memory or PCMCIA Flash card. The large storage capacity of these Flash disks makes them good candidates for another means of capturing a core dump. To configure a router for a core dump using a Flash disk, use the following command in global configuration mode:

Command	Purpose
`Router(config)# exception flash [procmem |iomem	all] device-name[:partition-number] [erase	no_erase]`	Configures the router for a core dump using a flash disk.
`Router(config)# exception core-file filename`	(Optional) Specifies the name to be used for the core dump file.

The **show flash all** EXEC command will list the devices you can use for the **exception flash** command.

Creating an Exception Memory Core Dump

To cause the router to create a core dump and reboot when certain memory size parameters are violated during the debugging process, use the following commands in global configuration mode:

As a debugging procedure, you can cause the router to create a core dump and reboot when certain memory size parameters are violated. The following **exception memory** commands are used to trigger a core dump:

Command	Purpose
`Router(config)# exception memory minimum bytes`	Triggers a core dump and system reload when the amount of free memory falls below the specified number of bytes. • Do not specify too low a memory value, as the router needs some amount of free memory to provide the core dump. • If you enter a size that is greater than the free memory (and the **exception dump** command has been configured), a core dump and router reload is generated after 60 seconds.
`Router(config)# memory check-interval seconds`	(Optional) Increases the interval at which memory will be checked. The default is 60 seconds, but much can happen in 60 seconds to mask the cause of corruption. Reducing the interval will increase CPU utilization (by around 12%) which will be acceptable in most cases, but will also increase the chance of getting a usable core. To make sure CPU utilization doesn't hit 100%, you should gradually decrease the interval on busy routers. The ideal interval is as low as possible without causing other system problems.
`Router(config)# exception memory fragment bytes`	Triggers a core dump and system reload when the amount of contiguous (non-fragmented) free memory falls below the specified number of bytes.
`Router(config)# exception core-file filename`	(Optional) Specifies the name to be used for the core dump file. The file usually must exist on the TFTP server, and be writable. Note that the file will be the same size as the amount of processor memory on the router.

Note that the **exception memory minimum** command is primarily useful if you anticipate running out of memory before a core dump can be triggered or other debugging can be performed (rapid memory leak); if the memory leak is gradual (slow drift), you have generally have time to perform debugging before the system runs out of memory and must be reloaded.

By default, the number of free memory bytes is checked every 60 seconds when these commands are configured. The frequency of this checking can be increased using the **memory check-interval** *seconds* command.

The **exception dump** *ip-address* command must be configured with these commands. If the **exception dump** command is not configured, the router reloads without triggering a core dump.

The following example configures the router to monitor the free memory. If the memory falls below 250000 bytes, the core dump is created and the router reloads.

```
exception dump 172.18.92.2
exception core-file memory.overrun
exception memory minimum 250000
```

Setting a Spurious Interrupt Core Dump

During the debugging process, you can configure the router to create a spurious interrupt core dump and reboot when a specified number of interrupts have occurred.

Caution Use the **exception spurious-interrupt** global configuration command only when directed by a technical support representative and only enable options requested by the technical support representative.

To enable and configure the crash information for spurious interrupts, use the following commands in global configuration mode:

Command	Purpose
`Router(config)#` **`exception spurious-interrupt`** *`number`*	Sets the maximum number of spurious interrupts to include in the core dump before reloading.

`Router(config)#` **`exception dump`** *`ip-address`*
or
`Router(config)#` **`exception flash`**
	Specifies the destination for the core dump file.

The following example configures a router to create a core dump with a limit of two spurious interrupts:

```
exception spurious-interrupt 2
exception dump 209.165.200.225
```

Enabling Debug Operations

Your router includes hardware and software to aid in troubleshooting internal problems and problems with other hosts on the network. The **debug** privileged EXEC mode commands start the console display of several classes of network events. The following commands describe in general the system debug message feature. Refer to the *Cisco IOS Debug Command Reference* for all information regarding **debug** commands. Also refer to the *Internetwork Troubleshooting Guide* publication for additional information.

To enable debugging operations, use the following commands:

Command	Purposes
`Router# show debugging`	Displays the state of each debugging option.
`Router# debug ?`	Displays a list and brief description of all the **debug** command options.
`Router# debug command`	Begins message logging for the specified **debug** command.
`Router# no debug command`	Turns message logging off for the specified **debug** command.

Caution The system gives high priority to debugging output. For this reason, debugging commands should be turned on only for troubleshooting specific problems or during troubleshooting sessions with technical support personnel. Excessive debugging output can render the system inoperable.

You can configure time-stamping of system **debug** messages. Time-stamping enhances real-time debugging by providing the relative timing of logged events. This information is especially useful when customers send debugging output to your technical support personnel for assistance. To enable time-stamping of system **debug** messages, use either of the following commands in global configuration mode:

Command	Purposes

`Router(config)# service timestamps debug uptime`
or
`Router(config)# service timestamps debug datetime [msec] [localtime] [show-timezone]`
	Enables time-stamping of system **debug** messages.

Normally, the messages are displayed only on the console terminal. Refer to the section "Setting the Syslog Destination" earlier in this chapter to change the output device.

Enabling Conditionally Triggered Debugging

When the Conditionally Triggered Debugging feature is enabled, the router generates debugging messages for packets entering or leaving the router on a specified interface; the router will not generate debugging output for packets entering or leaving through a different interface. You can specify the interfaces explicitly. For example, you may only want to see debugging messages for one interface or subinterface. You can also turn on debugging for all interfaces that meet specified condition. This feature is useful on dial access servers, which have a large number of ports.

Normally, the router will generate debugging messages for every interface, resulting in a large number of messages. The large number of messages consumes system resources, and can affect your ability to find the specific information you need. By limiting the number of debugging messages, you can receive messages related to only the ports you wish to troubleshoot.

Conditionally Triggered Debugging controls the output from the following protocol-specific **debug** commands:

- **debug aaa** {**accounting** | **authorization** | **authentication**}

- **debug dialer** {**events** | **packets**}
- **debug isdn** {**q921** | **q931**}
- **debug modem** {**oob** | **trace**}
- **debug ppp** {**all** | **authentication** | **chap** | **error** | **negotiation** | **multilink events** | **packet**}

Although this feature limits the output of the commands listed, it does not automatically enable the generation of debugging output from these commands. Debugging messages are generated only when the protocol-specific **debug** command is enabled. The **debug** command output is controlled through two processes:

- The protocol-specific **debug** commands specify which protocols are being debugged. For example, the **debug dialer events** command generates debugging output related to dialer events.
- The **debug condition** commands limit these debugging messages to those related to a particular interface. For example, the **debug condition username bob** command generates debugging output only for interfaces with packets that specify a username of bob.

To configure Conditionally Triggered Debugging, perform the tasks described in the following sections:

- Enabling Protocol-Specific debug Commands
- Enabling Conditional Debugging Commands
- Specifying Multiple Debugging Conditions

Enabling Protocol-Specific debug Commands

In order to generate any debugging output, the protocol-specific **debug** command for the desired output must be enabled. Use the **show debugging** command to determine which types of debugging are enabled. To display the current debug conditions, use the **show debug condition** command. To enable the desired protocol-specific **debug** commands, use the following commands in privileged EXEC mode:

Command	Purpose
`Router# show debugging`	Determines which types of debugging are enabled.
`Router# show debug condition [condition-id]`	Displays the current **debug** conditions.
`Router# debug protocol`	Enables the desired debugging commands.
`Router# no debug protocol`	Disables the debugging commands that are not desired.

If you do not want output, disable all the protocol-specific **debug** commands.

Enabling Conditional Debugging Commands

If no **debug condition** commands are enabled, all debugging output, regardless of the interface, will be displayed for the enabled protocol-specific **debug** commands.

The first **debug condition** command you enter enables conditional debugging. The router will display only messages for interfaces that meet one of the specified conditions. If multiple conditions are specified, the interface must meet at least one of the conditions in order for messages to be displayed.

To enable messages for interfaces specified explicitly or for interfaces that meet certain conditions, perform the tasks described in the following sections:

- Displaying Messages for One Interface
- Displaying Messages for Multiple Interfaces
- Limiting the Number of Messages Based on Conditions

Displaying Messages for One Interface

To disable debugging messages for all interfaces except one, use the following command in privileged EXEC mode:

Command	Purpose
`Router# **debug condition interface** *interface*`	Enables debugging output for only the specified interface.

To reenable debugging output for all interfaces, use the **no debug interface** command.

Displaying Messages for Multiple Interfaces

To enable debugging messages for multiple interfaces, use the following commands in privileged EXEC mode:

	Command	Purposes
Step 1	`Router# **debug condition interface** *interface*`	Enables debugging output for only the specified interface
Step 2	`Router# **debug condition interface** *interface*`	Enable debugging messages for additional interfaces. Repeat this task until debugging messages are enabled for all desired interfaces.

If you specify more than one interface by entering this command multiple times, debugging output will be displayed for all of the specified interfaces. To turn off debugging on a particular interface, use the **no debug interface** command. If you use the **no debug interface all** command or remove the last **debug interface** command, debugging output will be reenabled for all interfaces.

Limiting the Number of Messages Based on Conditions

The router can monitor interfaces to learn if any packets contain the specified value for one of the following conditions:

- username
- calling party number
- called party number

If you enter a condition, such as calling number, debug output will be stopped for all interfaces. The router will then monitor every interface to learn if a packet with the specified calling party number is sent or received on any interfaces. If the condition is met on an interface or subinterface, **debug** command output will be displayed for that interface. The debugging output for an interface is “triggered” when the condition has been met. The debugging output continues to be disabled for the other interfaces. If, at some later time, the condition is met for another interface, the debug output also will become enabled for that interface.

Once debugging output has been triggered on an interface, the output will continue until the interface goes down. However, the session for that interface might change, resulting in a new username, called party number, or calling party number. Use the **no debug interface** command to reset the debug trigger mechanism for a particular interface. The debugging output for that interface will be disabled until the interface meets one of the specified conditions.

To limit the number of debugging messages based on a specified condition, use the following command in privileged EXEC mode:

Command	Purpose
`Router# debug condition {username username	called dial-string	caller dial-string}`	Enables conditional debugging. The router will display only messages for interfaces that meet this condition.

To reenable the debugging output for all interfaces, enter the **no debug condition all** command.

Specifying Multiple Debugging Conditions

To limit the number of debugging messages based on more than one condition, use the following commands in privileged EXEC mode:

	Command	Purposes
Step 1	`Router# debug condition {username username	called dial-string	caller dial-string}`	Enables conditional debugging, and specifies the first condition.
Step 2	`Router# debug condition {username username	called dial-string	caller dial-string}`	Specifies the second condition. Repeat this task until all conditions are specified.

If you enter multiple **debug condition** commands, debugging output will be generated if an interface meets at least one of the conditions. If you remove one of the conditions using the **no debug condition** command, interfaces that meet only that condition no longer will produce debugging output. However, interfaces that meet a condition other than the removed condition will continue to generate output. Only if no active conditions are met for an interface will the output for that interface be disabled.

Conditionally Triggered Debugging Configuration Examples

In this example, four conditions have been set by the following commands:

- **debug condition interface serial 0**
- **debug condition interface serial 1**
- **debug condition interface virtual-template 1**
- **debug condition username fred**

The first three conditions have been met by one interface. The fourth condition has not yet been met:

```
Router# show debug condition

Condition 1: interface Se0 (1 flags triggered)
        Flags: Se0
Condition 2: interface Se1 (1 flags triggered)
        Flags: Se1
```

```
Condition 3: interface Vt1 (1 flags triggered)
        Flags: Vt1
Condition 4: username fred (0 flags triggered)
```

When any **debug condition** command is entered, debugging messages for conditional debugging are enabled. The following debugging messages show conditions being met on different interfaces as the serial 0 and serial 1 interfaces come up. For example, the second line of output indicates that serial interface 0 meets the username fred condition.

```
*Mar  1 00:04:41.647: %LINK-3-UPDOWN: Interface Serial0, changed state to up
*Mar  1 00:04:41.715: Se0 Debug: Condition 4, username fred triggered, count 2
*Mar  1 00:04:42.963: %LINEPROTO-5-UPDOWN: Line protocol on Interface Serial0, changed
state to up
*Mar  1 00:04:43.271: Vi1 Debug: Condition 3, interface Vt1 triggered, count 1
*Mar  1 00:04:43.271: %LINK-3-UPDOWN: Interface Virtual-Access1, changed state to up
*Mar  1 00:04:43.279: Vi1 Debug: Condition 4, username fred triggered, count 2
*Mar  1 00:04:43.283: Vi1 Debug: Condition 1, interface Se0 triggered, count 3
*Mar  1 00:04:44.039: %IP-4-DUPADDR: Duplicate address 172.27.32.114 on Ethernet 0,
sourced by 00e0.1e3e.2d41
*Mar  1 00:04:44.283: %LINEPROTO-5-UPDOWN: Line protocol on Interface Virtual-Access1,
changed state to up
*Mar  1 00:04:54.667: %LINK-3-UPDOWN: Interface Serial1, changed state to up
*Mar  1 00:04:54.731: Se1 Debug: Condition 4, username fred triggered, count 2
*Mar  1 00:04:54.735: Vi1 Debug: Condition 2, interface Se1 triggered, count 4
*Mar  1 00:04:55.735: %LINEPROTO-5-UPDOWN: Line protocol on Interface Serial1, changed
state to up
```

After a period of time, the **show debug condition** command displays the revised list of conditions:

```
Router# show debug condition

Condition 1: interface Se0 (2 flags triggered)
        Flags: Se0 Vi1
Condition 2: interface Se1 (2 flags triggered)
        Flags: Se1 Vi1
Condition 3: interface Vt1 (2 flags triggered)
        Flags: Vt1 Vi1
Condition 4: username fred (3 flags triggered)
        Flags: Se0 Vi1 Se1
```

Next, the serial 1 and serial 0 interfaces go down. When an interface goes down, conditions for that interface are cleared.

```
*Mar  1 00:05:51.443: %LINK-3-UPDOWN: Interface Serial1, changed state to down
*Mar  1 00:05:51.471: Se1 Debug: Condition 4, username fred cleared, count 1
*Mar  1 00:05:51.479: Vi1 Debug: Condition 2, interface Se1 cleared, count 3
*Mar  1 00:05:52.443: %LINEPROTO-5-UPDOWN: Line protocol on Interface Serial1, changed
state to down
*Mar  1 00:05:56.859: %LINK-3-UPDOWN: Interface Serial0, changed state to down
*Mar  1 00:05:56.887: Se0 Debug: Condition 4, username fred cleared, count 1
*Mar  1 00:05:56.895: Vi1 Debug: Condition 1, interface Se0 cleared, count 2
*Mar  1 00:05:56.899: Vi1 Debug: Condition 3, interface Vt1 cleared, count 1
*Mar  1 00:05:56.899: Vi1 Debug: Condition 4, username fred cleared, count 0
*Mar  1 00:05:56.903: %LINK-3-UPDOWN: Interface Virtual-Access1, changed state to down
*Mar  1 00:05:57.907: %LINEPROTO-5-UPDOWN: Line protocol on Interface Serial0, changed
state to down
*Mar  1 00:05:57.907: %LINEPROTO-5-UPDOWN: Line protocol on Interface Virtual-Access1,
changed state to down
```

The final **show debug condition** output is the same as the output before the interfaces came up:

```
Router# show debug condition

Condition 1: interface Se0 (1 flags triggered)
```

```
        Flags: Se0
Condition 2: interface Se1 (1 flags triggered)
        Flags: Se1
Condition 3: interface Vt1 (1 flags triggered)
        Flags: Vt1
Condition 4: username fred (0 flags triggered)
```

Using the Environmental Monitor

Some routers and access servers have an environmental monitor that monitors the physical condition of the router. If a measurement exceeds acceptable margins, a warning message is printed to the system console. The system software collects measurements once every 60 seconds, but warnings for a given test point are printed at most once every 4 hours. If the temperature measurements are out of specification more than the shutdown, the software shuts the router down (the fan will remain on). The router must be manually turned off and on after such a shutdown. You can query the environmental monitor using the **show environment** command at any time to determine whether a measurement is out of tolerance. Refer to the *Cisco IOS System Error Messages* publication for a description of environmental monitor warning messages.

On routers with an environmental monitor, if the software detects that any of its temperature test points have exceeded maximum margins, it performs the following steps:

1. Saves the last measured values from each of the six test points to internal nonvolatile memory.
2. Interrupts the system software and causes a shutdown message to be printed on the system console.
3. Shuts off the power supplies after a few milliseconds of delay.

The system displays the following message if temperatures exceed maximum margins, along with a message indicating the reason for the shutdown:

```
Router#
%ENVM-1-SHUTDOWN: Environmental Monitor initiated shutdown
%ENVM-2-TEMP: Inlet temperature has reached SHUTDOWN level at 64(C)
```

Refer to the hardware installation and maintenance publication for your router for more information about environmental specifications.

Packet Capture Infrastructure

Cisco IOS Embedded Packet Capture is an onboard packet capture facility that provides device management and tools for automation. This feature simplifies operations by allowing the devices to become active participants in the management and operation of the network. This feature facilitates troubleshooting, the gathering of information on packet format, application analysis, and security. To capture packet data, perform the tasks described in the following sections:

- Defining a Capture Buffer on the Device
- Defining a Capture Point
- Capturing Packet Data at a Trace Point into a Buffer
- Viewing the Contents of a Capture Buffer

Defining a Capture Buffer on the Device

The Capture Buffer is where the packet data will be contained. Capture Buffers are named and the user can define/configure how the buffer handles the data going into it. You can specify the size and type of buffer: linear or circular.

- A linear buffer will stop capturing automatically when full.
- A circular buffer will continue to capture packet data.

The maximum number of bytes to capture per packet can be limited to save space. Rate limiting can also be enabled to:

- Specify a max capture rate (in packets per second).
- Capture every "nth" packet.

The automatic "off" criteria can be defined to:

- Set to stop the capture after a specified time interval.
- Set to stop the capture after capturing a given number of packets.

Filters can also be set for packets being stored in a buffer via access control lists (ACLs).

Two types of data are stored in a capture buffer: Meta Data and Packet Data. Meta Data contains

- A timestamp of when it is added to a buffer.
- Direction: egress or ingress.
- The switch path it captured.
- Encapsulation type corresponding to input/output interface to allow the decoding of L2.
- Offset to network_start, to facilitate the decoding of L3 if complete L2 decoders are unavailable.
- L3 protocol ID, to facilitate the decoding of L3 if complete L2 decoders are unavailable.

The packet data starts from datagramstart and copyies a minimum of the per packet capture size or datagramsize.

The following example shows how to define a circular capture buffer, 256K bytes long (size), to include up to 256 bytes per packet (max-size), named pktrace1. This information is displayed using the **show monitor capture buffer all parameters** command.

```
Router# monitor capture buffer pktrace1 size 256 max-size 256 circular

Router# show monitor capture buffer all parameters
Capture buffer pktrace1 (circular buffer)
Buffer Size : 262144 bytes, Max Element Size : 256 bytes, Packets : 0
Allow-nth-pak : 0, Duration : 0 (seconds), Max packets : 0, pps : 0
Associated Capture Points:
Configuration:
monitor capture buffer pktrace1 size 256 max-size 256 circular
Router#
```

Defining a Capture Point

The Capture Point is a traffic transit point where the packet capture takes place. It is identified by a name. The following capture points are available: IPv4, IPv6 CEF, and process switching. Capture points can be interface specific or "all."

- Punt

- Drop
- From-us (meaning locally generated packets)

Capture points can also be specific to traffic direction:

- In (meaning capture at ingress)
- Out (meaning capture at egress)
- Both

Actions that can be performed on a capture point include association/disassociation with a packet capture buffer. A capture point can only be associated with one capture buffer and an ACL filter can also be applied. Multiple packet capture points may be active on a given interface; for example, BGP packets can be captured into one capture buffer and Open Shortest Path First (OSPF) packets into another.

The following example shows how to define a capture point:

```
Router# monitor capture point ip cef ipceffa0/1 fastEthernet 0/1 both
Router# monitor capture point ip cef local from-us

Router# show monitor capture point all
Status Information for Capture Point ipceffa0/1
IPv4 CEF
Switch Path: IPv4 CEF            , Capture Buffer: pktrace1
Status : Inactive

Configuration:
monitor capture point ip cef ipceffa0/1 FastEthernet0/1 both

Status Information for Capture Point local
IPv4 CEF
Switch Path: IPv4 From Us        , Capture Buffer: None
Status : Inactive

Configuration:
monitor capture point ip cef local from-us
```

Capturing Packet Data at a Trace Point into a Buffer

The following example shows how to capture packets to and from interface Fast Ethernet 1/0. The buffer has already been set up and the capture point for the interface has been created. Now you associate the two.

```
Router# monitor cap point associate ipceffa0/1 pktrace1

Router# show mon capture point all
Status Information for Capture Point ipceffa0/1
IPv4 CEF
Switch Path: IPv4 CEF            , Capture Buffer: pktrace1
Status : Inactive

Configuration:
monitor capture point ip cef ipceffa0/1 FastEthernet0/1 both
```

Note Multiple capture points can be associated with one buffer.

The following example shows how to start the capture point. The packets can be seen in the capture buffer.

```
Router# monitor capture point start ipceffa0/1
Router#
Mar 21 11:13:34.023: %BUFCAP-6-ENABLE: Capture Point ipceffa0/1 enabled.

Router# show monitor capture buffer all parameters
Capture buffer pktrace1 (circular buffer)
Buffer Size : 262144 bytes, Max Element Size : 256 bytes, Packets : 31
Allow-nth-pak : 0, Duration : 0 (seconds), Max packets : 0, pps : 0
Associated Capture Points:
Name : ipceffa0/1, Status : Active
Configuration:
monitor capture buffer pktrace1 size 256 max-size 256 circular
monitor capture point associate ipceffa0/1 pktrace1
```

The following example shows how to stop the capture point and export the data for analysis. Export writes the file using one of the supported file transfer options: FTP, HTTP, HTTPS, RCP, SCP, or TFTP:

```
Router# monitor capture point stop ipceffa0/1
iin-rtr1#
Mar 21 11:14:20.152: %BUFCAP-6-DISABLE: Capture Point ipceffa0/1 disabled.

Router# monitor capture buffer pktrace1 export
  tftp://88.1.88.9/pktrace1
!
Router#
```

Viewing the Contents of a Capture Buffer

The Packet Capture Infrastructure feature provides the ability to dump packets in ASCII. The following example shows an IPv4 ICMP echo reply packet from 10.0.0.2 to 10.0.0.1:

```
<timestamp>: IPv4 packet received on Ethernet0/0 in the IPv4 CEF LES switch path
029E28E0: AABBCC01 2D00AABB CC013000 08004500  *;L.-.*;L.0...E.
029E28F0: 00640001 0000FE01 A8950A00 00020A00  .d....~.(.......
029E2900: 00010000 D5C80001 00000000 00000000  ....UH..........
029E2910: B080ABCD ABCDABCD ABCDABCD ABCDABCD  0.+M+M+M+M+M+M+M
029E2920: ABCDABCD ABCDABCD ABCDABCD ABCDABCD  +M+M+M+M+M+M+M+M
029E2930: ABCDABCD ABCDABCD ABCDABCD ABCDABCD  +M+M+M+M+M+M+M+M
029E2940: ABCDABCD ABCDABCD ABCDABCD ABCDABCD  +M+M+M+M+M+M+M+M
029E2950: ABCD
```

The infrastructure provides the following filters for output:

- interface
- direction
- switch path
- L3 protocol
- time range
- packet size

The following example shows how to view the contents of the capture buffer called pktrace1. This output is displayed using the **show monitor capture buffer** *buffer_name* [**dump**] command. The **show** command supports two modes: the default mode and the dump mode. In the latter, the hex dump of the captured packet is also shown.

```
Router# show monitor capture buffer pktrace1 dump

11:13:00.593 EDT Mar 21 2007 : IPv4 Turbo      : Fa2/1 Fa0/1

65B6F500: 080020A2 44D90009 E94F8406 08004500  .. "DY..iO....E.
65B6F510: 00400F00 0000FE01 92AF5801 13025801  .@....~../X...X.
65B6F520: 58090800 4D1A1169 00000000 0005326C  X...M..i......2l
65B6F530: 01CCABCD ABCDABCD ABCDABCD ABCDABCD  .L+M+M+M+M+M+M+M
65B6F540: ABCDABCD ABCDABCD ABCDABCD ABCD00    +M+M+M+M+M+M+M.

…

11:13:20.593 EDT Mar 21 2007 : IPv4 Turbo      : Fa2/1 Fa0/1

65B6F500: 080020A2 44D90009 E94F8406 08004500  .. "DY..iO....E.
65B6F510: 00400F02 0000FE01 92AD5801 13025801  .@....~..-X...X.
65B6F520: 58090800 FEF91169 00000000 0005326C  X...~y.i......2l
65B6F530: 4FECABCD ABCDABCD ABCDABCD ABCDABCD  Ol+M+M+M+M+M+M+M
65B6F540: ABCDABCD ABCDABCD ABCDABCD ABCDFF    +M+M+M+M+M+M+M
```

Additionally, the contents of the capture buffer can be exported to an external location in pcap format. The is done using the **monitor capture buffer** [**name**] **export** [**location**] command. The contents can be exported to the following external locations:

- FTP
- HTTP
- HTTPS
- pram: Location to dump buffer
- RCP
- SCP

This document first published April 2001. Last updated July 2008 (revision 14).

To report errors in this document, send a detailed email to bug-doc@cisco.com

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

System Monitoring and Logging

Configuring System Logging Counts

Feature History

Release	Modification
12.2(8)T	This feature was introduced.

This document describes the error log count enhancement feature in Cisco IOS Release 12.2(8)T. It includes the following sections:

- Feature Overview, page 1
- Supported Platforms, page 2
- Supported Standards, MIBs, and RFCs, page 3
- Configuration Tasks, page 3
- Configuration Examples, page 4
- Command Reference, page 5

Feature Overview

The Cisco IOS logging facility allows you to save error messages locally or to a remote host. When these error messages exceed the capacity of the local buffer dedicated to storing them, the oldest messages are removed. To provide you with more information about messages that have occurred and may have been removed from the local buffer, an error log counter tabulates the occurrences of each error message, and time-stamps the most recent occurrence.

These messages are further sorted by message facility. Messages from each message facility are grouped together and totaled in the count. If a message is rate-limited, the count is incremented based on the actual messages that have occurred.

The **service timestamps** command configuration determines the format of the "Last Time" column in the **show logging** command output. Use the **service timestamps** command to configure the time-stamp format in the "Last Time" column.

Americas Headquarters:
Cisco Systems, Inc., 170 West Tasman Drive, San Jose, CA 95134-1706 USA

Benefits

- Provides detailed information regarding system messages, including the most recent time the message occurred.
- Alerts you to a potential problem with the system if you see the same error message occurring repeatedly.

Related Features and Technologies

- Cisco IOS Logging

Related Documents

- *Cisco IOS Configuration Fundamentals Configuration Guide*, Cisco IOS Release 12.2
- *Cisco IOS Configuration Fundamentals Command Reference*, Cisco IOS Release 12.2

Supported Platforms

- Cisco 800 series
- Cisco 806
- Cisco 820 series
- Cisco 828
- Cisco 1400 series
- Cisco 1600 series
- Cisco 1700 series
- Cisco 2500 series
- Cisco 2600 series
- Cisco 3620 series
- Cisco 3640 series
- Cisco 3660 series
- Cisco 7100 series
- Cisco 7200 series
- Cisco 7500 series
- Cisco 7700 series
- Cisco Catalyst 4000 Gateway Module
- Cisco CVA120
- Cisco ONS 15104
- Cisco Route Processor Module (RPM)
- Cisco SOHO 70 series

- Cisco SOHO 78
- Cisco uBR925 series cable access routers
- Cisco uBR7200 series universal broadband routers
- Cisco Universal Router Module (URM)
- Cisco VG200

Determining Platform Support Through Cisco Feature Navigator

Cisco IOS software is packaged in feature sets that support specific platforms. To get updated information regarding platform support for this feature, access Cisco Feature Navigator. Cisco Feature Navigator dynamically updates the list of supported platforms as new platform support is added for the feature.

Cisco Feature Navigator is a web-based tool that enables you to quickly determine which Cisco IOS software images support a specific set of features and which features are supported in a specific Cisco IOS image. You can search by feature or release. Under the release section, you can compare releases side by side to display both the features unique to each software release and the features in common.

To access Cisco Feature Navigator, you must have an account on Cisco.com. If you have forgotten or lost your account information, send a blank e-mail to cco-locksmith@cisco.com. An automatic check will verify that your e-mail address is registered with Cisco.com. If the check is successful, account details with a new random password will be e-mailed to you. Qualified users can establish an account on Cisco.com by following the directions at http://www.cisco.com/register.

Cisco Feature Navigator is updated regularly when major Cisco IOS software releases and technology releases occur. For the most current information, go to the Cisco Feature Navigator home page at the following URL:

http://www.cisco.com/go/fn

Supported Standards, MIBs, and RFCs

Standards

No new or modified standards are supported by this feature.

MIBs

No new MIBs are supported by this feature

To obtain lists of supported MIBs by platform and Cisco IOS release, and to download MIB modules, go to the Cisco MIB website on Cisco.com at the following URL:

http://www.cisco.com/public/sw-center/netmgmt/cmtk/mibs.shtml

RFCs

No new or modified RFCs are supported by this feature.

Configuration Tasks

See the following sections for configuration tasks for this feature. Each task in the list is identified as either required or optional.

- Enabling the Error Log Count Capability (required)

Enabling the Error Log Count Capability

To enable the error log count capability, use the following command in global configuration mode:

Command	Purpose
`Router(config)# logging count`	Enables the error log count capability.

Verifying the Error Log Count Capability

Enter the **show logging count** command to view information about syslog error messages.

```
Router# show logging count

Facility       Message Name                     Sev Occur   Last Time
=============================================================================
SYS            BOOTTIME                           6    1 00:00:12
SYS            RESTART                            5    1 00:00:11
SYS            CONFIG_I                           5    3 1d00h
-------------  -------------------------------  -----------------------------
SYS TOTAL                                              5

LINEPROTO      UPDOWN                             5   13 00:00:19
-------------  -------------------------------  -----------------------------
LINEPROTO TOTAL                                       13

LINK           UPDOWN                             3    1 00:00:18
LINK           CHANGED                            5   12 00:00:09
-------------  -------------------------------  -----------------------------
LINK TOTAL                                            13

SNMP           COLDSTART                          5    1 00:00:11
-------------  -------------------------------  -----------------------------
SNMP TOTAL                                             1
```

Configuration Examples

This section provides the following configuration example:

- Enabling the Error Log Count Capability Example

Enabling the Error Log Count Capability Example

In the following example, the error log count capability is enabled:

```
Router# logging count

Building configuration...
Current configuration : 2507 bytes
!
! Last configuration change at 14:53:38 UTC Tue Feb 5 2002
!
```

```
.
.
.
hostname router
!
logging count
logging buffered notifications
```

Command Reference

The following new and modified commands are pertinent to this feature. To see the command pages for these commands and other commands used with this feature, go to the *Cisco IOS Master Commands List*, Release 12.4, at http://www.cisco.com/univercd/cc/td/doc/product/software/ios124/124mindx/124index.htm.

New Command

- **logging count**

Modified Command

- **show logging**

Configuring CPU Threshold Notifications

The CPU Thresholding Notification feature notifies users when a predefined threshold of CPU usage is crossed by generating a Simple Network Management Protocol (SNMP) trap message for the top users of the CPU.

Feature History for the CPU Thresholding Notification Feature

Release	Modification
12.0(26)S	This feature was introduced.
12.3(4)T	This feature was integrated into Cisco IOS Release 12.3(4)T.
12.2(25)S	This feature was integrated into Cisco IOS Release 12.2(25)S.

Finding Support Information for Platforms and Cisco IOS Software Images

Use Cisco Feature Navigator to find information about platform support and Cisco IOS software image support. Access Cisco Feature Navigator at http://www.cisco.com/go/fn. You must have an account on Cisco.com. If you do not have an account or have forgotten your username or password, click **Cancel** at the login dialog box and follow the instructions that appear.

Contents

- Restrictions for CPU Thresholding Notification, page 2
- Information About CPU Thresholding Notification, page 2
- How to Configure CPU Thresholding Notification, page 2
- Configuration Examples for CPU Thresholding Notification, page 5
- Additional References, page 7
- Command Reference, page 8

Americas Headquarters:
Cisco Systems, Inc., 170 West Tasman Drive, San Jose, CA 95134-1706 USA

Restrictions for CPU Thresholding Notification

CPU utilization averages are computed by Cisco IOS software using a 4-millisecond Network-to-Management Interface (NMI) tick. In the unlikely event where the traffic rate is a multiple of this tick rate over a prolonged period of time, the CPU Thresholding Notification feature may not accurately measure the CPU load.

Information About CPU Thresholding Notification

The CPU Thresholding Notification feature allows you to configure CPU utilization thresholds that, when crossed, trigger a notification. Two types of CPU utilization threshold are supported:

- Rising Threshold, page 2
- Falling Threshold, page 2

Rising Threshold

A rising CPU utilization threshold specifies the percentage of CPU resources that, when exceeded for a configured period of time, triggers a CPU threshold notification.

Falling Threshold

A falling CPU utilization threshold specifies the percentage of CPU resources that, when CPU usage falls below this level for a configured period of time, triggers a CPU threshold notification.

How to Configure CPU Thresholding Notification

This section contains the following procedures:

- Enabling CPU Thresholding Notification, page 2
- Defining CPU Thresholding Notification, page 3
- Setting the Entry Limit and Size of CPU Utilization Statistics, page 4

Enabling CPU Thresholding Notification

To specify the recipient of SNMP notification operations and enable CPU thresholding notification, perform these steps:

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. snmp-server enable traps cpu threshold

4. **snmp-server host** *host-address* [**traps** | **informs**] [**version** {**1** | **2c** | **3** [**auth** | **noauth** | **priv**]}] *community-string* [**udp-port** *port*] **cpu** [*notification-type*] [**vrf** *vrf-name*]

DETAILED STEPS

	Command or Action	Purpose

Step 1	**`enable`**
	Example:
	`Router> enable`
		Enables privileged EXEC mode.
		• Enter your password if prompted.

Step 2	**`configure terminal`**
	Example:
	`Router# configure terminal`
		Enables global configuration mode.

Step 3	**`snmp-server enable traps cpu threshold`**
	Example:
	`Router(config)# snmp-server enable traps cpu threshold`
		Enables CPU thresholding violation notification as traps and inform requests.

Step 4	**snmp-server host** *host-address* [**traps** | **informs**] [**version** {**1** | **2c** | **3** [**auth** | **noauth** | **priv**]}] *community-string* [**udp-port** *port*] **cpu** [*notification-type*] [**vrf** *vrf-name*] **Example:** `Router(config)# snmp-server host 192.168.0.0 traps public cpu`	Sends CPU traps to the specified address.

Defining CPU Thresholding Notification

To define a rising and a falling CPU threshold notification, perform these steps:

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **process cpu threshold type** {**total** | **process** | **interrupt**} **rising** *percentage* **interval** *seconds* [**falling** *percentage* **interval** *seconds*]

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**process cpu threshold type** {**total** | **process** | **interrupt**} **rising** *percentage* **interval** *seconds* [**falling** *percentage* **interval** *seconds*] **Example:** `Router(config)# process cpu threshold type total rising 80 interval 5 falling 20 interval 5`	Sets the CPU thresholding notifications types and values. • In this example, the CPU utilization threshold is set to 80 percent for a rising threshold notification and 20 percent for a falling threshold notification, with a 5-second polling interval.

Setting the Entry Limit and Size of CPU Utilization Statistics

To set the process entry limit and the size of the history table for CPU utilization statistics, perform these steps:

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **process cpu statistics limit entry-percentage** *number* [**size** *seconds*]

DETAILED STEPS

	Command or Action	Purpose

Step 1	**`enable`**
	Example:
	`Router> enable`
		Enables privileged EXEC mode.
		• Enter your password if prompted.

Step 2	**`configure terminal`**
	Example:
	`Router# configure terminal`
		Enters global configuration mode.

Step 3	**`process cpu statistics limit entry-percentage`** *`number`* `[`**`size`** *`seconds`*`]`
	Example:
	`Router(config)# process cpu statistics limit entry-percentage 40 size 300`
		Sets the process entry limit and the size of the history table for CPU utilization statistics.
		• In this example, to generate an entry in the history table, a process must exceed 40 percent CPU utilization.
		• In this example, the duration of time for which the most recent history is saved in the history table is 300 seconds.

Configuration Examples for CPU Thresholding Notification

The following examples show how to set a rising and a falling CPU thresholding notification:

- Setting a Rising CPU Thresholding Notification: Example, page 5
- Setting a Falling CPU Thresholding Notification: Example, page 6

Setting a Rising CPU Thresholding Notification: Example

The following example shows how to set a rising CPU thresholding notification for total CPU utilization. When total CPU utilization exceeds 80 percent for a period of 5 seconds or longer, a rising threshold notification is sent.

```
Router(config)# process cpu threshold type total rising 80 interval 5
```

Note When the optional **falling** arguments (*percentage* and *seconds*) are not specified, they take on the same values as the **rising** arguments (*percentage* and *seconds*).

Setting a Falling CPU Thresholding Notification: Example

The following example shows how to set a falling CPU thresholding notification for total CPU utilization. When total CPU utilization, which at one point had risen above 80 percent and triggered a rising threshold notification, falls below 70 percent for a period of 5 seconds or longer, a falling threshold notification is sent.

```
Router(config)# process cpu threshold type total rising 80 interval 5 falling 70
interval 5
```

Note When the optional **falling** arguments (*percentage* and *seconds*) are not specified, they take on the same values as the **rising** arguments (*percentage* and *seconds*).

Additional References

For additional information related to the CPU Thresholding Notification feature, refer to the following references:

Related Documents

Related Topic	Document Title
SNMP traps	Configuration Fundamentals Command Reference

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
CISCO-PROCESS-MIB	To locate and download MIBs for selected platforms, Cisco IOS releases, and feature sets, use Cisco MIB Locator found at the following URL: http://www.cisco.com/go/mibs

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
Technical Assistance Center (TAC) home page, containing 30,000 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/public/support/tac/home.shtml

Command Reference

The following new and modified commands are pertinent to this feature. To see the command pages for these commands and other commands used with this feature, go to the *Cisco IOS Master Commands List*, Release 12.4, at http://www.cisco.com/univercd/cc/td/doc/product/software/ios124/124mindx/124index.htm.

- **process cpu statistics limit entry-percentage**
- **process cpu threshold type**
- **snmp-server enable traps cpu**
- **snmp-server host**

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

Defining Memory Threshold Notifications

First Published: August 21, 2003
Last Updated: May 2, 2008

The Memory Threshold Notifications feature allows you to reserve memory for critical notifications and to configure a router to issue notifications when available memory falls below a specified threshold.

Feature History for the Memory Threshold Notifications Feature

Release	Modification
12.2(18)S	This feature was introduced.
12.0(26)S	This feature was integrated into Cisco IOS Release 12.0(26) S.
12.3(4)T	This feature was integrated into Cisco IOS Release 12.3(4)T.
Cisco IOS XE Release 2.1	This feature was introduced on Cisco ASR 1000 Series Routers.

Finding Support Information for Platforms and Cisco IOS Software Images

Use Cisco Feature Navigator to find information about platform support and Cisco IOS software image support. Access Cisco Feature Navigator at http://www.cisco.com/go/fn. You must have an account on Cisco.com. If you do not have an account or have forgotten your username or password, click **Cancel** at the login dialog box and follow the instructions that appear.

Contents

- Information About Memory Threshold Notifications, page 2
- How to Define Memory Threshold Notifications, page 3
- Configuration Examples for Memory Threshold Notifications, page 4
- Additional References, page 6
- Command Reference, page 7

Americas Headquarters:
Cisco Systems, Inc., 170 West Tasman Drive, San Jose, CA 95134-1706 USA

Information About Memory Threshold Notifications

The Memory Threshold Notifications feature provides two ways to mitigate low-memory conditions on a router: notifications can be sent to indicate that free memory has fallen below a configured threshold, and memory can be reserved to ensure that sufficient memory is available to issue critical notifications. To implement the Memory Threshold Notifications feature, you should understand the following concepts:

- Memory Threshold Notifications, page 2
- Memory Reservation, page 2

Memory Threshold Notifications

Notifications are messages issued by the router. When you specify a memory threshold using the **memory free low-watermark** command, for example, the router issues a notification when available free memory falls below the specified threshold, and again once available free memory rises to 5 percent above the specified threshold. The following are examples of memory threshold notifications:

Available Free Memory Less Than the Specified Threshold

```
000029: *Aug 12 22:31:19.559: %SYS-4-FREEMEMLOW: Free Memory has dropped below 2000k
Pool: Processor  Free: 66814056  freemem_lwm: 204800000
```

Available Free Memory Recovered to More Than the Specified Threshold

```
000032: *Aug 12 22:33:29.411: %SYS-5-FREEMEMRECOVER: Free Memory has recovered 2000k
Pool: Processor  Free: 66813960  freemem_lwm: 0
```

Memory Reservation

Memory reservation for critical operations ensures that management processes, such as event logging, continue to function even when router memory is exhausted.

How to Define Memory Threshold Notifications

This section contains the following procedures:

- Setting a Low Free Memory Threshold, page 3
- Reserving Memory for Critical Notifications, page 3

Setting a Low Free Memory Threshold

To set a low free memory threshold, perform the following steps:

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **memory free low-watermark** {**processor** *threshold* | **io** *threshold*}

DETAILED STEPS

	Command or Action	Purpose

Step 1	`enable`
	Example:
	`Router> enable`
		Enables privileged EXEC mode.
		• Enter your password if prompted.

Step 2	`configure terminal`
	Example:
	`Router# configure terminal`
		Enters global configuration mode.

Step 3	**`memory free low-watermark processor`** *`threshold`*
	or
	`memory free low-watermark io` *`threshold`*
	Example:
	`Router(config)#` **`memory free low-watermark processor 20000`**
	or
	Example:
	`Router(config)#` **`memory free low-watermark io 20000`**
		Specifies a threshold in kilobytes of free processor or input/output (I/O) memory. To view acceptable values for the memory threshold, enter the following command:
		• **memory free low-watermark processor ?**
		or
		• **memory free low-watermark io ?**

Reserving Memory for Critical Notifications

When a router is overloaded by processes, the amount of available memory might fall to levels insufficient for it to issue critical notifications. To reserve a region of memory to be used by the router for the issuing of critical notifications, perform the following steps:

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **memory reserve critical** *kilobytes*

DETAILED STEPS

	Command or Action	Purpose

Step 1	`enable`
	Example:
	`Router> enable`
		Enables privileged EXEC mode.
		• Enter your password if prompted.

Step 2	`configure terminal`
	Example:
	`Router# configure terminal`
		Enters global configuration mode.

Step 3	`memory reserve critical` *`kilobytes`*
	Example:
	`Router(config)# memory reserve critical 1000`
		Reserves the specified amount of memory in kilobytes so that the router can issue critical notifications.
		• The amount of memory reserved for critical notifications cannot exceed 25 percent of total available memory.

Configuration Examples for Memory Threshold Notifications

The following examples show how to configure a router to issue notifications when available memory falls below a specified threshold and how to reserve memory for critical notifications:

- Setting a Low Free Memory Threshold: Examples, page 4
- Reserving Memory for Critical Notifications: Example, page 5

Setting a Low Free Memory Threshold: Examples

The following example specifies a threshold of 20000 KB of free processor memory before the router issues notifications:

Threshold for Free Processor Memory

```
Router(config)# memory free low-watermark processor 20000
```

The following example specifies a threshold of 20000 KB of free I/O memory before the router issues notifications:

Threshold for Free IO Memory

```
Router(config)# memory free low-watermark io 20000
```

If available free memory falls below the specified threshold, the router sends a notification message like this one:

```
000029: *Aug 12 22:31:19.559: %SYS-4-FREEMEMLOW: Free Memory has dropped below 20000k
Pool: Processor  Free: 66814056  freemem_lwm: 204800000
```

Once available free memory rises to above 5 percent of the threshold, another notification message like this is sent:

```
000032: *Aug 12 22:33:29.411: %SYS-5-FREEMEMRECOVER: Free Memory has recovered 20000k
Pool: Processor  Free: 66813960  freemem_lwm: 0
```

Reserving Memory for Critical Notifications: Example

The following example reserves 1000 KB of memory for critical notifications:

```
Router# memory reserved critical 1000
```

Note The amount of memory reserved for critical notifications cannot exceed 25 percent of total available memory.

Additional References

The following sections provide references related to the Memory Threshold Notifications feature:

Related Documents

Related Topic	Document Title
Logging system messages	"Troubleshooting and Fault Management" module

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
No new or modified MIBs are supported by this feature and support for existing MIBs has not been modified by this feature.	To locate and download MIBs for selected platforms, Cisco IOS releases, and feature sets, use Cisco MIB Locator found at the following URL: http://www.cisco.com/go/mibs

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 Support website provides extensive online resources, including documentation and tools for troubleshooting and resolving technical issues with Cisco products and technologies. To receive security and technical information about your products, you can subscribe to various services, such as the Product Alert Tool (accessed from Field Notices), the Cisco Technical Services Newsletter, and Really Simple Syndication (RSS) Feeds. Access to most tools on the Cisco Support website requires a Cisco.com user ID and password.	http://www.cisco.com/techsupport

Command Reference

The following commands are introduced or modified in the feature or features documented in this module. For information about these commands, see the *Cisco IOS Network Management Command Reference* at http://www.cisco.com/en/US/docs/ios/netmgmt/command/reference/nm_book.html. For information about all Cisco IOS commands, go to the Command Lookup Tool at http://tools.cisco.com/Support/CLILookup or to the Cisco IOS Master Commands List.

- **memory free low-watermark**
- **memory reserve critical**

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

Configuring the Event Tracer

Feature History

Release	Modification
12.0(18)S	This feature was introduced.
12.2(8)T	This feature was integrated into Cisco IOS Release 12.2(8)T.

This document describes the Event Tracer feature. It includes the following sections:

- Feature Overview, page 1
- Supported Platforms, page 3
- Supported Standards, MIBs, and RFCs, page 3
- Prerequisites, page 3
- Configuration Tasks, page 4
- Configuration Examples, page 7
- Command Reference, page 8

Feature Overview

The Event Tracer feature provides a binary trace facility for troubleshooting Cisco IOS software. This feature gives Cisco service representatives additional insight into the operation of the Cisco IOS software and can be useful in helping to diagnose problems in the unlikely event of an operating system malfunction or, in the case of redundant systems, route processor switchover.

Note This feature is intended for use as a software diagnostic tool and should be configured only under the direction of a Cisco Technical Assistance Center (TAC) representative.

Event tracing works by reading informational messages from specific Cisco IOS software subsystem components that have been preprogrammed to work with event tracing, and by logging messages from those components into system memory. Trace messages stored in memory can be displayed on the screen or saved to a file for later analysis.

Americas Headquarters:
Cisco Systems, Inc., 170 West Tasman Drive, San Jose, CA 95134-1706 USA

By default, trace messages saved to a file are saved in binary format without applying additional processing or formatting. Saving messages in binary format allows event tracing to collect informational messages faster and for a longer time prior to a system malfunction or processor switchover. Optionally, event trace messages can be saved in ASCII format for additional file processing.

The Event Tracer feature can support multiple traces simultaneously. To do this, the feature assigns a unique ID number to each instance of a trace. This way, all messages associated with a single instance of a trace get the same ID number. Event tracing also applies a timestamp to each trace message, which aids in identifying the message sequence.

The number of trace messages stored in memory for each instance of a trace is configurable up to 65536 entries. As the number of trace messages stored in memory approaches the configured limit, the oldest entries are overwritten with new messages, which continues until the event trace is terminated.

Event tracing can be configured in “one-shot” mode. This is where the current contents of memory for a specified component are discarded and a new trace begins. New trace messages are collected until the message limit is reached, at which point the trace is automatically terminated.

Benefits

Event tracing has a number of benefits to aid in system diagnosis:

Binary Data Format

Event information is saved in binary format without applying any formatting or processing of the information. This results in capturing event information more quickly and for a longer period of time in the moments leading up to a system malfunction or processor switchover. The ability to gather information quickly is also helpful in tracing events that generate a lot of data quickly.

File Storage

Information gathered by the event tracing can be written to a file where it can be saved for further analysis.

Optional ASCII Data Format

Event tracing provides an optional command to save the information in ASCII format.

Multiple Trace Capability

Event tracing can be configured to trace one or more components of the Cisco IOS software simultaneously, depending on the software version running on the networking device.

Restrictions

Event tracing provides a mechanism to help TAC representatives assist Cisco customers in diagnosing certain Cisco IOS software functions. Configuration of this feature on a networking device is recommended only under the direction of a TAC representative. This feature does not produce customer readable data; therefore, it requires the assistance of a TAC representative for proper configuration and analysis.

Supported Platforms

- Cisco 12000 Internet router

Platform Support Through Feature Navigator

Cisco IOS software is packaged in feature sets that support specific platforms. To get updated information regarding platform support for this feature, access Feature Navigator. Feature Navigator dynamically updates the list of supported platforms as new platform support is added for the feature.

Feature Navigator is a web-based tool that enables you to quickly determine which Cisco IOS software images support a specific set of features and which features are supported in a specific Cisco IOS image.

To access Feature Navigator, you must have an account on Cisco.com. If you have forgotten or lost your account information, send a blank e-mail to cco-locksmith@cisco.com. An automatic check will verify that your e-mail address is registered with Cisco.com. If successful, account details with a new random password will be e-mailed to you. If you want to establish an account on Cisco.com, go to http://www.cisco.com/register and follow the directions to establish an account.

Feature Navigator is updated when major Cisco IOS software releases and technology releases occur. As of May 2001, Feature Navigator supports M, T, E, S, and ST releases. You can access Feature Navigator at the following URL:

http://www.cisco.com/go/fn

Supported Standards, MIBs, and RFCs

Standards

None

MIBs

None

To obtain lists of supported MIBs by platform and Cisco IOS release, and to download MIB modules, go to the Cisco MIB website on Cisco.com at the following URL:

http://www.cisco.com/public/sw-center/netmgmt/cmtk/mibs.shtml

RFCs

None

Prerequisites

The list of software components that support event tracing can vary from one Cisco IOS software image to another. And in many cases, depending on the software component, the event tracing functionality is enabled or disabled by default. Knowing what software components support event tracing and knowing the existing state of the component configuration is important in deciding whether to configure event tracing.

To determine whether event tracing has been enabled or disabled by default for a specific component, follow these steps:

Step 1 Use the **monitor event-trace ?** command in global configuration mode to get a list of software components that support event tracing.

```
Router(config)# monitor event-trace ?
```

Step 2 Use the **show monitor event-trace** *component* **all** command to determine whether event tracing is enabled or disabled by default for the component.

```
Router# show monitor event-trace component all
```

Step 3 Use the **show monitor event-trace** *component* **parameters** command to find out the default size of the trace message file for the component.

```
Router# show monitor event-trace component parameters
```

This information can help you in determining your configuration options.

Configuration Tasks

See the following sections for configuration tasks for the Event Tracer feature. Each task in the list is identified as either required or optional.

- Configuring Event Tracing (Optional)
- Configuring the Event Trace Size (Optional)
- Configuring the Event Trace Message File (Optional)
- Verifying Event Trace Operation (Optional)

Follow the instructions in the "Prerequisites" section prior to configuring this feature. If the default configuration information meets your site requirements, no further configuration may be necessary, and you may proceed to the section "Verifying Event Trace Operation."

Configuring Event Tracing

In most cases where Cisco IOS software components support event tracing, the feature is configured by default. For some software components, event tracing is enabled, while for other components event tracing might be disabled. In some cases, a TAC representative may want to change the default settings.

To enable or disable event tracing, use the following commands in global configuration mode:

Command	Purpose

`Router(config)# monitor event-trace component enable`
or
`Router(config)# monitor event-trace component disable`
	Enables or disables event tracing for the specified Cisco IOS software component on the networking device.
	Note Component names are set in the system software and are not configurable. To obtain a list of software components supporting event tracing for this release, use the **monitor event-trace ?** command.

Configuring the Event Trace Size

In most cases where Cisco IOS software components support event tracing, the feature is configured by default. In some cases, such as directed by a TAC representative, you might need to change the size parameter to allow for writing more or fewer trace messages to memory.

To configure the message size parameter, use the following command in global configuration mode:

Command	Purpose
Router(config)# **monitor event-trace** *component* **size** *number*	Configures the size of the trace for the specified component. The number of messages that can be stored in memory for each instance of a trace is configurable up to 65536 entries.

Configuring the Event Trace Message File

To configure the file location where you want to save trace messages, use the following command in global configuration mode:

Command	Purpose
Router(config)# **monitor event-trace** *component* **dump-file** *filename*	Configures the file where the trace messages will be saved. The maximum length of the filename (path:filename) is 100 characters. The path can point to flash memory on the networking device or to a TFTP or FTP server.

Verifying Event Trace Operation

Note Depending on the software component, event tracing is enabled or disabled by default. In either case, the default condition will not be reflected in the output of the **show running-config** command; however, changing any of the settings for a command that has been enable or disabled by default will cause those changes to show up in the output of the **show running-config** command.

Step 1 If you made changes to the event tracing configuration, enter the **show running-config** command in privileged EXEC mode to verify the changes.

```
Router# show running-config
```

Step 2 Enter the **show monitor event-trace** *component* command to verify that event tracing has been enabled or disabled for a component.

In the following example, event tracing has been enabled for the IPC component. Notice that each trace message is numbered sequentially (for example, 3667) and is followed by a the timestamp (derived from the device uptime). Following the timestamp is the component specific message data.

```
Router# show monitor event-trace ipc

3667:  6840.016:Message type:3 Data=0123456789
3668:  6840.016:Message type:4 Data=0123456789
3669:  6841.016:Message type:5 Data=0123456789
3670:  6841.016:Message type:6 Data=0123456
```

To view trace information for all components enabled for event tracing, enter the **show monitor event-trace all-traces** command. In this example, separate output is provided for each event and message numbers are interleaved between the events.

```
Router# show monitor event-trace all-traces

Test1 event trace:
3667: 6840.016:Message type:3 Data=0123456789
3669: 6841.016:Message type:4 Data=0123456789
3671: 6842.016:Message type:5 Data=0123456789
3673: 6843.016:Message type:6 Data=0123456789

Test2 event trace:
3668: 6840.016:Message type:3 Data=0123456789
3670: 6841.016:Message type:4 Data=0123456789
3672: 6842.016:Message type:5 Data=0123456789
3674: 6843.016:Message type:6 Data=0123456789
```

Step 3 Verify that you have properly configured the filename for writing trace messages.

```
Router# monitor event-trace ipc dump
```

Troubleshooting Tips

Event Tracing Does Not Appear to Be Configured in the Running Configuration

Depending on the software component, event tracing is enabled or disabled by default. In either case, the default condition will not be reflected in output of the **show running-config** command; however, changing any of the settings for a command that has been enabled or disabled by default will cause those changes to show up in the output of the **show running-config** command. Changing the condition of the component back to its default state (enabled or disabled), will cause the entry not to appear in the configuration file.

Show Command Output Is Reporting “One or More Entries Lost “

The trace function is not locked while information is being displayed to the console, which means that new trace messages can accumulate in memory. If entries accumulate faster than they can be displayed, some messages can be lost; however, messages will continue to display on the console. If the number of lost messages is excessive, the **show** command will stop displaying messages.

Show Command Output Terminates Unexpectedly

The trace function is not locked while information is being displayed to the console, which means that new trace messages can accumulate in memory. If entries accumulate faster than they can be displayed, some messages can be lost. If the number of lost messages is excessive, the **show** command will stop displaying messages.

Show Command Output Is Reporting That “Tracing Currently Disabled, from EXEC Command”

The Cisco IOS software allows for the subsystem components to define whether support for event tracing is enabled or disabled at boot time. Event tracing allows users to enable or disable event tracing in two ways: using the **monitor event-trace** (EXEC) command in privileged EXEC mode or using the **monitor event-trace** (global) command in global configuration mode. To enable event tracing again in this case, you would enter the **enable** form of either of these commands.

Show Command Output Is Reporting That "Tracing Currently Disabled, from Config Mode"

The Cisco IOS software allows for the subsystem components to define whether support for event tracing is enabled or disabled at boot time. Event tracing allows users to disable event tracing in two ways: using the **monitor event-trace disable** (EXEC) command in privileged EXEC mode or using the **monitor event-trace disable** (global) command in global configuration mode. To enable event tracing again in this case, you would enter the **enable** form of either of these commands.

Event Trace Messages Are Not Being Saved in ASCII Format

By default, the **monitor event-trace** *component* **dump** and **monitor event-trace dump-traces** commands save trace messages in binary format. If you want to save trace messages in ASCII format, use either the **monitor event-trace** *component* **dump pretty** command to write the trace messages for a single event, or the **monitor event-trace dump-traces pretty** command to write trace messages for all event traces currently enabled on the networking device.

Configuration Examples

This section provides the following configuration examples:

- Configuring Event Tracing for One Component Example
- Configuring Event Tracing for Multiple Components Example
- Configuring the Event Trace Size Example
- Configuring the Event Trace Message File Example

Configuring Event Tracing for One Component Example

In the following example, the networking device has been configured to trace IPC component events:

```
monitor event-trace ipc enable
```

Configuring Event Tracing for Multiple Components Example

In the following example, the networking device has been configured to trace IPC and MBUS component events:

```
monitor event-trace ipc enable
monitor event-trace mbus enable
```

Configuring the Event Trace Size Example

In the following example, the size of the IPC trace is set to 4096 entries while the size of the MBUS trace is set to 8192 entries:

```
monitor event-trace ipc size 4096
monitor event-trace mbus size 8192
```

Configuring the Event Trace Message File Example

The following example identifies the files in which to write trace messages. In this example, event tracing has been enabled for both the IPC and MBUS components, the IPC trace messages are written to the ipcdump file in flash memory, while the MBUS trace message files are written to the mbusdump file on the TFTP server.

```
monitor event-trace ipc dump-file slot0:ipcdump
monitor event-trace mbus dump-file TFTP:mbusdump
```

Command Reference

The following new commands are pertinent to this feature. To see the command pages for these commands and other commands used with this feature, go to the *Cisco IOS Master Commands List*, Release 12.4, at http://www.cisco.com/univercd/cc/td/doc/product/software/ios124/124mindx/124index.htm.

New Commands

- **monitor event-trace (EXEC)**
- **monitor event-trace (global)**
- **monitor event-trace dump-traces**
- **show monitor event-trace**

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

Embedded Resource Manager (ERM)

First Published: December 07, 2004
Last Updated: February 27, 2008

The Embedded Resource Manager (ERM) feature allows you to monitor internal system resource utilization for specific resources such as the buffer, memory, and CPU. ERM monitors resource utilization from the perspective of various subsystems within the Cisco IOS software such as resource owners (ROs) and resource users (RUs). ERM allows you to configure threshold values for system resources.

The ERM infrastructure is designed to allow for granular monitoring on a task basis within the Cisco IOS software. Network administrators can define thresholds to create notifications according to the real-time resource consumption. ERM goes beyond simply monitoring for total CPU utilization. Through the use of ERM, network administrators and operators can gain a better understanding of the device's operational characteristics, leading to better insight into system scalability and improved system availability.

Finding Feature Information in This Module

Your Cisco IOS software release may not support all of the features documented in this module. To reach links to specific feature documentation in this module and to see a list of the releases in which each feature is supported, use the "Command Reference" section on page 54.

Finding Support Information for Platforms and Cisco IOS and Catalyst OS Software Images

Use Cisco Feature Navigator to find information about platform support and Cisco IOS and Catalyst OS software image support. To access Cisco Feature Navigator, go to http://www.cisco.com/go/cfn. An account on Cisco.com is not required.

Contents

- Prerequisites for Embedded Resource Manager, page 2
- Restrictions for Embedded Resource Manager, page 2
- Information About Embedded Resource Manager, page 2
- How to Configure Embedded Resource Manager, page 8

Americas Headquarters:
Cisco Systems, Inc., 170 West Tasman Drive, San Jose, CA 95134-1706 USA

- Configuration Examples for Embedded Resource Manager, page 46
- Additional References, page 53
- Command Reference, page 54
- Glossary, page 58

Prerequisites for Embedded Resource Manager

You must be running Cisco IOS Release 12.4(6)T or a later release to use the Packet Memory Reclamation functionality.

Restrictions for Embedded Resource Manager

Additional instructions from a Cisco technical support representative may be required.

Information About Embedded Resource Manager

ERM promotes resource availability by providing the infrastructure to track resource usage.

To configure threshold values for resource manager entities, you should understand the following concepts:

- Benefits of the Embedded Resource Manager, page 2
- Resource Accounting and Thresholds Tracking in ERM, page 3
- System Resources Monitored by the Embedded Resource Manager, page 4
- Resource Policy Templates, page 8

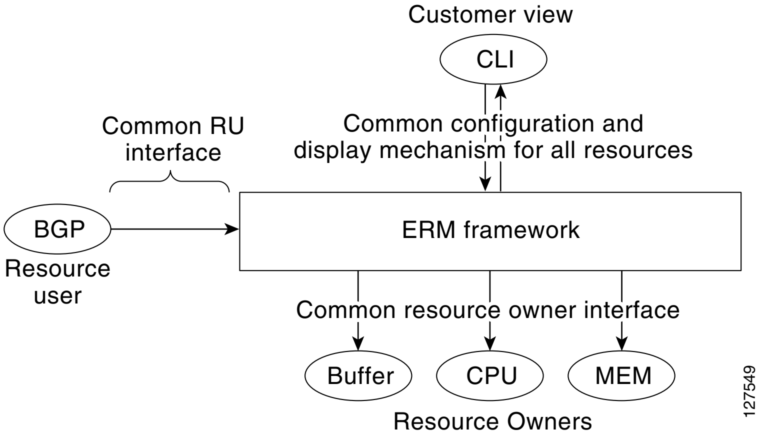
Benefits of the Embedded Resource Manager

The ERM framework tracks resource utilization and resource depletion by monitoring finite resources. Support for monitoring CPU, buffer, and memory utilization at a global or IOS-process level is available.

The ERM framework provides a mechanism to send notifications whenever the specified threshold values are exceeded by any resource user. This notification helps network administrators diagnose any CPU, buffer, and memory utilization issues.

The ERM architecture is illustrated in Figure 1.

Figure 1 ERM Architecture



ERM provides a framework for monitoring any finite resource within the Cisco IOS software and provides information that a user can analyze to better understand how network changes might impact system operation. ERM helps in addressing infrastructure problems such as reloads, memory allocation failure, and high CPU utilization by performing the following functions:

- Monitoring system resource usage.
- Setting the resource threshold at a granular level.
- Generating alerts when resource utilization reaches the specified level.
- Generating internal events using the Cisco IOS Embedded Event Manager feature.

Resource Accounting and Thresholds Tracking in ERM

ERM tracks the resource usage for each RU internally. A RU is a subsystem or process task within the Cisco IOS software; for example, the Open Shortest Path First (OSPF) hello process is a resource user. Threshold limits are used to notify network operators of specific conditions. The ERM infrastructure provides a means to notify the internal RU subsystem of threshold indications as well. The resource accounting is performed by individual ROs. ROs are part of the Cisco IOS software and are responsible for monitoring certain resources such as the memory, CPU, and buffer. When the utilization for each RU exceeds the threshold value you have set, the ROs send internal notifications to the RUs and to network administrators in the form of system logging (syslog) messages or Simple Network Management Protocol (SNMP) alerts.

You can set rising and falling values for critical, major, and minor levels of thresholds. When the resource utilization exceeds the rising threshold level, an Up notification is sent. When the resource utilization falls below the falling threshold level, a Down notification is sent.

ERM provides for three types of thresholds to be defined:

- The System Global Threshold is the point when the entire resource reaches a specified value. A notification is sent to all RUs once the threshold is exceeded.
- The User Local Threshold is the point when a specified RUs utilization exceeds the configured limit.
- The User Global Threshold is the point when the entire resource reaches a configured value. A notification is sent to the specified RU once the threshold is exceeded.

System Resources Monitored by the Embedded Resource Manager

ERM monitors CPU, buffer, and memory utilization at a global and task-based level. To avoid infrastructure issues and promote the availability of system resources, the resource owners described in the following sections are monitored:

- CPU Resource Owner, page 4
- Memory Resource Owner, page 5
- Buffer Resource Owner, page 7

CPU Resource Owner

The ERM feature uses the existing loadometer process to calculate the load information displayed by the **show processes cpu** command. This method generates a report of the extended load statistics and adds it to a circular buffer every five seconds. You can obtain a record of the load statistics for the past one minute through the CLI. This feature also provides an intelligent CPUHOG profiling mechanism that helps to reduce the time required to diagnose error conditions.

The functions described in the following sections help in load monitoring.

- Loadometer Process, page 4
- Scheduler, page 4
- Snapshot Management Using Event Trace, page 4
- Automatic CPUHOG Profiling, page 4

Loadometer Process

The loadometer process generates an extended load monitor report every five seconds. The loadometer function, which calculates process CPU usage percentage, is enhanced to generate the loadometer process reports.

Scheduler

The scheduler collects data when a process is executed, which enables the loadometer to generate reports. The scheduler collects data when the process is launched or when the process transfers control to the scheduler.

Snapshot Management Using Event Trace

Snapshot management manages the buffer in which snapshots of reports are stored. The snapshot management infrastructure stores, displays, and releases the snapshots.

Automatic CPUHOG Profiling

The timer Interrupt Service Routine (ISR) provides automatic CPUHOG profiling. The timer ISR begins profiling a process when it notices that the process has exceeded the configured value or a default of twice the maximum scheduling quantum (maximum time taken for the execution of a task).

On beginning the profiling, the timer ISR saves the interrupted program counter (pc) and return address (ra) in a preallocated buffer. This process provides information that can help the user analyze the CPUHOG.

The profiling continues until the CPUHOG is reported or the buffer is full. To analyze the computation of a long running process you must specify a process ID (PID) and a threshold to start the profiling. When this process takes up more than the specified time (in milliseconds), the profiling begins.

When the data belonging to a particular process exceeds the default size of the buffer, it is reported as a CPUHOG. The default size of the buffer is 1250 entries and can store up to five seconds of profiling data.

Memory Resource Owner

The Embedded Resource Manager feature enhances the memory manager in Cisco IOS devices. The enhancements are described in the following sections:

- Memory Usage History, page 5
- Memory Accounting, page 5
- Interface Wedging and Packet Memory Leaks, page 5
- Memory Resource Reclamation for Interfaces, page 6
- Memory Leak Reclamation, page 6
- I/O Memory, page 6

Memory Usage History

The Embedded Resource Manager feature helps in maintaining memory fragmentation information and thus reduces the need for maintenance of separate scripts for collecting such information.

Memory Accounting

ERM performs the accounting of information for memory by tracking the memory usage of individual RUs. When a process is created, a corresponding RU is also created, against which the usage of memory is recorded. The process of RU creation helps the user to migrate from a process-based accounting to a resource user-based accounting scheme for memory.

The memory RO maintains a global threshold and a per-RU memory usage threshold that can be configured through the ERM infrastructure. The memory RO also tracks the global free memory. When a particular RU's memory usage exceeds the global free memory, a notification is sent to the registered resource monitors (RMs). Similarly when a particular RU exceeds its threshold of memory usage, a notification is sent to that RU. These notifications are sent using the ERM infrastructure.

A memory RO has the intelligence to assign memory to a RU. When a memory RO receives an allocation request, the memory is assigned to the current RU. When a free request is received, the memory RO reduces the memory assigned to the RU.

Interface Wedging and Packet Memory Leaks

In certain situations, errors in the system accounting of incoming packets can occur, leading to a "memory leak" caused by the input queue. When there is a leak in an interface's input queue, gradually the queue reaches its maximum permitted value, causing the interface to become "wedged." A wedged interface may no longer process incoming packets. Packet memory leaks can cause interface input queue wedges.

The Packet Memory Reclamation functionality improves the infrastructure for preventing wedged interface input queues, and it provides a method for changing the defaults of that infrastructure. The Embedded Resource Manager provides the Packet Memory Reclamation functionality for "unwedging" interface input queues and configuring the system to detect and rectify packet leaks.

Note To use the Packet Memory Reclamation functionality, you must be running Cisco IOS Release 12.4(6)T or a later release. Additional troubleshooting (debugging) commands were introduced by this enhancement for use by technical support representatives in specific situations.

Memory Resource Reclamation for Interfaces

The Garbage Detection process works in conjunction with the Memory RO in achieving interface unwedging (for more details, see the *Memory Leak Dectector* feature guide that is part of the *Cisco IOS Configuration Fundamentals Configuration Guide*, Release 12.4T).

As part of the reclamation process, incoming packets that belong to a leaked input queue can be deallocated and reused. This feature provides a command (**critical rising**) that can be used to fine-tune memory resource reclamation.

Note Configuration of this feature will typically be needed only as part of a troubleshooting process with a Cisco Technical Support representative. Additional configuration tasks or special technical support commands may be required before this feature can be effectively used. Additional **memory debug leak internal service** commands are made available to Cisco Technical Support engineers for use in specific situations.

The deallocation procedure is triggered when a check is made to see if packets are using too much memory. Thresholds for the memory RO can be configured using a global policy of any level.

The purpose of configuring this memory policy is to find a balance between the utilization of the Memory Leak Detector (that can become resource intensive) and the need to detect packet memory leaks. Ideally, the system should perform deallocation only when it becomes absolutely necessary.

The **critical rising** command allows you to set a rising and falling threshold percentage for critical levels of I/O memory usage, and to specify an interval for those values. These values trigger the Memory Leak Detector process and, if needed, the deallocation procedure.

For example, if memory usage is more than that of the rising threshold of 75 percent of total I/O memory for more than 5 seconds, the “critical” notification is generated within the system and a callback is issued. As an action in the callback, a check is made to see if the packets are using too much memory. When the packets have used too much memory, the deallocation procedure begins. If the deallocation procedure does not bring memory utilization below the lower threshold value, the deallocation procedure is periodically reattempted. Once the memory usage falls below the configured threshold value, the periodic attempts to deallocate are stopped.

Memory Leak Reclamation

The Packet Memory Reclamation feature uses the ERM infrastructure to clean up and reclaim leaked Cisco IOS packet memory.

This feature uses the Memory Leak Detector process (sometimes referred to as the Garbage Detection or GD process) and the memory-manager RO functionality to reclaim packet memory.

I/O Memory

The I/O memory pool is one of the memory types in Cisco IOS software. The input queue buffers use memory from this pool for processing.

Buffer Resource Owner

The Embedded Resource Manager feature addresses the recurring problems of the Buffer Manager described in the following sections.

- Automatic Buffer Tuning, page 7
- Buffer Leak Detection, page 7
- Buffer Accounting, page 7
- Buffer Usage Thresholding, page 8

Automatic Buffer Tuning

The Embedded Resource Manager feature allows you to automatically tune the buffers using the **buffer tune automatic** command. The buffer RO tunes permanent memory in particle pools based on the usage of the buffer pool.

The buffer RO tracks the number of failures and the availability of memory in the buffer pool. When the number of failures increases above 1 percent of the buffer hits or when no memory is available in the buffer pool, the buffer RO performs an automatic tuning.

Note Ensure that there is sufficient free I/O memory or main memory using the first lines of the **show memory** command output before enabling automatic tuning of buffers.

Here are some keywords from the **buffer tune** command that can help you verify if you have sufficient I/O memory:

- **permanent**: take the number of total buffers in a pool and add 20 percent.
- **min-free**: set the **min-free** keyword to 20 to 30 percent of the permanent number of allocated buffers in the pool.
- **max-free**: set the **max-free** keyword to a value greater than the sum of permanent and minimum values.

However, when there is a traffic burst, the Cisco IOS device may not have enough time to create the new buffers and the number of failures may continue to increase.

The Embedded Resource Manager feature monitors the buffer pool every minute for tuning (that is, for number of hits, number of failures, and the number of counters created). When buffer tuning is enabled, the buffer RO automatically tunes the buffers when required.

Buffer Leak Detection

The Embedded Resource Manager feature allows Cisco IOS devices to detect and diagnose potential buffer leaks. All the buffers in a pool are linked so that they can be traced easily. The number of buffers allocated for incoming and outgoing packets in each buffer pool is tracked and can be displayed in the **show buffers leak** command output.

Buffer Accounting

The Embedded Resource Manager feature consists of mechanisms to account for the usage of buffers. All buffers are owned by the pool manager process (buffer RU). When a RU requests a buffer, the allocated buffer is allotted to that RU. When the RU returns the buffer, it is deducted from the RU’s account. The packet type from the output of the **show buffers usage** command indicates the RU to which the packet belongs.

Buffer Usage Thresholding

The Embedded Resource Manager feature provides a facility to manage high buffer utilization. The buffer manager RO registers as a RU with the memory RO. The buffer manager RU is set before a memory allocation is made for creating new buffers. The buffer manager also registers as an RO. When a buffer is allocated, the current RU (if any) is charged with the memory allocation. The buffer manager RO registers for the notifications from the memory manager for the processor and I/O memory pool. If the I/O memory pool is falling short of memory, the buffer manager tries to free the lists of all the buffer pools. If your Cisco IOS device does not support I/O memory, then it registers for notifications from the processor memory.

Cisco IOS software maintains a threshold per buffer pool. When a particular pool exceeds the specified threshold, ERM sends a notification to all the RUs in that pool, so that the RUs can take corrective measures. Thresholds are configured for public buffer pools only.

Global notification is set for every pool in the system; that is, one notification for all pools in the public pool and one notification for each pool in the private pool. Threshold notifications are sent to only those RUs that have registered with the ROs for getting notifications. A list of RUs that have registered with the RO is maintained by the RO. When the threshold of a particular RU is exceeded, then that RU is notified and marked notified. When the buffers are recovered, the notified RUs are moved back to the original list.

For example, an Ethernet driver RU is allocated buffers from some particular private pool. Another RU, Inter Processor Communication (IPC), is added to the list. In this case, when the pool runs low on buffers, the IPC RU gets a notification and it can take corrective measures.

You can configure threshold values as percentages of the total buffers available in the public pool. Total buffer is the sum of maximum allowed buffers and the permanent pools in the public buffer pool. If these values change due to buffer tuning, then the threshold values also change. For example, if the configuration requires that a notification be sent when the IPC RU is holding more than 40 percent of Ethernet buffers and the sum of permanent and maximum allowed for Ethernet buffers is 150 percent, then the Ethernet pool is notified when the IPC RU is holding 60 percent.

Resource Policy Templates

Resource owner policy is a template used by the ROs to associate a RU with a set of thresholds that are configured through the CLI. This template can be used to specify system global, user local, and per user global thresholds. A particular resource group or RU can have only one policy associated with it. The policy template for ROs is maintained by the ERM framework.

When a policy template is associated with a user type and its instance (RUs), the thresholds configured in that policy are applied based on the RU to RO relationship. This method ignores any RO configuration that may not be applicable to the RU.

How to Configure Embedded Resource Manager

This section contains the following procedures.

- Managing Resource Utilization by Defining Resource Policy, page 9 (required)
- Setting Expected Operating Ranges for Buffer Resources, page 10 (required)
- Setting Expected Operating Ranges for CPU Resources, page 12 (required)
- Setting Expected Operating Ranges for Memory Resources, page 17 (required)

- Enabling Automatic Tuning of Buffers, page 21 (required)
- Managing Memory Usage History, page 22 (required)
- Configuring a CPU Process to Be Included in the Extended Load Monitor Report, page 23 (required)
- Managing Extended CPU Load Monitoring, page 23 (required)
- Managing Automatic CPUHOG Profiling, page 24 (required)
- Applying a Policy to Resource Users, page 25 (optional)
- Setting a Critical Rising Threshold for Global I/O Memory, page 27 (optional)
- Verifying ERM Operations, page 29 (optional)
- Troubleshooting Tips, page 44 (optional)

Managing Resource Utilization by Defining Resource Policy

Perform this task to configure a resource policy for ERM.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **resource policy**
4. **policy** *policy-name* [**global** | **type** *resource-user-type*]

DETAILED STEPS

	Command or Action	Purpose

Step 1	**`enable`**
	Example:
	`Router> enable`
		Enables privileged EXEC mode.
		• Enter your password if prompted.

Step 2	**`configure terminal`**
	Example:
	`Router# configure terminal`
		Enters global configuration mode.

	Command or Action	Purpose
Step 3	**resource policy** **Example:** `Router(config)# resource policy`	Enters ERM configuration mode.
Step 4	**policy** *policy-name* [**global** | **type** *resource-user-type*] **Example:** `Router(config-erm)# policy policy1 type iosprocess`	Enters ERM policy configuration mode to configure a resource policy. • The *policy-name* argument identifies the name of the resource policy. • The **global** keyword is used when you are configuring a system global policy. • The **type** keyword indicates that you are configuring either a user local or per user global policy. The *resource-user-type* argument identifies the name of the resource user type you want to attach the policy to.

Setting Expected Operating Ranges for Buffer Resources

Perform this task to configure threshold values for buffer RO.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **resource policy**
4. **policy** *policy-name* [**global** | **type** *resource-user-type*]
5. **system**
   or
   slot *slot-number*
6. **buffer public**
7. **critical rising** *rising-threshold-value* [**interval** *interval-value*] [**falling** *falling-threshold-value* [**interval** *interval-value*]] [**global**]
   or
   major rising *rising-threshold-value* [**interval** *interval-value*] [**falling** *falling-threshold-value* [**interval** *interval-value*]] [**global**]
   or
   minor rising *rising-threshold-value* [**interval** *interval-value*] [**falling** *falling-threshold-value* [**interval** *interval-value*]] [**global**]
8. **exit**

DETAILED STEPS

	Command or Action	Purpose

Step 1	**`enable`**
	Example:
	`Router> enable`
		Enables privileged EXEC mode.
		• Enter your password if prompted.

Step 2	**`configure terminal`**
	Example:
	`Router# configure terminal`
		Enters global configuration mode.

Step 3	**`resource policy`**
	Example:
	`Router(config)# resource policy`
		Enters ERM configuration mode.

Step 4	**`policy`** *`policy-name`* `[`**`global`** `|` **`type`** *`resource-user-type`*`]`
	Example:
	`Router(config-erm)# policy policy1 type iosprocess`
		Configures a resource policy and enters ERM policy configuration mode.
		• The *policy-name* argument identifies the name of the resource policy.
		• The **global** keyword is used when you are configuring a system global policy.
		• The **type** keyword indicates that you are configuring either a user local or per user global policy. The *resource-user-type* argument identifies the name of the resource user type you want to attach the policy to.

Step 5	**`system`**
	or
	`slot` *`slot-number`*
	Example:
	`Router(config-erm-policy)# system`
	or
	Example:
	`Router(config-erm-policy)# slot 1`
		Enters policy node configuration mode with the **system** command.
		Enters ERM slot configuration mode with the **slot** *slot-number* command. This command is available only in distributed platforms like the Route Switch Processor (RSP).

Step 6	**`buffer public`**
	Example:
	`Router(config-policy-node)# buffer public`
		Enters buffer owner configuration mode.
		Allows you to set the rising and falling values for the critical, major, and minor thresholds.

	Command or Action	Purpose

Step 7	**critical rising** *rising-threshold-value* [**interval** *interval-value*] [**falling** *falling-threshold-value* [**interval** *interval-value*]] [**global**]
	or
	major rising *rising-threshold-value* [**interval** *interval-value*] [**falling** *falling-threshold-value* [**interval** *interval-value*]] [**global**]
	or
	minor rising *rising-threshold-value* [**interval** *interval-value*] [**falling** *falling-threshold-value* [**interval** *interval-value*]] [**global**]
	Example:
	`Router(config-owner-buffer)# critical rising 40 falling 20 interval 10 global`
	or
	Example:
	`Router(config-owner-buffer)# major rising 30 falling 15 interval 10 global`
	or
	Example:
	`Router(config-owner-buffer)# minor rising 20 falling 10 interval 10 global`
		Allows you to set the rising and falling threshold values for critical, major, and minor levels of buffer usage count for the public buffer pools.
		Note If you had configured a global policy in Step 4, you do not need to give the **global** keyword while setting the threshold values in Step 7. However, if you have configured a user local or per user global policy (by not specifying the **global** keyword) in Step 4, enter the **global** keyword in Step 7 if you want to configure a per user global threshold.

Step 8	**exit** **Example:** `Router(config-owner-buffer)# exit`	Exits buffer owner configuration mode.

Setting Expected Operating Ranges for CPU Resources

Perform this task to configure threshold values for the CPU RO.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **resource policy**
4. **policy** *policy-name* [**global** | **type** *resource-user-type*]
5. **system**
   or
   slot *slot-number*
6. **cpu interrupt**

7. **critical rising** *rising-threshold-value* [**interval** *interval-value*] [**falling** *falling-threshold-value* [**interval** *interval-value*]] **global**
   or
   major rising *rising-threshold-value* [**interval** *interval-value*] [**falling** *falling-threshold-value* [**interval** *interval-value*]] **global**
   or
   minor rising *rising-threshold-value* [**interval** *interval-value*] [**falling** *falling-threshold-value* [**interval** *interval-value*]] **global**
8. **exit**
9. **cpu process**
10. **critical rising** *rising-threshold-value* [**interval** *interval-value*] [**falling** *falling-threshold-value* [**interval** *interval-value*]] [**global**]
    or
    major rising *rising-threshold-value* [**interval** *interval-value*] [**falling** *falling-threshold-value* [**interval** *interval-value*]] [**global**]
    or
    minor rising *rising-threshold-value* [**interval** *interval-value*] [**falling** *falling-threshold-value* [**interval** *interval-value*]] [**global**]
11. **exit**
12. **cpu total**
13. **critical rising** *rising-threshold-value* [**interval** *interval-value*] [**falling** *falling-threshold-value* [**interval** *interval-value*]] **global**
    or
    major rising *rising-threshold-value* [**interval** *interval-value*] [**falling** *falling-threshold-value* [**interval** *interval-value*]] **global**
    or
    minor rising *rising-threshold-value* [**interval** *interval-value*] [**falling** *falling-threshold-value* [**interval** *interval-value*]] **global**
14. **exit**

DETAILED STEPS

	Command or Action	Purpose

Step 1	**`enable`**
	Example:
	`Router> enable`
		Enables privileged EXEC mode.
		• Enter your password if prompted.

Step 2	**`configure terminal`**
	Example:
	`Router# configure terminal`
		Enters global configuration mode.

Step 3	**`resource policy`**
	Example:
	`Router(config)# resource policy`
		Enters ERM configuration mode.

	Command or Action	Purpose

Step 4	**`policy`** *`policy-name`* `[`**`global`**` | `**`type`**
	`resource-user-type``]`
	Example:
	`Router(config-erm)# policy policy1 type iosprocess`
		Configures a resource policy and enters ERM policy configuration mode.
		- The *policy-name* argument identifies the name of the resource policy.
		- The **global** keyword is used when you are configuring a system global policy.
		- The **type** keyword indicates that you are configuring either a user local or per user global policy. The *resource-user-type* argument identifies the name of the resource user type you want to attach the policy to.

Step 5	**`system`**
	or
	`slot` *`slot-number`*
	Example:
	`Router(config-erm-policy)# system`
	or
	Example:
	`Router(config-erm-policy)# slot 1`
		Enters policy node configuration mode with the **system** command.
		Enters ERM slot configuration mode with the **slot** *slot-number* command. This command is available only in distributed platforms like the RSP.

Step 6	**`cpu interrupt`**
	Example:
	`Router(config-policy-node)# cpu interrupt`
		(Optional) Enters CPU owner configuration mode.
		Allows you to set the rising and falling values for the critical, major, and minor thresholds.

	Command or Action	Purpose

Step 7	**critical rising** *rising-threshold-value* [**interval** *interval-value*] [**falling** *falling-threshold-value* [**interval** *interval-value*]] **global**
	or
	major rising *rising-threshold-value* [**interval** *interval-value*] [**falling** *falling-threshold-value* [**interval** *interval-value*]] **global**
	or
	minor rising *rising-threshold-value* [**interval** *interval-value*] [**falling** *falling-threshold-value* [**interval** *interval-value*]] **global**
	Example:
	`Router(config-owner-cpu)# critical rising 40 falling 20 interval 10 global`
	or
	Example:
	`Router(config-owner-cpu)# major rising 30 falling 15 interval 10 global`
	or
	Example:
	`Router(config-owner-cpu)# minor rising 20 falling 10 interval 10 global`
		Allows you to set the rising and falling threshold values for critical, major, and minor levels of percentages of CPU interrupt utilization.
		Note If you had configured a global policy in Step 4, you do not need to give the **global** keyword while setting the threshold values in Step 7. However, if you have configured a user local or per user global policy (by not specifying the **global** keyword) in Step 4, enter the **global** keyword in Step 7 if you want to configure a per user global threshold.
		For interrupt CPU utilization, you can configure either global thresholds or per user global thresholds. Hence, you must enter the **global** keyword either in Step 4 or in Step 7.

Step 8	**exit** **Example:** `Router(config-owner-cpu)# exit`	Exits the CPU owner configuration mode.
Step 9	**cpu process** **Example:** `Router(config-policy-node)# cpu process`	(Optional) Enters CPU owner configuration mode. Allows you to set the rising and falling values for the critical, major, and minor thresholds.

	Command or Action	Purpose

Step 10	**critical rising** *rising-threshold-value* [**interval** *interval-value*] [**falling** *falling-threshold-value* [**interval** *interval-value*]] [**global**]
	or
	major rising *rising-threshold-value* [**interval** *interval-value*] [**falling** *falling-threshold-value* [**interval** *interval-value*]] [**global**]
	or
	minor rising *rising-threshold-value* [**interval** *interval-value*] [**falling** *falling-threshold-value* [**interval** *interval-value*]] [**global**]
	Example:
	`Router(config-owner-cpu)# critical rising 40 falling 20 interval 10 global`
	or
	Example:
	`Router(config-owner-cpu)# major rising 30 falling 15 interval 10 global`
	or
	Example:
	`Router(config-owner-cpu)# minor rising 20 falling 10 interval 10 global`
		Allows you to set the rising and falling threshold values for critical, major, and minor levels of percentages of process CPU utilization.
		Note If you had configured a global policy in Step 4, you do not need to give the **global** keyword while setting the threshold values in Step 10. However, if you have configured a user local or per user global policy (by not specifying the **global** keyword) in Step 4, enter the **global** keyword in Step 10 if you want to configure a per user global threshold.
		For process CPU utilization, you can configure global thresholds, per user global thresholds or user local thresholds.

Step 11	**exit** **Example:** `Router(config-owner-cpu)# exit`	Exits the CPU owner configuration mode.
Step 12	**cpu total** **Example:** `Router(config-policy-node)# cpu total`	(Optional) Enters CPU owner configuration mode. Allows you to set the rising and falling values for the critical, major, and minor thresholds.

	Command or Action	Purpose

Step 13	**critical rising** *rising-threshold-value* [**interval** *interval-value*] [**falling** *falling-threshold-value* [**interval** *interval-value*]] **global**
	or
	major rising *rising-threshold-value* [**interval** *interval-value*] [**falling** *falling-threshold-value* [**interval** *interval-value*]] **global**
	or
	minor rising *rising-threshold-value* [**interval** *interval-value*] [**falling** *falling-threshold-value* [**interval** *interval-value*]] **global**
	Example:
	`Router(config-owner-cpu)# critical rising 40 falling 20 interval 10 global`
	or
	Example:
	`Router(config-owner-cpu)# major rising 30 falling 15 interval 10 global`
	or
	Example:
	`Router(config-owner-cpu)# minor rising 20 falling 10 interval 10 global`
		Allows you to set the rising and falling threshold values for critical, major, and minor levels of percentages of total CPU utilization.
		Note If you had configured a global policy in Step 4, you do not need to give the **global** keyword while setting the threshold values in Step 13. However, if you have configured a user local or per user global policy (by not specifying the **global** keyword) in Step 4, enter the **global** keyword in Step 13 if you want to configure a per user global threshold.
		For total CPU utilization, you can configure either global thresholds or per user global thresholds. Hence, you must enter the **global** keyword either in Step 4 or in Step 13.

Step 14	**exit** **Example:** `Router(config-owner-cpu)# exit`	Exits CPU owner configuration mode.

Setting Expected Operating Ranges for Memory Resources

Perform this task to configure threshold values for the memory RO.

Note When the Packet Memory Reclamation functionality is enabled, and the violation of the configured threshold value for the memory RO occurs, the system verifies whether the memory is hogged by the buffers. If 70 percent of the memory is used by the buffers, the system activates the Memory Leak Detector process (sometimes referred to as the “Garbage Detection” or “GD” process) to clean up the memory. (For more details, see the Memory Leak Dectector feature guide that is part of the *Cisco IOS Configuration Fundamentals Configuration Guide*, Release 12.4T).

SUMMARY STEPS

1. **enable**
2. **configure terminal**

3. **resource policy**
4. **policy** *policy-name* [**global** | **type** *resource-user-type*]
5. **system**
or
slot *slot-number*
6. **memory io**
7. **critical rising** *rising-threshold-value* [**interval** *interval-value*] [**falling** *falling-threshold-value* [**interval** *interval-value*]] [**global**]
or
major rising *rising-threshold-value* [**interval** *interval-value*] [**falling** *falling-threshold-value* [**interval** *interval-value*]] [**global**]
or
minor rising *rising-threshold-value* [**interval** *interval-value*] [**falling** *falling-threshold-value* [**interval** *interval-value*]] [**global**]
8. **exit**
9. **memory processor**
10. **critical rising** *rising-threshold-value* [**interval** *interval-value*] [**falling** *falling-threshold-value* [**interval** *interval-value*]] [**global**]
or
major rising *rising-threshold-value* [**interval** *interval-value*] [**falling** *falling-threshold-value* [**interval** *interval-value*]] [**global**]
or
minor rising *rising-threshold-value* [**interval** *interval-value*] [**falling** *falling-threshold-value* [**interval** *interval-value*]] [**global**]
11. **exit**

DETAILED STEPS

	Command or Action	Purpose

Step 1	**`enable`**
	Example:
	`Router> enable`
		Enables privileged EXEC mode.
		• Enter your password if prompted.

Step 2	**`configure terminal`**
	Example:
	`Router# configure terminal`
		Enters global configuration mode.

Step 3	**`resource policy`**
	Example:
	`Router(config)# resource policy`
		Enters ERM configuration mode.

	Command or Action	Purpose
Step 4	`policy` *`policy-name`* `[`**`global`**`	`**`type`** *`resource-user-type`*`]` **Example:** `Router(config-erm)# policy policy1 type iosprocess`	Configures a resource policy and enters ERM policy configuration mode. • The *policy-name* argument identifies the name of the resource policy. • The **global** keyword is used when you are configuring a system global policy. • The **type** keyword indicates that you are configuring either a user local or per user global policy. The *resource-user-type* argument identifies the name of the resource user type you want to attach the policy to.

Step 5	**`system`**
	or
	`slot` *`slot-number`*
	Example:
	`Router(config-erm-policy)# system`
	or
	Example:
	`Router(config-erm-policy)# slot 1`
		Enters policy node configuration mode with the **system** command.
		Enters ERM slot configuration mode with the **slot** *slot-number* command. This command is available only in distributed platforms like the RSP.

Step 6	**`memory io`**
	Example:
	`Router(config-policy-node)# memory io`
		(Optional) Enters memory owner configuration mode.
		Allows you to set the rising and falling values for the critical, major, and minor thresholds.

	Command or Action	Purpose

Step 7	**critical rising** *rising-threshold-value* [**interval** *interval-value*] [**falling** *falling-threshold-value* [**interval** *interval-value*]] [**global**]
	or
	major rising *rising-threshold-value* [**interval** *interval-value*] [**falling** *falling-threshold-value* [**interval** *interval-value*]] [**global**]
	or
	minor rising *rising-threshold-value* [**interval** *interval-value*] [**falling** *falling-threshold-value* [**interval** *interval-value*]] [**global**]
	Example:
	`Router(config-owner-memory)# critical rising 40 falling 20 interval 10 global`
	or
	Example:
	`Router(config-owner-memory)# major rising 30 falling 15 interval 10 global`
	or
	Example:
	`Router(config-owner-memory)# minor rising 20 falling 10 interval 10 global`
		Allows you to set the rising and falling threshold values for critical, major, and minor levels of percentages of I/O memory usage.
		Note If you had configured a global policy in Step 4, you do not need to give the **global** keyword while setting the threshold values in Step 7. However, if you have configured a user local or per user global policy (by not specifying the **global** keyword) in Step 4, enter the **global** keyword in Step 7 if you want to configure a per user global threshold.

Step 8	**exit** **Example:** `Router(config-owner-memory)# exit`	Exits memory owner configuration mode.
Step 9	**memory processor** **Example:** `Router(config-policy-node)# memory processor`	(Optional) Enters memory owner configuration mode. Allows you to set the rising and falling values for the critical, major, and minor thresholds.

	Command or Action	Purpose

Step 10	**critical rising** *rising-threshold-value* [**interval** *interval-value*] [**falling** *falling-threshold-value* [**interval** *interval-value*]] [**global**]
	or
	major rising *rising-threshold-value* [**interval** *interval-value*] [**falling** *falling-threshold-value* [**interval** *interval-value*]] [**global**]
	or
	minor rising *rising-threshold-value* [**interval** *interval-value*] [**falling** *falling-threshold-value* [**interval** *interval-value*]] [**global**]
	Example:
	`Router(config-owner-memory)# critical rising 40 falling 20 interval 10 global`
	or
	Example:
	`Router(config-owner-memory)# major rising 30 falling 15 interval 10 global`
	or
	Example:
	`Router(config-owner-memory)# minor rising 20 falling 10 interval 10 global`
		Allows you to set the rising and falling threshold values for critical, major, and minor levels of percentages of processor memory usage.
		Note If you had configured a global policy in Step 4, you do not need to give the **global** keyword while setting the threshold values in Step 10. However, if you have configured a user local or per user global policy (by not specifying the **global** keyword) in Step 4, enter the global keyword in Step 10 if you want to configure a per user global threshold.

Step 11	**exit** **Example:** `Router(config-owner-memory)# exit`	Exits memory owner configuration mode.

Enabling Automatic Tuning of Buffers

Perform this task to enable automatic tuning of buffers.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **buffer tune automatic**

DETAILED STEPS

	Command or Action	Purpose

Step 1	**`enable`**
	Example:
	`Router> enable`
		Enables privileged EXEC mode.
		• Enter your password if prompted.

Step 2	**`configure terminal`**
	Example:
	`Router# configure terminal`
		Enters global configuration mode.

Step 3	**`buffer tune automatic`**
	Example:
	`Router(config)# buffer tune automatic`
		Enables automatic tuning of buffers.

Managing Memory Usage History

Perform this task to change the number of hours for which the memory log is maintained.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **memory statistics history table** *number-of-hours*

DETAILED STEPS

	Command or Action	Purpose

Step 1	**`enable`**
	Example:
	`Router> enable`
		Enables privileged EXEC mode.
		• Enter your password if prompted.

Step 2	**`configure terminal`**
	Example:
	`Router# configure terminal`
		Enters global configuration mode.

Step 3	**`memory statistics history table`** *`number-of-hours`*
	Example:
	`Router(config)# memory statistics history table 48`
		Changes the time (number of hours) for which the memory log is maintained.

Configuring a CPU Process to Be Included in the Extended Load Monitor Report

Perform this task to configure a process (or processes) to be included in the extended load monitor report.

SUMMARY STEPS

1. **enable**
2. **monitor processes cpu extended** *process-id-list*

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**monitor processes cpu extended** *process-id-list* **Example:** `Router# monitor processes cpu extended 1`	Enables the specified process or processes to be monitored for the extended CPU load. You can specify a maximum of eight processes to be monitored.

Managing Extended CPU Load Monitoring

Perform this task to change the history size in the collection report for extended CPU load.

Restrictions

You cannot disable this feature completely. If the command is not configured, the default behavior is to collect a one-minute history. The one-minute history is equivalent to collecting history for a history size 12.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **process cpu extended history** *history-size*

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**process cpu extended history** *history-size* **Example:** `Router(config)# process cpu extended history 24`	Enables you to change the history size of the extended collection report. If the command is not configured, the default behavior is to collect a one-minute history, which is equivalent to collecting history for history size 12.

Managing Automatic CPUHOG Profiling

Perform this task to enable automatic profiling of CPUHOGs by the CPU Resource Owner. The CPU Resource Owner predicts when a process could hog CPU and begins profiling that process at the same time. This function is enabled by default.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **processes cpu autoprofile hog**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**processes cpu autoprofile hog** **Example:** `Router(config)# processes cpu autoprofile hog`	Enables automatic profiling of CPUHOG processes. This function is enabled by default.

Applying a Policy to Resource Users

Perform this task to apply a policy or policy template to RUs or resource groups.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **resource policy**
4. **policy** *policy-name* [**global** | **type** *resource-user-type*]
5. **exit**
6. **user** {*resource-instance-name resource-user-type resource-policy-name* | **global** *global-policy-name* | **group** *resource-group-name* **type** *resource-user-type*}
7. **instance** *instance-name*
8. **policy** *policy-name*

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**resource policy** **Example:** `Router(config)# resource policy`	Enters ERM configuration mode.
Step 4	**policy** *policy-name* [**global** | **type** *resource-user-type*] **Example:** `Router(config-erm)# policy policy1 type iosprocess`	Configures a resource policy and enters ERM policy configuration mode. • The *policy-name* argument identifies the name of the resource policy. • The **global** keyword is used when you are configuring a system global policy. • The **type** keyword indicates that you are configuring either a user local or per user global policy. The *resource-user-type* argument identifies the name of the resource user type you want to attach the policy to.
Step 5	**exit** **Example:** `Router(config-erm)# exit`	Exits ERM policy configuration mode.

	Command or Action	Purpose

Step 6	**user** {*resource-instance-name resource-user-type resource-policy-name* | **global** *global-policy-name* | **group** *resource-group-name* **type** *resource-user-type*}
	Example:
	`Router(config-erm)# user group lowPrioUsers type iosprocess`
		Applies a policy system wide (global thresholding), a group of users (group thresholding), or a particular user.
		Note When you apply a group policy to a group of RUs by giving the **group** keyword in this command, the Cisco IOS router enters the resource group configuration mode. Go to Step 7 if you want to add RUs to the resource group. Got to Step 8 if you want to apply a policy to the resource group.
		• The *resource-instance-name* argument identifies the name of the RU to which you are applying a policy.
		• The *resource-user-type-name* argument identifies the type of RU.
		• The *resource-policy-name* argument identifies the name resource policy you are applying to the individual RU.
		• The *global-policy-name* argument identifies the name of the global policy you are trying to apply.
		• The *resource-group-name* argument identifies the name of the resource group.

Step 7	**instance** *instance-name* **Example:** `Router(config-res-group)# instance http`	Adds an RU to a resource group. The *instance-name* argument specifies the RU or instance name. **Note** All the RUs added by this command will be grouped together under the resource group and the same thresholding policy will be applied to all the RUs. For example, if you have created a resource group **lowPrioUsers** in Step 6, then all the RUs you add in Step 7 will be part of the resource group **lowPrioUsers** and the same policy is applied to all the RUs.
Step 8	**policy** *policy-name* **Example:** `Router(config-res-group)# policy group-policy1`	Specifies the policy you want to apply to the resource group you created in Step 6. The *policy-name* argument specifies the name of the group policy. This command helps you to set the same threshold policy to a group of RUs grouped under a resource group. For example, if you have some low-priority tasks or RUs like **http** and **snmp** and you want to set a threshold not on these individual RUs, but as a group; then add these RUs to the **lowPrioUsers** group using Step 7 and then apply a threshold policy using Step 8. In this case, if you have set a minor rising threshold of 10 percent (this 10 percent threshold is applied to both **http** and **snmp** in the **lowPrioUsers** group), then a notification is sent to **lowPrioUsers** resource group when the accumulated usage exceeds the 10 percent mark. That is, if http uses 4 percent and snmp uses 7 percent, a notification will be sent to all the RUs in the **lowPrioUsers** resource group.

Setting a Critical Rising Threshold for Global I/O Memory

Perform this task to specify a critical rising threshold value for the global I/O memory pool. If global I/O memory resource consumption meets or exceeds this value, the Memory Leak Detector process will be automatically triggered. This configuration is only needed if you are experiencing a problem and you want to change (fine tune) how often the automatic process occurs (for example, set the threshold lower so that deallocation check occurs more frequently).

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **resource policy**
4. **policy** *policy-name* [**global** | **type** *resource-user-type*]
5. **system**
   or
   slot *slot-number*
6. **memory io**
7. **critical rising** *rising-threshold-value* [**interval** *interval-value*] [**falling** *falling-threshold-value* [**interval** *interval-value*]] [**global**]
8. **exit**

DETAILED STEP

	Command or Action	Purpose

Step 1	`enable`
	Example:
	`Router> enable`
		Enables privileged EXEC mode.
		• Enter your password if prompted.

Step 2	`configure terminal`
	Example:
	`Router# configure terminal`
		Enters global configuration mode.

Step 3	`resource policy`
	Example:
	`Router(config)# resource policy`
		Enters ERM configuration mode.

	Command or Action	Purpose
Step 4	**policy** *policy-name* [**global** | **type** *resource-user-type*] **Example:** `Router(config-erm)# policy policy1 type iosprocess`	Configures a resource policy and enters ERM policy configuration mode. • The *policy-name* argument identifies the name of the resource policy. • The **global** keyword is used when you are configuring a system global policy. • The **type** keyword indicates that you are configuring either a user local or per-user global policy. The *resource-user-type* argument identifies the name of the resource user type you want to attach the policy to.

Step 5	**system**
	or
	slot *slot-number*
	Example:
	`Router(config-erm-policy)# system`
	or
	Example:
	`Router(config-erm-policy)# slot 1`
		Enters policy node configuration mode with the **system** command.
		Enters ERM slot configuration mode with the **slot** *slot-number* command. This command is available only in distributed platforms like RSP.

Step 6	**memory io** **Example:** `Router(config-policy-node)# memory io`	(Optional) Enters memory owner configuration mode. Allows you to set the rising and falling values for the critical, major, and minor thresholds.

	Command or Action	Purpose
Step 7	**critical rising** *rising-threshold-value* [**interval** *interval-value*] [**falling** *falling-threshold-value* [**interval** *interval-value*]] [**global**] **Example:** `Router(config-owner-memory)# critical rising 75 falling 65 interval 10 global`	Allows you to set the rising and falling threshold values for critical levels as percentages of the I/O memory pool, and set the interval of time that must pass before these values are registered. • If the amount of memory held by the resource user exceeds the rising threshold value, a rising threshold notification is generated. • If the falling threshold notification is generated before the interval has passed, then the rising notification is not sent. • The interval following the rising threshold signifies this time period in seconds. • If the amount of memory held by the resource user falls below the falling threshold, the falling threshold notification is sent. • The optional **global** keyword indicates that the threshold is being set on the global memory consumption, not on the memory used by the particular resource user in which the configuration is being applied. **Note** If you had configured a global policy in Step 4, you do not need to give the **global** keyword while setting the threshold values in Step 7. However, if you have configured a user local or per-user global policy (by not specifying the **global** keyword) in Step 4, enter the **global** keyword in Step 7 if you want to configure a per user global threshold.
Step 8	**exit** **Example:** `Router(config-owner-memory)# exit`	Exits memory owner configuration mode.

Verifying ERM Operations

To verify the various ERM operations, perform the following steps.

SUMMARY STEPS

1. **show buffers leak** [**resource user**]
2. **show buffers tune**
3. **show buffers usage** [**pool** *pool-name*]
4. **show memory** [**processor** | **io**] **fragment** [**detail**]
5. **show memory statistics history table**
6. **show monitor event-trace cpu-report** {**brief** {**all** [**detail**] | **back** *time* | **clock** *time* | **from-boot** [*seconds* | **detail**] | **latest** [**detail**]} | **handle** *handle-number*}

7. **show processes cpu autoprofile hog**
8. **show processes cpu extended** [**history**]
9. **show resource all** [**brief** | **detailed**]
10. **show resource database**
11. **show resource owner** {*resource-owner-name* | **all**} **user** {*resource-user-type-name* | **all**} [**brief** | **detailed** | **triggers**]
12. **show resource relationship user** *resource-user-type*
13. **show resource user** {**all** | *resource-user-type*} [**brief** | **detailed**]

DETAILED STEPS

Step 1 **show buffers leak** [**resource user**]

Use this command without the optional keywords to display the details of all the buffers that are older than one minute in the system, for example:

```
Router# show buffers leak

Header   DataArea  Pool  Size  Link  Enc    Flags      Input     Output  User

6488F464  E000084 Small    74    0    0       10       None       None EEM ED Sy
6488FB5C  E000304 Small    74    0    0       10       None       None EEM ED Sy
648905D0  E0006C4 Small    61    0    0        0       None       None EEM ED Sy
648913C0  E000BC4 Small    74    0    0       10       None       None EEM ED Sy
6489173C  E000D04 Small    74    0    0       10       None       None EEM ED Sy
648921B0  E0010C4 Small    60    0    0        0       None       None Init
6489252C  E001204 Small   103    0    0       10       None       None EEM ED Sy
64892C24  E001484 Small    74    0    0       10       None       None EEM ED Sy
64892FA0  E0015C4 Small    74    0    0       10       None       None EEM ED Sy
64893A14  E001984 Small    74    0    0       10       None       None EEM ED Sy
64893D90  E001AC4 Small    61    0    0        0       None       None EEM ED Sy
64894804  E001E84 Small    61    0    0        0       None       None EEM ED Sy
6517CB64  E32F944 Small    74    0    0       10       None       None EEM ED Sy
6517D25C  E176D44 Small    74    0    0       10       None       None EEM ED Sy
6517D5D8  E176E84 Small    74    0    0       10       None       None EEM ED Sy
6517D954  E209A84 Small    74    0    0       10       None       None EEM ED Sy
6517E744  E209D04 Small    61    0    0        0       None       None EEM ED Sy
6517EE3C  E29CBC4 Small    61    0    0        0       None       None EEM ED Sy
65180324  E177844 Small    74    0    0       10       None       None EEM ED Sy
65180D98  E177C04 Small    61    0    0        0       None       None EEM ED Sy
65E1F3A0  E4431A4 Small   102    0    0        0       None       None EEM ED Sy
64895278  E002644 Middl   191    0    0       10       None       None EEM ED Sy
64895CEC  E003004 Middl   173    0    0       10       None       None EEM ED Sy
64896068  E003344 Middl   176    0    0       10       None       None EEM ED Sy
648963E4  E003684 Middl   191    0    0       10       None       None EEM ED Sy
64896E58  E004044 Middl   109    0    0       10       None       None EEM ED Sy
64897C48  E004D44 Middl   194    0    0       10       None       None EEM ED Sy
65181F04  E330844 Middl   173    0    0       10       None       None EEM ED Sy
65183070  E3C3644 Middl   105    0    0       10       None       None EEM ED Sy
65DF9558  E4746E4 Middl   107    0    0        0       None       None EEM ED Sy
65DFA6C4  E475724 Middl   116    0    0        0       None       None EEM ED Sy
65DFADBC  E475DA4 Middl   115    0    0        0       None       None EEM ED Sy
65DFC620  E477464 Middl   110    0    0        0       None       None EEM ED Sy
64C64AE0        0 FS He     0    0    3        0       None       None Init
64C64E5C        0 FS He     0    0    3        0       None       None Init
64C651D8        0 FS He     0    0    3        0       None       None Init
64C65554        0 FS He     0    0    0        0       None       None Init
64C658D0        0 FS He     0    0    0        0       None       None Init
```

```
64C65C4C        0 FS He     0    0    0        0       None       None Init
64C65FC8        0 FS He     0    0    0        0       None       None Init
64C66344        0 FS He     0    0    0        0       None       None Init
64D6164C        0 FS He     0    0    0        0       None       None Init
64EB9D10        0 FS He     0    0    0        0       None       None Init
6523EE14        0 FS He     0    0    0        0       None       None Init
65413648        0 FS He     0    0    0        0       None       None Init
```

Use this command with the optional keywords to display the details of the buffers of a specified RU that are older than one minute in the system, for example:

```
Router# show buffers leak resource user

Resource User:  EEM ED Syslog count:       32
Resource User:           Init count:        2
Resource User:         *Dead* count:        2
Resource User: IPC Seat Manag count:       11
Resource User:      XDR mcast count:        2
```

Step 2 **show buffers tune**

Use this command to display the details of automatic tuning of buffers, for example:

```
Router# show buffers tune

Tuning happened for the pool Small

Tuning happened at 20:47:25
Oldvalues
permanent:50  minfree:20  maxfree:150
Newvalues
permanet:61  minfree:15  maxfree:76

Tuning happened for the pool Middle
Tuning happened at 20:47:25
Oldvalues
permanent:25  minfree:10  maxfree:150
Newvalues
permanet:36  minfree:9  maxfree:45
```

Step 3 **show buffers usage** [**pool** *pool-name*]

Use this command without the optional keyword and argument to display the details of the buffer usage pattern in a specified buffer pool, for example:

```
Router# show buffers usage

Statistics for the Small pool
Caller pc    : 0x626BA9E0 count:       20
Resource User: EEM ED Sys count:       20
Caller pc    : 0x60C71F8C count:        1
Resource User:       Init count:        1
Number of Buffers used by packets generated by system:   62
Number of Buffers used by incoming packets:               0

Statistics for the Middle pool
Caller pc    : 0x626BA9E0 count:       12
Resource User: EEM ED Sys count:       12
Number of Buffers used by packets generated by system:   41
Number of Buffers used by incoming packets:               0

Statistics for the Big pool
Number of Buffers used by packets generated by system:   50
Number of Buffers used by incoming packets:               0
```

```
Statistics for the VeryBig pool
Number of Buffers used by packets generated by system:   10
Number of Buffers used by incoming packets:               0

Statistics for the Large pool
Number of Buffers used by packets generated by system:    0
Number of Buffers used by incoming packets:               0

Statistics for the Huge pool
Number of Buffers used by packets generated by system:    0
Number of Buffers used by incoming packets:               0

Statistics for the IPC pool
Number of Buffers used by packets generated by system:    2
Number of Buffers used by incoming packets:               0

Statistics for the Header pool
Number of Buffers used by packets generated by system:  511
Number of Buffers used by incoming packets:               0

Statistics for the FS Header pool
Caller pc    : 0x608F68FC count:        9
Resource User:       Init count:       12
Caller pc    : 0x61A21D3C count:        1
Caller pc    : 0x60643FF8 count:        1
Caller pc    : 0x61C526C4 count:        1
Number of Buffers used by packets generated by system:   28
Number of Buffers used by incoming packets:               0
```

Use this command with the optional keyword and argument to display the details of the buffer usage pattern in a small buffer pool, for example:

```
Router# show buffers usage pool small

Statistics for the Small pool
Caller pc    : 0x626BA9E0 count:       20
Resource User: EEM ED Sys count:       20
Caller pc    : 0x60C71F8C count:        1
Resource User:       Init count:        1
Number of Buffers used by packets generated by system:   62
Number of Buffers used by incoming packets:               0
```

Step 4 **show memory** [**processor** | **io**] **fragment** [**detail**]

Use this command without the optional keywords to display the block details of every allocated block for both I/O memory and processor memory, for example:

```
Router# show memory fragment

Processor memory

Free memory size : 211014448 Number of free blocks:      139
Allocator PC Summary for allocated blocks in pool: Processor

    PC          Total     Count  Name
0x6189A438     318520       1  RTPSPI
0x6205711C     237024       2  CCH323_CT
0x6080BE38      98416       2  Exec
0x606AD988      80256       1  Init
0x618F68A8      73784       1  CCSIP_UDP_SOCKET
0x6195AD04      67640       1  QOS_MODULE_MAIN
0x606488C8      65592       1  CEF: Adjacency chunk
0x60635620      65592       1  CEF: 16 path chunk pool
0x615ECE58      65592       1  XTagATM VC chunk
```

```
0x6165ACF8      65592       1  eddri_self_event
0x608DE168      65592       1  MallocLite
0x60857920      51020      11  Normal
0x6203BF88      42480       4  IPv6 CEF fib tables
0x60DC7F14      32824       1  PPP Context Chunks
.
.
.
I/O memory

Free memory size : 14700024 Number of free blocks:       52
Allocator PC Summary for allocated blocks in pool: I/O

    PC          Total   Count   Name
0x60857934    3936000      60  FastEthernet0/
0x60857898     524800       8  FastEthernet0/0
0x601263CC      29120       7  Init
0x6082DB28       9408      23  *Packet Data*
0x60126344       8448       4  Init

Allocator PC Summary for free blocks in pool: I/O

    PC          Total    Count   Name
0x608C5730   29391444       1  (coalesced)
0x608FC1F4       5376      28  (fragment)
0x6082DB28       4288      14  (fragment)
```

Use this command with the **detail** optional keyword to display the block details of every allocated block for both I/O memory and processor memory, for example:

```
Router# show memory fragment detail

Processor memory

Free memory size : 211038812 Number of free blocks:      139
 Address      Bytes      Prev     Next Ref     PrevF    NextF Alloc PC  what
644AAB70 0000001032 644AAB20 644AAFAC 001  -------- -------- 620450F8  Index Table Block
644AAFAC 0000000028 644AAB70 644AAFFC 000  0        6448CB5C 607B2ADC  NameDB String
644AAFFC 0000000076 644AAFAC 644AB07C 001  -------- -------- 60818DE0  Init
6448CB0C 0000000028 6448CABC 6448CB5C 001  -------- -------- 607F8380  Cond Debug 
definition
6448CB5C 0000000028 6448CB0C 6448CBAC 000  644AAFAC 6489F158 607B2ADC  NameDB String
6448CBAC 0000000028 6448CB5C 6448CBFC 001  -------- -------- 607F8380  Cond Debug 
definition
6489EF8C 0000000408 6489DBCC 6489F158 001  -------- -------- 60857920  Normal
6489F158 0000000064 6489EF8C 6489F1CC 000  6448CB5C 6448CABC 607B2ADC  NameDB String
6489F1CC 0000005004 6489F158 648A058C 001  -------- -------- 60857920  Normal
6448CA6C 0000000028 6448C9AC 6448CABC 001  -------- -------- 607D72FC  Parser Linkage
6448CABC 0000000028 6448CA6C 6448CB0C 000  6489F158 644949C8 607B2ADC  NameDB String
6448CB0C 0000000028 6448CABC 6448CB5C 001  -------- -------- 607F8380  Cond Debug 
definition
64494978 0000000028 64494928 644949C8 001  -------- -------- 607D72FC  Parser Linkage
644949C8 0000000028 64494978 64494A18 000  6448CABC 654F2868 607B2ADC  NameDB String
64494A18 0000000028 644949C8 64494A68 001  -------- -------- 607D72FC  Parser Linkage
654F27E8 0000000076 654F2768 654F2868 001  -------- -------- 60818DE0  Init
654F2868 0000000076 654F27E8 654F28E8 000  644949C8 654F1BE8 60818DE0  Init
.
.
.
I/O memory

Free memory size : 14700024 Number of free blocks:       52
 Address      Bytes      Prev     Next Ref     PrevF    NextF Alloc PC  what
0E000000 0000000056 00000000 0E00006C 000  0        E176F4C  00000000  (fragment)
```

```
0E00006C 0000000268 0E000000 0E0001AC 001  -------- -------- 6082DB28  *Packet Data*
0E176E0C 0000000268 0E176CCC 0E176F4C 001  -------- -------- 6082DB28  *Packet Data*
0E176F4C 0000000076 0E176E0C 0E176FCC 000  E000000  E209F4C  6082DB28  (fragment)
0E176FCC 0000002060 0E176F4C 0E17780C 001  -------- -------- 60126344  Init
0E209E0C 0000000268 0E209CCC 0E209F4C 001  -------- -------- 6082DB28  *Packet Data*
0E209F4C 0000000076 0E209E0C 0E209FCC 000  E176F4C  E29CF4C  6082DB28  (fragment)
0E209FCC 0000002060 0E209F4C 0E20A80C 001  -------- -------- 60126344  Init
0E29CE0C 0000000268 0E29CCCC 0E29CF4C 001  -------- -------- 6082DB28  *Packet Data*
0E29CF4C 0000000076 0E29CE0C 0E29CFCC 000  E209F4C  E32FF4C  6082DB28  (fragment)
0E29CFCC 0000002060 0E29CF4C 0E29D80C 001  -------- -------- 60126344  Init
0E32FE0C 0000000268 0E32FCCC 0E32FF4C 001  -------- -------- 6082DB28  *Packet Data*
0E32FF4C 0000000076 0E32FE0C 0E32FFCC 000  E29CF4C  0        6082DB28  (fragment)
0E32FFCC 0000002060 0E32FF4C 0E33080C 001  -------- -------- 60126344  Init
0E177FCC 0000004108 0E177E4C 0E17900C 001  -------- -------- 601263CC  Init
0E17900C 0000000140 0E177FCC 0E1790CC 000  0        E18910C  601263CC  (fragment)
```

Use this command with **detail** optional keyword to display the block details of every allocated block for processor memory, for example:

```
Router# show memory processor fragment detail

Processor memory

Free memory size : 65566148 Number of free blocks:      230
 Address      Bytes     Prev     Next Ref     PrevF    NextF Alloc PC  what
645A8148 0000000028 645A80F0 645A8194 001  -------- -------- 60695B20  Init
645A8194 0000000040 645A8148 645A81EC 000  0        200B4300 606B9614  NameDB String
645A81EC 0000000260 645A8194 645A8320 001  -------- -------- 607C2D20  Init
200B42B4 0000000028 200B4268 200B4300 001  -------- -------- 62366C80  Init
200B4300 0000000028 200B42B4 200B434C 000  645A8194 6490F7E8 60976574  AAA Event Data
200B434C 0000002004 200B4300 200B4B50 001  -------- -------- 6267D294  Coproc Request
Structures
6490F79C 0000000028 6490F748 6490F7E8 001  -------- -------- 606DDA04  Parser Linkage
6490F7E8 0000000028 6490F79C 6490F834 000  200B4300 6491120C 606DD8D8  Init
6490F834 0000006004 6490F7E8 64910FD8 001  -------- -------- 607DF5BC  Process Stack
649111A0 0000000060 64911154 6491120C 001  -------- -------- 606DE82C  Parser Mode
6491120C 0000000028 649111A0 64911258 000  6490F7E8 500770F0 606DD8D8  Init
64911258 0000000200 6491120C 64911350 001  -------- -------- 603F0E38  Init
.
20000000 0000000828 5C3AEB24 2000036C 001  -------- -------- 60734010  *Packet Header*
6500BF94 0000000828 6500BC28 6500C300 001  -------- -------- 60734010  *Packet Header*
6500C300 0004760912 6500BF94 50000000 000  5C3AEB24 2C42E310 6071253C  (coalesced)
50000000 0000000828 6500C300 5000036C 001  -------- -------- 60734010  *Packet Header*
2C42E0B4 0000000556 2C429430 2C42E310 001  -------- -------- 60D4A0B4  Virtual Exec
2C42E310 0062725312 2C42E0B4 00000000 000  6500C300 0        6071253C  (coalesced)
```

Use this command with **detail** optional keyword to display the block details of every allocated block for I/O memory, for example:

```
Router# show memory io fragment detail

0E3F8BAC 0000000204 0E3F8AAC 0E3F8CAC 001  -------- -------- 608C5730  test memory
0E3F8CAC 0000000204 0E3F8BAC 0E3F8DAC 000  0        E3F8AAC  608C5730  test memory
0E3F8DAC 0000000204 0E3F8CAC 0E3F8EAC 001  -------- -------- 608C5730  test memory
0E3F89AC 0000000204 0E3F88AC 0E3F8AAC 001  -------- -------- 608C5730  test memory
0E3F8AAC 0000000204 0E3F89AC 0E3F8BAC 000  E3F8CAC  E3F88AC  608C5730  test memory
0E3F8BAC 0000000204 0E3F8AAC 0E3F8CAC 001  -------- -------- 608C5730  test memory
0E3F87AC 0000000204 0E3F86AC 0E3F88AC 001  -------- -------- 608C5730  test memory
0E3F88AC 0000000204 0E3F87AC 0E3F89AC 000  E3F8AAC  E3F86AC  608C5730  test memory
0E3F89AC 0000000204 0E3F88AC 0E3F8AAC 001  -------- -------- 608C5730  test memory
0E3F85AC 0000000204 0E3F826C 0E3F86AC 001  -------- -------- 608C5730  test memory
0E3F86AC 0000000204 0E3F85AC 0E3F87AC 000  E3F88AC  0        608C5730  test memory
0E3F87AC 0000000204 0E3F86AC 0E3F88AC 001  -------- -------- 608C5730  test memory
0E3F4E6C 0000000268 0E3F4D2C 0E3F4FAC 000  0        E3F5BEC  608C5730  test memory
```

```
0E3F5BEC 0000000268 0E3F5AAC 0E3F5D2C 000  E3F4E6C  E3EE56C  608C5730  test memory
0E3EE46C 0000000204 0E3EE12C 0E3EE56C 001  -------- -------- 608C5730  test memory
0E3EEFAC 0000000204 0E3EEE6C 0E3EF0AC 001  -------- -------- 608C5730  test memory
0E3F06EC 0000000204 0E3F03AC 0E3F07EC 001  -------- -------- 608C5730  test memory
0E3F8DAC 0000000204 0E3F8CAC 0E3F8EAC 001  -------- -------- 608C5730  test memory
```

Step 5 **show memory statistics history table**

Use this command to display the history of memory consumption, for example:

```
Router# show memory statistics history table

History for Processor memory

Time: 15:48:56.806
Used(b): 422748036 Largest(b): 381064952 Free blocks :291
Maximum memory users for this period
Process Name           Holding   Num Alloc
Virtual Exec             26992          37
TCP Protocols            14460           6
IP Input                  1212           1

Time: 14:42:54.506
Used(b): 422705876 Largest(b): 381064952 Free blocks :296
Maximum memory users for this period
Process Name           Holding   Num Alloc
Exec                  400012740          24
Dead                   1753456          90
Pool Manager            212796         257

Time: 13:37:26.918
Used(b): 20700520 Largest(b): 381064952 Free blocks :196
Maximum memory users for this period
Process Name           Holding   Num Alloc
Exec                      8372           5

Time: 12:39:44.422
Used(b): 20701436 Largest(b): 381064952 Free blocks :193

Time: 11:46:25.135
Used(b): 20701436 Largest(b): 381064952 Free blocks :193
Maximum memory users for this period
Process Name           Holding   Num Alloc
CDP Protocol              3752          25
.
.
.
History for I/O memory

Time: 15:48:56.809
Used(b):  7455520 Largest(b): 59370080 Free blocks :164

Time: 14:42:54.508
Used(b):  7458064 Largest(b): 59370080 Free blocks :165
Maximum memory users for this period
Process Name           Holding   Num Alloc
Pool Manager            141584         257

Time: 13:37:26.920
Used(b):  7297744 Largest(b): 59797664 Free blocks :25

Time: 12:39:44.424
Used(b):  7297744 Largest(b): 59797664 Free blocks :25
.
```

```
.
.
Time: 09:38:53.040
Used(b):  7297744 Largest(b): 59797664 Free blocks :25

Time: 01:02:05.533
Used(b):  7308336 Largest(b): 59797664 Free blocks :23

Time: 00:00:17.937
Used(b):  7308336 Largest(b): 59797664 Free blocks :23
Maximum memory users for this period
Process Name           Holding   Num Alloc
Init                   7296000         214
Pool Manager               816           3
```

Step 6 **show monitor event-trace cpu-report** {**brief** {**all** [**detail**] | **back** *time* | **clock** *time* | **from-boot** [*seconds* | **detail**] | **latest** [**detail**]} | **handle** *handle-number*}

Use this command to view a brief CPU report details for event tracing on a networking device, for example:

```
Router# show monitor event-trace cpu-report brief all

Timestamp   : Handle Name               Description
00:01:07.320:  1     CPU                None
```

Use this command to view a brief CPU report details for event tracing on a networking device, for example:

```
Router# show monitor event-trace cpu-report handle 1

00:01:07.320:  1     CPU                None
################################################################################
Global Statistics
-----------------
5 sec CPU util 0%/0% Timestamp 21:03:56
Queue Statistics
----------------
             Exec Count    Total CPU  Response Time       Queue Length
                                       (avg/max)            (avg/max)
Critical           1          0          0/0                   1/1
High               5          0          0/0                   1/1
Normal           178          0          0/0                   2/9
Low               15          0          0/0                   2/3
Common Process Information
-------------------------------
 PID   Name         Prio Style
-------------------------------
  10 AAA high-capacit M  New
 133 RADIUS TEST CMD  M  New
  47 VNM DSPRM MAIN   H  New
  58 TurboACL         M  New
  97 IP Background    M  New
  99 CEF: IPv4 proces L  New
 112 X.25 Background  M  New
 117 LFDp Input Proc  M  New
   3 Init             M  Old
CPU Intensive processes
-------------------------------------------------------------------------------
 PID Total       Exec    Quant         Burst  Burst size  Schedcall  Schedcall
     CPUms      Count   avg/max        Count avg/max(ms)      Count  Per avg/max
-------------------------------------------------------------------------------
   3   820          6   136/236         1     24/24              18  887/15172
Priority Suspends
```

```
------------------------------------
 PID Exec Count Prio-Susps
------------------------------------
   3          6          1

Latencies
-------------------------
 PID     Exec Count Latency
                     avg/max
-------------------------
  10          1 15192/15192
 133          1 15192/15192
  58          1 15192/15192
 112          1 15192/15192
 117          1 15192/15192
  99          1 15172/15172
  47          1 15172/15172
  97          1 15172/15172
################################################################################
################################################################################
Global Statistics
-----------------
5 sec CPU util 0%/0% Timestamp 00:00:00
Queue Statistics
----------------
         Exec Count  Total CPU    Response Time           Queue Length
                                    (avg/max)                (avg/max)
Critical      0          0              0/0                     0/0
High          0          0              0/0                     0/0
Normal        0          0              0/0                     0/0
Low           0          0              0/0                     0/0

Common Process Information
-------------------------------
 PID Name            Prio Style
-------------------------------

CPU Intensive processes
-------------------------------------------------------------------------------
 PID Total       Exec    Quant         Burst  Burst size  Schedcall  Schedcall
     CPUms      Count   avg/max        Count avg/max(ms)      Count Per avg/max
-------------------------------------------------------------------------------
Priority Suspends
------------------------------------
 PID Exec Count Prio-Susps
------------------------------------
Latencies
-------------------------
 PID Exec Count   Latency
                  avg/max
-------------------------
################################################################################
```

Step 7 **show processes cpu autoprofile hog**

Use this command to view the CPUHOG autoprofile data, for example:

```
Router# show processes cpu autoprofile hog

0x6075DD40 0x60755638
0x6075DD24 0x60755638
0x6075563C 0x60755638
0x60755638 0x60755638
0x60755638 0x60755638
```

```
0x6075DD10 0x60755638
0x6075DD40 0x60755638
0x6075DD40 0x60755638
0x6075563C 0x60755638
0x6075DCE0 0x60755638
0x6075DD44 0x60755638
0x6075DCCC 0x60755638
0x6075DD10 0x60755638
.
.
.
0x6075DD3C 0x60755638
0x6075DD38 0x60755638
0x6075DD10 0x60755638
0x6075DCCC 0x60755638
0x6075DCDC 0x60755638
0x6075563C 0x60755638
0x6075DD3C 0x60755638
0x6075DD20 0x60755638
0x6075DD58 0x60755638
0x6075DD1C 0x60755638
0x6075DD10 0x60755638
0x6075DCDC 0x60755638
0x6075DCF8 0x60755638
```

Step 8 **show processes cpu extended** [**history**]

Use this command to view an extended CPU load report, for example:

```
Router# show processes cpu extended

#################################################################################
Global Statistics
-----------------
5 sec CPU util 0%/0% Timestamp 21:03:56
Queue Statistics
----------------

             Exec Count  Total CPU    Response Time          Queue Length
                                       (avg/max)              (avg/max)
Critical           1          0          0/0                    1/1
High               5          0          0/0                    1/1
Normal           178          0          0/0                    2/9
Low               15          0          0/0                    2/3
Common Process Information
-------------------------------
 PID Name            Prio Style
-------------------------------
CPU Intensive processes
--------------------------------------------------------------------------------
 PID Total       Exec    Quant         Burst  Burst size  Schedcall   Schedcall
     CPUms      Count   avg/max        Count avg/max(ms)       Count Per avg/max
--------------------------------------------------------------------------------
Priority Suspends
------------------------------------
 PID Exec Count Prio-Susps
------------------------------------
Latencies
-------------------------
 PID Exec Count   Latency
                  avg/max
-------------------------

#################################################################################
```

Step 9 **show resource all** [**brief** | **detailed**]

Use this command without the optional keywords to display the resource details, for example:

```
Router# show resource all

Resource Owner: cpu
Resource User Type: iosprocess
Resource User: Init(ID: 0x1000001)
    RUID Runtime(ms)   Invoked      uSecs   5Sec   1Min   5Min Res Usr
16777217           0         0          0  0.00%  0.00%  0.00% Init
  Resource User: Scheduler(ID: 0x1000002)
    RUID Runtime(ms)   Invoked      uSecs   5Sec   1Min   5Min Res Usr
16777218           0         0          0  0.00%  0.00%  0.00% Scheduler
  Resource User: Dead(ID: 0x1000003)
    RUID Runtime(ms)   Invoked      uSecs   5Sec   1Min   5Min Res Usr
16777219           0         0          0  0.00%  0.00%  0.00% Dead
  Resource User: Interrupt(ID: 0x1000004)
    RUID Runtime(ms)   Invoked      uSecs   5Sec   1Min   5Min Res Usr
16777220           0         0          0  0.00%  0.00%  0.00% Interrupt
  Resource User: Memory RO RU(ID: 0x1000005)
    RUID Runtime(ms)   Invoked      uSecs   5Sec   1Min   5Min Res Usr
16777221           0         0          0  0.00%  0.00%  0.00% Memory RO RU
  Resource User: Chunk Manager(ID: 0x1000006)
    RUID Runtime(ms)   Invoked      uSecs   5Sec   1Min   5Min Res Usr
16777222           0        13          0  0.00%  0.00%  0.00% Chunk Manager
  Resource User: Load Meter(ID: 0x1000007)
    RUID Runtime(ms)   Invoked      uSecs   5Sec   1Min   5Min Res Usr
16777223        2872     36029         79  0.00%  0.00%  0.00% Load Meter
  Resource User: Check heaps(ID: 0x1000009)
    RUID Runtime(ms)   Invoked      uSecs   5Sec   1Min   5Min Res Usr
16777225      352744     33446      10546  0.00%  0.20%  0.17% Check heaps
  Resource User: Pool Manager(ID: 0x100000A)
    RUID Runtime(ms)   Invoked      uSecs   5Sec   1Min   5Min Res Usr
16777226           0         1          0  0.00%  0.00%  0.00% Pool Manager
  Resource User: Buffer RO RU(ID: 0x100000B)
    RUID Runtime(ms)   Invoked      uSecs   5Sec   1Min   5Min Res Usr
16777227           0         0          0  0.00%  0.00%  0.00% Buffer RO RU
  Resource User: Timers(ID: 0x100000C)
    RUID Runtime(ms)   Invoked      uSecs   5Sec   1Min   5Min Res Usr
16777228           0         2          0  0.00%  0.00%  0.00% Timers
  Resource User: Serial Background(ID: 0x100000D)
    RUID Runtime(ms)   Invoked      uSecs   5Sec   1Min   5Min Res Usr
16777229           0         2          0  0.00%  0.00%  0.00% Serial Backgroun
  Resource User: AAA_SERVER_DEADTIME(ID: 0x100000E)
    RUID Runtime(ms)   Invoked      uSecs   5Sec   1Min   5Min Res Usr
16777230           0         1          0  0.00%  0.00%  0.00% AAA_SERVER_DEADT
  Resource User: AAA high-capacity counters(ID: 0x100000F)
    RUID Runtime(ms)   Invoked      uSecs   5Sec   1Min   5Min Res Usr
16777231           0         2          0  0.00%  0.00%  0.00% AAA high-capacit
  Resource User: Policy Manager(ID: 0x1000010)
    RUID Runtime(ms)   Invoked      uSecs   5Sec   1Min   5Min Res Usr
16777232           0         1          0  0.00%  0.00%  0.00% Policy Manager
  Resource User: Crash writer(ID: 0x1000011)
    RUID Runtime(ms)   Invoked      uSecs   5Sec   1Min   5Min Res Usr
16777233           0         1          0  0.00%  0.00%  0.00% Crash writer
  Resource User: RO Notify Timers(ID: 0x1000012)
    RUID Runtime(ms)   Invoked      uSecs   5Sec   1Min   5Min Res Usr
16777234           0         1          0  0.00%  0.00%  0.00% RO Notify Timers
  Resource User: RMI RM Notify Watched Policy(ID: 0x1000013)
    RUID Runtime(ms)   Invoked      uSecs   5Sec   1Min   5Min Res Usr
16777235           0         1          0  0.00%  0.00%  0.00% RMI RM Notify Wa
  Resource User: EnvMon(ID: 0x1000014)
```

```
    RUID Runtime(ms)   Invoked      uSecs   5Sec   1Min   5Min Res Usr
16777236       11164     92859        120  0.00%  0.00%  0.00% EnvMon
  Resource User: IPC Dynamic Cache(ID: 0x1000015)
    RUID Runtime(ms)   Invoked      uSecs   5Sec   1Min   5Min Res Usr
16777237           0      3004          0  0.00%  0.00%  0.00% IPC Dynamic Cach
  Resource User: IPC Periodic Timer(ID: 0x1000017)
    RUID Runtime(ms)   Invoked      uSecs   5Sec   1Min   5Min Res Usr
16777239           0    180082          0  0.00%  0.00%  0.00% IPC Periodic Tim
  Resource User: IPC Managed Timer(ID: 0x1000018)
    RUID Runtime(ms)   Invoked      uSecs   5Sec   1Min   5Min Res Usr
16777240         572     79749          7  0.00%  0.00%  0.00% IPC Managed Time
  Resource User: IPC Deferred Port Closure(ID: 0x1000019)
    RUID Runtime(ms)   Invoked      uSecs   5Sec   1Min   5Min Res Usr
16777241           4    180088          0  0.00%  0.00%  0.00% IPC Deferred Por
  Resource User: IPC Seat Manager(ID: 0x100001A)
    RUID Runtime(ms)   Invoked      uSecs   5Sec   1Min   5Min Res Usr
16777242       97560   1408799         69  0.23%  0.02%  0.00% IPC Seat Manager
  Resource User: IPC Session Service(ID: 0x100001B)
    RUID Runtime(ms)   Invoked      uSecs   5Sec   1Min   5Min Res Usr
16777243           0         1          0  0.00%  0.00%  0.00% IPC Session Serv
  Resource User: ARP Input(ID: 0x100001C)
    RUID Runtime(ms)   Invoked      uSecs   5Sec   1Min   5Min Res Usr
16777244          20      3082          6  0.00%  0.00%  0.00% ARP Input
  Resource User: EEM ED Syslog(ID: 0x100001D)
    RUID Runtime(ms)   Invoked      uSecs   5Sec   1Min   5Min Res Usr
16777245           0        49          0  0.00%  0.00%  0.00% EEM ED Syslog
  Resource User: DDR Timers(ID: 0x100001E)
    RUID Runtime(ms)   Invoked      uSecs   5Sec   1Min   5Min Res Usr
16777246           0         2          0  0.00%  0.00%  0.00% DDR Timers
  Resource User: Dialer event(ID: 0x100001F)
    RUID Runtime(ms)   Invoked      uSecs   5Sec   1Min   5Min Res Usr
16777247           0         2          0  0.00%  0.00%  0.00% Dialer event
  Resource User: Entity MIB API(ID: 0x1000020)
    RUID Runtime(ms)   Invoked      uSecs   5Sec   1Min   5Min Res Usr
16777248          28        16       1750  0.00%  0.00%  0.00% Entity MIB API
.
.
.
Resource User: draco-oir-process:slot 2(ID: 0x100011E)
Getbufs  Retbufs  Holding  RU Name
0        0        0        draco-oir-proces

  Resource User: SCP async: Draco-LC4(ID: 0x1000125)
Getbufs  Retbufs  Holding  RU Name
35849    243101   4294760044 SCP async: Draco

  Resource User: IFCOM Msg Hdlr(ID: 0x1000127)
Getbufs  Retbufs  Holding  RU Name
2        2        0        IFCOM Msg Hdlr

  Resource User: IFCOM Msg Hdlr(ID: 0x1000128)
Getbufs  Retbufs  Holding  RU Name
28       28       0        IFCOM Msg Hdlr

  Resource User: Exec(ID: 0x100012C)
Getbufs  Retbufs  Holding  RU Name
912      912      0        Exec

Resource Owner: test_mem
 Resource User Type: test_process
 Resource User Type: mem_rut
Resource Owner: test_cpu
 Resource User Type: test_process
 Resource User Type: cpu_rut
```

Step 10 **show resource database**

Use this command to display the resource database details, for example:

```
Router# show resource database

List of all Resource Owners :
Owner: cpu                       Id:0x1
Owner's list of monitors is empty.
Owner: memory                    Id:0x2
Owner's list of monitors is empty.
Owner: Buffer                    Id:0x3
Owner's list of monitors is empty.
Owner: test_mem                  Id:0x4
Owner's list of monitors is empty.
Owner: test_cpu                  Id:0x5
Owner's list of monitors is empty.
Owner: test_RO0                  Id:0x7
Owner's list of monitors is empty.
Owner: test_RO1                  Id:0x8
Owner's list of monitors is empty.
Owner: test_RO2                  Id:0x9
Owner's list of monitors is empty.
Owner: test_RO3                  Id:0xA
Owner's list of monitors is empty.
.
.
.
Resource Monitor: test_ROM0, ID: 0x1B
 Not Watching any Relations.
 Not Watching any Policies.
Resource Monitor: test_ROM1, ID: 0x1C
 Not Watching any Relations.
 Not Watching any Policies.
Resource Monitor: test_ROM2, ID: 0x1D
 Not Watching any Relations.
 Not Watching any Policies.
```

Step 11 **show resource owner** {*resource-owner-name* | **all**} **user** {*resource-user-type-name* | **all**} [**brief** | **detailed** | **triggers**]

Use this command to display the resource owner details, for example:

```
Router# show resource owner all user all

Resource Owner: cpu
 Resource User Type: iosprocess
  Resource User: Init(ID: 0x1000001)
    RUID Runtime(ms)   Invoked      uSecs   5Sec   1Min   5Min Res Usr
16777217           0         0          0  0.00%  0.00%  0.00% Init
  Resource User: Scheduler(ID: 0x1000002)
    RUID Runtime(ms)   Invoked      uSecs   5Sec   1Min   5Min Res Usr
16777218           0         0          0  0.00%  0.00%  0.00% Scheduler
  Resource User: Dead(ID: 0x1000003)
    RUID Runtime(ms)   Invoked      uSecs   5Sec   1Min   5Min Res Usr
16777219           0         0          0  0.00%  0.00%  0.00% Dead
  Resource User: Interrupt(ID: 0x1000004)
    RUID Runtime(ms)   Invoked      uSecs   5Sec   1Min   5Min Res Usr
16777220           0         0          0  0.00%  0.00%  0.00% Interrupt
  Resource User: Memory RO RU(ID: 0x1000005)
    RUID Runtime(ms)   Invoked      uSecs   5Sec   1Min   5Min Res Usr
16777221           0         0          0  0.00%  0.00%  0.00% Memory RO RU
  Resource User: Chunk Manager(ID: 0x1000006)
    RUID Runtime(ms)   Invoked      uSecs   5Sec   1Min   5Min Res Usr
16777222           4         3       1333  0.00%  0.00%  0.00% Chunk Manager
```

```
  Resource User: Load Meter(ID: 0x1000007)
    RUID Runtime(ms)   Invoked      uSecs   5Sec   1Min   5Min Res Usr
16777223           4       292         13  0.00%  0.00%  0.00% Load Meter
  Resource User: Check heaps(ID: 0x1000009)
    RUID Runtime(ms)   Invoked      uSecs   5Sec   1Min   5Min Res Usr
16777225         376       192       1958  0.00%  0.02%  0.00% Check heaps
  Resource User: Pool Manager(ID: 0x100000A)
    RUID Runtime(ms)   Invoked      uSecs   5Sec   1Min   5Min Res Usr
16777226           0         1          0  0.00%  0.00%  0.00% Pool Manager
  Resource User: Buffer RO RU(ID: 0x100000B)
    RUID Runtime(ms)   Invoked      uSecs   5Sec   1Min   5Min Res Usr
16777227           0         0          0  0.00%  0.00%  0.00% Buffer RO RU
  Resource User: Timers(ID: 0x100000C)
    RUID Runtime(ms)   Invoked      uSecs   5Sec   1Min   5Min Res Usr
16777228           0         2          0  0.00%  0.00%  0.00% Timers
  Resource User: Serial Background(ID: 0x100000D)
    RUID Runtime(ms)   Invoked      uSecs   5Sec   1Min   5Min Res Usr
16777229           0         2          0  0.00%  0.00%  0.00% Serial Backgroun
  Resource User: ALARM_TRIGGER_SCAN(ID: 0x100000E)
    RUID Runtime(ms)   Invoked      uSecs   5Sec   1Min   5Min Res Usr
16777230           0       268          0  0.00%  0.00%  0.00% ALARM_TRIGGER_SC
  Resource User: AAA_SERVER_DEADTIME(ID: 0x100000F)
    RUID Runtime(ms)   Invoked      uSecs   5Sec   1Min   5Min Res Usr
16777231           0         1          0  0.00%  0.00%  0.00% AAA_SERVER_DEADT
  Resource User: AAA high-capacity counters(ID: 0x1000010)
    RUID Runtime(ms)   Invoked      uSecs   5Sec   1Min   5Min Res Usr
.
.
.
Resource User Type: test_RUT143

Resource User Type: test_RUT144
Resource User Type: test_RUT145

Resource User Type: test_RUT146
Resource User Type: test_RUT147
```

Step 12 **show resource relationship user** *resource-user-type*

Use this command to display the relationship details between different resource owners, for example:

```
Router# show resource relationship

Resource User Type: iosprocess (ID: 0x1)
 -> Resource Owner: cpu (ID: 0x1)
 -> Resource Owner: memory (ID: 0x2)
 -> Resource Owner: Buffer (ID: 0x3)
 -> Resource User: Init (ID: 0x1000001)
 -> Resource User: Scheduler (ID: 0x1000002)
 -> Resource User: Dead (ID: 0x1000003)
 -> Resource User: Interrupt (ID: 0x1000004)
 -> Resource User: Memory RO RU (ID: 0x1000005)
 -> Resource User: Chunk Manager (ID: 0x1000006)
 -> Resource User: Load Meter (ID: 0x1000007)
 -> Resource User: Check heaps (ID: 0x1000009)
 -> Resource User: Pool Manager (ID: 0x100000A)
 -> Resource User: Buffer RO RU (ID: 0x100000B)
 -> Resource User: Timers (ID: 0x100000C)
 -> Resource User: Serial Background (ID: 0x100000D)
 -> Resource User: ALARM_TRIGGER_SCAN (ID: 0x100000E)
 -> Resource User: AAA_SERVER_DEADTIME (ID: 0x100000F)
 -> Resource User: AAA high-capacity counters (ID: 0x1000010)
 -> Resource User: Policy Manager (ID: 0x1000011)
 -> Resource User: Crash writer (ID: 0x1000012)
 -> Resource User: RO Notify Timers (ID: 0x1000013)
 -> Resource User: RMI RM Notify Watched Policy (ID: 0x1000014)
 -> Resource User: EnvMon (ID: 0x1000015)
 -> Resource User: OIR Handler (ID: 0x1000016)
 -> Resource User: IPC Dynamic Cache (ID: 0x1000017)
 -> Resource User: IPC Zone Manager (ID: 0x1000018)
 -> Resource User: IPC Periodic Timer (ID: 0x1000019)
 -> Resource User: IPC Managed Timer (ID: 0x100001A)
 -> Resource User: IPC Deferred Port Closure (ID: 0x100001B)
 -> Resource User: IPC Seat Manager (ID: 0x100001C)
 -> Resource User: IPC Session Service (ID: 0x100001D)
 -> Resource User: Compute SRP rates (ID: 0x100001E)
 -> Resource User: ARP Input (ID: 0x100001F)
 -> Resource User: DDR Timers (ID: 0x1000020)
 -> Resource User: Dialer event (ID: 0x1000021)
 -> Resource User: Entity MIB API (ID: 0x1000022)
 -> Resource User: SERIAL A'detect (ID: 0x1000023)
 -> Resource User: GraphIt (ID: 0x1000024)
 -> Resource User: HC Counter Timers (ID: 0x1000025)
 -> Resource User: Critical Bkgnd (ID: 0x1000026)
 -> Resource User: Net Background (ID: 0x1000027)
 -> Resource User: Logger (ID: 0x1000028)
.
.
.
Resource User Type: test_RUT141 (ID: 0x92)
 -> Resource Owner: test_RO0 (ID: 0x7)
 Resource User Type: test_RUT142 (ID: 0x93)
 -> Resource Owner: test_RO0 (ID: 0x7)
 Resource User Type: test_RUT143 (ID: 0x94)
 -> Resource Owner: test_RO0 (ID: 0x7)
 Resource User Type: test_RUT144 (ID: 0x95)
 -> Resource Owner: test_RO0 (ID: 0x7)
 Resource User Type: test_RUT145 (ID: 0x96)
 -> Resource Owner: test_RO0 (ID: 0x7)
 Resource User Type: test_RUT146 (ID: 0x97)
 -> Resource Owner: test_RO0 (ID: 0x7)
 Resource User Type: test_RUT147 (ID: 0x98)
 -> Resource Owner: test_RO0 (ID: 0x7)
 Resource User Type: test_RUT148 (ID: 0x99)
 -> Resource Owner: test_RO0 (ID: 0x7)
 Resource User Type: test_RUT149 (ID: 0x9A)
 -> Resource Owner: test_RO0 (ID: 0x7)
```

Step 13 **show resource user** {**all** | *resource-user-type*} [**brief** | **detailed**]

Use this command to display the relationship details between different ROs, for example:

```
Router# show resource user all

Resource User Type: iosprocess
Resource Grp: Init
Resource Owner: memory
Processor memory
Allocated   Freed  Holding   Blocks
27197780  8950144 18247636     6552

I/O memory
Allocated   Freed  Holding   Blocks
 7296000     9504  7286496      196

Resource Owner: cpu
    RUID Runtime(ms)   Invoked      uSecs   5Sec   1Min   5Min Res Usr
16777224       14408       116     124206 100.40%  8.20%  1.70% Init
```

```
Resource Owner: Buffer
Getbufs  Retbufs  Holding  RU Name
332      60       272      Init

Resource User: Init
Resource User: Scheduler
Resource Owner: memory
Processor memory
Allocated   Freed  Holding   Blocks
   77544        0    77544        2

Resource Owner: cpu
    RUID Runtime(ms)   Invoked      uSecs   5Sec   1Min   5Min Res Usr
16777218           0         0          0  0.00%  0.00%  0.00% Scheduler
Resource Owner: Buffer
Getbufs  Retbufs  Holding  RU Name
0        0        0        Scheduler

Resource User: Dead
Resource Owner: memory
Processor memory
Allocated   Freed  Holding   Blocks
 1780540      260  1780280      125
.
.
.

 Resource User: BGP Scanner
  Resource Owner: memory
Processor memory
Allocated   Freed  Holding   Blocks
    9828     9828        0        0

  Resource Owner: cpu
    RUID Runtime(ms)   Invoked      uSecs   5Sec   1Min   5Min Res Usr
16777406         660       659       1001  0.00%  0.00%  0.00% BGP Scanner
  Resource Owner: Buffer
Getbufs  Retbufs  Holding  RU Name
0        0        0        BGP Scanner
Resource User Type: test_process
Resource User Type: mem_rut
Resource User Type: cpu_rut
```

Troubleshooting Tips

To trace and troubleshoot the notification and registration activities for resources using the Embedded Resource Manager feature, use the following suggested techniques.

- Enable debugging of resource registration using the **debug resource policy registration** command in privileged EXEC mode.
- Enable debugging of resource manager notification using the **debug resource policy notification** command in privileged EXEC mode.

SUMMARY STEPS

1. **enable**
2. **debug resource policy registration**
3. **debug resource policy notification** [**owner** *resource-owner-name*]

DETAILED STEPS

	Command or Action	Purpose

Step 1	**`enable`**
	Example:
	`Router> enable`
		Enables privileged EXEC mode.
		• Enter your password if prompted.

Step 2	**`debug resource policy registration`**
	Example:
	`Router# debug resource policy registration`
		Enables debugging on resource policy registration.

Step 3	**`debug resource policy notification`** `[`**`owner`** *`resource-owner-name`*`]`
	Example:
	`Router# debug resource policy notification owner cpu`
		Enables notification debugging on ROs.

Examples

Use the **debug resource policy registration** command to trace the resource manager registration information, for example:

```
Router# debug resource policy registration

Registrations debugging is on

When a Resource User is created
*Mar  3 09:35:58.304: resource_user_register: RU: ruID: 0x10000B8, rutID: 0x1, rg_ID: 0x0
name: usrr1

When a Resource User is deleted
*Mar  3 09:41:09.500: resource_user_unregister: RU: ruID: 0x10000B8, rutID: 0x1, rg_ID:
0x0 name: usrr1
```

Use the **debug resource policy notification** [**owner** *resource-owner-name*] command to trace the resource policy notification information, for example:

```
Router# debug resource policy notification

Enabled notif. debugs on all owners
```

When a threshold is exceeded, you would see these messages:

```
*Mar  3 09:50:44.081: Owner: 'memory' initiated a notification:
*Mar  3 09:50:44.081: %SYS-4-RESMEMEXCEED: Resource user usrr1 has exceeded the Major
memory threshold
Pool: Processor Used: 42932864 Threshold :42932860
*Mar  3 09:50:46.081: Notification from Owner: 'memory' is dispatched for User: 'usrr1'
(ID: 0x10000B9)
*Mar  3 09:50:46.081: %SYS-4-RESMEMEXCEED: Resource user usrr1 has exceeded the Major
memory threshold
Pool: Processor Used: 42932864 Threshold :42932860

Router# no debug resource policy notification

Disabled notif. debugs on all owners
```

```
Router# debug resource policy notification owner cpu

Enabled notif. debugs on owner 'cpu'

Router# no debug resource policy notification owner cpu

Disabled notif. debugs on owner 'cpu'

Router# debug resource policy notification owner memory

Enabled notif. debugs on owner 'memory'

Router# no debug resource policy notification owner memory

Disabled notif. debugs on owner 'memory'

Router# debug resource policy notification owner Buffer

Enabled notif. debugs on owner 'Buffer'

Router# no debug resource policy notification owner Buffer

Disabled notif. debugs on owner 'Buffer'
```

Configuration Examples for Embedded Resource Manager

This section provides the following configuration examples:

- Managing Resource Utilization by Defining Resource Policy: Example, page 46
- Setting Expected Operating Ranges for Resource Owners: Example, page 47
- Applying a Policy: Example, page 52
- Setting a System Global Thresholding Policy for I/O Memory: Example, page 53

Managing Resource Utilization by Defining Resource Policy: Example

The following example shows how to configure a global resource policy with the policy name system-global-pc1:

```
configure terminal
resource policy
policy system-global-pc1 global
```

The following example shows how to configure a per user global resource policy with the policy name per-user-global-pc1 and the resource type as iosprocess:

```
configure terminal
resource policy
policy per-user-global-pc1 type iosprocess
```

The following example shows how to configure a user local resource policy with the policy name user-local-pc1 and the resource type as iosprocess:

```
configure terminal
resource policy
policy user-local-pc1 type iosprocess
```

Setting Expected Operating Ranges for Resource Owners: Example

The following example shows how to configure various thresholds for buffer, CPU, and memory ROs.

Configuring System Global Thresholding Policy for Buffer RO

The following example shows how to configure a global policy with the policy name as system-global-pc1 for public buffer with critical threshold values of 90 percent as rising at an interval of 12 seconds, 20 percent as falling at an interval of 10 seconds, major threshold values of 70 percent as rising at an interval of 12 seconds, 15 percent as falling at an interval of 10 seconds, and minor threshold values of 60 percent as rising at an interval of 12 seconds, 10 percent as falling at an interval of 10 seconds:

```
configure terminal
resource policy
policy system-global-pc1 global
system
buffer public
critical rising 90 interval 12 falling 20 interval 10
major rising 70 interval 12 falling 15 interval 10
minor rising 60 interval 12 falling 10 interval 10
```

Configuring Per User Global Thresholding Policy for Buffer RO

The following example shows how to configure a per user global policy with the policy name as per-user-global-pc1 for public buffer with critical threshold values of 90 percent as rising at an interval of 12 seconds, 20 percent as falling at an interval of 10 seconds, major threshold values of 70 percent as rising at an interval of 12 seconds, 15 percent as falling at an interval of 10 seconds, and minor threshold values of 60 percent as rising at an interval of 12 seconds, 10 percent as falling at an interval of 10 seconds:

```
configure terminal
resource policy
policy per-user-global-pc1 type iosprocess
system
buffer public
critical rising 90 interval 12 falling 20 interval 10 global
major rising 70 interval 12 falling 15 interval 10 global
minor rising 60 interval 12 falling 10 interval 10 global
```

Configuring User Local Thresholding Policy for Buffer RO

The following example shows how to configure a user local policy with the policy name as user-local-pc1 for public buffer with critical threshold values of 90 percent as rising at an interval of 12 seconds, 20 percent as falling at an interval of 10 seconds, major threshold values of 70 percent as rising at an interval of 12 seconds, 15 percent as falling at an interval of 10 seconds, and minor threshold values of 60 percent as rising at an interval of 12 seconds, 10 percent as falling at an interval of 10 seconds:

```
configure terminal
resource policy
policy user-local-pc1 type iosprocess
system
buffer public
critical rising 70 interval 12 falling 20 interval 10
major rising 70 interval 12 falling 15 interval 10
minor rising 60 interval 12 falling 10 interval 10
```

Configuring System Global Thresholding Policy for I/O Memory RO

The following example shows how to configure a global policy with the policy name as system-global-pc1 for I/O memory with critical threshold values of 90 percent as rising at an interval of 12 seconds, 20 percent as falling at an interval of 10 seconds, major threshold values of 70 percent as rising at an interval of 12 seconds, 15 percent as falling at an interval of 10 seconds, and minor threshold values of 60 percent as rising at an interval of 12 seconds, 10 percent as falling at an interval of 10 seconds:

```
configure terminal
resource policy
policy system-global-pc1 global
system
memory io
critical rising 90 interval 12 falling 20 interval 10
major rising 70 interval 12 falling 15 interval 10
minor rising 60 interval 12 falling 10 interval 10
```

Configuring Per User Global Thresholding Policy for I/O Memory RO

The following example shows how to configure a per user global policy with the policy name as per-user-global-pc1 for I/O memory with critical threshold values of 90 percent as rising at an interval of 12 seconds, 20 percent as falling at an interval of 10 seconds, major threshold values of 70 percent as rising at an interval of 12 seconds, 15 percent as falling at an interval of 10 seconds, and minor threshold values of 60 percent as rising at an interval of 12 seconds, 10 percent as falling at an interval of 10 seconds:

```
configure terminal
resource policy
policy per-user-global-pc1 type iosprocess
system
memory io
critical rising 90 interval 12 falling 20 interval 10 global
major rising 70 interval 12 falling 15 interval 10 global
minor rising 60 interval 12 falling 10 interval 10 global
```

Configuring User Local Thresholding Policy for I/O Memory RO

The following example shows how to configure a user local policy with the policy name as user-local-pc1 for I/O memory with critical threshold values of 90 percent as rising at an interval of 12 seconds, 20 percent as falling at an interval of 10 seconds, major threshold values of 70 percent as rising at an interval of 12 seconds, 15 percent as falling at an interval of 10 seconds, and minor threshold values of 60 percent as rising at an interval of 12 seconds, 10 percent as falling at an interval of 10 seconds:

```
configure terminal
resource policy
policy user-local-pc1 type iosprocess
system
memory io
critical rising 90 interval 12 falling 20 interval 10
major rising 70 interval 12 falling 15 interval 10
minor rising 60 interval 12 falling 10 interval 10
```

Configuring System Global Thresholding Policy for Processor Memory RO

The following example shows how to configure a user system global policy with the policy name as system-global-pc1 for processor memory with critical threshold values of 90 percent as rising at an interval of 12 seconds, 20 percent as falling at an interval of 10 seconds, major threshold values of 70 percent as rising at an interval of 12 seconds, 15 percent as falling at an interval of 10 seconds, and minor threshold values of 60 percent as rising at an interval of 12 seconds, 10 percent as falling at an interval of 10 seconds:

```
configure terminal
resource policy
policy system-global-pc1 global
system
memory processor
critical rising 90 interval 12 falling 20 interval 10
major rising 70 interval 12 falling 15 interval 10
minor rising 60 interval 12 falling 10 interval 10
```

Configuring Per User Global Thresholding Policy for Processor Memory RO

The following example shows how to configure a per user global policy with the policy name as user-global-pc1 and the resource type as iosprocess for processor memory with critical threshold values of 90 percent as rising at an interval of 12 seconds, 20 percent as falling at an interval of 10 seconds, major threshold values of 70 percent as rising at an interval of 12 seconds, 15 percent as falling at an interval of 10 seconds, and minor threshold values of 60 percent as rising at an interval of 12 seconds, 10 percent as falling at an interval of 10 seconds:

```
configure terminal
resource policy
policy user-global-pc1 type iosprocess
system
memory processor
critical rising 90 interval 12 falling 20 interval 10
major rising 70 interval 12 falling 15 interval 10
minor rising 60 interval 12 falling 10 interval 10
```

Configuring User Local Thresholding Policy for Processor Memory RO

The following example shows how to configure a user local policy with the policy name as user-local-pc1 and the resource type as iosprocess for processor memory with critical threshold values of 90 percent as rising at an interval of 12 seconds, 20 percent as falling at an interval of 10 seconds, major threshold values of 70 percent as rising at an interval of 12 seconds, 15 percent as falling at an interval of 10 seconds, and minor threshold values of 60 percent as rising at an interval of 12 seconds, 10 percent as falling at an interval of 10 seconds:

```
configure terminal
resource policy
policy user-local-pc1 type iosprocess
system
memory processor
critical rising 90 interval 12 falling 20 interval 10
major rising 70 interval 12 falling 15 interval 10
minor rising 60 interval 12 falling 10 interval 10
```

Configuring System Global Thresholding Policy for Interrupt CPU RO

The following example shows how to configure a global policy with the policy name as system-global-pc1 for interrupt CPU with critical threshold values of 90 percent as rising at an interval of 12 seconds, 20 percent as falling at an interval of 10 seconds, major threshold values of 70 percent as rising at an interval of 12 seconds, 15 percent as falling at an interval of 10 seconds, and minor threshold values of 60 percent as rising at an interval of 12 seconds, 10 percent as falling at an interval of 10 seconds:

```
configure terminal
resource policy
policy system-global-pc1 global
```

```
system
cpu interrupt
critical rising 90 interval 12 falling 20 interval 10
major rising 70 interval 12 falling 15 interval 10
minor rising 60 interval 12 falling 10 interval 10
```

Configuring Per User Global Thresholding Policy for Interrupt CPU RO

The following example shows how to configure a per user global policy with the policy name as per-user-global-pc1 and the resource type as iosprocess for interrupt CPU with critical threshold values of 90 percent as rising at an interval of 12 seconds, 20 percent as falling at an interval of 10 seconds, major threshold values of 70 percent as rising at an interval of 12 seconds, 15 percent as falling at an interval of 10 seconds, and minor threshold values of 60 percent as rising at an interval of 12 seconds, 10 percent as falling at an interval of 10 seconds:

```
configure terminal
resource policy
policy per-user-global-pc1 type iosprocess
system
cpu interrupt
critical rising 90 interval 12 falling 20 interval 10 global
major rising 70 interval 12 falling 15 interval 10 global
minor rising 60 interval 12 falling 10 interval 10 global
```

Configuring User Local Thresholding Policy for Interrupt CPU RO

The following example shows how to configure a user local policy with the policy name as user-local-pc1 and the resource type as iosprocess for interrupt CPU with critical threshold values of 90 percent as rising at an interval of 12 seconds, 20 percent as falling at an interval of 10 seconds, major threshold values of 70 percent as rising at an interval of 12 seconds, 15 percent as falling at an interval of 10 seconds, and minor threshold values of 60 percent as rising at an interval of 12 seconds, 10 percent as falling at an interval of 10 seconds:

```
configure terminal
resource policy
policy user-local-pc1 global type iosprocess
system
cpu interrupt
critical rising 90 interval 12 falling 20 interval 10
major rising 70 interval 12 falling 15 interval 10
minor rising 60 interval 12 falling 10 interval 10
```

Configuring System Global Thresholding Policy for Process CPU RO

The following example shows how to configure a global policy with the policy name as system-global-pc1 for process CPU with critical threshold values of 90 percent as rising at an interval of 12 seconds, 20 percent as falling at an interval of 10 seconds, major threshold values of 70 percent as rising at an interval of 12 seconds, 15 percent as falling at an interval of 10 seconds, and minor threshold values of 60 percent as rising at an interval of 12 seconds, 10 percent as falling at an interval of 10 seconds:

```
configure terminal
resource policy
policy system-global-pc1 global
system
cpu process
critical rising 90 interval 12 falling 20 interval 10
major rising 70 interval 12 falling 15 interval 10
minor rising 60 interval 12 falling 10 interval 10
```

Configuring Per User Global Thresholding Policy for Process CPU RO

The following example shows how to configure a per user global policy with the policy name as per-user-global-pc1 and the resource type as iosprocess for process CPU with critical threshold values of 90 percent as rising at an interval of 12 seconds, 20 percent as falling at an interval of 10 seconds, major threshold values of 70 percent as rising at an interval of 12 seconds, 15 percent as falling at an interval of 10 seconds, and minor threshold values of 60 percent as rising at an interval of 12 seconds, 10 percent as falling at an interval of 10 seconds:

```
configure terminal
resource policy
resource policy per-user-global-pc1 type iosprocess
system
cpu process
critical rising 90 interval 12 falling 20 interval 10 global
major rising 70 interval 12 falling 15 interval 10 global
minor rising 60 interval 12 falling 10 interval 10 global
```

Configuring User Local Thresholding Policy for Process CPU RO

The following example shows how to configure a user local policy with the policy name as user-local-pc1 and the resource type as iosprocess for process CPU with critical threshold values of 90 percent as rising at an interval of 12 seconds, 20 percent as falling at an interval of 10 seconds, major threshold values of 70 percent as rising at an interval of 12 seconds, 15 percent as falling at an interval of 10 seconds, and minor threshold values of 60 percent as rising at an interval of 12 seconds, 10 percent as falling at an interval of 10 seconds:

```
configure terminal
resource policy
policy user-local-pc1 global type iosprocess
system
cpu process
critical rising 90 interval 12 falling 20 interval 10
major rising 70 interval 12 falling 15 interval 10
minor rising 60 interval 12 falling 10 interval 10
```

Configuring System Global Thresholding Policy for Total CPU RO

The following example shows how to configure a global policy with the policy name as system-global-pc1 for total CPU with critical threshold values of 90 percent as rising at an interval of 12 seconds, 20 percent as falling at an interval of 10 seconds, major threshold values of 70 percent as rising at an interval of 12 seconds, 20 percent as falling at an interval of 10 seconds, and minor threshold values of 60 percent as rising at an interval of 12 seconds, 10 percent as falling at an interval of 10 seconds:

```
configure terminal
resource policy
policy system-global-pc1 global
system
cpu total
critical rising 90 interval 12 falling 20 interval 10
major rising 70 interval 12 falling 15 interval 10
minor rising 60 interval 12 falling 10 interval 10
```

Configuring Per User Global Thresholding Policy for Total CPU RO

The following example shows how to configure a per user global policy with the policy name as per-user-global-pc1 and the resource type as iosprocess for total CPU with critical threshold values of 90 percent as rising at an interval of 12 seconds, 20 percent as falling at an interval of 10 seconds, major threshold values of 70 percent as rising at an interval of 12 seconds, 15 percent as falling at an interval of 10 seconds, and minor threshold values of 60 percent as rising at an interval of 12 seconds, 10 percent as falling at an interval of 10 seconds:

```
configure terminal
resource policy
policy per-user-global-pc1 type iosprocess
system
cpu total
critical rising 90 interval 12 falling 20 interval 10 global
major rising 70 interval 12 falling 15 interval 10 global
minor rising 60 interval 12 falling 10 interval 10 global
```

Configuring User Local Thresholding Policy for Total CPU RO

The following example shows how to configure a user local policy with the policy name as user-local-pc1 and the resource type as iosprocess for total CPU with critical threshold values of 90 percent as rising at an interval of 12 seconds, 20 percent as falling at an interval of 10 seconds, major threshold values of 70 percent as rising at an interval of 12 seconds, 15 percent as falling at an interval of 10 seconds, and minor threshold values of 60 percent as rising at an interval of 12 seconds, 10 percent as falling at an interval of 10 seconds:

```
configure terminal
resource policy
policy user-local-pc1 type iosprocess
system
cpu total
critical rising 90 interval 12 falling 20 interval 10
major rising 70 interval 12 falling 15 interval 10
minor rising 60 interval 12 falling 10 interval 10
```

Applying a Policy: Example

The following example shows how to apply a per user thresholding policy for the resource instance EXEC, resource user type iosprocess, and policy name policy-test1:

```
configure terminal
resource policy
policy policy-test1 type iosprocess
exit
user EXEC iosprocess policy-test1
```

The following example shows how to apply a global thresholding policy with the policy name global-global-test1:

```
configure terminal
resource policy
policy global-global-test1 global
exit
user global global-global-test1
```

The following example shows how to apply a group thresholding policy with the group name gr1 and resource type as iosprocess:

```
configure terminal
resource policy
policy group-test1
exit

user group gr1 type iosprocess
```

```
instance http
policy group-test1
```

Setting a System Global Thresholding Policy for I/O Memory: Example

The following example shows the configuration of a global memory thresholding policy for I/O memory. In this example, the policy is given the name “system-global-io”, and the threshold for critical I/O memory usage is defined as being usage of over 90 percent of the globally available I/O memory pool for 12 consecutive seconds.

The critical falling threshold is also defined in this example (less than 20 percent of the globally available I/O memory pool for 10 seconds or more); however, only the critical rising level will affect when the automatic deallocation procedure is triggered.

```
configure terminal
resource policy
policy system-global-io global
system
memory io
critical rising 90 interval 12 falling 20 interval 10
```

Additional References

The following sections provide references related to the Embedded Resource Manager.

Related Documents

Related Topic	Document Title
Configuration fundamentals commands: complete command syntax, command modes, command history, defaults, usage guidelines, and examples	*Cisco IOS Configuration Fundamentals Command Reference*, Release 12.4T
Network management commands: complete command syntax, command modes, command history, defaults, usage guidelines, and examples	*Cisco IOS Network Management Command Reference*, Release 12.4T
Embedded Event Manager configuration tasks	*Cisco IOS Embedded Event Manager Configuration Guide*, Release 12.4T
Memory Leak Detector	The *Memory Leak Dectector* feature guide (part of the *Cisco IOS Configuration Fundamentals Configuration Guide,* Release 12.4T)

Standards

Standards	Title
No new or modified standards are supported by this feature.	—

MIBs

MIBs	MIBs Link
CISCO-ERM-MIB.my	To locate and download MIBs for selected platforms, Cisco IOS releases, and feature sets, use Cisco MIB Locator found at the following URL: http://www.cisco.com/go/mibs

RFCs

RFCs	Title
No new or modified RFCs are supported by this feature.	—

Technical Assistance

Description	Link
The Cisco Support website provides extensive online resources, including documentation and tools for troubleshooting and resolving technical issues with Cisco products and technologies. To receive security and technical information about your products, you can subscribe to various services, such as the Product Alert Tool (accessed from Field Notices), the Cisco Technical Services Newsletter, and Really Simple Syndication (RSS) Feeds. Access to most tools on the Cisco Support website requires a Cisco.com user ID and password.	http://www.cisco.com/techsupport

Command Reference

The following commands are introduced or modified in the feature or features documented in this module. For information about these commands, see the *Cisco IOS Network Management Command Reference* at http://www.cisco.com/en/US/docs/ios/netmgmt/command/reference/br_book.html. For information about all Cisco IOS commands, go to the Command Lookup Tool at http://tools.cisco.com/Support/CLILookup or to the *Cisco IOS Master Commands List*.

- **debug resource policy notification**
- **debug resource policy registration**
- **policy (ERM)**
- **show snmp mib bulkstat transfer**
- **snmp mib bulkstat object-list**
- **snmp mib bulkstat schema**

- **snmp mib bulkstat transfer**
- **show monitor event trace**
- **show monitor event trace cpu-report**
- **show process cpu**
- **show process cpu autoprophile hog**
- **show process cpu extended**
- **show resource all**
- **show resource database**
- **show resource owner**
- **show resource relationship**
- **show resource user**
- **snmp-server enable traps resource-policy**
- **user (ERM)**

Feature Information for Embedded Resource Manager

Table 1 lists the features in this module and provides links to specific configuration information. Only features that were introduced or modified in Cisco IOS Release 12.3(14)T or a later release appear in the table.

Not all commands may be available in your Cisco IOS software release. For release information about a specific command, see the command reference documentation.

Use Cisco Feature Navigator to find information about platform support and software image support. Cisco Feature Navigator enables you to determine which Cisco IOS and Catalyst OS software images support a specific software release, feature set, or platform. To access Cisco Feature Navigator, go to http://www.cisco.com/go/cfn. An account on Cisco.com is not required.

Note Table 1 lists only the Cisco IOS software release that introduced support for a given feature in a given Cisco IOS software release train. Unless noted otherwise, subsequent releases of that Cisco IOS software release train also support that feature.

Table 1 Feature Information for Embedded Resource Manager

Feature Name	Releases	Feature Information
Embedded Resource Manager	12.3(14)T 12.2(33)SRB 12.2(33)SB	The Embedded Resource Manager (ERM) feature allows you to monitor internal system resource utilization for finite resources such as the buffer, memory, and CPU. ERM monitors resource utilization from the perspective of various subsystems within the Cisco IOS software such as resource owners (ROs) and resource users (RUs). ERM allows you to configure threshold values for system resources, leading to better insight into system scalability and improved system availability. The following sections provide information about this feature: • Prerequisites for Embedded Resource Manager, page 2 • How to Configure Embedded Resource Manager, page 8

Table 1 Feature Information for Embedded Resource Manager (continued)

Feature Name	Releases	Feature Information
Embedded Resource Manager MIB	12.2(33)SRB 12.2(33)SB	The ERM MIB feature introduces MIB support for the Embedded Resource Manager (ERM) feature. The ERM feature tracks resource usage information for every registered resource owner and resource user. ERM ensures efficient usage of available resources.The ERM MIB feature allows you to monitor the usage of resources by gathering resource usage information using MIB objects. The network manager can use the information collected by the ERM MIB objects to ensure the optimal use of the resources. The following command was introduced by this feature: **snmp-server enable traps resource-policy**.
Packet Memory Reclamation	12.4(6)T	The Packet Memory Reclamation functionality utilizes the ERM infrastructure to cleanup and reclaim leaked Cisco IOS packet memory using the Memory Leak Detector process (sometimes referred to as the “Garbage Detection” or “GD” process). The following section provides information about this feature: • Memory Resource Owner, page 5

Glossary

CPUHOG—Each process is allocated a quantum of time, which is equivalent to 200 ms. If a process is running for more than 2 seconds, the process is hogging the CPU. This condition is called CPUHOG.

RM—resource usage monitors. Applications that wants to monitor resource utilization of resources by the resource users.

RO—resource owners. Provides resources to the resource users. For example, CPU, buffer, memory and so on.

RU—resource users. Applications or clients (like HTTP, SNMP, telnet, and so on) that use the resources and receive notifications to throttle when thresholds exceed the current values.

Note Refer to *Internetworking Terms and Acronyms* for terms not included in this glossary.

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

Embedded Event Manager Overview

First Published: October 31, 2005
Last Update: July 11, 2008

Embedded Event Manager (EEM) is a distributed and customized approach to event detection and recovery offered directly in a Cisco IOS device. EEM offers the ability to monitor events and take informational, corrective, or any desired EEM action when the monitored events occur or when a threshold is reached. An EEM policy is an entity that defines an event and the actions to be taken when that event occurs.

This module contains a technical overview of EEM. EEM can be used alone, or with other network management technologies to help monitor and maintain your network. Before you begin to implement EEM, it is important that you understand the information presented in this module.

Finding Feature Information in This Module

Your Cisco IOS software release may not support all of the features documented in this module. To reach links to specific feature documentation in this module and to see a list of the releases in which each feature is supported, use the "Feature Information for Embedded Event Manager Overview" section on page 16.

Finding Support Information for Platforms and Cisco IOS and Catalyst OS Software Images

Use Cisco Feature Navigator to find information about platform support and Cisco IOS and Catalyst OS software image support. To access Cisco Feature Navigator, go to http://www.cisco.com/go/cfn. An account on Cisco.com is not required.

Contents

- Information About Embedded Event Manager, page 2
- Where to Go Next, page 14
- Additional References, page 14
- Feature Information for Embedded Event Manager Overview, page 16

Americas Headquarters:
Cisco Systems, Inc., 170 West Tasman Drive, San Jose, CA 95134-1706 USA

Information About Embedded Event Manager

To use EEM in your network, you should understand the following concepts:

- Embedded Event Manager, page 2
- Embedded Event Manager 1.0, page 3
- Embedded Event Manager 2.0, page 4
- Embedded Event Manager 2.1, page 4
- Embedded Event Manager 2.1 (Software Modularity), page 5
- Embedded Event Manager 2.2, page 5
- Embedded Event Manager 2.3, page 6
- Embedded Event Manager 2.4, page 6
- Event Detectors, page 7
- Embedded Event Manager Actions, page 11
- Embedded Event Manager Environment Variables, page 11
- Embedded Event Manager Policy Creation, page 13

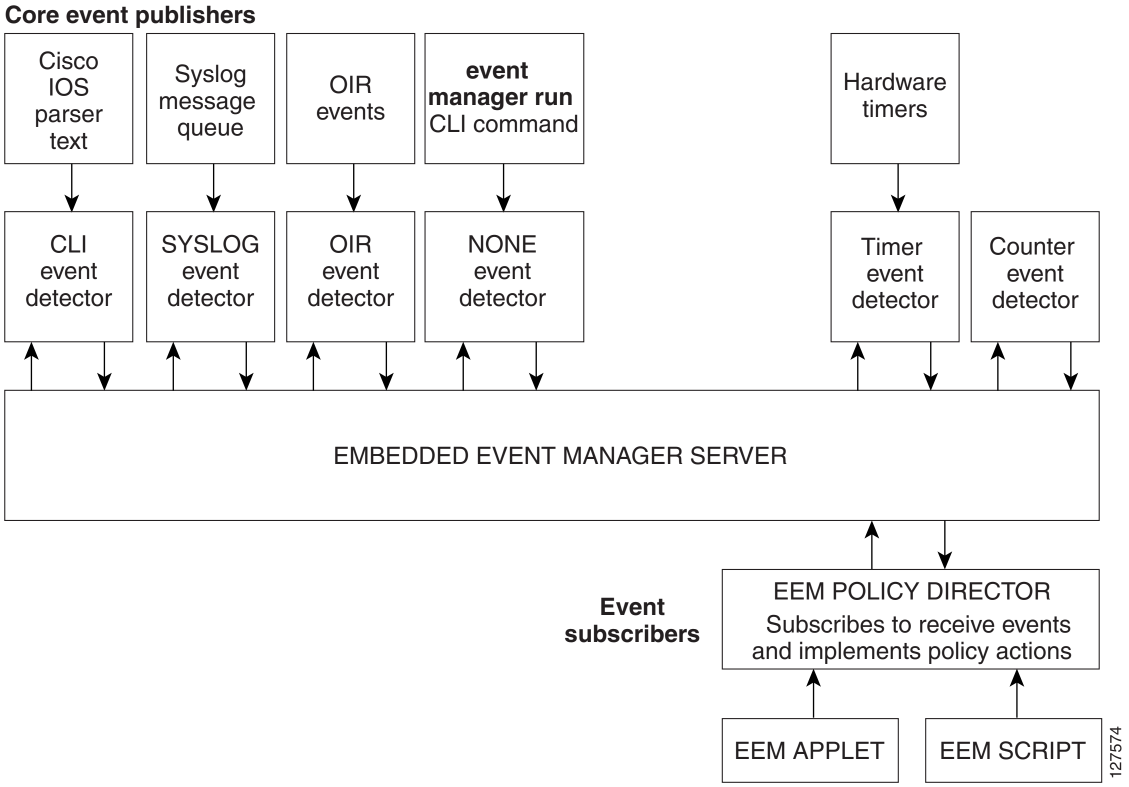
Embedded Event Manager

Event tracking and management has traditionally been performed by devices external to the networking device. Embedded Event Manager (EEM) has been designed to offer event management capability directly in Cisco IOS devices. The on-device, proactive event management capabilities of EEM are useful because not all event management can be done off router because some problems compromise communication between the router and the external network management device. Capturing the state of the router during such situations can be invaluable in taking immediate recovery actions and gathering information to perform root-cause analysis. Network availability is also improved if automatic recovery actions are performed without the need to fully reboot the routing device.

EEM is a flexible, policy-driven framework that supports in-box monitoring of different components of the system with the help of software agents known as event detectors. Figure 1 shows the relationship between the EEM server, core event publishers (event detectors), and the event subscribers (policies). Basically, event publishers screen events and publish them when there is a match on an event specification that is provided by the event subscriber. Event detectors notify the EEM server when an event of interest occurs. The EEM policies that are configured using the Cisco IOS command-line interface (CLI) then implement recovery on the basis of the current state of the system and the actions specified in the policy for the given event.

EEM offers the ability to monitor events and take informational or corrective action when the monitored events occur or when a threshold is reached. An EEM policy is an entity that defines an event and the actions to be taken when that event occurs. There are two types of EEM policies: an applet or a script. An applet is a simple form of policy that is defined within the CLI configuration. A script is a form of policy that is written in Tool Command Language (Tcl).

Figure 1 Embedded Event Manager Core Event Detectors



Embedded Event Manager 1.0

EEM 1.0 is supported in Cisco IOS Releases 12.0(26)S and 12.3(4)T and later releases and introduced Embedded Event Manager. EEM 1.0 introduced the following event detectors:

- SNMP—The Simple Network Management Protocol (SNMP) event detector allows a standard SNMP MIB object to be monitored and an event to be generated when the object matches specified values or crosses specified thresholds.
- Syslog—The syslog event detector allows for screening syslog messages for a regular expression pattern match.

EEM 1.0 introduced the following actions:

- Generating prioritized syslog messages.
- Generating a Cisco Networking Services (CNS) event for upstream processing by CNS devices.
- Reloading the Cisco IOS software.
- Switching to a secondary processor in a fully redundant hardware configuration.

Embedded Event Manager 2.0

EEM 2.0 is supported in Cisco IOS Release 12.2(25)S and later releases and introduced some new features. EEM 2.0 introduced the following event detectors:

- Application-Specific—The application-specific event detector allows any Embedded Event Manager policy to publish an event.
- Counter—The counter event detector publishes an event when a named counter crosses a specified threshold.
- Interface Counter—The interface counter event detector publishes an event when a generic Cisco IOS interface counter for a specified interface crosses a defined threshold.
- Timer—The timer event detector publishes events for the following four different types of timers: absolute-time-of-day, countdown, watchdog, and CRON.
- Watchdog System Monitor (IOSWDSysMon)—The Cisco IOS watchdog system monitor event detector publishes an event when CPU or memory utilization for a Cisco IOS process crosses a threshold.

EEM 2.0 introduced the following actions:

- Setting or modifying a named counter.
- Publishing an application-specific event
- Generating an SNMP trap.

The ability to run a Cisco defined sample policy written using Tool Command Language (Tcl) was introduced. A sample policy was provided that could be stored in the system policy directory.

Embedded Event Manager 2.1

EEM 2.1 is supported in Cisco IOS Release 12.3(14)T, 12.2(18)SXF5, 12.2(28)SB, 12.2(33)SRA, and later releases, and introduced some new features. EEM 2.1 introduced the following new event detectors:

- CLI—The CLI event detector screens command-line interface (CLI) commands for a regular expression match.
- None—The none event detector publishes an event when the Cisco IOS **event manager run** CLI command executes an EEM policy.
- OIR—The online insertion and removal (OIR) event detector publishes an event when a particular hardware insertion or removal event occurs.

EEM 2.1 introduced the following actions:

- Executing a Cisco IOS CLI command.
- Requesting system information when an event occurs.
- Sending a short e-mail.
- Manually running an EEM policy.

EEM 2.1 also permits multiple concurrent policies to be run using the new **event manager scheduler script** command. Support for SNMP event detector rate-based events is provided as is the ability to create policies using Tool Command Language (Tcl).

Embedded Event Manager 2.1 (Software Modularity)

EEM 2.1 (Software Modularity) is supported in Cisco IOS Release 12.2(18)SXF4 and later releases on Cisco IOS Software Modularity images. EEM 2.1 (Software Modularity) introduced the following event detectors:

- GOLD—The Generic Online Diagnostic (GOLD) event detector publishes an event when a GOLD failure event is detected on a specified card and subcard.
- System Manager—The system manager event detector generates events for Cisco IOS Software Modularity process start, normal or abnormal stop, and restart events. The events generated by the system manager allows policies to change the default behavior of the process restart.
- Watchdog System Monitor (WDSysMon)—The Cisco IOS Software Modularity watchdog system monitor event detector detects infinite loops, deadlocks, and memory leaks in Cisco IOS Software Modularity processes.

EEM 2.1 for Software Modularity introduced the ability to display EEM reliability metric data for processes.

Note EEM 2.1 for Software Modularity images also supports the resource and RF event detectors introduced in EEM 2.2, but it does not support the enhanced object tracking event detector or the actions to read and set tracked objects.

Embedded Event Manager 2.2

EEM 2.2 is supported in Cisco IOS Release 12.4(2)T, 12.2(31)SB3, 12.2(33)SRB, and later releases, and introduced some new features. EEM 2.2 introduced the following event detectors:

- Enhanced Object Tracking—The enhanced object tracking event detector publishes an event when the tracked object changes. Enhanced object tracking provides complete separation between the objects to be tracked and the action to be taken by a client when a tracked object changes.
- Resource—The resource event detector publishes an event when the Embedded Resource Manager (ERM) reports an event for the specified policy.
- RF—The redundancy framework (RF) event detector publishes an event when one or more RF events occur during synchronization in a dual Route Processor (RP) system. The RF event detector can also detect an event when a dual RP system continuously switches from one RP to another RP (referred to as a ping-pong situation).

EEM 2.2 introduced the following actions:

- Reading the state of a tracked object.
- Setting the state of a tracked object.

Embedded Event Manager 2.3

EEM 2.3 is supported in Cisco IOS Release 12.2(33)SXH and later releases for the Cisco Catalyst 6500 Series switches and introduces enhancements to the Generic Online Diagnostics (GOLD) Event Detector on that product.

- The **event gold** command was enhanced with the addition of the **action-notify**, **testing-type**, **test-name**, **test-id**, **consecutive-failure**, **platform-action**, and **maxrun** keywords for improved reaction to GOLD test failures and conditions.
- The following platform-wide GOLD Event Detector information can be accessed through new read-only EEM built-in environment variables:
  - Boot-up diagnostic level
  - Card index, name, serial number
  - Port counts
  - Test counts
- The following test-specific GOLD Event Detector information can be accessed through new read-only EEM built-in environment variables (available to EEM applets only):
  - Test name, attribute, total run count
  - Test result per test, port, or device
  - Total failure count, last fail time
  - Error code
  - Occurrence of consecutive failures

These enhancements result in reduced mean time to recovery (MTTR) and higher availability through improved automation and fault detection.

Embedded Event Manager 2.4

EEM 2.4 is supported in Cisco IOS Release 12.4(20)T and later releases, and introduced several new features. EEM 2.4 introduced the following event detectors:

- SNMP Notification—The SNMP notification event detector provides the ability to intercept SNMP trap and inform messages coming into the router. An SNMP notification event is generated when an incoming SNMP trap or inform message matches specified values or crosses specified thresholds.
- RPC—The remote procedure call (RPC) event detector provides the ability to invoke EEM policies from outside the router over an encrypted connection using Secure Shell (SSH). The RPC event detector uses Simple Object Access Protocol (SOAP) data encoding for exchanging XML-based messages. This event detector can be used to run EEM policies and then receive output in a SOAP XML-formatted reply.

EEM 2.4 added enhancements to the following event detectors:

- Interface counter rate-based trigger—This feature adds the ability for an interface event to be triggered based on a rate of change over a period of time. A rate can be specified both for the entry value and the exit value. This feature copies the rate-based functionality that currently exists for the SNMP event detector.

- SNMP delta value—The difference between the monitored Object Identifier (OID) value at the beginning of the monitored period and the actual OID value when the event is published will be provided in the **event_reqinfo** data for both the SNMP event detector and the Interface Counter event detector.

EEM 2.4 introduced the following actions:

- Multiple event support—The ability to run multiple events was introduced. and **show event manager** commands were enhanced to show multiple events.
- Support for parameters—The *parameter* argument has been added to the **event manager run** command. A maximum of 15 parameters can be used.
- Display of Job IDs and completion status—Some of the **show event manager** commands were enhanced to display Job IDs and completion status.
- Bytecode support—Tcl 8 defines a specialized bytecode language (BCL) and includes a just-in-time compiler that translates Tcl scripts to BCL. Byte sequence is executed by a "virtual machine," Tcl_ExecuteByteCode(), or TEBC for short, as often as needed. Currently EEM accepts file extensions, such as *.tcl for user policies and *.tm for system policies. Tcl standard extension for bytecode scripts are *.tbc. Now EEM will accept *.tbc as valid EEM policies.
- Registration substitution enhancement—Supports replacing multiple parameters in the event registration statement lines with a single environment variable.
- Tcl package support

EEM 2.4 introduced the following new commands:

- **attribute (EEM)**
- **correlate**
- **event manager detector rpc**
- **event manager directory user repository**
- **event manager update user policy**
- **event manager scheduler clear**
- **event manager update user policy**
- **event owner, event rpc**
- **event snmp-notification**
- **show event manager detector**
- **show event manager version**
- **trigger (EEM).**

Event Detectors

Embedded Event Manager (EEM) uses software programs known as *event detectors* to determine when an EEM event occurs. Event detectors are separate systems that provide an interface between the agent being monitored, for example Simple Network Management Protocol (SNMP), and the EEM policies where an action can be implemented. Some event detectors are available on every Cisco IOS release, but most event detectors have been introduced in a specific release. For details of which event detector is supported in each Cisco IOS release, see the EEM Event Detectors Available by Cisco IOS Release concept in the "Writing Embedded Event Manager Policies Using the Cisco IOS CLI" or the "Writing Embedded Event Manager Policies Using Tcl" modules. EEM contains the following event detectors.

Application-Specific Event Detector

The application-specific event detector allows any Embedded Event Manager policy to publish an event. When an EEM policy publishes an event it must use an EEM subsystem number of 798 with any event type. If an existing policy is registered for subsystem 798 and a specified event type, a second policy of the same event type will trigger the first policy to run when the specified event is published.

CLI Event Detector

The CLI event detector screens command-line interface (CLI) commands for a regular expression match. When a match is found, an event is published. The match logic is performed on the fully expanded CLI command after the command is successfully parsed and before it is executed. The CLI event detector supports three publish modes:

- Synchronous publishing of CLI events—The CLI command is not executed until the EEM policy exits, and the EEM policy can control whether the command is executed. The read/write variable, _exit_status, allows you to set the exit status at policy exit for policies triggered from synchronous events. If _exit_status is 0, the command is skipped, if _exit_status is 1, the command is run.
- Asynchronous publishing of CLI events—The CLI event is published, and then the CLI command is executed.
- Asynchronous publishing of CLI events with command skipping—The CLI event is published, but the CLI command is not executed.

Counter Event Detector

The counter event detector publishes an event when a named counter crosses a specified threshold. There are two or more participants that affect counter processing. The counter event detector can modify the counter, and one or more subscribers define the criteria that cause the event to be published. After a counter event has been published, the counter monitoring logic can be reset to start monitoring the counter immediately or it can be reset when a second threshold—called an exit value—is crossed.

Enhanced Object Tracking Event Detector

The enhanced object tracking (EOT) event detector publishes an event when the status of a tracked object changes. Object tracking was first introduced into the Hot Standby Router Protocol (HSRP) as a simple tracking mechanism that allowed you to track the interface line-protocol state only. If the line-protocol state of the interface went down, the HSRP priority of the router was reduced, allowing another HSRP router with a higher priority to become active.

Object tracking was enhanced to provide complete separation between the objects to be tracked and the action to be taken by a client when a tracked object changes. Thus, several clients such as HSRP, VRRP, or GLBP can register their interest with the tracking process, track the same object, and each take different action when the object changes. Each tracked object is identified by a unique number that is specified on the tracking command-line interface (CLI). Client processes use this number to track a specific object. The tracking process periodically polls the tracked objects and notes any change of value. The changes in the tracked object are communicated to interested client processes, either immediately or after a specified delay. The object values are reported as either up or down.

Enhanced object tracking is now integrated with EEM to allow EEM to report on a status change of a tracked object and to allow enhanced object tracking to track EEM objects. A new type of tracking object—a stub object—is created. The stub object can be manipulated using the existing CLI commands that already allow tracked objects to be manipulated.

GOLD Event Detector

The GOLD event detector publishes an event when a GOLD failure event is detected on a specified card and subcard.

Interface Counter Event Detector

The interface counter event detector publishes an event when a generic Cisco IOS interface counter for a specified interface crosses a defined threshold. A threshold can be specified as an absolute value or an incremental value. If the incremental value is set to 50, for example, an event would be published when the interface counter increases by 50.

After an interface counter event has been published, the interface counter monitoring logic is reset using two methods. The interface counter is reset either when a second threshold—called an exit value—is crossed or when an elapsed period of time occurs.

None Event Detector

The none event detector publishes an event when the Cisco IOS **event manager run** CLI command executes an EEM policy. EEM schedules and runs policies on the basis of an event specification that is contained within the policy itself. An EEM policy must be identified and registered to be permitted to run manually before the **event manager run** command will execute.

OIR Event Detector

The online insertion and removal (OIR) event detector publishes an event when one of the following hardware insertion or removal events occurs:

- A card is removed.
- A card is inserted.

Route Processors (RPs), line cards, or feature cards can be monitored for OIR events.

Resource Event Detector

The resource event detector publishes an event when the Embedded Resource Manager (ERM) reports an event for the specified policy. The ERM infrastructure tracks resource depletion and resource dependencies across processes and within a system to handle various error conditions. The error conditions are handled by providing an equitable sharing of resources between various applications. The ERM framework provides a communication mechanism for resource entities and allows communication between these resource entities from numerous locations. The ERM framework also helps in debugging CPU and memory-related issues. The ERM monitors system resource usage to better understand scalability needs by allowing you to configure threshold values for resources such as the CPU, buffers, and memory. The ERM event detector is the preferred method for monitoring resources in Cisco IOS software but the ERM event detector is not supported in Software Modularity images. For more details about ERM, go to "Embedded Resource Manager" module.

RF Event Detector

The redundancy framework (RF) event detector publishes an event when one or more RF events occur during synchronization in a dual Route Processor (RP) system. The RF event detector can also detect an event when a dual RP system continuously switches from one RP to another RP (referred to as a ping-pong situation).

RPC Event Detector

The remote procedure call (RPC) event detector provides the ability to invoke EEM policies from outside the router over an encrypted connection using Secure Shell (SSH). The RPC event detector uses Simple Object Access Protocol (SOAP) data encoding for exchanging XML-based messages. This event detector can be used to run EEM policies and then receive output in a SOAP XML-formatted reply.

SNMP Event Detector

The SNMP event detector allows a standard SNMP MIB object to be monitored and an event to be generated when the object matches specified values or crosses specified thresholds.

SNMP Notification Event Detector

The SNMP notification event detector provides the ability to intercept SNMP trap and inform messages coming into the router. An SNMP notification event is generated when an incoming SNMP trap or inform message matches specified values or crosses specified thresholds.

Syslog Event Detector

The syslog event detector allows for screening syslog messages for a regular expression pattern match. The selected messages can be further qualified, requiring that a specific number of occurrences be logged within a specified time. A match on a specified event criteria triggers a configured policy action.

System Manager Event Detector

The system manager event detector generates events for Cisco IOS Software Modularity process start, normal or abnormal stop, and restart events. The events generated by the system manager allows policies to change the default behavior of the process restart.

Timer Event Detector

The timer event detector publishes events for the following four different types of timers:

- An absolute-time-of-day timer publishes an event when a specified absolute date and time occurs.
- A countdown timer publishes an event when a timer counts down to zero.
- A watchdog timer publishes an event when a timer counts down to zero and then the timer automatically resets itself to its initial value and starts to count down again.
- A CRON timer publishes an event using a UNIX standard CRON specification to indicate when the event is to be published. A CRON timer never publishes events more than once per minute.

Watchdog System Monitor (IOSWDSysMon) Event Detector for Cisco IOS

The Cisco IOS watchdog system monitor event detector publishes an event when one of the following occurs:

- CPU utilization for a Cisco IOS task crosses a threshold.
- Memory utilization for a Cisco IOS task crosses a threshold.

Note Cisco IOS processes are now referred to as tasks to distinguish them from Cisco IOS Software Modularity processes.

Two events may be monitored at the same time, and the event publishing criteria can be specified to require one event or both events to cross their specified thresholds.

Watchdog System Monitor (WDSysMon) Event Detector for Cisco IOS Software Modularity

The Cisco IOS Software Modularity watchdog system monitor event detector detects infinite loops, deadlocks, and memory leaks in Cisco IOS Software Modularity processes.

Embedded Event Manager Actions

The CLI-based corrective actions that are taken when event detectors report events enable a powerful on-device event management mechanism. Some EEM actions are available on every Cisco IOS release, but most EEM actions have been introduced in a specific release. For details of which EEM action is supported in each Cisco IOS release, see the EEM Actions Available by Cisco IOS Release concept in the "Writing Embedded Event Manager Policies Using the Cisco IOS CLI" or the "Writing Embedded Event Manager Policies Using Tcl" modules. EEM supports the following actions:

- Executing a Cisco IOS command-line interface (CLI) command.
- Generating a CNS event for upstream processing by Cisco CNS devices.
- Setting or modifying a named counter.
- Switching to a secondary processor in a fully redundant hardware configuration.
- Requesting system information when an event occurs.
- Sending a short e-mail.
- Manually running an EEM policy.
- Publishing an application-specific event.
- Reloading the Cisco IOS software.
- Generating an SNMP trap.
- Generating prioritized syslog messages.
- Reading the state of a tracked object.
- Setting the state of a tracked object.

EEM action CLI commands contain an EEM action label that is a unique identifier that can be any string value. Actions are sorted and run in ascending alphanumeric (lexicographical) key sequence using the label as the sort key. If you are using numbers as labels be aware that alphanumerical sorting will sort 10.0 after 1.0, but before 2.0, and in this situation we recommend that you use numbers such as 01.0, 02.0, and so on, or use an initial letter followed by numbers.

Embedded Event Manager Environment Variables

EEM allows environment variables to be used in EEM policies. Tool Command Language (Tcl) allows global variables to be defined that are known to all procedures within a Tcl script. EEM allows environment variables to be defined using a CLI command, the **event manager environment** command, for use within an EEM policy. All EEM environment variables are automatically assigned to Tcl global variables before a Tcl script is run. There are three different types of environment variables associated with Embedded Event Manager:

- User-defined—Defined by you if you create an environment variable in a policy that you have written.
- Cisco-defined—Defined by Cisco for a specific sample policy.
- Cisco built-in (available in EEM applets)—Defined by Cisco and can be read only or read/write. The read only variables are set by the system before an applet starts to execute. The single read/write variable, _exit_status, allows you to set the exit status at policy exit for policies triggered from synchronous events.

Cisco-defined environment variables (see Table 1) and Cisco system-defined environment variables may apply to one specific event detector or to all event detectors. Environment variables that are user-defined or defined by Cisco in a sample policy are set using the **event manager environment** command. Variables that are used in the EEM policy must be defined before you register the policy. A Tcl policy contains a section called "Environment Must Define" that can be defined to check that any required environment variables are defined before the policy runs.

Cisco built-in environment variables are a subset of the Cisco-defined environment variables and the built-in variables are available to EEM applets only. The built-in variables can be read-only or can be read and write, and these variables may apply to one specific event detector or to all event detectors. For more details and a table listing the Cisco system-defined variables, see the ""Writing Embedded Event Manager Policies Using the Cisco IOS CLI" module.

Note Cisco-defined environment variables begin with an underscore character (_). We strongly recommend that customers avoid the same naming convention to prevent naming conflicts.

Table 1 describes the Cisco-defined variables used in the sample EEM policies. Some of the environment variables do not have to be specified for the corresponding sample policy to run and these are marked as optional.

Table 1 Cisco-Defined Environmental Variables and Examples

Environment Variable	Description	Example
_config_cmd1	The first configuration command that is executed.	**interface Ethernet1/0**
_config_cmd2	(Optional) The second configuration command that is executed.	**no shutdown**
_crash_reporter_debug	(Optional) A value that identifies whether debug information for tm_crash_reporter.tcl will be enabled.	1
_crash_reporter_url	The URL location to which the crash report is sent.	http://www.yourdomain.com/fm/interface_tm.cgi
_cron_entry	A CRON specification that determines when the policy will run. See the "Writing Embedded Event Manager Policies Using Tcl" module for more information about how to specify a cron entry.	0-59/1 0-23/1 * * 0-7
_email_server	A Simple Mail Transfer Protocol (SMTP) mail server used to send e-mail.	mailserver.yourdomain.com
_email_to	The address to which e-mail is sent.	engineer@yourdomain.com
_email_from	The address from which e-mail is sent.	devtest@yourdomain.com
_email_cc	The address to which the e-mail is be copied.	manager@yourdomain.com
_show_cmd	The CLI **show** command to be executed when the policy is run.	**show version**

Table 1 Cisco-Defined Environmental Variables and Examples (continued)

Environment Variable	Description	Example
_syslog_pattern	A regular expression pattern match string that is used to compare syslog messages to determine when the policy runs.	.*UPDOWN.*FastEthernet 0/0.*
_tm_fsys_usage_cron	(Optional) A CRON specification that is used in the **event_register** keyword extension. If unspecified, the _tm_fsys_usage.tcl policy is triggered once per minute.	0-59/1 0-23/1 * * 0-7
_tm_fsys_usage_debug	(Optional) When this variable is set to a value of 1, disk usage information is displayed for all entries in the system.	1
_tm_fsys_usage_ freebytes	(Optional) Free byte threshold for systems or specific prefixes. If free space falls below a given value, a warning is displayed.	disk2:98000000
_tm_fsys_usage_percent	(Optional) Disk usage percentage thresholds for systems or specific prefixes. If disk usage percentage exceeds a given percentage, a warning is displayed. If unspecified, the default disk usage percentage is 80 percent for all systems.	nvram:25 disk2:5

Embedded Event Manager Policy Creation

EEM is a policy driven process in which the EEM policy engine receives notifications when faults and other events occur in the Cisco IOS software system. Embedded Event Manager policies implement recovery based on the current state of the system and the actions specified in the policy for a given event. Recovery actions are triggered when the policy is run.

Although there are some EEM CLI configuration and **show** commands, EEM is implemented through the creation of policies. An EEM policy is an entity that defines an event and the actions to be taken when that event occurs. There are two types of EEM policies: an applet or a script. An applet is a simple form of policy that is defined within the CLI configuration. A script is a form of policy that is written in Tcl.

The creation of an EEM policy involves:

- Selecting the event for which the policy is run.
- Defining the event detector options associated with logging and responding to the event.
- Defining the environment variables, if required.
- Choosing the actions to be performed when the event occurs.

There are two ways to create an EEM policy. The first method is to write applets using CLI commands, and the second method is to write Tcl scripts. Cisco provides enhancements to Tcl in the form of Tcl command extensions that facilitate the development of EEM policies. Scripts are defined off the networking device using an ASCII editor. The script is then copied to the networking device and registered with EEM. When a policy is registered with the Embedded Event Manager, the software examines the policy and registers it to be run when the specified event occurs. Policies can be unregistered or suspended. Both types of policies can be used to implement EEM in your network.

For details on writing EEM policies using the Cisco IOS CLI, go to "Writing Embedded Event Manager Policies Using the Cisco IOS CLI" module.

For details on writing EEM policies using Tcl, go to "Writing Embedded Event Manager Policies Using Tcl" module.

Where to Go Next

- If you want to write EEM policies using the Cisco IOS CLI, see the "Writing Embedded Event Manager Policies Using the Cisco IOS CLI" module.
- If you want to write EEM policies using Tcl, see the "Writing Embedded Event Manager Policies Using Tcl" module.

Additional References

The following sections provide references related to EEM.

Related Documents

Related Topic	Document Title
Network Management commands (including EEM commands): complete command syntax, defaults, command mode, command history, usage guidelines, and examples	*Cisco IOS Network Management Command Reference*
Embedded Event Manager policy writing using the CLI	Writing Embedded Event Manager Policies Using the Cisco IOS CLI module
Embedded Event Manager policy writing using Tcl	Writing Embedded Event Manager Policies Using Tcl module
Embedded Resource Manager	Embedded Resource Manager module

Standards

Standard	Title
No new or modified standards are supported, and support for existing standards has not been modified.	—

MIBs

MIB	MIBs Link
CISCO-EMBEDDED-EVENT-MGR-MIB	To locate and download MIBs for selected platforms, Cisco IOS releases, and feature sets, use Cisco MIB Locator found at the following URL: http://www.cisco.com/go/mibs

RFCs

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

Technical Assistance

Description	Link
The Cisco Support website provides extensive online resources, including documentation and tools for troubleshooting and resolving technical issues with Cisco products and technologies. To receive security and technical information about your products, you can subscribe to various services, such as the Product Alert Tool (accessed from Field Notices), the Cisco Technical Services Newsletter, and Really Simple Syndication (RSS) Feeds. Access to most tools on the Cisco Support website requires a Cisco.com user ID and password.	http://www.cisco.com/techsupport

Feature Information for Embedded Event Manager Overview

Table 2 lists the features in this module and provides links to specific configuration information. Only features that were introduced or modified in Cisco IOS Releases 12.3(14)T, 12.2(25)S, 12.0(26)S, 12.2(18)SXF4, 12.2(28)SB, 12.2(33)SRA, 12.2(33)SXH or a later release appear in the table.

Not all commands may be available in your Cisco IOS software release. For release information about a specific command, see the command reference documentation.

Use Cisco Feature Navigator to find information about platform support and software image support. Cisco Feature Navigator enables you to determine which Cisco IOS and Catalyst OS software images support a specific software release, feature set, or platform. To access Cisco Feature Navigator, go to http://www.cisco.com/go/cfn. An account on Cisco.com is not required.

Note Table 2 lists only the Cisco IOS software release that introduced support for a given feature in a given Cisco IOS software release train. Unless noted otherwise, subsequent releases of that Cisco IOS software release train also support that feature.

Table 2 Feature Information for Embedded Event Manager Overview

Feature Name	Releases	Feature Configuration Information
Embedded Event Manager 1.0	12.0(26)S 12.3(4)T	EEM 1.0 introduced Embedded Event Manager applet creation with the SNMP and Syslog event detectors. EEM 1.0 also introduced the following actions: generating prioritized syslog messages, generating a CNS event for upstream processing by Cisco CNS devices, reloading the Cisco IOS software, and switching to a secondary processor in a fully redundant hardware configuration. The following sections provide information about this feature: • Embedded Event Manager 1.0, page 3 • Event Detectors, page 7 • Embedded Event Manager Actions, page 11 • Embedded Event Manager Policy Creation, page 13 The following commands were introduced by this feature: **action cns-event**, **action force-switchover**, **action reload**, **action syslog**, **debug event manager**, **event manager applet**, **event snmp**, **event syslog**, **show event manager policy registered**.

Table 2 Feature Information for Embedded Event Manager Overview (continued)

Feature Name	Releases	Feature Configuration Information
Embedded Event Manager 2.0	12.2(25)S Cisco IOS XE Release 2.1	EEM 2.0 introduced the application-specific event detector, the counter event detector, the interface counter event detector, the timer event detector, and the IOSWDSysMon event detector. New actions include setting and modifying a named counter, publishing an application-specific event, and generating an SNMP trap. The ability to define environment variables and to run a sample EEM policy (included in the software) written using Tcl was introduced. The following sections provide information about this feature: • Embedded Event Manager 2.0, page 4 • Event Detectors, page 7 • Embedded Event Manager Actions, page 11 • Embedded Event Manager Environment Variables, page 11 • Embedded Event Manager Policy Creation, page 13 The following commands were introduced by this feature: **action counter**, **action publish-event**, **action snmp-trap**, **event application**, **event counter**, **event interface**, **event ioswdsysmon**, **event manager environment**, **event manager history size**, **event manager policy**, **event manager scheduler suspend**, **event timer**, **show event manager environment, show event manager history events, show event manager history traps, show event manager policy available**, **show event manager policy pending**.

Table 2 Feature Information for Embedded Event Manager Overview (continued)

Feature Name	Releases	Feature Configuration Information
Embedded Event Manager 2.1	12.3(14)T 12.2(18)SXF5 12.2(28)SB 12.2(33)SRA Cisco IOS XE Release 2.1	EEM 2.1 introduced some new event detectors and actions with new functionality to allow EEM policies to be run manually and the ability to run multiple concurrent policies. Support for Simple Network Management Protocol (SNMP) event detector rate-based events was provided as was the ability to create policies using Tool Command Language (Tcl). The following sections provide information about this feature: • Embedded Event Manager 2.1, page 4 • Event Detectors, page 7 • Embedded Event Manager Actions, page 11 • Embedded Event Manager Environment Variables, page 11 • Embedded Event Manager Policy Creation, page 13 The following commands were introduced or modified by this feature: **action cli**, **action counter**, **action info**, **action mail**, **action policy**, **debug event manager**, **event cli**, **event manager directory user**, **event manager policy**, **event manager run**, **event manager scheduler script**, **event manager session cli username**, **event none**, **event oir**, **event snmp**, **event syslog**, **set** (EEM), **show event manager directory user**, **show event manager policy registered**, **show event manager session cli username**.

Table 2 Feature Information for Embedded Event Manager Overview (continued)

Feature Name	Releases	Feature Configuration Information
Embedded Event Manager 2.1 (Software Modularity)	12.2(18)SXF4 Cisco IOS Software Modularity images	EEM 2.1 for Software Modularity images introduced the GOLD, system manager, and WDSysMon (Cisco IOS Software Modularity watchdog) event detectors, and the ability to display Cisco IOS Software Modularity processes and process metrics. The following sections provide information about this feature: • Embedded Event Manager 2.1 (Software Modularity), page 5 • Event Detectors, page 7 • Embedded Event Manager Actions, page 11 • Embedded Event Manager Environment Variables, page 11 • Embedded Event Manager Policy Creation, page 13 The following commands were introduced by this feature: **event gold**, **event process**, **show event manager metric process**. **Note** EEM 2.1 for Software Modularity images also supports the resource and RF event detectors introduced in EEM 2.2, but it does not support the enhanced object tracking event detector or the actions to read and set tracked objects.
Embedded Event Manager 2.2	12.4(2)T 12.2(31)SB3 12.2(33)SRB Cisco IOS XE Release 2.1	EEM 2.2 introduced the enhanced object tracking, resource, and RF event detectors. The actions of reading and setting the state of a tracked object were also introduced. The following sections provide information about this feature: • Embedded Event Manager 2.2, page 5 • Event Detectors, page 7 • Embedded Event Manager Actions, page 11 • Embedded Event Manager Environment Variables, page 11 • Embedded Event Manager Policy Creation, page 13 The following commands were introduced or modified by this feature: **action track read**, **action track set**, **default-state**, **event resource**, **event rf**, **event track**, **show track**, **track stub-object**.

Table 2 Feature Information for Embedded Event Manager Overview (continued)

Feature Name	Releases	Feature Configuration Information
Embedded Event Manager 2.3	12.2(33)SXH 12.2(33)SB	EEM 2.3 is supported in Cisco IOS Release 12.2(33)SXH and later releases for the Cisco Catalyst 6500 Series switches and introduces enhancements to the Generic Online Diagnostics (GOLD) Event Detector on that product. The following sections provide information about this feature: • Embedded Event Manager 2.3, page 6 • Event Detectors, page 7 • Embedded Event Manager Actions, page 11 • Embedded Event Manager Environment Variables, page 11 • Embedded Event Manager Policy Creation, page 13 The **event gold** command was enhanced in addition to the Tcl keywords—**action-notify**, **testing-type**, **test-name**, **test-id**, **consecutive-failure**, **platform-action**, and **maxrun**—for improved reaction to GOLD test failures and conditions.
Embedded Event Manager 2.4	12.4(20)T	EEM 2.4 is supported in Cisco IOS Release 12.4(20)T and later releases, and introduced several new features. The following sections provide information about this feature: • Embedded Event Manager 2.4, page 6 • Event Detectors, page 7 • Embedded Event Manager Actions, page 11 • Embedded Event Manager Environment Variables, page 11 • Embedded Event Manager Policy Creation, page 13 The following commands were introduced by this feature: **attribute (EEM)**, **correlate**, **event manager detector rpc**, **event manager directory user repository**, **event manager update user policy**, **event manager scheduler clear**, **event manager update user policy**, **event owner**, **event rpc**, **event snmp-notification**, **show event manager detector**, **show event manager version**, **trigger (EEM)**.
Embedded Syslog Manager	Cisco IOS XE Release 2.1	This feature was introduced on Cisco ASR 1000 Series Routers.

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

Configuring the XML Interface to Syslog Messages

The XML Interface to Syslog Messages features provides command-line interface (CLI) commands for enabling syslog messages to be sent in an Extensible Markup Language (XML) format. Logs in a standardized XML format can be more readily used in external customized monitoring tools.

Specifications for the XML Interface to Syslog Messages Feature

Feature History	
Release	**Modification**
12.2(15)T	This feature was introduced.
Supported Platforms[1]	
All platforms that support standard system message logging. For details, see Cisco Feature Navigator.	

[1] For image and plaform support details and updates, see Cisco Feature Navigator. Access Cisco Feature Navigator at http://www.cisco.com/go/fn. You must have an account on Cisco.com. If you do not have an account or have forgotten your username or password, click Cancel at the login dialog box and follow the instructions that appear.

Contents

- Information About the XML Interface to Syslog Messages Feature, page 2
- How to Configure XML Formatting of Syslog Messages, page 4
- Configuration Examples for XML Formatting of Syslog Messages, page 6
- Additional References, page 8
- Command Reference, page 9
- Glossary, page 10

Americas Headquarters:
Cisco Systems, Inc., 170 West Tasman Drive, San Jose, CA 95134-1706 USA

Information About the XML Interface to Syslog Messages Feature

To configure the XML Interface to Syslog Messages feature, you must understand the following concepts:

- Cisco IOS System Message Logging
- XML-Formatted System Message Logging
- System Logging Message Formatting

Cisco IOS System Message Logging

The Cisco IOS system message logging (syslog) process allows the system to report and save important error and notifications messages, either locally or to a remote logging server. These syslog messages include messages in a standardized format (often called system error messages) and output from **debug** commands. These messages are generated during network operation to assist users and Cisco TAC engineers with identifying the type and severity of a problem, or to aid users in monitoring router activity. Syslog messages can be sent to the console, a monitor (TTY and Telnet connections), the system buffer, or to remote hosts.

Note The system message logging process in Cisco IOS software is abbreviated as “syslog”. The messages generated by this process are called “syslog messages”. However, syslog messages are also referred to in Cisco IOS documenation as “system error messages” or “SEMs”. Note that syslog messages are not restricted to error conditions, and can reflect purely informational messages.

XML-Formatted System Message Logging

XML, a derivative of SGML, provides a representation scheme to structuralize consistently formatted data such as that found in syslog messages.

The XML Interface to Syslog Messages features provides CLI commands for enabling syslog messages to be sent in an XML format. Logs in a standardized XML format can be more readily used in external customized monitoring tools. Within the Cisco IOS software, a closed set of meaningful XML tags are defined and, when enabled, applied to the syslog messages sent to the console, monitor, buffer, or to remote hosts.

Two system logging formats exist in Cisco IOS software: the standard logging format and the XML logging format. This means that you can specify that the standard syslog messages be sent to one remote host while the XML-formatted syslog messages are sent to another host. Similarly, if logging messages are sent to the system buffer, the XML logging buffer is separate from the standard logging buffer, and you can have the standard and XML logging buffers running at the same time.

The XML logging process is dependant on the standard logging process. In most cases, settings for the standard logging process carry over to the XML logging process. For example, the severity level for the **logging buffered xml** command is determined by the level set for the standard **logging buffered** command (or, if not set, by the default severity level for the standard buffer). Similarly, the default size of the XML logging buffer is the same as the standard logging buffer’s default (the default buffer size varies by platform).

System Logging Message Formatting

System logging messages take the following format:

```
%<facility>-<severity>-<mnemonic>: <message-text>
```

For example:

```
%LINK-5-CHANGED: Interface Serial3/3, changed state to administratively down
```

Usually, these messages are proceeded by additional text, such as the timestamp and message sequence number:

```
<sequence-number>: <date or system-up-time> <time>:%<facility>-<severity>-<mnemonic>:
<message-text>
```

For example:

```
000013: Mar 18 14:52:10.039:%LINK-5-CHANGED: Interface Serial3/3, changed state to
administratively down
```

Note The timestamp format used in system logging messages is determined by the **service timestamps** global configuration mode command. The **service sequence-numbers** global configuration command enables or disables the leading sequence number. An asterix (*) before the time indicates that the time may be incorrect because the system clock has not synchronized to a reliable time source.

Table 1 shows the XML tags applied to syslog messages (the XML formatting):

Table 1 XML Tags used for Syslog Message Fields

Tag Applied	Delimited Item
<ios-log-msg></ios-log-message>	Entire syslog message.
<facility></facility>	Facility Name. FACILITY is a code consisting of two or more uppercase letters that indicate the facility to which the message refers. A facility can be a hardware device, a protocol, or a module of the system software.
<severity></severity>	Severity Value. SEVERITY is a single-digit code from 0 to 7 that reflects the severity of the condition. The lower the number, the more serious the situation.
<msg-id></msg-id>	Mnemonic. The MNEMONIC is a code (usually an abbreviated description) that uniquely identifies the type of error or event.
<seq></seq>	The error sequence number.
<time></time>	The timestamp, including date and time, or the system uptime (time since last reboot).

Table 1 XML Tags used for Syslog Message Fields

Tag Applied	Delimited Item
<args></args>	The variables within the message text. The full “human readable” text of the message is not retained in XML. Only the variables are extracted and formatted. The variables within a system error message are identified with brackets (`[chars]`, `[hex]`, `[int]`, and so on) in Cisco IOS documentation. For example: %LINK-5-CHANGED: : Interface [chars], changed state to [chars] For the complete text of syslog messages, see the *Cisco IOS System Error Messages* document, available on Cisco.com.
<arg id=“x”></arg>	A specific argument. “x” is a sequential variable I.D. number, starting with zero.

The following example shows a syslog message in standard format, followed by the same message with XML formatting applied:

Standard Syslog Message Format

000013: *Oct 11 14:52:10.039: %SYS-5-CONFIG_I: Configured from console by vty0 (172.19.208.14)

XML Syslog Message Format

```
<ios-log-msg><facility>SYS</facility><severity>5</severity><msg-id>CONFIG_I</msg-id><seq>0
00013</seq><time>*Oct 11 14:52:10.039</time><args><arg id=“0”>console</arg><arg
id=“1”>vty0 (172.19.208.14)</arg></args></ios-log-msg>
```

Note System logging messages include debugging messages when debugging is enabled on the router and logging is configured to record severity level 7 messages. However, debugging messages do not use the system logging message format. XML formatting will not, therefore, be applied to these messages.

How to Configure XML Formatting of Syslog Messages

Enabling logging in an XML format consists of simply using the appropriate logging command to indicate where syslog messages should be sent, followed by the **xml** keyword. Standard system message logging is enabled by default, but XML formatting of these messages is disabled by default.

As mentioned previously, the XML-formatted logging process is separate than (but dependant on) the standard logging process, so you can configure XML-formatted logging in addition to standard logging if the destination is a remote host or the system buffer.

COMMAND SUMMARY

To enable XML formatting for syslog messages, use one of the following commands in global configuration mode:

- **logging console xml**
- **logging monitor xml**
- **logging buffered xml**
- **logging host** {*ip-address* | *host-name*} **xml**

To view the status of logging and the contents of the XML logging buffer, use the **show logging xml** command in EXEC mode. To clear the contents of the XML logging buffer, use the **clear logging xml** command in EXEC mode.

COMMAND DETAILS

Command or Action	Purpose

`logging console xml [severity-level]`
Example:
`Router(config)#logging console xml informational`
	Enables system message logging to the console connections in XML format.
	Messages at or numerically below the severity level will be logged. The default severity level varies by platform, but is generally level 7 (“debugging”), meaning that messages at all severity levels (0 through 7) are logged.

`logging monitor xml [severity-level]`
Example:
`Router(config)#logging monitor xml 6`
	Enables system message logging to the monitor connections (all available TTY or Telnet connections) in XML format.
	Messages at or numerically below the severity level will be logged. The default severity level varies by platform, but is generally level 7 (“debugging”), meaning that messages at all severity levels (0 through 7) are logged.
	Note that the display of logging messages is often disabled by default, meaning that messages will not be displayed when you log into the terminal until you issue the **terminal monitor** EXEC mode command.

Command or Action	Purpose
logging buffered xml [*xml-buffer-size*] **Example:** Router(config)#logging buffered xml 14336	Enables system message logging to the system buffer in XML format. The severity level for logged messages is determined by the setting of the **logging buffered** command. If the **logging buffered** command has not been used, the default severity level for that command is used. The default severity level varies by platform, but is generally level 7("debugging") , meaning that messages at all severity levels (0 through 7) are logged. For more information on severity levels, see the documentation of the **logging buffered** command. The default XML logging buffer size varies by platform. (The size of the XML logging buffer is the same as the standard logging buffer's default.) The valid range for the XML buffer size is 4096 to 2147483647 bytes (4 Kilobytes to 2 Gigabytes).
logging host {*ip-address* | *host-name*} **xml** **Example:** Router(config)#logging host 209.165.202.132 xml Router(config)#logging host 209.165.201.20 xml	Enables system message logging in XML format to the specified host. By issuing this command more than once, you build a list of syslog servers that receive logging messages. **Note** To send standard logging output to one host and XML-formatted logging output to another host, you must specify a different IP address (or host name) in the **logging host** (standard) command. The default severity level varies by platform, but is generally level 5("notifications") , meaning that messages at severity levels 0 through 7 are logged. To specify the severity level for logging to all remote hosts, use the **logging trap** command.

Configuration Examples for XML Formatting of Syslog Messages

In the following example, logging is enabled and then logging to the standard buffer and to the XML buffer is enabled. The last two **show logging** commands compare the difference between the standard syslog buffer and the XML syslog buffer.

```
Router#show logging
Syslog logging: disabled (10 messages dropped, 5 messages rate-limited, 6 flush)
    Console logging: disabled
    Monitor logging: disabled
    Buffer logging: disabled, xml disabled
    Logging Exception size (8192 bytes)
    Count and timestamp logging messages: disabled
    Trap logging: level informational, 31 message lines logged
Router#config terminal
Enter configuration commands, one per line.  End with CNTL/Z.
Router(config)#logging on
Router(config)#logging buffered
Router(config)#end
Router#show logging
Syslog logging: enabled (10 messages dropped, 5 messages rate-limited, 6 flushed)
    Console logging: disabled
    Monitor logging: disabled
    Buffer logging: level debugging, 1 messages logged, xml disabled
    Logging Exception size (8192 bytes)
    Count and timestamp logging messages: disabled
    Trap logging: level informational, 32 message lines logged

Log Buffer (8192 bytes):

1w0d: %SYS-5-CONFIG_I: Configured from console by console
Router#config terminal
Enter configuration commands, one per line.  End with CNTL/Z.
Router(config)#logging buffered xml
Router(config)#end
Router#show logging
Syslog logging: enabled (10 messages dropped, 5 messages rate-limited, 6 flushes, 0
overruns, xml enabled)
    Console logging: disabled
    Monitor logging: disabled
    Buffer logging: level debugging, 2 messages logged, xml enabled (1 messages logged)
    Logging Exception size (8192 bytes)
    Count and timestamp logging messages: disabled
    Trap logging: level informational, 33 message lines logged

Log Buffer (8192 bytes):

1w0d: %SYS-5-CONFIG_I: Configured from console by console
1w0d: %SYS-5-CONFIG_I: Configured from console by console
Router#show logging xml
<syslog-logging status="enabled" msg-dropped="10" msg-rate-limited="5" flushes="6"
overruns="0"><xml>enabled</xml></syslog-logging>
    <console-logging>disabled</console-logging>
    <monitor-logging>disabled</monitor-logging>
    <buffer-logging level="debugging" messages-logged="2"><xml
messages-logged="1">enabled</xml></buffer-logging>
    <logging-exception size="8192 bytes"></logging-exception>
    <count-and-timestamp-logging status="disabled"></count-and-timestamp-logging>
    <trap-logging level="informational" messages-lines-logged="33"></trap-logging>

<log-xml-buffer size="8192 bytes"></log-xml-buffer>

<ios-log-msg><facility>SYS</facility><severity>5</severity><msg-id>CONFIG_I</msg-id><time>
1w0d</time><args><arg id="0">console</arg>
Router#
```

Additional References

For additional information related to XML Interface to Syslog Messages feature, refer to the following references:

Related Documents

Related Topic	Document Title
system message logging	*"Cisco IOS Configuration Fundamentals Configuration Guide, Release 12.2";* *Troubleshooting and Fault Management* chapter
System Error Messages (SEMs)	"Cisco IOS System Error Messages, Release 12.2"
Debug-level System Messages	"Cisco IOS Debug Command Reference, Release 12.2"

Standards

XML is not currently an Internet Standard. The XML 1.0 Recommendation ("Extensible Markup Language (XML) 1.0 (Second Edition)") is defined at http://www.w3.org/TR/. See also RFC 3076.

MIBs

No relevant MIBs are associated with this feature.

RFCs

RFCs[1]	Title
RFC 3470	"Guidelines for the Use of Extensible Markup Language (XML) within IETF Protocols" (Status: BEST CURRENT PRACTICE)

[1] Not all supported RFCs are listed.

Technical Assistance

Description	Link
Technical Assistance Center (TAC) The Cisco TAC home page contains 30,000 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/public/support/tac/home.shtml
System Error Message Decoder tool For help with researching and resolving your Cisco IOS error messages, try the Cisco IOS Error Message Decoder tool. This tool is made available by the Cisco Technical Assistance Center (TAC) for registered Cisco.com users.	http://www.cisco.com/cgi-bin/Support/Errordecoder/home.pl

Command Reference

The following new and modified commands are pertinent to this feature. To see the command pages for these commands and other commands used with this feature, go to the *Cisco IOS Master Commands List*, Release 12.4, at http://www.cisco.com/univercd/cc/td/doc/product/software/ios124/124mindx/124index.htm.

- **clear logging xml**
- **logging buffered xml**
- **logging console xml**
- **logging host**
- **logging monitor xml**
- **show logging xml**

Note The **logging host** command replaced the **logging** command in Release 12.2(15)T.

Glossary

Note Refer to the *Internetworking Terms and Acronyms* for terms not included in this glossary.

console — In the context of this feature, specifies the connection (CTY or console line) to the console port of the router. Typically, this is a terminal attached directly to the console port, or a PC with a terminal emulation program. Corresponds to the **show terminal** command.

monitor — In the context of this feature, specifies the TTY (TeleTYpe) line connection at a line port. In other words, the "monitor" keyword corresponds to a TTY line connection or a Telnet (terminal emulation) connection. TTY lines (also called ports) communicate with peripheral devices such as terminals, modems, and serial printers. An example of a TTY connection is a PC with a terminal emulation program connected to the device using a dial-up modem.

SEMs—Abbreviation for system error messages. "System error messages" is a term sometimes used for messages generated by the system logging (syslog) process. Syslog messages use a standardized format, and come in 8 severity levels, from "emergencies" (level 0) to "debugging" (level 7). The term "system error message" is actually misleading, as these messages can include notifications of router activity beyond "errors" (such as informational notices).

syslog—Abbreviation for the system message logging process in Cisco IOS software. Also used to identify the messages generated, as in "syslog messages." Technically, the term "syslog" refers only to the process of logging messages to a remote host or hosts, but is commonly used to refer to all Cisco IOS system logging processes.

trap — A trigger in the system software for sending error messages. In the context of this feature, "trap logging" means logging messages to a remote host. The remote host is actually a syslog host from the perspective of the device sending the trap messages, but because the receiving device typically provides collected syslog data to other devices, the receiving device is also referred to as a "syslog server."

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

SNMP Support

Configuring SNMP Support

First Published: December 20, 2006
Last Updated: July 24, 2008

Simple Network Management Protocol (SNMP) is an application-layer protocol that provides a message format for communication between SNMP managers and agents. SNMP provides a standardized framework and a common language that is used for monitoring and managing devices in a network.

This document discusses how to enable an SNMP agent on a Cisco device and how to control the sending of SNMP notifications from the agent. For information about using SNMP management systems, see the appropriate documentation for your network management system (NMS) application.

For a complete description of the router monitoring commands mentioned in this document, see the *Cisco IOS Network Management Command Reference*. To locate documentation of other commands that appear in this document, use the *Cisco IOS Command Reference Master Index* or search online.

Finding Feature Information in This Module

Your Cisco IOS software release may not support all of the features documented in this module. To reach links to specific feature documentation in this module and to see a list of the releases in which each feature is supported, use the "Feature Information for Configuring SNMP Support" section on page 47.

Finding Support Information for Platforms and Cisco IOS and Catalyst OS Software Images

Use Cisco Feature Navigator to find information about platform support and Cisco IOS and Catalyst OS software image support. To access Cisco Feature Navigator, go to http://www.cisco.com/go/cfn. An account on Cisco.com is not required.

Americas Headquarters:
Cisco Systems, Inc., 170 West Tasman Drive, San Jose, CA 95134-1706 USA

Contents

- Restrictions for Configuring SNMP Support, page 2
- Information About Configuring SNMP Support, page 2
- How to Configure SNMP Support, page 13
- Configuration Examples for SNMP Support, page 58
- Additional References, page 63
- Command References, page 65
- Feature Information for Configuring SNMP Support, page 68
- Glossary, page 71

Restrictions for Configuring SNMP Support

Not all Cisco platforms are supported on the features described in this module. Use Cisco Feature Navigator to find information about platform support and Cisco IOS software image support.

Information About Configuring SNMP Support

To configure SNMP support on your network, you should understand the following concepts:

- Components of SNMP, page 2
- SNMP Notifications, page 4
- MIBs and RFCs, page 6
- Versions of SNMP, page 6
- Detailed Interface Registration Information, page 8
- SNMP Support for VPNs, page 9
- MIB Persistence, page 10
- Circuit Interface Identification Persistence, page 11
- Event MIB, page 11
- Expression MIB, page 12
- SNMP Notification Logging, page 13

Components of SNMP

SNMP is an application-layer protocol that provides a message format for communication between SNMP managers and agents. SNMP provides a standardized framework and a common language used for monitoring and managing devices in a network.

The SNMP framework is made up of three parts:

- SNMP manager
- SNMP agent
- MIB

SNMP Manager

The SNMP manager is a system that controls and monitors the activities of network hosts using SNMP. The most common managing system is an NMS. The term NMS can be applied either to a dedicated device used for network management or to the applications used on such a device. Several network management applications are available for use with SNMP and range from simple command-line applications to applications that use GUIs, such as the CiscoWorks2000 products.

SNMP Agent

The SNMP agent is the software component within a managed device that maintains the data for the device and reports this data, as needed, to managing systems. The agent resides on the routing device (router, access server, or switch). To enable an SNMP agent on a Cisco routing device, you must define the relationship between the manager and the agent.

Note Although it is possible to configure a Cisco router to be an SNMP agent, this practice is not recommended. Commands that an agent needs to control the SNMP process are available through the Cisco IOS command-line interface (CLI) without additional configuration.

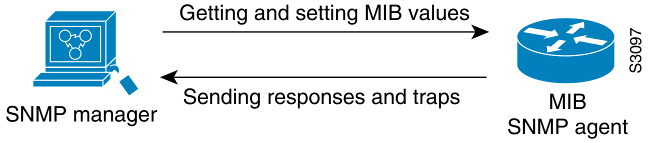
MIB

A MIB is a virtual information storage area for network management information and consists of collections of managed objects. Within a MIB are collections of related objects defined in MIB modules. MIB modules are written in the SNMP MIB module language, as defined in STD 58, RFC 2578, RFC 2579, and RFC 2580 (see the "MIBs and RFCs" section for an explanation of RFC and STD documents). Individual MIB modules are also referred to as MIBs; for example, the Interfaces Group MIB (IF-MIB) is a MIB module within the MIB on your system.

An SNMP agent contains MIB variables whose values the SNMP manager can request or change through Get or Set operations. A manager can get a value from an agent or store a value in that agent. The agent gathers data from the MIB, the repository for information about device parameters and network data. The agent can also respond to manager requests to get or set data.

Figure 1 illustrates the communications between the SNMP manager and agent. A manager sends an agent requests to get and set MIB values. The agent responds to these requests. Independent of this interaction, the agent can send the manager unsolicited notifications (traps or informs) to notify the manager about network conditions.

Figure 1 Communication Between an SNMP Agent and Manager



SNMP Notifications

A key feature of SNMP is its capability to generate unsolicited notifications from an SNMP agent.

Traps and Informs

Unsolicited (asynchronous) notifications can be generated as traps or inform requests (informs). Traps are messages alerting the SNMP manager to a condition on the network. Informs are traps that include a request for confirmation of receipt from the SNMP manager. Notifications can indicate improper user authentication, restarts, the closing of a connection, loss of connection to a neighbor router, or other significant events.

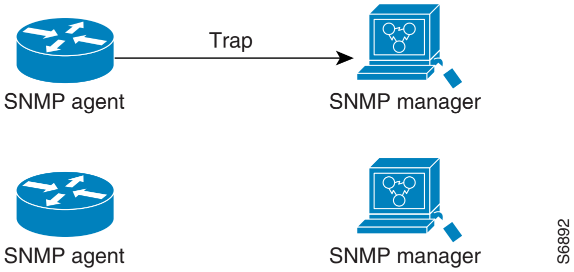
Traps are less reliable than informs because the receiver does not send an acknowledgment when it receives a trap. The sender does not know if the trap was received. An SNMP manager that receives an inform acknowledges the message with an SNMP response protocol data unit (PDU). If the sender never receives a response, the inform can be sent again. Thus, informs are more likely to reach their intended destination.

Traps are often preferred even though they are less reliable because informs consume more resources in the router and in the network. Unlike a trap, which is discarded as soon as it is sent, an inform must be held in memory until a response is received or the request times out. Also, traps are sent only once, whereas an inform may be resent several times. The retries increase traffic and contribute to higher overhead on the network. Use of traps and informs requires a trade-off between reliability and resources. If it is important that the SNMP manager receives every notification, use informs, but if traffic volume or memory usage are concerns and receipt of every notification is not required, use traps.

Figure 2 through Figure 5 illustrate the differences between traps and informs.

Figure 2 shows that an agent successfully sends a trap to an SNMP manager. Although the manager receives the trap, it does not send an acknowledgment. The agent has no way of knowing that the trap reached its destination.

Figure 2 Trap Successfully Sent to SNMP Manager



In Figure 3, the agent successfully sends an inform to the manager. When the manager receives the inform, a response is sent to the agent and the agent knows that the inform reached its destination. Notice that in this example the traffic generated is twice as much as in the interaction shown in Figure 2.

Figure 3 Inform Request Successfully Sent to SNMP Manager

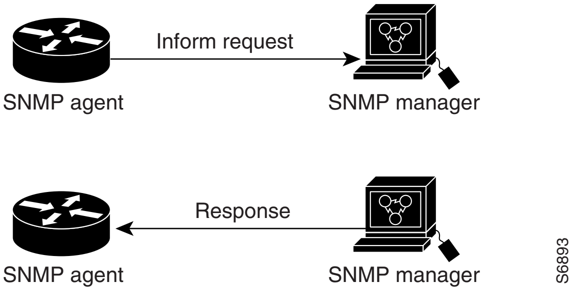


Figure 4 shows an agent sending a trap to a manager that the manager does not receive. The agent has no way of knowing that the trap did not reach its destination. The manager never receives the trap because traps are not resent.

Figure 4 Trap Unsuccessfully Sent to SNMP Manager

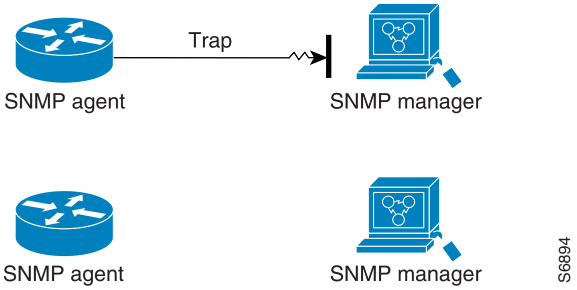
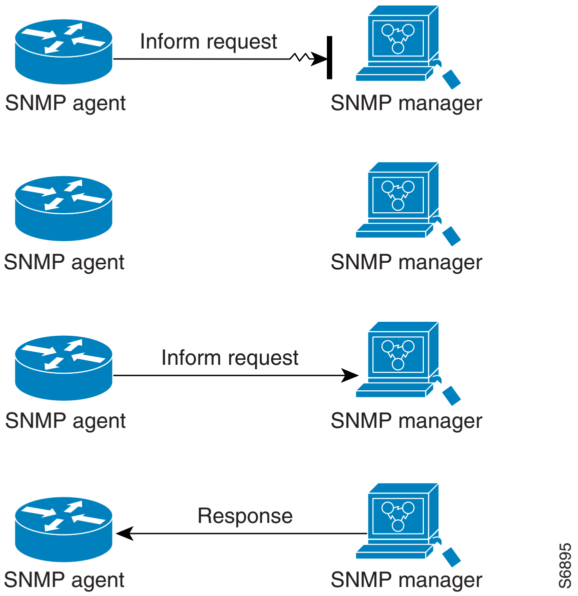


Figure 5 shows an agent sending an inform to a manager that does not reach the manager. Because the manager did not receive the inform, it does not send a response. After a period of time, the agent resends the inform. The manager receives the inform from the second transmission and replies. In this example, more traffic is generated than in the scenario shown in Figure 4 but the notification reaches the SNMP manager.

Figure 5 Inform Unsuccessfully Sent to SNMP Manager



MIBs and RFCs

MIB modules typically are defined in RFC documents submitted to the Internet Engineering Task Force (IETF), an international standards body. RFCs are written by individuals or groups for consideration by the Internet Society and the Internet community as a whole, usually with the intention of establishing a recommended Internet standard. Before being given RFC status, recommendations are published as Internet Draft (I-D) documents. RFCs that have become recommended standards are also labeled as standards (STD) documents. You can learn about the standards process and the activities of the IETF at the Internet Society website at http://www.isoc.org. You can read the full text of all RFCs, I-Ds, and STDs referenced in Cisco documentation at the IETF website at http://www.ietf.org.

The Cisco implementation of SNMP uses the definitions of MIB II variables described in RFC 1213 and definitions of SNMP traps described in RFC 1215.

Cisco provides its own private MIB extensions with every system. Cisco enterprise MIBs comply with the guidelines described in the relevant RFCs unless otherwise noted in the documentation. You can find the MIB module definition files and list of MIBs supported on each Cisco platform on the Cisco MIB website on Cisco.com.

Versions of SNMP

Cisco IOS software supports the following versions of SNMP:

- SNMPv1—Simple Network Management Protocol: a full Internet standard, defined in RFC 1157. (RFC 1157 replaces the earlier versions that were published as RFC 1067 and RFC 1098.) Security is based on community strings.
- SNMPv2c—The community-string based Administrative Framework for SNMPv2. SNMPv2c (the “c” is for “community”) is an experimental Internet protocol defined in RFC 1901, RFC 1905, and RFC 1906. SNMPv2c is an update of the protocol operations and data types of SNMPv2p (SNMPv2 Classic) and uses the community-based security model of SNMPv1.

- SNMPv3—Version 3 of SNMP. SNMPv3 is an interoperable standards-based protocol defined in RFCs 3413 to 3415. SNMPv3 provides secure access to devices by authenticating and encrypting packets over the network.

  The security features provided in SNMPv3 are as follows:

  - Message integrity—Ensuring that a packet has not been tampered with in transit.
  - Authentication—Determining that the message is from a valid source.
  - Encryption—Scrambling the contents of a packet to prevent it from being learned by an unauthorized source.

Both SNMPv1 and SNMPv2c use a community-based form of security. The community of SNMP managers able to access the agent MIB is defined by an IP address access control list (ACL) and password.

SNMPv2c support includes a bulk retrieval mechanism and detailed error message reporting to management stations. The bulk retrieval mechanism supports the retrieval of tables and large quantities of information, minimizing the number of round trips required. The SNMPv2c improved error handling support includes expanded error codes that distinguish different types of errors; these conditions are reported through a single error code in SNMPv1. The following three types of exceptions are also reported: no such object, no such instance, and end of MIB view.

SNMPv3 is a security model in which an authentication strategy is set up for a user and the group in which the user resides. A security level is the permitted level of security within a security model. A combination of a security model and a security level determines which security mechanism is employed when handling an SNMP packet.

Three security models are available: SNMPv1, SNMPv2c, and SNMPv3. Table 1 lists the combinations of security models and levels and their meanings.

Table 1 SNMP Security Models and Levels

Model	Level	Authentication	Encryption	What Happens
v1	noAuthNoPriv	Community String	No	Uses a community string match for authentication.
v2c	noAuthNoPriv	Community String	No	Uses a community string match for authentication.
v3	noAuthNoPriv	Username	No	Uses a username match for authentication.

Table 1 SNMP Security Models and Levels (continued)

Model	Level	Authentication	Encryption	What Happens
v3	authNoPriv	Message Digest 5 (MD5) or Secure Hash Algorithm (SHA)	No	Provides authentication based on the HMAC-MD5 or HMAC-SHA algorithms.
v3	authPriv	MD5 or SHA	Data Encryption Standard (DES)	Provides authentication based on the HMAC-MD5 or HMAC-SHA algorithms. Provides DES 56-bit encryption in addition to authentication based on the CBC-DES (DES-56) standard.

Note SNMPv2p (SNMPv2 Classic) is not supported in Cisco IOS Release 11.2 and later releases. SNMPv2c replaces the Party-based Administrative and Security Framework of SNMPv2p with a Community-based Administrative Framework. SNMPv2c retained the bulk retrieval and error handling capabilities of SNMPv2p.

You must configure an SNMP agent to use the version of SNMP supported by the management station. An agent can communicate with multiple managers, however, and you can configure Cisco IOS software to support communications with one management station using the SNMPv1 protocol, one using the SNMPv2c protocol, and another using SMNPv3.

SNMPv3 supports RFCs 1901 to 1908, 2104, 2206, 2213, 2214, and 2271 to 2275. For additional information about SNMPv3, see RFC 2570, *Introduction to Version 3 of the Internet-standard Network Management Framework* (this is not a standards document).

Detailed Interface Registration Information

The Interface Index Display for SNMP feature introduces new commands and command modifications that allow advanced users of SNMP to view information about the interface registrations directly on the managed agent. You can display MIB information from the agent without using an external NMS.

Note For the purposes of this document, the agent is a routing device running Cisco IOS software.

This feature addresses three objects in the Interfaces MIB: ifIndex, ifAlias, and ifName. For a complete definition of these objects, see the IF-MIB.my file available from the Cisco SNMPv2 MIB website at ftp://ftp.cisco.com/pub/mibs/v2/.

Interface Index

The ifIndex object (ifEntry 1) is called the Interface Index. The Interface Index is a unique value greater than zero that identifies each interface or subinterface on the managed device. This value becomes the interface index identification number.

The CLI command **show snmp mib ifmib ifindex** allows you to view the SNMP Interface Index Identification numbers assigned to interfaces and subinterfaces. An NMS is not required.

Interface Alias

The ifAlias object (ifXEntry 18) is called the Interface Alias. The Interface Alias is a user-specified description of an interface used for SNMP network management. The ifAlias is an object in the Interfaces Group MIB (IF-MIB) that can be set by a network manager to "name" an interface. The ifAlias value for an interface or subinterface can be set using the **description** command in interface configuration mode or subinterface configuration mode or by using a Set operation from an NMS. Previously, ifAlias descriptions for subinterfaces were limited to 64 characters. (The OLD-CISCO-INTERFACES-MIB allows up to 255 characters for the locIfDescr MIB variable, but this MIB does not support subinterfaces.) A new CLI command, **snmp ifmib ifalias long**, configures the system to handle IfAlias descriptions of up to 256 characters. IfAlias descriptions appear in the output of the CLI **show interfaces** command.

Interface Name

The ifName object (ifXEntry 1) is the textual name of the interface. The purpose of the ifName object is to cross reference the CLI representation of a given interface. The value of this object is the name of the interface as assigned by the local device and is suitable for use in CLI commands. If there is no local name or this object is otherwise not applicable, this object contains a zero-length string. No commands introduced by this feature affect the ifName object, but it is discussed here to show its relation to the ifIndex and ifAlias objects.

The **show snmp mib** command shows all objects in the MIB on a Cisco device (similar to a mibwalk). The objects in the MIB tree are sorted using lexical ordering, meaning that object identifiers are sorted in sequential, numerical order. Lexical ordering is important when using the GetNext operation from an NMS because these operations take an object identifier (OID) or a partial OID as input and return the next object from the MIB tree based on the lexical ordering of the tree.

SNMP Support for VPNs

The SNMP Support for VPNs feature allows SNMP traps and informs to be sent and received using virtual private network (VPN) routing/forwarding (VRF) tables. In particular, this feature adds support to Cisco IOS software for the sending and receiving of SNMP traps and informs specific to individual VPNs.

A VPN is a network that provides high connectivity transfers on a shared system with the same usage guidelines as a private network. A VPN can be built on the Internet over IP, Frame Relay, or ATM networks.

A VRF stores per-VPN routing data. It defines the VPN membership of a customer site attached to the network access server (NAS). A VRF consists of an IP routing table, a derived Cisco Express Forwarding table, and guidelines and routing protocol parameters that control the information that is included in the routing table.

The SNMP Support for VPNs feature provides configuration commands that allow users to associate SNMP agents and managers with specific VRFs. The specified VRF is used for sending SNMP traps and informs and responses between agents and managers. If a VRF is not specified, the default routing table for the VPN is used.

Support for VPNs allows you to configure an SNMP agent to accept only SNMP requests from a certain set of VPNs. With this configuration, service providers can provide network management services to their customers, so customers can manage all user VPN devices.

MIB Persistence

The MIB Persistence features allow the SNMP data of a MIB to be persistent across reloads; that is, MIB information retains the same set object values each time a networking device reboots. MIB Persistence is enabled by issuing the **snmp mib persist** command, and the MIB data of all MIBs that have had persistence enabled using this command is then written to NVRAM by issuing the **write mib-data** command. All modified MIB data must be written to NVRAM using the **write mib-data** command.

Both Event and Expression MIBs allow you to configure a value for an object and to set up object definitions. Both also allow rows of data to be modified while the row is in an active state.

Scalar objects are stored every time they are changed, and table entries are stored only if the row is in an active state. The Event MIB has two scalar objects and nine tables to be persisted into NVRAM. Following are the tables:

- mteEventNotificationTable
- mteEventSetTable
- mteEventTable
- mteObjectsTable
- mteTriggerBooleanTable
- mteTriggerDeltaTable
- mteTriggerExistenceTable
- mteTriggerTable
- mteTriggerThresholdTable

The Expression MIB has two scalar objects and three tables to be stored in NVRAM. The scalar objects are expResourceDeltaMinimum and expResourceDeltaWildcardInstanceMaximum. Following are the tables:

- expExpressionTable
- expNameTable
- expObjectTable

Writing MIB data to NVRAM may take several seconds. The length of time depends on the amount of MIB data.

Event MIB Persistence and Expression MIB Persistence both allow MIB objects to be saved from reboot to reboot, allowing long-term monitoring of specific devices and interfaces and configurations of object values that are preserved across reboots.

Circuit Interface Identification Persistence

The Circuit Interface MIB (CISCO-CIRCUIT-INTERFACE-MIB) provides a MIB object (cciDescr) that can be used to identify individual circuit-based interfaces for SNMP monitoring. The Circuit Interface Identification Persistence for SNMP feature maintains this user-defined name of the circuit across reboots, allowing the consistent identification of circuit interfaces. Circuit Interface Identification Persistence is enabled using the **snmp mib persist circuit** global configuration command.

Cisco IOS Release 12.2(2)T introduces the Circuit Interface Identification Persistence for SNMP feature. The Circuit Interface MIB (CISCO-CIRCUIT-INTERFACE-MIB) provides a MIB object (cciDescr) that can be used to identify individual circuit-based interfaces for SNMP monitoring. The Cisco Circuit Interface MIB was introduced in Cisco IOS Release 12.1(3)T.

The Circuit Interface Identification Persistence for SNMP feature maintains the user-defined name of the circuit (defined in the cciDescr object) across reboots, allowing for the consistent identification of circuits.

The Circuit Interface Identification Persistence for SNMP feature is a supplement to the Interface Index Persistence feature introduced in Cisco IOS Release 12.1(3)T and in Cisco IOS Release 12.0(11)S. Circuit Interface Identification Persistence is enabled with the **snmp mib persist circuit** global configuration command. Use this command if you need to consistently identify circuits using SNMP across reboots. This command is disabled by default because this feature uses NVRAM.

In addition, the **show snmp mib ifmib ifindex** EXEC mode command allows you to display the Interfaces MIB ifIndex values directly on your system without an NMS; the **show snmp mib** EXEC mode command allows you to display a list of the MIB module identifiers registered directly on your system with an NMS. And the **snmp ifmib ifalias long** command allows you to specify a description for interfaces or subinterfaces of up to 256 characters in length. Prior to the introduction of this command, ifAlias descriptions for SNMP management were limited to 64 characters.

Event MIB

The Event MIB provides the ability to monitor MIB objects on a local or remote system using SNMP and initiate simple actions whenever a trigger condition is met; for example, an SNMP trap can be generated when an object is modified. When the notifications are triggered through events, the Network Management System (NMS) does not need to constantly poll managed devices to track changes.

By allowing the SNMP notifications to take place only when a specified condition is met, Event MIB reduces the load on affected devices and improves the scalability of network management solutions.

The Event MIB operates based on event, object lists configured for the event, event action, trigger, and trigger test.

Events

The event table defines the activities to be performed when an event is triggered. These activities include sending a notification and setting a MIB object. The event table has supplementary tables for additional objects that are configured according to event action. If the event action is set to notification, notifications are sent out whenever the object configured for that event is modified.

Object List

The objects table lists objects that can be added to notifications based on trigger, trigger test type, or the event that sends a notification. The Event MIB allows wildcarding, which enables you to monitor multiple instances of an object. To specify a group of object identifiers, you can use the wildcard option.

Trigger

The trigger table defines conditions to trigger events. The trigger table lists the objects to be monitored and associates each trigger with an event. An event occurs when a trigger is activated. To create a trigger, you should configure a trigger entry in the mteTriggerTable of the Event MIB. This trigger entry specifies object identifier of the object to be monitored. Each trigger is configured to monitor a single object or a group of objects specified by a wildcard (*). The Event MIB process checks the state of the monitored object at specified intervals.

Trigger Test

The trigger table has supplementary tables for additional objects that are configured based on the type of test performed for a trigger. For each trigger entry type such as existence, theshold, or boolean, the corresponding tables (existence, threshold, and boolean tables) are populated with the information required to perform the test. Event MIB allows you to set event triggers based on existence, threshold, and boolean trigger types. When the specified test on an object returns a value of *true*, the trigger is activated. You can configure Event MIB to send out notifications to the interested host when a trigger is activated.

Expression MIB

The Expression MIB allows you to create expressions based on a combination of objects. The expressions are evaluated according to the sampling method. The Expression MIB supports the following types of object sampling:

- Absolute
- Delta
- Changed

If there are no delta or change values in an expression, the expression is evaluated when a requester attempts to read the value of expression. In this case, all requesters get a newly calculated value.

For expressions with delta or change values, evaluation is performed for every sampling. In this case, requesters get the value as of the last sample period.

Absolute Sampling

Absolute sampling uses the value of the MIB object during sampling.

Delta Sampling

Delta sampling is used for expressions with counters that are identified based on delta (difference) from one sample to the next. Delta sampling requires the application to do continuous sampling, because it uses the value of the last sample.

Changed Sampling

Changed sampling uses the changed value of the object since the last sample.

SNMP Notification Logging

Systems that support SNMP often need a mechanism for recording notification information. This mechanism protects against notifications being lost because they exceeded retransmission limits. The Notification Log MIB provides a common infrastructure for other MIBs in the form of a local logging function. The SNMP Notification Logging feature adds Cisco IOS CLI commands to change the size of the notification log, to set the global ageout value for the log, and to display logging summaries at the command line. The Notification Log MIB improves notification tracking and provides a central location for tracking all MIBs.

Note The Notification Log MIB supports notification logging on the default log only.

How to Configure SNMP Support

There is no specific command that you use to enable SNMP. The first **snmp-server** command that you enter enables the supported versions of SNMP. All other configurations are optional.

Perform the following tasks to configure SNMP support.

- Setting Up System Information, page 12 (optional)
- Configuring SNMP Versions 1 and 2, page 13
- Configuring SNMP Version 3, page 16 (optional)
- Configuring a Router As an SNMP Manager, page 20 (optional)
- Enabling the SNMP Agent Shutdown Mechanism, page 24 (optional)
- Defining the Maximum SNMP Agent Packet Size, page 24 (optional)
- Limiting the Number of TFTP Servers Used via SNMP, page 25 (optional)
- Disabling the SNMP Agent, page 26 (optional)
- Configuring SNMP Notifications, page 27 (optional)
- Configuring Interface Index Display and Interface Indexes and Configuration of Long Name Support, page 34 (optional)
- Configuring SNMP Support for VPNs, page 38 (optional)
- Configuring MIB Persistence, page 39 (optional)
- Configuring Event MIB, page 42 (optional)
- Configuring Expression MIB, page 54 (optional)

Setting Up System Information

You can set the system contact, location, and serial number of the SNMP agent so that these descriptions can be accessed through the configuration file. Although the configuration items described below are optional, setting up this basic information is recommended because it may be useful when troubleshooting your configuration. In addition, the first **snmp-server** command that you issue enables SNMP on the device.

Perform this task as needed.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **snmp-server contact** *text*
4. **snmp-server location** *text*
5. **snmp-server chassis-id** *number*

DETAILED STEPS

	Command or Action	Purpose

Step 1	**`enable`**
	Example:
	`Router> enable`
		Enables privileged EXEC mode.
		• Enter your password if prompted.

Step 2	**`configure terminal`**
	Example:
	`Router# configure terminal`
		Enters global configuration mode.

Step 3	**`snmp-server contact`** *`text`*
	Example:
	`Router(config)# snmp-server contact NameOne`
		Sets the system contact string.

Step 4	**`snmp-server location`** *`text`*
	Example:
	`Router(config)# snmp-server location LocationOne`
		Sets the system location string.

Step 5	**`snmp-server chassis-id`** *`number`*
	Example:
	`Router(config)# snmp-server chassis-id 987654`
		Sets the system serial number.

Configuring SNMP Versions 1 and 2

When you configure SNMP versions 1 and 2, you can optionally create or modify views for community strings to limit which MIB objects an SNMP manager can access.

Perform the following tasks when configuring SNMP version 1 or version 2.

- Creating or Modifying an SNMP View Record, page 13 (optional)
- Creating or Modifying Access Control for an SNMP Community, page 14 (required)
- Configuring a Recipient of an SNMP Trap Operation, page 15 (required)

Prerequisites

- An established SNMP community string that defines the relationship between the SNMP manager and the agent
- A host defined to be the recipient of SNMP notifications

Creating or Modifying an SNMP View Record

You can assign views to community strings to limit which MIB objects an SNMP manager can access. You can use a predefined view or create your own view. If you are using a predefined view or no view at all, skip this task.

Perform this task to create or modify an SNMP view record.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **snmp-server view** *view-name oid-tree* {**included** | **excluded**}
4. **no snmp-server view** *view-name oid-tree* {**included** | **excluded**}

DETAILED STEPS

	Command or Action	Purpose

Step 1	**`enable`**
	Example:
	`Router> enable`
		Enables privileged EXEC mode.
		• Enter your password if prompted.

Step 2	**`configure terminal`**
	Example:
	`Router# configure terminal`
		Enters global configuration mode.

	Command or Action	Purpose

Step 3	**snmp-server view** *view-name oid-tree* {**included** | **excluded**}
	Example:
	`Router(config)# snmp-server view mib2 mib-2 included`
		This example creates a view that includes all objects in the MIB-II subtree.
		• You can enter this command multiple times for the same view record. Later lines take precedence when an object identifier is included in two or more lines

Step 4	**no snmp-server view** *view-name oid-tree* {**included** | **excluded**}
	Example:
	`Router(config)# no snmp-server view mib2 mib-2 included`
		Removes a server view.

Creating or Modifying Access Control for an SNMP Community

Use an SNMP community string to define the relationship between the SNMP manager and the agent. The community string acts like a password to regulate access to the agent on the router. Optionally, you can specify one or more of the following characteristics associated with the string:

- An access list of IP addresses of the SNMP managers that are permitted to use the community string to gain access to the agent.
- A MIB view, which defines the subset of all MIB objects accessible to the given community.
- Read and write or read-only permission for the MIB objects accessible to the community.

Perform this task to create or modify a community string.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **snmp-server community** *string* [**view** *view-name*] [**ro** | **rw**] [**ipv6** *nacl*] [*access-list-number*]
4. **no snmp-server community** *string*

DETAILED STEPS

	Command or Action	Purpose

Step 1	**`enable`**
	Example:
	`Router> enable`
		Enables privileged EXEC mode.
		• Enter your password if prompted.

Step 2	**`configure terminal`**
	Example:
	`Router# configure terminal`
		Enters global configuration mode.

Step 3	**snmp-server community string** [**view** *view-name*] [**ro** | **rw**] [**ipv6** *nacl*] [*access-list-number*] **Example:** `Router(config)# snmp-server community comaccess ro 4`	Defines the community access string. • You can configure one or more community strings.
Step 4	**no snmp-server community** *string* **Example:** `Router(config)# no snmp-server community comaccess`	Removes the community string from the configuration.

Configuring a Recipient of an SNMP Trap Operation

SNMP traps are unreliable because the receiver does not send acknowledgments when it receives traps. The sender does not know if the traps were received. However, a SNMP entity that receives an inform acknowledges the message with a SNMP response protocol data unit (PDU). If the sender never receives the response, the inform can be sent again. Thus, informs are more likely to reach their intended destination.

Compared to traps, informs consume more resources in the agent and in the network. Unlike a trap, which is discarded as soon as it is sent, an inform must be held in memory until a response is received or the request times out. Also, traps are sent only once; an inform may be sent several times. The retries increase traffic and overhead on the network.

If you do not enter a **snmp-server host** command, no notifications are sent. To configure the router to send SNMP notifications, you must enter at least one **snmp-server host** command. If you enter the command without keywords, all trap types are enabled for the host.

To enable multiple hosts, you must issue a separate **snmp-server host** command for each host. You can specify multiple notification types in the command for each host.

When multiple **snmp-server host** commands are given for the same host and type of notification, each succeeding command overwrites the previous command. Only the last **snmp-server host** command will be in effect. For example, if you enter an **snmp-server host inform** command for a host and then enter another **snmp-server host inform** command for the same host, the second command replaces the first.

The **snmp-server host** command is used in conjunction with the **snmp-server enable** command. Use the **snmp-server enable** command to specify which SNMP notifications are sent globally. For a host to receive most notifications, at least one **snmp-server enable** command and the **snmp-server host** command for that host must be enabled.

Some notification types cannot be controlled with the **snmp-server enable** command. For example, some notification types are always enabled and others are enabled by a different command. For example, the linkUpDown notifications are controlled by the **snmp trap link-status** command. These notification types do not require an **snmp-server enable** command.

A *notification-type* option's availability depends on the router type and Cisco IOS software features supported on the router. For example, the envmon notification type is available only if the environmental monitor is part of the system. To see what notification types are available on your system, use the command help (?) at the end of the **snmp-server host** command.

Perform this task to configure the recipient of an SNMP trap operation.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **snmp-server host** *host-id* [**traps** | **informs**][**version** {**1** | **2c** | **3** [**auth** | **noauth** | **priv**]}] *community-string* [**udp-port** *port-number*] [*notification-type*]

DETAILED STEPS

	Command or Action	Purpose

Step 1	**`enable`**
	Example:
	`Router> enable`
		Enables privileged EXEC mode.
		• Enter your password if prompted.

Step 2	**`configure terminal`**
	Example:
	`Router# configure terminal`
		Enters global configuration mode.

Step 3	**`snmp-server host`** *`host-id`* `[`**`traps`**` | `**`informs`**`][`**`version`**` {`**`1`**` | `**`2c`**` | `**`3`** `[`**`auth`**` | `**`noauth`**` | `**`priv`**`]}]` *`community-string`* `[`**`udp-port`** *`port-number`*`]` `[`*`notification-type`*`]`
	Example:
	`Router(config)# snmp-server host 172.16.1.27 version 2c public`
		Specifies whether you want the SNMP notifications sent as traps or informs, the version of SNMP to use, the security level of the notifications (for SNMPv3), and the recipient (host) of the notifications.

Configuring SNMP Version 3

When you configure SNMP version 3 and you want to use the SNMPv3 security mechanism for handling SNMP packets, you must establish SNMP groups and users with passwords.

Perform the following tasks to configure SNMP version 3.

- Specifying SNMP-Server Group Names, page 17 (required)
- Configuring SNMP Server Users, page 19 (required)

Specifying SNMP-Server Group Names

SNMPv3 is a security model. A security model is an authentication strategy that is set up for a user and the group in which the user resides.

No default values exist for authentication or privacy algorithms when you configure the **snmp-server group** command. Also, no default passwords exist. For information about specifying a MD5 password, see the documentation for the **snmp-server user** command.

Perform this task to specify a new SNMP group or a table that maps SNMP users to SNMP views.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **snmp group** [*groupname* {**v1** | **v2c** | **v3** [**auth** | **noauth** | **priv**]}][**read** *readview*] [**write** *writeview*] [**notify** *notifyview*] [**access** *access-list*]
4. **exit**
5. **show snmp group**

DETAILED STEPS

	Command or Action	Purpose

Step 1	**`enable`**
	Example:
	`Router> enable`
		Enables privileged EXEC mode.
		• Enter your password if prompted.

Step 2	**`configure terminal`**
	Example:
	`Router# configure terminal`
		Enters global configuration mode.

Step 3	**`snmp group`** `[`*`groupname`* `{`**`v1`** `|` **`v2c`** `|` **`v3`** `[`**`auth`** `|` **`noauth`** `|` **`priv`**`]}][`**`read`** *`readview`*`] [`**`write`** *`writeview`*`] [`**`notify`** *`notifyview`*`] [`**`access`** *`access-list`*`]`
	Example:
	`Router(config)# snmp-server group group1 v3 auth access lmnop`
		Configures the SNMP server group *group1*, enabling user authentication for members of the named access list *lmnop*.

Step 4	**`exit`**
	Example:
	`Router(config)# exit`
		Exits global configuration mode.

Step 5	**`show snmp group`**
	Example:
	`Router# show snmp group`
		Displays information about each SNMP group on the network.

Examples

The following example shows information about each SNMP group on the network:

```
Router# show snmp group

groupname: ILMI                           security model:v1
readview : *ilmi                          writeview: *ilmi
notifyview: <no notifyview specified>
row status: active

groupname: ILMI                           security model:v2c
readview : *ilmi                          writeview: *ilmi
notifyview: <no notifyview specified>
row status: active

groupname: public                         security model:v1
readview : <no readview specified>        writeview: <no writeview specified>
notifyview: *tv.FFFFFFFF.FFFFFFFF.FFFFFFFF0F
row status: active


groupname: public                         security model:v2c
readview : <no readview specified>        writeview: <no writeview specified>
notifyview: *tv.FFFFFFFF.FFFFFFFF.FFFFFFFF0F
row status: active
```

Configuring SNMP Server Users

To configure a remote user, specify the IP address or port number for the remote SNMP agent of the device where the user resides. Also, before you configure remote users for a particular agent, configure the SNMP engine ID, using the **snmp-server engineID** command with the remote option. The remote agent’s SNMP engine ID is required when computing the authentication and privacy digests from the password. If the remote engine ID is not configured first, the configuration command will fail.

For the *privpassword* and *auth-password* arguments, the minimum length is one character; the recommended length is at least eight characters, and should include both letters and numbers.

SNMP passwords are localized using the SNMP engine ID of the authoritative SNMP engine. For informs, the authoritative SNMP agent is the remote agent. You must configure the remote agent’s SNMP engine ID in the SNMP database before you can send proxy requests or informs to it.

Passwords and Digests

No default values exist for authentication or privacy algorithms when you configure the command. Also, no default passwords exist. The minimum length for a password is one character, although we recommend using at least eight characters for security. If you forget a password, you cannot recover it and will need to reconfigure the user. You can specify either a plain text password or a localized MD5 digest.

If you have the localized MD5 or SHA digest, you can specify that string instead of the plain text password. The digest should be formatted as aa:bb:cc:dd where aa, bb, and cc are hexadecimal values. Also, the digest should be exactly 16 octets in length.

Perform this task to add a new user to an SNMP group.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **snmp-server user** *username groupname* [**remote** *ip-address* [**udp-port** *port*]] {**v1** | **v2c** | **v3** [**encrypted**] [**auth** {**md5** | **sha**} *auth-password*]} [**access** *access-list*]
4. **exit**
5. **show snmp user** [*username*]

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**snmp-server user** *username groupname* [**remote** *ip-address* [**udp-port** *port*]] {**v1** | **v2c** | **v3** [**encrypted**] [**auth** {**md5** | **sha**} *auth-password*]} [**access** *access-list*] **Example:** `Router(config)# snmp-server user user1 group1 v3 auth md5 password123`	Configures a new user to an SNMP group with the plain text password "password123" for the user "user1" in the SNMPv3 group "group1".
Step 4	**exit** **Example:** `Router(config)# exit`	Exits global configuration mode and returns to privileged EXEC mode.
Step 5	**show snmp user** [*username*] **Example:** `Router# show snmp user user123`	Displays the information about the configured characteristics of an SNMP user.

Examples

The following example shows the information about the configured characteristics of the SNMP user1:

```
Router# show snmp user user1

User name: user1
Engine ID: 00000009020000000C025808
storage-type: nonvolatile       active access-list: 10
Rowstatus: active
Authentication Protocol: MD5
Privacy protocol: DES
Group name: group1
```

Configuring a Router As an SNMP Manager

The SNMP manager feature allows a router to act as a network management station—an SNMP client. As an SNMP manager, the router can send SNMP requests to agents and receive SNMP responses and notifications from agents. When the SNMP manager process is enabled, the router can query other SNMP agents and process incoming SNMP traps.

Security Considerations

Most network security policies assume that routers will accept SNMP requests, send SNMP responses, and send SNMP notifications.

With the SNMP manager functionality enabled, the router may also send SNMP requests, receive SNMP responses, and receive SNMP notifications. Your security policy implementation may need to be updated prior to enabling this feature.

SNMP requests typically are sent to User Datagram Protocol (UDP) port 161. SNMP responses are typically sent from UDP port 161. SNMP notifications are typically sent to UDP port 162.

SNMP Sessions

Sessions are created when the SNMP manager in the router sends SNMP requests, such as informs, to a host or receives SNMP notifications from a host. One session is created for each destination host. If there is no further communication between the router and host within the session timeout period, the session will be deleted.

The router tracks statistics, such as the average round-trip time required to reach the host, for each session. Using the statistics for a session, the SNMP manager in the router can set reasonable timeout periods for future requests, such as informs, for that host. If the session is deleted, all statistics are lost. If another session with the same host is later created, the request timeout value for replies will return to the default value.

Sessions consume memory. A reasonable session timeout value should be large enough that regularly used sessions are not prematurely deleted, yet small enough such that irregularly used or one-time sessions are purged expeditiously.

Enabling the SNMP Manager

Perform this task to enable the SNMP manager process and to set the session timeout value.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **snmp-server manager**
4. **snmp-server manager session-timeout** *seconds*
5. **exit**
6. **show snmp**
7. **show snmp sessions** [**brief**]
8. **show snmp pending**

DETAILED STEPS

	Command or Action	Purpose

Step 1	**`enable`**
	Example:
	`Router> enable`
		Enables privileged EXEC mode.
		• Enter your password if prompted.

Step 2	**`configure terminal`**
	Example:
	`Router# configure terminal`
		Enters global configuration mode.

Step 3	**`snmp-server manager`**
	Example:
	`Router(config)# snmp-server manager`
		Enables the SNMP manager.

Step 4	**`snmp-server manager session-timeout`** *`seconds`*
	Example:
	`Router(config)# snmp-server manager session-timeout 30`
		(Optional) Changes the session timeout value.

Step 5	**`exit`**
	Example:
	`Router(config)# exit`
		Exits global configuration mode.

Step 6	**`show snmp`**
	Example:
	`Router# show snmp`
		(Optional) Displays the status of SNMP communications.

Step 7	**`show snmp sessions`** `[`**`brief`**`]`
	Example:
	`Router# show snmp sessions`
		(Optional) Displays displays the status of SNMP sessions.

Step 8	**`show snmp pending`**
	Example:
	`Router# show snmp pending`
		(Optional) Displays the current set of pending SNMP requests.

Examples

The following example shows the status of SNMP communications:

```
Router# show snmp

Chassis: 01506199

37 SNMP packets input
    0 Bad SNMP version errors
    4 Unknown community name
    0 Illegal operation for community name supplied
    0 Encoding errors
    24 Number of requested variables
    0 Number of altered variables
    0 Get-request PDUs
    28 Get-next PDUs
    0 Set-request PDUs

78 SNMP packets output
    0 Too big errors (Maximum packet size 1500)
    0 No such name errors
    0 Bad values errors
    0 General errors
    24 Response PDUs
    13 Trap PDUs


SNMP logging: enabled
    Logging to 172.17.58.33.162, 0/10, 13 sent, 0 dropped.

SNMP Manager-role output packets
    4 Get-request PDUs
    4 Get-next PDUs
    6 Get-bulk PDUs
    4 Set-request PDUs
    23 Inform-request PDUs
    30 Timeouts
    0 Drops

SNMP Manager-role input packets
    0 Inform response PDUs
    2 Trap PDUs
    7 Response PDUs
    1 Responses with errors

SNMP informs: enabled
    Informs in flight 0/25 (current/max)
    Logging to 172.17.217.141.162
        4 sent, 0 in-flight, 1 retries, 0 failed, 0 dropped
    Logging to 172.17.58.33.162
        0 sent, 0 in-flight, 0 retries, 0 failed, 0 dropped
```

The following example displays the status of SNMP sessions:

```
Router# show snmp sessions

Destination: 172.17.58.33.162, V2C community: public
  Round-trip-times: 0/0/0 (min/max/last)
  packets output
    0 Gets, 0 GetNexts, 0 GetBulks, 0 Sets, 4 Informs
    0 Timeouts, 0 Drops
  packets input
    0 Traps, 0 Informs, 0 Responses (0 errors)

Destination: 172.17.217.141.162, V2C community: public, Expires in 575 secs
  Round-trip-times: 1/1/1 (min/max/last)
  packets output
    0 Gets, 0 GetNexts, 0 GetBulks, 0 Sets, 4 Informs
    0 Timeouts, 0 Drops
 packets input
    0 Traps, 0 Informs, 4 Responses (0 errors)
```

The following example shows the current set of pending SNMP requests:

```
Router# show snmp pending
```

```
req id: 47, dest: 172.17.58.33.161, V2C community: public, Expires in 5 secs

req id: 49, dest: 172.17.58.33.161, V2C community: public, Expires in 6 secs

req id: 51, dest: 172.17.58.33.161, V2C community: public, Expires in 6 secs

req id: 53, dest: 172.17.58.33.161, V2C community: public, Expires in 8 secs
```

Enabling the SNMP Agent Shutdown Mechanism

Using SNMP packets, a network management tool can send messages to users on virtual terminals and on the console. This facility operates in a similar fashion to the **send** EXEC command; however, the SNMP request that causes the message to be issued to the users also specifies the action to be taken after the message is delivered. One possible action is a shutdown request. After a system is shut down, typically it is reloaded. Because the ability to cause a reload from the network is a powerful feature, it is protected by the **snmp-server system-shutdown** global configuration command. If you do not issue this command, the shutdown mechanism is not enabled.

Perform this task to enable the SNMP agent shutdown mechanism.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **snmp-server system-shutdown**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**snmp-server system-shutdown** **Example:** `Router(config)# snmp-server system-shutdown`	Enables system shutdown using the SNMP message reload feature.

Defining the Maximum SNMP Agent Packet Size

You can define the maximum packet size permitted when the SNMP agent is receiving a request or generating a reply.

Perform this task to set the maximum permitted packet size.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **snmp-server packetsize** *byte-count*

DETAILED STEPS

	Command or Action	Purpose

Step 1	`enable`
	Example:
	`Router> enable`
		Enables privileged EXEC mode.
		• Enter your password if prompted.

Step 2	`configure terminal`
	Example:
	`Router# configure terminal`
		Enters global configuration mode.

Step 3	**`snmp-server packetsize`** *`byte-count`*
	Example:
	`Router(config)# snmp-server packetsize 512`
		Establishes the maximum packet size.

Limiting the Number of TFTP Servers Used via SNMP

You can limit the number of TFTP servers used for saving and loading configuration files via SNMP by using an access list. Limiting the use of TFTP servers in this way conserves system resources and centralizes the operation for manageability.

Perform this task to limit the number of TFTP servers.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **snmp-server tftp-server-list** *number*

DETAILED STEPS

	Command or Action	Purpose

Step 1	**`enable`**
	Example:
	`Router> enable`
		Enables privileged EXEC mode.
		• Enter your password if prompted.

Step 2	**`configure terminal`**
	Example:
	`Router# configure terminal`
		Enters global configuration mode.

Step 3	**`snmp-server tftp-server-list`** *`number`*
	Example:
	`Router(config)# snmp-server tftp-server-list 12`
		Limits the number of TFTP servers used for configuration file copies via SNMP to the servers in an access list.

Troubleshooting Tips

To monitor SNMP trap activity in real time for the purposes of troubleshooting, use the SNMP **debug** commands, including the **debug snmp packet** EXEC command. For documentation of SNMP **debug** commands, see the *Cisco IOS Debug Command Reference.*

Disabling the SNMP Agent

Perform this task to disable any version of an SNMP agent.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **no snmp-server**

DETAILED STEPS

	Command or Action	Purpose

Step 1	**`enable`**
	Example:
	`Router> enable`
		Enables privileged EXEC mode.
		• Enter your password if prompted.

Step 2	**`configure terminal`**
	Example:
	`Router# configure terminal`
		Enters global configuration mode.

Step 3	**`no snmp-server`**
	Example:
	`Router(config)# no snmp-server`
		Disables SNMP agent operation.

Configuring SNMP Notifications

To configure a router to send SNMP traps or informs, perform the tasks described in the following sections:

- Configuring the Router to Send SNMP Notifications, page 27 (required)
- Changing Notification Operation Values, page 30 (optional)
- Controlling Individual RFC 1157 SNMP Traps, page 31 (optional)
- Configuring SNMP Notification Log Options, page 32 (optional)

Note Most Cisco IOS commands use the word "traps" in their command syntax. Unless there is an option within the command to specify either traps or informs, the keyword **traps** should be taken to mean traps, informs, or both. Use the **snmp-server host** command to specify whether you want SNMP notifications to be sent as traps or informs.

The SNMP Proxy manager must be available and enabled on a device for informs to be used. The SNMP Proxy manager is shipped with PLUS software images only.

Configuring the Router to Send SNMP Notifications

Perform this task to configure the router to send traps or informs to a host.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **snmp-server engineID remote** *remote-ip-address remote-engineID*
4. **snmp-server user** *username groupname* [**remote** *host* [**udp-port** *port*] {**v1** | **v2c** | **v3** [**encrypted**] [**auth** {**md5** | **sha**} *auth-password*]} [**access** *access-list*]

5. **snmp group** *groupname* {**v1** | **v2c** | **v3** {**auth** | **noauth** | **priv**}} [**read** *readview*] [**write** *writeview*] [**notify** *notifyview*] [**access** *access-list*]
6. **snmp-server host** *host* [**traps** | **informs**] [**version** {**1** | **2c** | **3** [**auth** | **noauth** | **priv**]}] *community-string* [*notification-type*]
7. **snmp-server enable traps** [*notification-type* [*notification-options*]]

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. - Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**snmp-server engineID remote** *remote-ip-address remote-engineID* **Example:** `Router(config)# snmp-server engineID remote 172.16.20.3 80000009030000B064EFE100`	Specifies the SNMP engine ID and configures the VRF name traps-vrf for SNMP communications with the remote device at 172.16.20.3.
Step 4	**snmp-server user** *username groupname* [**remote** *host* [**udp-port** *port*] {**v1** | **v2c** | **v3** [**encrypted**] [**auth** {**md5** | **sha**} *auth-password*]} [**access** *access-list*] **Example:** `Router(config)# snmp-server user abcd public remote 172.16.20.3 v3 encrypted auth md5 publichost remotehostusers`	Configures an SNMP user to be associated with the host created in Step 3. **Note** You cannot configure a remote user for an address without first configuring the engine ID for that remote host. This restriction is imposed in the design of these commands; if you try to configure the user before the host, you will receive a warning message and the command will not be executed.
Step 5	**snmp group** *groupname* {**v1** | **v2c** | **v3** {**auth** | **noauth** | **priv**}} [**read** *readview*] [**write** *writeview*] [**notify** *notifyview*] [**access** *access-list*] **Example:** `Router(config)# snmp group GROUP1 v2c auth read viewA write viewA notify viewB`	Configures an SNMP group.

	Command or Action	Purpose
Step 6	**snmp-server host** *host* [**traps** | **informs**] [**version** {**1** | **2c** | **3** [**auth** | **noauth** | **priv**]}] *community-string* [*notification-type*] **Example:** `Router(config)# snmp-server host myhost.host3.com informs version 3 public`	Specifies whether you want the SNMP notifications sent as traps or informs, the version of SNMP to use, the security level of the notifications (for SNMPv3), and the recipient (host) of the notifications. • The **snmp-server host** command specifies which hosts will receive SNMP notifications, and whether you want the notifications sent as traps or informs.
Step 7	**snmp-server enable traps** [*notification-type* [*notification-options*]] **Example:** `Router(config)# snmp-server enable traps bgp`	Enables sending of traps or informs and specifies the type of notifications to be sent. • If a *notification-type* is not specified, all supported notification will be enabled on the router. • To discover which notifications are available on your router, enter the **snmp-server enable traps ?** command. • The **snmp-server enable traps** command globally enables the production mechanism for the specified notification types (such as Border Gateway Protocol [BGP] traps, config traps, entity traps, Hot Standby Router Protocol [HSRP] traps, and so on).

Changing Notification Operation Values

You can specify a value other than the default for the source interface, message (packet) queue length for each host, or retransmission interval.

Perform this task to change notification operation values as needed.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **snmp-server trap-source** *interface*
4. **snmp-server queue-length** *length*
5. **snmp-server trap-timeout** *seconds*
6. **snmp-server informs** [**retries** *retries*] [**timeout** *seconds*] [**pending** *pending*]

DETAILED STEPS

Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.

Step 3	**`snmp-server trap-source`** *`interface`*
	Example:
	`Router(config)# snmp-server trap-source ethernet 2/1`
		Sets the IP address for the Ethernet interface in slot2, port 1 as the source for all SNMP notifications.

Step 4	**`snmp-server queue-length`** *`length`*
	Example:
	`Router(config)# snmp-server queue-length 50`
		Establishes the message queue length for each notification.
		• This example shows the queue length set to 50 entries.

Step 5	**`snmp-server trap-timeout`** *`seconds`*
	Example:
	`Router(config)# snmp-server trap-timeout 30`
		Defines how often to resend notifications on the retransmission queue.

Step 6	**`snmp-server informs`** [**`retries`** *`retries`*] [**`timeout`** *`seconds`*] [**`pending`** *`pending`*] **Example:** `Router(config)# snmp-server informs retries 10 timeout 30 pending 100`	Configures inform-specific operation values. • This example sets the maximum number of times to resend an inform, the number of seconds to wait for an acknowledgment before resending, and the maximum number of informs waiting for acknowledgments at any one time.

Controlling Individual RFC 1157 SNMP Traps

Starting with Cisco IOS Release 12.1(3)T, you can globally enable or disable authenticationFailure, linkUp, linkDown, warmStart, and coldStart traps or informs individually. (These traps constitute the "generic traps" defined in RFC 1157.) Note that linkUp and linkDown notifications are enabled by default on specific interfaces but will not be sent unless they are enabled globally.

Perform this task to enable the authenticationFailure, linkUp, linkDown, warmStart, and coldStart notification types.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **snmp-server enable traps snmp [authentication**] [**linkup**] [**linkdown**] [**warmstart**] [**coldstart**]
4. **interface** *type slot/port*
5. **no snmp-server link status**

DETAILED STEPS

Step 1	**`enable`**
	Example:
	`Router> enable`
		Enables privileged EXEC mode.
		• Enter your password if prompted.

Step 2	**`configure terminal`**
	Example:
	`Router# configure terminal`
		Enters global configuration mode.

Step 3	**`snmp-server enable traps snmp`** `[`**`authentication`**`] [`**`linkup`**`] [`**`linkdown`**`] [`**`warmstart`**`] [`**`coldstart`**`]`
	Example:
	`Router(config)# snmp-server enable traps snmp`
		Enables RFC 1157 generic traps.
		• When used without any of the optional keywords, enables authenticationFailure, linkUp, linkDown, warmStart, and coldStart traps.
		• When used with keywords, enables only the trap types specified. For example, to globally enable only linkUp and linkDown SNMP traps or informs for all interfaces, use the **snmp-server enable traps snmp linkup linkdown** form of this command.

Step 4	**`interface`** *`type slot/port`*
	Example:
	`Router(config)# interface Ethernet 0/1`
		Enters interface configuration mode for a specific interface.

Step 5	**`no snmp-server link status`**
	Example:
	`Router(config-if)# no snmp-server link status`
		Disables the sending of linkUp and linkDown notifications.

Configuring SNMP Notification Log Options

Perform this task to configure SNMP notification log options. These options allow you to control the log size and timing values. The SNMP log can become very large and long if left unmodified.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **snmp mib notification-log default**
4. **snmp mib notification-log globalageout** *seconds*
5. **snmp mib notification-log globalsize** *size*
6. **exit**
7. **show snmp mib notification-log**

DETAILED STEPS

Step 1	**`enable`**
	Example:
	`Router> enable`
		Enables privileged EXEC mode.
		- Enter your password if prompted.

Step 2	**`configure terminal`**
	Example:
	`Router# configure terminal`
		Enters global configuration mode.

Step 3	**`snmp mib notification-log default`**
	Example:
	`Router(config)# snmp mib notification-log default`
		Creates an unnamed SNMP notification log.

Step 4	**`snmp mib notification-log globalageout`** *`seconds`*
	Example:
	`Router(config)# snmp mib notification-log globalageout 20`
		Sets the maximum amount of time SNMP notification log entries remain in the system memory.
		- In this example, the system is configured to delete entries in the SNMP notification log that were logged more than 20 minutes ago.

Step 5	**`snmp mib notification-log globalsize`** *`size`*
	Example:
	`Router(config)# snmp mib notification-log globalsize 600`
		Sets the maximum number of entries that can be stored in all SNMP notification logs.

Step 6	**`exit`**
	Example:
	`Router(config)# exit`
		Exits global configuration mode.

Step 7	**`show snmp mib notification-log`**
	Example:
	`Router# show snmp mib notification-log`
		Displays information about the state of the local SNMP notification logging.

Examples

This example shows information about the state of local SNMP notification logging:

```
Router# show snmp mib notification-log

GlobalAgeout 20, GlobalEntryLimit 600
Total Notifications logged in all logs 0
Log Name"", Log entry Limit 600, Notifications logged 0
Logging status enabled
Created by cli
```

Configuring Interface Index Display and Interface Indexes and Configuration of Long Name Support

The display of Interface Indexes lets advanced users of SNMP view information about the interface registrations directly on a managed agent. An external NMS is not required.

Configuration of Long Alias Names for the interfaces lets users configure the ifAlias (the object defined in the MIB whose length is restricted to 64) up to 255 bytes.

Prerequisites

SNMP is enabled on your system.

Restrictions

The Interface Index Display and Interface Alias Long Name Support feature is not supported on all Cisco platforms. Use Cisco Feature Navigator to find information about platform support and Cisco IOS software image support.

Perform this task to configure the IF-MIB to retain ifAlias values of longer than 64 characters and to configure the ifAlias values for an interface.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **snmp ifmib ifalias long**
4. **interface** *type number*
5. **description** *text-string*
6. **exit**
7. **show snmp mib**
8. **show snmp mib ifmib ifindex** [*interface-type*] [*slot/*] [*port-adapter/*] [*port*]

DETAILED STEPS

Step 1	**`enable`**
	Example:
	`Router> enable`
		Enables privileged EXEC mode.
		• Enter your password if prompted.

Step 2	**`configure terminal`**
	Example:
	`Router# configure terminal`
		Enters global configuration mode.

Step 3	**`snmp ifmib ifalias long`**
	Example:
	`Router(config)# snmp ifmib ifalias long`
		Configures the Interfaces MIB (IF-MIB) on the system to return ifAlias values of longer than 64 characters to a Network Management System.

Step 4	**`interface`** *`type number`*
	Example:
	`Router(config)# interface ethernet 2/4`
		Enters interface configuration mode.
		• The form of this command varies depending on the interface being configured.

Step 5	**`description`** *`text-string`*
	Example:
	`Router(config)# description This text string description can be up to 256 characters long`
		Configures a free-text description of the specified interface.
		• This description can be up to 256 characters in length and is stored as the ifAlias object value in the IF-MIB.

Step 6	**`exit`**
	Example:
	`Router(config)# exit`
		Exits global configuration mode.

Step 7	**`show snmp mib`**
	Example:
	`Router# show snmp mib`
		Displays a list of the MIB module instance identifiers registered on your system.
		• The resulting display could be lengthy.

Step 8	**`show snmp mib ifmib ifindex`** `[`*`interface-type`*`] [`*`slot`*`/] [`*`port-adapter`*`/] [`*`port`*`]`
	Example:
	`Router# show snmp mib ifmib ifIndex Ethernet 2/0`
		Displays the Interfaces MIB ifIndex values registered on your system for all interfaces or the specified interface.

Note To verify that the ifAlias values of longer than 64 characters, perform an SNMP MIB walk for the ifMIB ifAlias variable from an NMS and verify that the entire description is displayed in the values for ifXEntry.18.

The description for interfaces also appears in the output of the **more system:running config** privileged EXEC mode command.

Examples

The following example shows a list of the MIB module instance identifiers registered on your system. The resulting display could be lengthy. Only a small portion is shown here.

```
Router# show snmp mib

system.1
system.2
sysUpTime
system.4
system.5
system.6
system.7
system.8
sysOREntry.2
sysOREntry.3
sysOREntry.4
interfaces.1
```

```
ifEntry.1
ifEntry.2
ifEntry.3
ifEntry.4
ifEntry.5
ifEntry.6
ifEntry.7
ifEntry.8
ifEntry.9
ifEntry.10
ifEntry.11

 --More--

captureBufferEntry.2
captureBufferEntry.3
captureBufferEntry.4
captureBufferEntry.5
captureBufferEntry.6
captureBufferEntry.7
capture.3.1.1
eventEntry.1
eventEntry.2
eventEntry.3
eventEntry.4
eventEntry.5
eventEntry.6

eventEntry.7
logEntry.1
logEntry.2
logEntry.3
logEntry.4
rmon.10.1.1.2
rmon.10.1.1.3
rmon.10.1.1.4
rmon.10.1.1.5
rmon.10.1.1.6
rmon.10.1.1.7
rmon.10.2.1.2
rmon.10.2.1.3
rmon.10.3.1.2

--More--
```

The following example shows output for the Interfaces MIB ifIndex values registered on a system for a specific interface:

```
Router# show snmp mib ifmib ifIndex Ethernet2/0

Ethernet2/0: Ifindex = 2
```

The following example shows output for the Interfaces MIB ifIndex values registered on a system for all interfaces:

```
Router# show snmp mib ifmib ifindex

ATM1/0: Ifindex = 1

ATM1/0-aal5 layer: Ifindex = 12

ATM1/0-atm layer: Ifindex = 10
```

```
ATM1/0.0-aal5 layer: Ifindex = 13

ATM1/0.0-atm subif: Ifindex = 11

ATM1/0.9-aal5 layer: Ifindex = 32

ATM1/0.9-atm subif: Ifindex = 31

ATM1/0.99-aal5 layer: Ifindex = 36

ATM1/0.99-atm subif: Ifindex = 35

Ethernet2/0: Ifindex = 2

Ethernet2/1: Ifindex = 3

Ethernet2/2: Ifindex = 4

Ethernet2/3: Ifindex = 5

Null0: Ifindex = 14

Serial3/0: Ifindex = 6

Serial3/1: Ifindex = 7

Serial3/2: Ifindex = 8

Serial3/3: Ifindex = 9
```

Troubleshooting Tips

An alternative to using the ifAlias value for the identification of interfaces across reboots is to use the cciDescr object in the Cisco Circuit Interface MIB (CISCO-CIRCUIT-INTERFACE-MIB.my). This MIB object can be used only for circuit-based interfaces such as ATM or Frame Relay interfaces. Cisco IOS Release 12.2(2)T introduced the Circuit Interface Identification Persistence for SNMP feature, which maintains the user-defined name of the circuit (defined in the cciDescr object) across reboots, allowing for the consistent identification of circuit-based interfaces.

Configuring SNMP Support for VPNs

This section describes how to configure SNMP support for VPNs. The SNMP Support for VPNs feature provides configuration commands that allow users to associate SNMP agents and managers with specific VRFs. The specified VRF is used to send SNMP traps and informs and responses between agents and managers. If a VRF is not specified, the default routing table for the VPN is used.

Support for VPNs allows users to configure an SNMP agent to only accept SNMP requests from a certain set of VPNs. With this configuration, providers can provide network management services to their customers who then can manage all user VPN devices.

Restrictions

- This feature is not supported on all Cisco platforms. Use Cisco Feature Navigator to find information about platform support and Cisco IOS software image support.

- Not all MIBs are VPN aware. For more information about VPN aware MIBs see the following URL: http://www.cisco.com/univercd/cc/td/doc/product/software/ios123/123newft/123t/123t_2/gtsnmpvp.htm

Perform this task to configure SNMP over a specific VPN.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **snmp-server host** *host-address* [**traps** | **informs**][**version** {**1** | **2c** | **3** [**auth** | **noauth** | **priv**]}] *community-string* [**udp-port** *port*] [*notification-type*] [**vrf** *vrf-name*]
4. **snmp-server engineID remote** *ip-address* [**udp-port** *udp-port-number*] [**vrf** *vrf-name*] *engineid-string*
5. **exit**
6. **show snmp-server host**

DETAILED STEPS

Step 1	**`enable`**
	Example:
	`Router> enable`
		Enables privileged EXEC mode.
		• Enter your password if prompted.

Step 2	**`configure terminal`**
	Example:
	`Router# configure terminal`
		Enters global configuration mode.

Step 3	**snmp-server host** *host-address* [**traps** | **informs**][**version** {**1** | **2c** | **3** [**auth** | **noauth** |**priv**]}] *community-string* [**udp-port** *port*] [*notification-type*] [**vrf** *vrf-name*] **Example:** `Router(config)# snmp-server host company.com public vrf trap-vrf`	Specifies the recipient of an SNMP notification operation and specifies the VRF table to be used for the sending of SNMP notifications.
Step 4	**snmp-server engineID remote** *ip-address* [**udp-port** *udp-port-number*] [**vrf** *vrf-name*] *engineid-string* **Example:** `Router(config)# snmp-server engineID remote 172.16.20.3 vrf traps-vrf 80000009030000B064EFE100`	Configures a name for the remote SNMP engine on a router when configuring SNMP over a specific VPN for a remote SNMP user.

Step 5	**`exit`**
	Example:
	`Router(config)# exit`
		Exits global configuration mode.

Step 6	**`show snmp-server host`**
	Example:
	`Router(config)# show snmp-server host`
		Displays the SNMP configuration and verifies that the SNMP Support for VPNs feature is configured properly.

Configuring MIB Persistence

The MIB Persistence features allow the SNMP data of a MIB to be persistent across reloads; that is, MIB information retains the same set of object values each time a networking device reboots. The following sections contain tasks for using Distributed Management Event and Expression MIB persistence.

- Enabling and Disabling Event MIB Persistence, page 40 (optional)
- Enabling and Disabling Expression MIB Persistence, page 41 (optional)

Prerequisites

- SNMP is configured on your networking device
- Values for Event MIB and Expression MIB have been configured

Restrictions

- If the number of MIB objects to persist increases, NVRAM storage capacity may be strained. Occasionally, the time taken to write MIB data to NVRAM may be longer than expected.
- The Distributed Management Event MIB Persistence feature is not supported on all Cisco platforms. Use Cisco Feature Navigator to find information about platform support and Cisco IOS software image support.

Enabling and Disabling Event MIB Persistence

Perform this task to configure Event MIB Persistence.

Note Event MIB Persistence is disabled by default.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **snmp mib persist event**
4. **no snmp mib persist event**

5. **exit**
6. **write mib-data**
7. **copy running-config startup-config**

DETAILED STEPS

Step 1	**`enable`**
	Example:
	`Router> enable`
		Enables privileged EXEC mode.
		- Enter your password if prompted.

Step 2	**`configure terminal`**
	Example:
	`Router# configure terminal`
		Enters global configuration mode.

Step 3	**`snmp mib persist event`**
	Example:
	`Router(config)# snmp mib persist event`
		Enables MIB Persistence for Event MIB.

Step 4	**`no snmp mib persist event`**
	Example:
	`Router(config)# no snmp mib persist event`
		(Optional) Disables MIB Persistence for Event MIB.

Step 5	**`exit`**
	Example:
	`Router(config)# exit`
		Exits global configuration mode.

Step 6	**`write mib-data`**
	Example:
	`Router(config)# write mib-data`
		Saves Event MIB Persistence configuration data to NVRAM.

Step 7	**`copy running-config startup-config`**
	Example:
	`Router(config)# copy running-config startup-config`
		Copies the running configuration to the startup configuration.

Enabling and Disabling Expression MIB Persistence

Perform this task to configure Expression MIB Persistence.

Note Expression MIB Persistence is disabled by default.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **snmp mib persist expression**
4. **no snmp mib persist expression**
5. **exit**
6. **write mib-data**
7. **copy running-config startup-config**
8. **more system:running-config**

DETAILED STEPS

Step 1	**`enable`**
	Example:
	`Router> enable`
		Enables privileged EXEC mode.
		• Enter your password if prompted.

Step 2	**`configure terminal`**
	Example:
	`Router# configure terminal`
		Enters global configuration mode.

Step 3	**`snmp mib persist expression`**
	Example:
	`Router(config)# snmp mib persist expression`
		Enables MIB Persistence for Expression MIB.

Step 4	**`no snmp mib persist expression`**
	Example:
	`Router(config)# no snmp mib persist expression`
		(Optional) Disables MIB Persistence for Expression MIB.

Step 5	**`exit`**
	Example:
	`Router(config)# exit`
		Exits global configuration mode.

Step 6	**`write mib-data`**
	Example:
	`Router(config)# write mib-data`
		Saves Expression MIB Persistence configuration data to NVRAM.

Step 7	**copy running-config startup-config** **Example:** `Router(config)# copy running-config startup-config`	Copies the running configuration to the startup configuration.
Step 8	**more system:running-config** **Example:** `Router(config)# more system:running-config`	Displays the currently running configuration. • Use this command to verify MIB persistence configuration.

Configuring Event MIB

Event MIB can be configured using SNMP directly. In this procedure, the Event MIB is configured to monitor the delta values of ifInOctets for all interfaces once per minute. If any of the samples exceed the specified threshold, a trap notification will be sent.

However, in the Cisco IOS Release 12.4(20)T, the Event MIB feature is enhanced to add CLIs to configure events, event action, and trigger.

This section contains the following tasks to configure Event MIB:

- Configuring Scalar Variables, page 42
- Configuring Event MIB Object List, page 43
- Configuring Event, page 44
- Configuring Event Action, page 45
- Configuring Event Trigger, page 47
- Configuring Existence Trigger Test, page 49
- Configuring Boolean Trigger Test, page 50
- Configuring Threshold Trigger Test, page 52

Configuring Scalar Variables

Perform this task to configure scalar variables for Event MIB.

Prerequisites

To configure the scalar variables for Event MIB, you should be familiar with the Event MIB scalar variables.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **snmp mib event sample minimum** *value*
4. **snmp mib event sample instance maximum** *value*
5. **exit**

DETAILED STEPS

	Command or Action	Purpose

Step 1	**`enable`**
	Example:
	`Router> enable`
		Enables privileged EXEC mode.
		• Enter your password if prompted.

Step 2	**`configure terminal`**
	Example:
	`Router# configure terminal`
		Enters global configuration mode.

Step 3	**`snmp mib event sample minimum`** *`value`*
	Example:
	`Router(config)# snmp mib event sample minimum 10`
		Sets the minimum value for object sampling.

Step 4	**`snmp mib event sample instance maximum`** *`value`*
	Example:
	`Router(config)# snmp mib event sample instance maximum 50`
		Sets the maximum value for object instance sampling.

Step 5	**`exit`**
	Example:
	`Router(config)# exit`
		Exits global configuration mode.

Configuring Event MIB Object List

To configure Event MIB, you need to set up a list of objects that can be added to notifications according to trigger, trigger test, or the event.

Prerequisites

To configure the Event MIB object list, you should be familiar with the Event MIB objects and object identifiers, which can be added to notifications according to event, trigger, or the trigger test.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **snmp mib event object list owner** *object-list-owner* **name** *object-list-name* **number** *object-number*
4. **object id** *object-identifier*
5. **wildcard**
6. **exit**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** Router> enable	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** Router# configure terminal	Enters global configuration mode.
Step 3	**snmp mib event object list owner** *object-list-owner* **name** *object-list-name* **number** *object-number* **Example:** Router(config)# snmp mib event object list owner john name objectA number 10	Configures the Event MIB object list.
Step 4	**object id** *object-identifier* **Example:** Router(config-event-objlist)# object id ifInOctets	Specifies the object identifier for the object configured for the event.
Step 5	**wildcard** **Example:** Router(config-event-objlist)# wildcard	(Optional) Starts a wildcarded search for object identifiers. By specifying a partial object identifier, you can obtain a list of object identifiers.
Step 6	**exit** **Example:** Router(config-event-objlist)# exit	Exits object list configuration mode.

Configuring Event

Perform this task to configure a management event.

Prerequisites

To configure a management event, you should be familiar with the SNMP MIB events and object identifiers.

SUMMARY STEPS

1. **enable**
2. **config terminal**
3. **snmp mib event owner** *event-owner* **name** *event-name*
4. **description** *event-description*

5. **object id** *object-identifier*
6. **enable**
7. **exit**

DETAILED STEPS

	Command or Action	Purpose

Step 1	`enable`
	Example:
	`Router> enable`
		Enables privileged EXEC mode.
		• Enter your password if prompted.

Step 2	`configure terminal`
	Example:
	`Router# configure terminal`
		Enters global configuration mode.

Step 3	**`snmp mib event owner`** *`event-owner`* **`name`** *`event-name`*
	Example:
	`Router(config)# snmp mib event owner john event EventA`
		Enters the event configuration mode.

Step 4	**`description`** *`event-description`*
	Example:
	`Router(config-event)# description eventA is an RMON event.`
		Describes the function and use of the event.

Step 5	**`object id`** *`object-identifier`*
	Example:
	`Router(config-event)# object id ifInOctets`
		Specifies the object identifier of the object.
		Note When the event action information is set to **notification**, the object identifier specifies the notification type to be sent out. If the event action information is configured as **set**, the object identifier identifies the object to be set.

Step 6	`enable`
	Example:
	`Router(config-event)# enable`
		Enables the event.
		Note The event can be executed during an event trigger only if it is enabled.

Step 7	`exit`
	Example:
	`Router(config-event)# exit`
		Exits event configuration mode.

Configuring Event Action

By configuring an event action, you can define the actions that an application can perform during an event trigger. The actions for an event include sending a notification, setting a MIB object and so on. You can set the event action information to either **set** or **notification**. The actions for the event can be configured only in the event configuration mode.

The following sections contain the tasks to configure event action:

- Configuring Action Notification, page 46
- Configuring Action Set, page 46

Configuring Action Notification

Perform this task to set the notification action for the event.

SUMMARY STEPS

1. **action notification**
2. **object** *object-id*
3. **exit**

DETAILED STEPS

	Command or Action	Purpose

Step 1	**`action notification`**
	Example:
	`Router(config-event)# action notification`
		Sets the notification action for an event.
		Note If the event action is set to notification, a notification is generated whenever an object associated with an event is modified.

Step 2	**`object`** *`object-id`*
	Example:
	`Router(config-event-action-notification)# object ifInOctets`
		Configures object for action notification. When the object specified is modified, a notification will be sent to the host system.

Step 3	**`exit`**
	Example:
	`Router(config-event-action-notification)# exit`
		Exits action notification configuration mode.

Configuring Action Set

Perform this task to set actions for an event.

SUMMARY STEPS

1. **action set**
2. **object wildcard**
3. **value** *integer-value*
4. **exit**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**action set** **Example:** `Router(config-event)# action set`	Enters action set configuration mode.
Step 2	**object wildcard** **Example:** `Router(config-event-action-set)# object wildcard`	Enables wildcarded search for the objects based on the object identifiers assigned to each object.
Step 3	**value** *integer-value* **Example:** `Router(config-event-action-set)# value 10`	Sets a value for the object.
Step 4	**exit** **Example:** `Router(config-event-action-set)# exit`	Exits action set configuration mode.

Configuring Event Trigger

By configuring an event trigger, you can list the objects to monitor, and associate each trigger to an event.

Perform this task to configure an event trigger.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **snmp mib event trigger owner** *trigger-owner* **name** *trigger-name*
4. **description** *trigger-description*
5. **frequency** *seconds*
6. **object list owner** *object-list-owner* **name** *object-list-name*
7. **object id** *object-identifier*
8. **wildcard**
9. **sample** [**absolute**] [**delta**] [**changed**]
10. **enable**
11. **exit**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. - Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**snmp mib event trigger owner** *trigger-owner* **name** *trigger-name* **Example:** `Router(config)# snmp mib event trigger owner john name EventTriggerA`	Enables event trigger configuration mode for the specified event trigger.
Step 4	**description** *trigger-description* **Example:** `Router(config-event-trigger)# description EventTriggerA is an RMON alarm.`	Describes the function and use of the event trigger.
Step 5	**frequency** *seconds* **Example:** `Router(config-event-trigger)# frequency 120`	Configures the waiting time (number of seconds) between trigger samples.
Step 6	**object list owner** *object-list-owner* **name** *object-list-name* **Example:** `Router(config-event-trigger)# object list owner john name ObjectListA`	Specifies the list of objects that can be added to notifications.
Step 7	**object id** *object-identifier* **Example:** `Router(config-event-trigger)# object id ifInOctets`	Configures object identifiers for an event trigger.
Step 8	**wildcard** **Example:** `Router(config-event-trigger)# wildcard`	(Optional) Enables wildcarded search for the object.

	Command or Action	Purpose
Step 9	**sample**[**absolute**][**delta**][**changed**] **Example:** `Router(config-event-trigger)# sample absolute`	Enables the specified sampling method for the object. This example uses the absolute sampling method. You can specify any of the three sampling methods; absolute, delta, and changed. • Absolute sampling—Uses the value of the MIB object during sampling. • Delta sampling—Considers the last sampling value maintained in the application. Delta sampling requires the applications to do continuous sampling. • Changed sampling—Uses the changed value of the object since the last sample.
Step 10	**enable** **Example:** `Router(config-event-trigger)# enable`	Enables the event trigger.
Step 11	**exit** **Example:** `Router(config-event-trigger)# exit`	Exits event trigger configuration mode.

Configuring Existence Trigger Test

Perform this task to configure trigger parameters for the test existence trigger type.

You should configure this trigger type in the event trigger configuration mode.

SUMMARY STEPS

1. **test existence**
2. **object list owner** *object-list-owner* **name** *object-list-name*
3. **event owner** *event-owner* **name** *event-name*
4. **type** [**present**] [**absent**] [**changed**]
5. **startup** [**present**] [**absent**]
6. **exit**

DETAILED STEPS

	Command or Action	Purpose

Step 1	**`test existence`**
	Example:
	`Router(config-event-trigger)# test existence`
		Enables test existence configuration mode.

Step 2	**`event owner`** *`event-owner`* **`name`** *`event-name`*
	Example:
	`Router(config-event-trigger-existence)# event owner John name EventA`
		Configures event for existence trigger test.

Step 3	**`object list owner`** *`object-list-owner`* **`name`** *`object-list-name`*
	Example:
	`Router(config-event-trigger-existence)# object list owner John name ObjectListA`
		Configures the list of objects for Existence trigger test.

Step 4	**`type`** `[`**`present`**`][`**`absent`**`][`**`changed`**`]`
	Example:
	`Router(config-event-trigger-existence)# type present`
		Performs the specified type of existence test. This example uses the present test type.
		There are three types of existence tests; present, absent and changed.
		• Present—Setting type to present tests if the objects that appear during the event trigger exist.
		• Absent—Setting type to absent tests if the objects that disappear during the event trigger exist.
		• Changed—Setting type to changed tests if the objects that changed during the event trigger exist.

Step 5	**`startup`** `[`**`present`**`][`**`absent`**`]`
	Example:
	`Router(config-event-trigger-existence)# startup present`
		Triggers an event if the test is performed successfully.

Step 6	**`exit`**
	Example:
	`Router(config-event-trigger-existence)# exit`
		Exits existence trigger test configuration mode.

Configuring Boolean Trigger Test

Perform this task to configure trigger parameters for Boolean trigger type. You should configure this trigger test in the event trigger configuration mode.

SUMMARY STEPS

1. **test boolean**
2. **comparison** [**unequal** |**equal** | **less** | **lessOrEqual** | **greater** | **greaterOrEqual**]

3. **object list owner** *object-list-owner* **name** *object-list-name*
4. **event owner** *event-owner* **name** *event-name*
5. **value** *integer-value*
6. **startup**
7. **exit**

DETAILED STEPS

	Command or Action	Purpose

Step 1	**`test boolean`**
	Example:
	`Router(config-event-trigger)# test boolean`
		Enables Boolean trigger test configuration mode.

Step 2	**`comparison`** `[`**`unequal`**`|`**`equal`**`|`**`less`**`|`**`lessOrEqual`**`|`**`greater`**`|`**`greaterOrEqual`**`]`
	Example:
	`Router(config-event-trigger-boolean)# comparison unequal`
		Performs the specified Boolean comparison test. The value for the Boolean comparison test can be set to unequal, equal, less, lessOrEqual, greater, or greaterOrEqual.

Step 3	**`value`** *`integer-value`*
	Example:
	`Router(config-event-trigger-boolean)# value 10`
		Sets a value for the Boolean trigger test.

Step 4	**`object list owner`** *`object-list-owner`* **`name`** *`object-list-name`*
	Example:
	`Router(config-event-trigger-boolean)# object list owner John name ObjectListA`
		Configures the list of objects for Boolean trigger test.

Step 5	**`event owner`** *`event-owner`* **`name`** *`event-name`*
	Example:
	`Router(config-event-trigger-boolean)# event owner John name EventA`
		Configures event for the Boolean trigger type.

Step 6	**`startup`**
	Example:
	`Router(config-event-trigger-boolean)# startup`
		Triggers an event if the test is performed successfully.

Step 7	**`exit`**
	Example:
	`Router(config-event-trigger-boolean)# exit`
		Exits Boolean trigger test configuration mode.

Configuring Threshold Trigger Test

Perform this task to configure trigger parameters for the threshold trigger test. You should configure this trigger test in the event trigger configuration mode.

SUMMARY STEPS

1. **test threshold**
2. **object list owner** *object-list-owner* **name** *object-list-name*
3. **rising** *integer-value*
4. **rising event owner** *event-owner* **name** *event-name*
5. **falling** *integer-value*
6. **falling event owner** *event-owner* **name** *event-name*
7. **delta rising** *integer-value*
8. **delta rising event owner** *event-owner* **name** *event-name*
9. **delta falling** *integer-value*
10. **delta falling event owner** *event-owner* **name** *event-name*
11. **startup** [**rising|falling|rising-or-falling**]
12. **exit**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**test threshold** **Example:** `Router(config-event-trigger)# test threshold`	Enables threshold trigger test configuration mode.
Step 2	**object list owner** *object-list-owner* **name** *object-list-name* **Example:** `Router(config-event-trigger-threshold)# object list owner John name ObjectListA`	Configures the list of objects for threshold trigger test.
Step 3	**rising** *integer-value* **Example:** `Router(config-event-trigger-threshold)# rising 100`	Sets the rising threshold to the specified value.
Step 4	**rising event owner** *event-owner* **name** *event-name* **Example:** `Router(config-event-trigger-threshold)# rising event owner John name EventA`	Configures event for Threshold trigger test for rising threshold.

	Command or Action	Purpose
Step 5	**falling** *integer-value* **Example:** `Router(config-event-trigger-threshold)# falling 50`	Sets the falling threshold to the specified value.
Step 6	**falling event owner** *event-owner* **name** *event-name* **Example:** `Router(config-event-trigger-threshold)# falling event owner Jane name EventB`	Configures event for Threshold trigger test for falling threshold.
Step 7	**delta rising** *integer-value* **Example:** `Router(config-event-trigger-threshold)# delta rising 30`	Sets the delta rising threshold to the specified value when the sampling method specified for the event trigger is delta.
Step 8	**delta rising event owner** *event-owner* **name** *event-name* **Example:** `Router(config-event-trigger-threshold)# delta rising event owner Jack name EventC`	Configures event for Threshold trigger test for delta rising threshold.
Step 9	**delta falling** *integer-value* **Example:** `Router(config-event-trigger-threshold)# delta falling 10`	Sets the delta falling threshold to the specified value when the sampling method specified for the event trigger is delta.
Step 10	**delta falling event owner** *event-owner* **name** *event-name* **Example:** `Router(config-event-trigger-threshold)# delta falling event owner John name EventAA`	Configures event for Threshold target test for delta falling threshold.
Step 11	**startup** [**rising**|**falling**|**rising-or-falling**] **Example:** `Router(config-event-trigger-threshold)# startup rising`	Triggers an event when the threshold trigger test conditions are met.
Step 12	**exit** **Example:** `Router(config-event-trigger-threshold)# exit`	Exits threshold trigger test configuration mode.

Configuring Expression MIB

Expression MIB can be configured using SNMP directly. However, in the Cisco IOS Release 12.4(20)T, Expression MIB feature is enhanced to add CLIs to configure expressions. You should be familiar with expressions, object identifiers and sampling methods before configuring Expression MIB.

The following sections contain the tasks to configure Expression MIB:

- Configuring Expression MIB Scalar Objects, page 54
- Configuring Expressions, page 55

Configuring Expression MIB Scalar Objects

Expression MIB has the following scalar objects:

- expResourceDeltaMinimum
- expResourceDeltaWildcardInstanceMaximum

Perform this task to configure Expression MIB scalar objects.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **snmp mib expression delta minimum** *seconds*
4. **snmp mib expression delta wildcard maximum** *number-of-instances*
5. **exit**

DETAILED STEPS

Step 1	`enable`
	Example:
	`Router> enable`
		Enables privileged EXEC mode.
		• Enter your password if prompted.

Step 2	`configure terminal`
	Example:
	`Router# configure terminal`
		Enters global configuration mode.

Step 3	**snmp mib expression delta minimum** *seconds* **Example:** `Router(config)# snmp mib expression delta minimum 20`	(Optional) Sets the minimum delta interval in seconds. **Note** Application may use larger values for this minimum delta interval to lower the impact of constantly computing deltas. For larger delta sampling intervals, the application samples less often and has less overhead. By using this command, you can enforce a lower overhead for all expressions created after the delta interval is set.
Step 4	**snmp mib expression delta wildcard maximum** *number-of-instances* **Example:** `Router(config)# snmp mib expression delta maximum 120`	(Optional) Limits the maximum number of dynamic instance entries for wildcarded delta objects in expressions. For a given delta expression, the number of dynamic instances is the number of values that meet all criteria to exist, times the number of delta values in the expression. There is no preset limit for the instance entries and it is dynamic based on a system's resources.

Step 5	`exit`
	Example:
	`Router(config)# exit`
		Exits global configuration mode.

Configuring Expressions

Perform this task to configure an expression.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **snmp mib expression owner** *expression-owner* **name** *expression-name*
4. **description** *expression-description*
5. **expression** *expression*
6. **delta interval** *seconds*
7. **value type** [**counter32** | **unsigned32** | **timeticks** | **integer32** | **ipaddress** | **octetstring** | **objectid** | **counter64**]
8. **enable**
9. **object** *object-number* **id** *object-identifier*

10. **wildcard**
11. **prefix object** *object-id*
12. **discontinuity object** *discontinuity-object-id* [**wildcard**] [**type timeticks | timestamp | date-and-time**]
13. **conditional object** *conditional-object-id*
14. **sample** [**absolute**] [**delta**] [**changed**]
15. **exit**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**snmp mib expression owner** *expression-owner* **name** *expression-name* **Example:** `Router(config-expression)# snmp mib expression owner John name ExpA`	Enables the expression to be configured.
Step 4	**description** *expression-description* **Example:** `Router(config-expression)# description this expression is created for the sysLocation MIB object`	Configures description for expression.
Step 5	**expression** *expression* **Example:** `Router(config-expression)# expression ($1+$2)*800/$3`	Configures the expression to be evaluated. **Note** The expression are in ANSI C syntax. However, the variables in an expression are defined as combination of the dollar sign ($) and an integer that corresponds to the object number of the object used in evaluating the expression.
Step 6	**delta interval** *seconds* **Example:** `Router(config-expression)# delta interval 180`	Configures the sampling interval for objects in the expression if the sampling method is delta.
Step 7	**value type** [**counter32** | **unsigned32** | **timeticks** | **integer32** |**ipaddress** | **octetstring** | **objectid** | **counter64**] **Example:** `Router(config-expression)# value type`	Sets the specified value type for expression.

	Command or Action	Purpose
Step 8	**enable** **Example:** `Router(config-expression)# enable`	Enables expression for evaluation.
Step 9	**object** *object-number* **id** *object-identifier* **Example:** `Router(config-expression)# object 2 id ifInOctets`	Configures the objects that are used for evaluating an expression. The object number is used to associate the object with the variables in the Expression. The variable corresponding to the object is $ and the object number. Thus the variable in the example used here corresponds to $10.
Step 10	**wildcard** **Example:** `Router(config-expression-object)# wildcard`	(Optional) Enables wildcarded search for objects used in evaluating expression.
Step 11	**prefix object** *object-id* **Example:** `Router(config-expression-object)# prefix object 0.2.2`	(Optional) Sets an object prefix. The prefix object assists an application in determining the instance indexing to use while evaluating expression.
Step 12	**discontinuity object** *discontinuity-object-id* [**wildcard**][**type timeticks** |**timestamp** | **date-and-time**] **Example:** `Router(config-expression-object)# discontinuity object sysUpTime`	(Optional) Configures the discontinuity properties for the object if the object sampling type is set to delta or changed. The discontinuity object ID supports normal checking for a discontinuity in a counter. • Using the wildcard keyword, you can enable wildcarded search for the objects with discontinuity properties. • Using the type keyword, you can set value for objects with discontinuity properties.
Step 13	**conditional object** *conditional-object-id* **[wildcard]** **Example:** `Router(config-expression-object)# conditional object mib-2.90.1.3.1.1.2.3.112.99.110.4.101.120.112.53`	(Optional) Configures the conditional object identifier. • Using the wildcard keyword, you can enable wildcarded search for the conditional objects with discontinuity properties.

	Command or Action	Purpose
Step 14	**sample** [**absolute**] [**delta**] [**changed**] **Example:** `Router(config-expression-object)# sample delta`	Enables the specified sampling method for the object. This example uses the delta sampling method. You can set any of the three sampling methods; absolute, delta, and changed. • Absolute sampling—Uses the value of the MIB object during sampling. • Delta sampling—Uses the last sampling value maintained in the application. This method requires the applications to do continuous sampling. • Changed sampling—Uses the changed value of the object since the last sample.
Step 15	**exit** **Example:** `Router(config-expression-object)# exit`	Exits expression object configuration mode.

Configuration Examples for SNMP Support

This section provides the following configuration examples:

- Configuring SNMPv1, SNMPv2c, and SNMPv3: Example, page 58
- Configuring IfAlias Long Name Support: Example, page 59
- Configuring SNMP Support for VPNs: Example, page 60
- Enabling Event MIB Persistence: Example, page 61
- Enabling Expression MIB Persistence: Example, page 61
- Configuring Event MIB: Example, page 61
- Configuring Expression MIB: Example, page 63

Configuring SNMPv1, SNMPv2c, and SNMPv3: Example

The following example shows how to enable SNMPv1, SNMPv2c, and SNMPv3. The configuration permits any SNMP manager to access all objects with read-only permissions using the community string named public. This configuration does not cause the router to send traps.

```
snmp-server community public
```

The following example shows how to permit SNMP access to all objects with read-only permission using the community string named public. The router also will send ISDN traps to the hosts 172.16.1.111 and 172.16.1.33 using SNMPv1 and to the host 172.16.1.27 using SNMPv2c. The community string named public is sent with the traps.

```
snmp-server community public
snmp-server enable traps isdn
snmp-server host 172.16.1.27 version 2c public
snmp-server host 172.16.1.111 version 1 public
snmp-server host 172.16.1.33 public
```

The following example shows how to allow read-only access for all objects to members of access list 4 that specify the comaccess community string. No other SNMP managers have access to any objects. SNMP Authentication Failure traps are sent by SNMPv2c to the host host3.com using the community string named public.

```
snmp-server community comaccess ro 4
snmp-server enable traps snmp authentication
snmp-server host host3.com version 2c public
```

The following example shows how to send Entity MIB inform notifications to the host host3.com. The community string is restricted. The first line enables the router to send Entity MIB notifications in addition to any traps or informs previously enabled. The second line specifies that the notifications should be sent as informs, specifies the destination of these informs, and overwrites previous **snmp-server host** commands for the host host3.com.

```
snmp-server enable traps entity
snmp-server host informs host3.com restricted entity
```

The following example shows how to send the SNMP and Cisco environmental monitor enterprise-specific traps to address 172.30.2.160:

```
snmp-server enable traps
snmp-server host 172.30.2.160 public snmp envmon
```

The following example shows how to enable the router to send all traps to the host myhost.host3.com using the community string public:

```
snmp-server enable traps
snmp-server host myhost.host3.com public
```

The following example shows a configuration in which no traps are sent to a host. The BGP traps are enabled for all hosts, but only the ISDN traps are enabled to be sent to a host.

```
snmp-server enable traps bgp
snmp-server host bob public isdn
```

The following example shows how to enable a router to send all informs to the host myhost.host3.com using the community string named public:

```
snmp-server enable traps
snmp-server host myhost.host3.com informs version 2c public
```

In the following example, the SNMP manager is enabled and the session timeout is set to a value greater than the default:

```
snmp-server manager
snmp-server manager session-timeout 1000
```

Configuring IfAlias Long Name Support: Example

In the following example a long description is applied to the Ethernet interface in slot 1, port adapter 0, and port 0:

```
Router# configure terminal
Router(config)# interface Ethernet1/0/0
Router(config-if)# description ethernet1/0/0 this is a test of a description that exceeds 
64 characters in length
Router(config-if)# ip address 192.168.134.55 255.255.255.0
Router(config-if)# no ip directed-broadcast
Router(config-if)# no ip route-cache distributed
```

Assuming that ifAlias long name support is not yet enabled (the default), the following example shows the results of a mibwalk operation from an NMS:

```
***** SNMP QUERY STARTED *****
 .
 .
 .
 ifXEntry.18.10 (octets) (zero-length)
 ifXEntry.18.11 (octets) ethernet1/0/0 this is a test of a description that exceeds 64 ch
 ifXEntry.18.12 (octets) (zero-length)
 .
 .
 .
```

The following output shows the description that is displayed at the CLI:

```
Router# show interface Ethernet0/0/0

Ethernet1/0/0 is administratively down, line protocol is down
  Hardware is Lance, address is 0010.7b4d.7046 (bia 0010.7b4d.7046)
  Description: ethernet1/0/0 this is a test of a description that exceeds 64 chh
  MTU 1500 bytes, BW 10000 Kbit, DLY 1000 usec,
     reliability 252/255, txload 1/255, rxload 1/255
 .
 .
 .
```

In the following example, ifAlias long name support is enabled and the description is displayed again:

```
Router(config)# snmp ifmib ifalias long
Router(config)# interface Ethernet1/0/0
Router(config-if)# description ethernet1/0/0 this is a test of a description that exceeds
64 characters in length
Router(config)# end
Router# show interface Ethernet1/0/0

Ethernet1/0/0 is administratively down, line protocol is down
  Hardware is Lance, address is 0010.7b4d.7046 (bia 0010.7b4d.7046)
  Description: ethernet1/0/0 this is a test of a description that exceeds 64 characters in
length
  MTU 1500 bytes, BW 10000 Kbit, DLY 1000 usec,
     reliability 252/255, txload 1/255, rxload 1/255
 .
 .
 .
***** SNMP QUERY STARTED *****
 .
 .
 .
 ifXEntry.18.10 (octets) (zero-length)
 ifXEntry.18.11 (octets) ethernet1/0/0 this is a test of a description that exceeds 64
characters in length
 ifXEntry.18.12 (octets) (zero-length)
 .
 .
 .
```

Configuring SNMP Support for VPNs: Example

In the following example all SNMP notifications are sent to xyz.com over the VRF named trap-vrf:

```
Router(config)# snmp-server host xyz.com vrf trap-vrf
```

In the following example the VRF named “traps-vrf” is configured for the remote server 172.16.20.3:

```
Router(config)# snmp-server engineID remote 172.16.20.3 vrf traps-vrf
80000009030000B064EFE100
```

Enabling Event MIB Persistence: Example

The following example shows how to enable Event MIB Persistence using the **snmp mib persist event** command in global configuration mode:

```
Router(config)# snmp mib persist event
Router# write mib-data
```

Enabling Expression MIB Persistence: Example

The following example shows how to enable Expression MIB Persistence using the **snmp mib persist expression** command in global configuration mode:

```
Router(config)# snmp mib persist expression
Router# write mib-data
```

Configuring Event MIB: Example

The following example shows how to configure scalar variables for an event:

```
Router# configure terminal
Router(config)# snmp mib event sample minimum 10
Router(config)# snmp mib event sample instance maximum 50
Router(config)# exit
```

The following example shows how to configure object list for an event:

```
Router# configure terminal
Router(config)# snmp mib event object list owner john name objectA number 1
Router(config-event-objlist)# object id ifInOctets
Router(config-event-objlist)# wildcard
Router(config-event-objlist)# exit
```

The following example shows how to configure an event:

```
Router# configure terminal
Router(config)# snmp mib event owner john event EventA
Router(config-event)# description eventA is an RMON event.
Router(config-event)# object id ifInOctets
Router(config-event)# enable
Router(config-event)# exit
```

The following example shows how to set the notification action for an event:

```
Router#(config-event)# action notification
Router(config-event-action-notification)# object id ifInOctets
Router(config-event-action-notification)# exit
```

The following example shows how to set actions for an event:

```
Router#(config-event)# action set
Router#(config-event-action-set)# object wildcard
Router#(config-event-action-set)# value 10
Router(config-event-action-set)# exit
```

The following example shows how to configure trigger for an event:

```
Router# configure terminal
Router(config)# snmp mib event trigger owner john name EventTriggerA
Router(config-event-trigger)# description EventTriggerA is an RMON alarm.
Router(config-event-trigger)# frequency 120
Router(config-event-trigger)# object list owner john name ObjectListA
Router(config-event-trigger)# object id ifInOctets
Router(config-event-trigger)# wildcard
Router(config-event-trigger)# sample absolute
Router(config-event-trigger)# enable
Router(config-event-trigger)# exit
```

The following example shows how to configure existence trigger test:

```
Router(config-event-trigger)# test existence
Router(config-event-trigger-existence)# event owner John name EventA
Router(config-event-trigger-existence)# object list owner John name ObjectListA
Router(config-event-trigger-existence)# type present
Router(config-event-trigger-existence)# startup present
Router(config-event-trigger-existence)# exit
```

The following example shows how to configure Boolean trigger test:

```
Router(config-event-trigger)# test boolean
Router(config-event-trigger-boolean)# event owner John name EventA
Router(config-event-trigger-boolean)# object list owner John name ObjectListA
Router(config-event-trigger-boolean)# comparison unequal
Router(config-event-trigger-boolean)# value 10
Router(config-event-trigger-boolean)# startup
Router(config-event-trigger-boolean)# exit
```

The following example shows how to configure threshold trigger test:

```
Router(config-event-trigger)# test threshold
Router(config-event-trigger-threshold)# object list owner John name ObjectListA
Router(config-event-trigger-threshold)# rising 100
Router(config-event-trigger-threshold)# rising event owner John name EventA
Router(config-event-trigger-threshold)# falling 50
Router(config-event-trigger-threshold)# falling event owner Jane name EventA
Router(config-event-trigger-threshold)# delta rising 30
Router(config-event-trigger-threshold)# delta rising event owner Jack name EventA
Router(config-event-trigger-threshold)# delta falling 10
Router(config-event-trigger-threshold)# delta falling event owner John name EventA
Router(config-event-trigger-threshold)# startup rising
Router(config-event-trigger-threshold)# exit
```

Configuring Expression MIB: Example

The following example shows how to configure Expression MIB using the **snmp mib expression** command in global configuration mode:

```
Router(config)# snmp mib expression owner pcn name exp6
Router(config-expression)# expression ($1+$2)*800/$3
Router(config-expression)# delta interval 120
Router(config-expression)# enable
Router(config-expression)# object 2 id ifInOctets
Router(config-expression-object)# wildcard
Router(config-expression-object)# sample delta
Router(config-expression-object)# conditional object
mib-2.90.1.3.1.1.2.3.112.99.110.4.101.120.112.53 wildcard
Router(config-expression-object)# object 2 id ifOutOctets
Router(config-expression-object)# wildcard
Router(config-expression-object)# sample delta
Router(config-expression-object)# exit
```

Additional References

The following sections provide references related to configuring SNMP support.

Related Documents

Related Topic	Document Title
SNMP commands: complete command syntax, command mode, command history, defaults, usage guidelines, and examples	*Cisco IOS Network Management Command Reference*
Cisco IOS implementation of RFC 1724, RIP Version 2 MIB Extensions	*RIPv2 Monitoring with SNMP Using the RFC 1724 MIB Extensions* feature module
DSP Operational State Notifications for notifications to be generated when a digital signaling processor (DSP) is used	*DSP Operational State Notifications* feature module

Standards

Standard	Title
CBC-DES (DES-56) standard	*Symmetric Encryption Protocol*
STD: 58	*Structure of Management Information Version 2 (SMIv2)*

MIBs

MIB	MIBs Link
• Circuit Interface Identification MIB • Cisco SNMPv2 • Ethernet-like Interfaces MIB • Event MIB • Expression MIB Support for Delta, Wildcarding, and Aggregation • Interfaces Group MIB (IF-MIB) • Interfaces Group MIB Enhancements • MIB Enhancements for Universal Gateways and Access Servers • MSDP MIB • NTP MIB • Response Time Monitor MIB	To locate and download MIBs for selected platforms, Cisco IOS releases, and feature sets, use Cisco MIB Locator found at the following URL: http://www.cisco.com/go/mibs

RFCs

RFC	Title
RFC 1067	*A Simple Network Management Protocol*
RFC 1091	*Telnet terminal-type option*
RFC 1098	*Simple Network Management Protocol (SNMP)*
RFC 1157	*Simple Network Management Protocol (SNMP)*
RFC 1213	*Management Information Base for Network Management of TCP/IP-based internets:MIB-II*
RFC 1215	*Convention for defining traps for use with the SNMP*
RFC 1901	*Introduction to Community-based SNMPv2*
RFC 1905	*Common Management Information Services and Protocol over TCP/IP (CMOT)*
RFC 1906	*Telnet X Display Location Option*
RFC 1908	*Simple Network Management Protocol (SNMP)*
RFC 2104	*HMAC: Keyed-Hashing for Message Authentication*
RFC 2206	*RSVP Management Information Base using SMIv2*
RFC 2213	*Integrated Services Management Information Base using SMIv2*
RFC 2214	*Integrated Services Management Information Base Guaranteed Service Extensions using SMIv2*
RFC 2271	*An Architecture for Describing SNMP Management Frameworks*
RFC 2570	*Introduction to Version 3 of the Internet-standard Network Management Framework*

RFC	Title
RFC 2578	*Structure of Management Information Version 2 (SMIv2)*
RFC 2579	*Textual Conventions for SMIv2*
RFC 2580	*Conformance Statements for SMIv2*
RFC 2981	*Event MIB*
RFC 2982	*Distributed Management Expression MIB*
RFC 3413	*SNMPv3 Applications*
RFC 3415	*View-based Access Control Model (VACM) for the Simple Network Management Protocol (SNMP)*

Technical Assistance

Description	Link
The Cisco Support website provides extensive online resources, including documentation and tools for troubleshooting and resolving technical issues with Cisco products and technologies. To receive security and technical information about your products, you can subscribe to various services, such as the Product Alert Tool (accessed from Field Notices), the Cisco Technical Services Newsletter, and Really Simple Syndication (RSS) Feeds. Access to most tools on the Cisco Support website requires a Cisco.com user ID and password.	http://www.cisco.com/techsupport

Command References

The following commands are introduced or modified in the feature or features documented in this module. For information about these commands, see the *Cisco IOS Network Management Command Reference* at http://www.cisco.com/en/US/docs/ios/netmgmt/command/reference/nm_book.html. For information about all Cisco IOS commands, use the Command Lookup Tool at http://tools.cisco.com/Support/CLILookup or the *Cisco IOS Master Command List, All Releases*, at http://www.cisco.com/en/US/docs/ios/mcl/allreleasemcl/all_book.html.

- **action notification**
- **action set**
- **comparison**
- **conditional object**
- **debug snmp detail**
- **delta falling event owner**
- **delta falling**
- **delta interval**
- **delta rising event owner**

- **delta rising**
- **description (event)**
- **description (expression)**
- **description (trigger)**
- **discontinuity object**
- **enable (event)**
- **enable (expression)**
- **event owner**
- **expression**
- **falling (threshold trigger test)**
- **falling event owner**
- **frequency (event trigger)**
- **object (expression)**
- **object id (action notification)**
- **object id (action set)**
- **object id (event trigger)**
- **object list (test existence)**
- **object list (test boolean)**
- **object list (test threshold)**
- **object wildcard**
- **rising (threshold trigger test)**
- **rising event owner**
- **sample (event-trigger)**
- **sample (expression)**
- **show snmp stats OID**
- **snmp mib event object list**
- **snmp mib event owner**
- **snmp mib event sample instance maximum**
- **snmp mib event sample minimum**
- **snmp mib event trigger**
- **snmp mib expression delta minimum**
- **snmp mib expression delta wildcard maximum**
- **snmp mib expression owner**
- **startup (test existence)**
- **startup (test boolean)**
- **startup (test threshold)**
- **test boolean**
- **test existence**

- **test threshold**
- **type (event trigger)**
- **value (event)**
- **value (action set)**
- **value type**
- **wildcard (event)**
- **wildcard (expression)**

Feature Information for Configuring SNMP Support

Table 19 lists the features in this module and provides links to specific configuration information. Only features that were introduced or modified in Cisco IOS Release 12.(1) or a later release appear in the table.

For information about a feature in this technology that is not documented here, see the SNMP Features Roadmap.

Not all commands may be available in your Cisco IOS software release. For release information about a specific command, see the command reference documentation.

Use Cisco Feature Navigator to find information about platform support and software image support. Cisco Feature Navigator enables you to determine which Cisco IOS and Catalyst OS software images support a specific software release, feature set, or platform. To access Cisco Feature Navigator, go to http://www.cisco.com/go/cfn. An account on Cisco.com is not required.

Note Table 19 lists only the Cisco IOS software release that introduced support for a given feature in a given Cisco IOS software release train. Unless noted otherwise, subsequent releases of that Cisco IOS software release train also support that feature.

Table 2 Feature Information for Configuring SNMP Support

Feature Name	Releases	Feature Information
Distributed Management Event and Expression MIB Persistence	12.0(5)T 12.0(12)S 12.1(3)T 12.2(4)T 12.2(4)T3	The MIB Persistence features allow the SNMP data of a MIB to be persistent across reloads; that is, MIB information retains the same set object values each time a networking device reboots. MIB Persistence is enabled by using the **snmp mib persist** command, and the MIB data of all MIBs that have had persistence enabled using this command is then written to NVRAM storage by using the **write mib-data** command. Any modified MIB data must be written to NVRAM memory using the **write mib-data** command. The following sections provide information about this module: • "MIB Persistence" section on page 10 • "Configuring MIB Persistence" section on page 39

Table 2 Feature Information for Configuring SNMP Support (continued)

Feature Name	Releases	Feature Information
Interface Index Display and Interface Alias Long Name Support for SNMP	12.2(2)T	The Interface Index Display for SNMP feature introduces new commands and command modifications that allow advanced users of SNMP to view information about the interface registrations directly on the managed agent. You can display MIB information from the agent without using an external NMS. This feature addresses three objects in the Interfaces MIB: *ifIndex*, *ifAlias*, and *ifName*. For complete definitions of these objects, see the IF-MIB.my file available from the Cisco SNMPv2 MIB website at ftp://ftp.cisco.com/pub/mibs/v2/. The following sections provide information about this feature: • "Detailed Interface Registration Information" section on page 8 • "Configuring Interface Index Display and Interface Indexes and Configuration of Long Name Support" section on page 34
SNMP Notification Logging	12.0(22)S 12.2(13)T	The SNMP Notification Logging feature adds Cisco IOS CLI commands to change the size of the notification log, to set the global ageout value for the log, and to display logging summaries at the command line. The following sections provide information about this feature: • "SNMP Notification Logging" section on page 11 • "Configuring SNMP Notifications" section on page 27
SNMP Support for VPNs	12.0(23)S 12.2(2)T 12.2(33)SXH 12.2(33)SB	The SNMP Support for VPNs feature allows SNMP traps and informs to be sent and received using VRF tables. In particular, this feature adds support to Cisco IOS software for sending and receiving SNMP traps and informs specific to individual VPNs. The following sections provide information about this feature: • "SNMP Support for VPNs" section on page 9 • "Configuring SNMP Support for VPNs" section on page 38
Circuit Interface Identification Persistence for SNMP feature	12.1(3)T	This feature can be used to identify individual circuit-based interfaces for SNMP monitoring. The following section provides information about this feature: • "Circuit Interface Identification Persistence" section on page 11
Circuit Interface Identification MIB	Cisco IOS XE Release 2.1	This feature was introduced on Cisco ASR 1000 Series Routers.
Distributed Management Event MIB Conformance to RFC 2981	Cisco IOS XE Release 2.1	This feature was introduced on Cisco ASR 1000 Series Routers.

Table 2 Feature Information for Configuring SNMP Support (continued)

Feature Name	Releases	Feature Information
SNMP (Simple Network Management Protocol)	Cisco IOS XE Release 2.1	This feature was introduced on Cisco ASR 1000 Series Routers.
SNMP Version 3	Cisco IOS XE Release 2.1	This feature was introduced on Cisco ASR 1000 Series Routers.
SNMPv2C	Cisco IOS XE Release 2.1	This feature was introduced on Cisco ASR 1000 Series Routers.
SNMP Diagnostics	12.4(20)T	The SNMP Diagnostics feature adds Cisco IOS CLI commands to display the object identifiers that are recently requested by the network management system, and to display the SNMP debug messages. The following section provides the list of commands added to this feature: • "Command References" section on page 65
Event MIB and Expression MIB CLIs	12.4(20)T	The CLIs to configure Event MIB and Expression MIB are introduced on Cisco ASR 1000 Series routers. The following section provides information about configuring Event MIB: • "Configuring Event MIB" section on page 42 The following section provides information about configuring Expression MIB: • "Configuring Expression MIB" section on page 54 The following section the list commands used for configuring Event MIB and Expression MIB: • "Command References" section on page 65

Glossary

ifAlias—SNMP Interface Alias. The ifAlias is an object in the Interfaces MIB (IF-MIB). The ifAlias is an alias name for the interface as specified by a network manager that provides a nonvolatile description for the interface. For a complete definition, see the IF-MIB.my file.

ifIndex—SNMP Interface Index. The ifIndex is an object in the Interfaces MIB (IF-MIB). The ifIndex is a unique integer assigned to every interface (including subinterfaces) on the managed system when the interface registers with the IF-MIB. For a complete definition, see the IF-MIB.my file.

OID—MIB object identifier. An object identifier is expressed as a series of integers or text strings. Technically, the numeric form is the *object name* and the text form is the *object descriptor*. In practice, both are called object identifiers, or OIDs. For example, the object name for the interfaces MIB is 1.3.6.1.2.1.2, and the object descriptor is 'iso.internet.mgmt.mib-2.interfaces' but either can be referred to as the OID. An OID can also be expressed as a combination of the two, such as iso.internet.2.1.2.

Note See *Internetworking Terms and Acronyms* for terms not included in this glossary.

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

Cisco Discovery Protocol (CDP)

Using Cisco Discovery Protocol

First Published: February 1, 1995
Last Updated: May 2, 2008

Cisco Discovery Protocol is a Layer 2 media independent and network independent protocol that runs on Cisco devices and enables networking applications to “learn” about nearby, directly connected devices. This protocol facilitates the management of Cisco devices by discovering these devices, determining how they are configured, and allowing systems using different network-layer protocols to learn about one another.

This module describes Cisco Discovery Protocol and how it functions with Simple Network Management Protocol (SNMP).

Finding Feature Information in This Module

Your Cisco IOS software release may not support all of the features documented in this module. To reach links to specific feature documentation in this module and to see a list of the releases in which each feature is supported, use the “Feature Information for Using Cisco Discovery Protocol” section on page 16.

Finding Support Information for Platforms and Cisco IOS and Catalyst OS Software Images

Use Cisco Feature Navigator to find information about platform support and Cisco IOS and Catalyst OS software image support. To access Cisco Feature Navigator, go to http://www.cisco.com/go/cfn. An account on Cisco.com is not required.

Contents

- Prerequisites for Using Cisco Discovery Protocol, page 2
- Restrictions for Using Cisco Discovery Protocol, page 2
- Information About Using Cisco Discovery Protocol, page 2
- How to Use Cisco Discovery Protocol, page 5
- Configuration Examples for Using Cisco Discovery Protocol, page 14
- Additional References, page 15
- Feature Information for Using Cisco Discovery Protocol, page 16

Americas Headquarters:
Cisco Systems, Inc., 170 West Tasman Drive, San Jose, CA 95134-1706 USA

Prerequisites for Using Cisco Discovery Protocol

- Interfaces must support Subnetwork Access Protocol (SNAP) headers

Restrictions for Using Cisco Discovery Protocol

- Cisco Discovery Protocol functions only on Cisco devices.
- Cisco Discovery Protocol is not supported on Frame Relay multipoint subinterfaces.

Information About Using Cisco Discovery Protocol

To use Cisco Discovery Protocol, you should understand the following concepts:

- VLAN Trunking Protocol, page 2
- Type-Length-Value Fields, page 2
- Cisco Discovery Protocol, page 3
- Benefits of Cisco Discovery Protocol, page 4

VLAN Trunking Protocol

VLAN Trunking Protocol (VTP) is a discovery technique that is used by switches. A switch advertises its management domain on its trunk ports, its configuration revision number, and its known VLANs and their specific parameters. A VTP domain is made up of a single device or interconnected devices that share the same VTP domain name. A switch can be in only one VTP domain.

Type-Length-Value Fields

Type-Length-Value (TLV) fields are blocks of information embedded in Cisco Discovery Protocol advertisements. Information in advertisements varies, and the TLV frame format allows for extending advertisements when needed. Table 1 summarizes the TLV definitions.

Table 1 Type-Length-Value Definitions for Cisco Discovery Protocol v2

TLV	Definition
Device-ID TLV	Identifies the device name in the form of a character string.
Address TLV	Contains the network addresses of both receiving and sending devices.
Port-ID TLV	Identifies the port on which a Cisco Discovery Protocol packet is sent.
Capabilities TLV	Identifies the device type, which indicates the functional capability of the device; for example, a switch.
Version TLV	Contains information about the software release that is running on the device.

Table 1 Type-Length-Value Definitions for Cisco Discovery Protocol v2 (continued)

TLV	Definition
Platform TLV	Identifies the hardware platform of the device; for example, Cisco 4500.
IP Network Prefix TLV	Contains a list of network prefixes to which the sending device can forward IP packets. A prefix includes the interface protocol and port number; for example, Eth 1/0.
VTP Management Domain TLV	Advertises the system's configured VTP management domain name. Used by network operators to verify VTP domain configuration in adjacent network nodes.
Native VLAN TLV	Indicates, per interface, the assumed VLAN for untagged packets on the interface. Cisco Discovery Protocol learns the native VLAN for an interface. This field is implemented only for interfaces that support the IEEE 802.1Q protocol.
Full/Half Duplex TLV	Indicates the duplex configuration of the Cisco Discovery Protocol broadcast interface. Used by network operators to diagnose connectivity problems between adjacent network devices.

Cisco Discovery Protocol

Cisco Discovery Protocol is a Layer 2 media independent and network independent protocol that networking applications use to learn about nearby, directly connected devices. It is enabled by default. Each device configured for Cisco Discovery Protocol advertises at least one address at which it can receive messages and sends periodic advertisements (messages) to the well-known multicast address 01:00:0C:CC:CC:CC. Devices discover each other by listening at that address. They also listen to messages to learn when interfaces on the other devices are up or go down.

Advertisements contain time-to-live information, which indicates the length of time a receiving device should hold Cisco Discovery Protocol information before discarding it. Advertisements supported and configured in the Cisco IOS software are sent by default every 60 seconds on interfaces that support SNAP headers. Cisco devices never forward Cisco Discovery Protocol packets. Each Cisco device that supports Cisco Discovery Protocol stores the information it receives in a table. Information in the table is refreshed each time an advertisement is received, and information about a device is discarded after three advertisements from that device are missed.

The information contained in Cisco Discovery Protocol advertisements varies by the type of device and the version of the operating system running on it. Following is some of the information that Cisco Discovery Protocol can learn:

- Cisco IOS version running on a Cisco device
- Hardware platform of the device
- IP addresses of the interfaces on devices
- Locally connected devices advertising Cisco Discovery Protocol
- Interfaces active on a Cisco device, including encapsulation type
- Hostname
- Duplex setting

- VTP domain
- Native VLAN

Cisco Discovery Protocol Version 2 provides more intelligent device tracking features than those available in version 1. One feature is an enhanced reporting mechanism for more rapid error tracking, which helps to reduce network downtime. Errors reported include mismatched native VLAN IDs (IEEE 802.1Q) on connected ports and mismatched port duplex states between connected devices. Messages about reported errors can be sent to the console or to a logging server.

Cisco IOS Cisco Discovery Protocol **show** commands can provide detailed output on VTP management domains and duplex modes of neighboring devices, counters related to Cisco Discovery Protocol, and VLAN IDs of connecting ports.

Using Cisco Discovery Protocol with SNMP

Using Cisco Discovery Protocol with SNMP allows network management applications to learn the device type and the SNMP agent address of neighboring devices and to send SNMP queries to those devices.

The SNMP management application learns protocol addresses and types of neighboring devices by retrieving the Cisco Discovery Protocol tables from SNMP agents in those devices. When enabled, the network management module (NMM) SNMP agent discovers neighboring devices and builds its local cache with information about these devices. A management workstation can retrieve this cache by sending SNMP requests to access the CISCO-CDP-MIB.

Cisco Discovery Protocol and On-Demand Routing Support for ATM PVCs

In Cisco IOS Release 12.2(8)T Cisco Discovery Protocol and On-Demand Routing (ODR) support was added for ATM point-to-point permanent virtual circuits (PVCs). ODR uses Cisco Discovery Protocol to propagate IP address information in hub-and-spoke topologies. When ODR is enabled, spoke routers automatically advertise their subnets using Cisco Discovery Protocol.

Cisco Discovery Protocol is disabled by default on ATM PVC interfaces. To enable Cisco Discovery Protocol, use the **cdp run** command in global configuration mode and the **cdp enable** command in interface configuration mode command on both ends of the PVC. To enable ODR, use the **router odr** command in global configuration mode on the hub router and turn off all dynamic routing protocols in the spoke routers. For details on configuring ODR, see to the “Configuring On-Demand Routing” section of the *Cisco IOS IP Routing Protocols Configuration Guide*.

Cisco Discovery Protocol Support in IPv6

Cisco Discovery Protocol in IPv6 functions the same as and offers the same benefits as in IPv4. IPv6 enhancements allow Cisco Discovery Protocol to exchange IPv6 and neighbor addressing information, provide IPv6 information to network management products, and provide troubleshooting tools.

Benefits of Cisco Discovery Protocol

Cisco Discovery Protocol provides the following benefits:

- Allows systems using different network-layer protocols to learn about one another
- Facilitates managing Cisco devices by discovering them and how they are configured
- Assists with troubleshooting TLV fields

- Functions with SNMP to learn SNMP agent addresses and to send SNMP queries

How to Use Cisco Discovery Protocol

To configure Cisco Discovery Protocol, perform the following optional tasks:

- Disabling and Enabling Cisco Discovery Protocol on a Supported Device, page 5
- Disabling and Enabling Cisco Discovery Protocol on a Supported Interface, page 7
- Setting the Transmission Timer and Hold Time, page 9
- Disabling and Reenabling Cisco Discovery Protocol Version-2 Advertisements, page 10
- Monitoring and Maintaining Cisco Discovery Protocol, page 12

Disabling and Enabling Cisco Discovery Protocol on a Supported Device

Perform the following tasks to disable and enable Cisco Discovery Protocol on a supported device:

- Disabling Cisco Discovery Protocol on a Supported Device, page 5
- Enabling Cisco Discovery Protocol on a Supported Device, page 6

Disabling Cisco Discovery Protocol on a Supported Device

Perform this task to disable Cisco Discovery Protocol a on Cisco device.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **no cdp run**
4. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. - Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**no cdp run** **Example:** `Router(config)# no cdp run`	Disables Cisco Discovery Protocol on a supported device.

Step 4	**end**
	Example:
	`Router(config)# end`
		Returns the command-line interface (CLI) to privileged EXEC mode.

Enabling Cisco Discovery Protocol on a Supported Device

Perform this task to enable Cisco Discovery Protocol a on Cisco device.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **cdp run**
4. **end**

DETAILED STEPS

	Command or Action	Purpose

Step 1	**`enable`**
	Example:
	`Router> enable`
		Enables privileged EXEC mode.
		- Enter your password if prompted.

Step 2	**`configure terminal`**
	Example:
	`Router# configure terminal`
		Enters global configuration mode.

Step 3	**`cdp run`**
	Example:
	`Router(config)# cdp run`
		Enables Cisco Discovery Protocol on a supported device.

Step 4	**`end`**
	Example:
	`Router(config)# end`
		Returns the CLI to privileged EXEC mode.

Disabling and Enabling Cisco Discovery Protocol on a Supported Interface

Cisco Discovery Protocol is enabled by default on supported interfaces on a Cisco device. To disable or reenable Cisco Discovery Protocol on an interface, perform these tasks:

- Disabling Cisco Discovery Protocol on a Supported Interface, page 7
- Enabling Cisco Discovery Protocol on a Supported Interface, page 8

Disabling Cisco Discovery Protocol on a Supported Interface

Perform this task to disable Cisco Discovery Protocol on a supported interface.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **interface** *type number* [*name-tag*]
4. **no cdp enable**
5. **end**

DETAILED STEPS

Step 1	**`enable`**
	Example:
	`Router> enable`
		Enables privileged EXEC mode.
		- Enter your password if prompted.

Step 2	**`configure terminal`**
	Example:
	`Router# configure terminal`
		Enters global configuration mode.

Step 3	**`interface`** *`type number`* `[`*`name-tag`*`]`
	Example:
	`Router(config)# interface ethernet 0/1`
		Configures Ethernet interface 0/1 and places the CLI in interface configuration mode.

Step 4	**`no cdp enable`**
	Example:
	`Router(config-if)# no cdp enable`
		Disables Cisco Discovery Protocol on Ethernet interface 0/1.

Step 5	**`end`**
	Example:
	`Router(config-if)# end`
		Returns the CLI to privileged EXEC mode.

Enabling Cisco Discovery Protocol on a Supported Interface

Perform this task to enable Cisco Discovery Protocol on a supported interface.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **interface** *type number* [*name-tag*]
4. **cdp enable**
5. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**interface** *type number* [*name-tag*] **Example:** `Router(config)# interface ethernet 0/1`	Configures Ethernet interface 0/1 and places the CLI in interface configuration mode.
Step 4	**cdp enable** **Example:** `Router(config-if)# cdp enable`	Enables Cisco Discovery Protocol on Ethernet interface 0/1.

Step 5	**end**
	Example:
	`Router(config-if)# end`
		Returns the CLI to privileged EXEC mode.

Setting the Transmission Timer and Hold Time

Perform this task to set the frequency of Cisco Discovery Protocol transmissions and the hold time for Cisco Discovery Protocol packets.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **cdp timer** *seconds*
4. **cdp holdtime** *seconds*
5. **end**

DETAILED STEPS

	Command	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**cdp timer** *seconds* **Example:** `Router(config)# cdp timer 30`	Specifies 30 seconds as the frequency of transmission of Cisco Discovery Protocol packets.
Step 4	**cdp holdtime** *seconds* **Example:** `Router(config)# cdp holdtime 90`	Specifies 90 seconds as the amount of time a receiving device should hold information before discarding it.

Step 5	**end**
	Example:
	`Router(config)# end`
		Returns the CLI to privileged EXEC mode.

Disabling and Reenabling Cisco Discovery Protocol Version-2 Advertisements

The broadcasting of Cisco Discovery Protocol Version 2 advertisements is enabled by default on Cisco devices. To disable or reenable this broadcasting, perform these tasks.

- Disabling Cisco Discovery Protocol Version-2 Advertisements, page 10
- Enabling Cisco Discovery Protocol Version-2 Advertisements, page 11

Disabling Cisco Discovery Protocol Version-2 Advertisements

Perform this task to disable Cisco Discovery Protocol version-2 advertisements.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **no cdp advertise-v2**
4. **end**

DETAILED STEPS

	Command or Action	Purpose

Step 1	**`enable`**
	Example:
	`Router> enable`
		Enables privileged EXEC mode.
		• Enter your password if prompted.

Step 2	**`configure terminal`**
	Example:
	`Router# configure terminal`
		Enters global configuration mode.

Step 3	**`no cdp advertise-v2`**
	Example:
	`Router(config)# no cdp advertise-v2`
		Disables the broadcasting of Cisco Discovery Protocol version 2 advertisements.

Step 4	**`end`**
	Example:
	`Router(config)# end`
		Returns the CLI to privileged EXEC mode.

Enabling Cisco Discovery Protocol Version-2 Advertisements

Perform this task to enable Cisco Discovery Protocol version-2 advertisements.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **cdp advertise-v2**
4. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**cdp advertise-v2** **Example:** `Router(config)# cdp advertise-v2`	Enables the broadcasting of Cisco Discovery Protocol version 2 advertisements.

Step 4	**end**
	Example:
	`Router(config)# end`
		Returns the CLI to privileged EXEC mode.

Monitoring and Maintaining Cisco Discovery Protocol

To monitor and maintain Cisco Discovery Protocol on your device, perform the following steps. This task and all steps are optional, and you do not have to perform the steps in the sequence shown.

SUMMARY STEPS

1. **enable**
2. **clear cdp counters**
3. **clear cdp table**
4. **show cdp**
5. **show cdp entry** *device-name* [**protocol** | **version**]
6. **show cdp interface** [*type number*]
7. **show cdp neighbors** [*type number*] [**detail**]
8. **show cdp traffic**
9. **show debugging**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. - Enter your password if prompted.
Step 2	**clear cdp counters** **Example:** `Router# clear cdp counters`	Resets the traffic counters to zero.
Step 3	**clear cdp table** **Example:** `Router# clear cdp table`	Deletes the Cisco Discovery Protocol table of information about neighbors.
Step 4	**show cdp** **Example:** `Router# show cdp`	Displays the interval between advertisements, the number of seconds the advertisement is valid for a given port, and the version of the advertisement.
Step 5	**show cdp entry** *device-name* [**protocol** | **version**] **Example:** `Router# show cdp entry device-name protocol`	Displays information about a specific neighbor. Display can be limited to protocol or version information.
Step 6	**show cdp interface** [*type number*] **Example:** `Router# show cdp interface`	Displays information about interfaces on which Cisco Discovery Protocol is enabled.
Step 7	**show cdp neighbors** [*type number*] [**detail**] **Example:** `Router# show cdp neighbors`	Displays the type of device that has been discovered, the name of the device, the number and type of the local interface (port), the number of seconds the Cisco Discovery Protocol advertisement is valid for the interface, the device type, the device product number, and the port ID. Issuing the **detail** keyword displays information about the native VLAN ID, the duplex mode, and the VTP domain name associated with neighbor devices.
Step 8	**show cdp traffic** **Example:** `Router# show cdp traffic`	Displays Cisco Discovery Protocol counters, including the number of packets sent and received and checksum errors.
Step 9	**show debugging** **Example:** `Router# show debugging`	Displays information about the types of debugging that are enabled for the router.

Configuration Examples for Using Cisco Discovery Protocol

The following examples show how to set transmission timer and hold time values and show commands you can issue to monitor and maintain Cisco Discovery Protocol.

- Setting the Transmission Timer and Hold Time: Example, page 14
- Monitoring and Maintaining Cisco Discovery Protocol: Example, page 14

Setting the Transmission Timer and Hold Time: Example

In the following example, the timer is set to send updates every 30 seconds and a **show cdp interface** command is issued to show that the update is effective:

```
cdp timer 30
end
show cdp interface
Serial0 is up, line protocol is up
Encapsulation is HDLC
Sending CDP packets every 30 seconds
Holdtime is 180 seconds
```

In the following example, the hold time is set to 90 seconds and a **show cdp interface** command is issued to show that the update is effective:

```
cdp holdtime 90
end
show cdp interface
Serial0 is up, line protocol is up
Encapsulation is HDLC
Sending CDP packets every 30 seconds
Holdtime is 90 seconds
```

Monitoring and Maintaining Cisco Discovery Protocol: Example

The following example shows a typical series of commands you can issue to view Cisco Discovery Protocol information:

```
Router# show cdp

Global CDP information:
        Sending CDP packets every 60 seconds
        Sending a holdtime value of 180 seconds
        Sending CDPv2 advertisements is enabled

Router# show cdp neighbors

Capability Codes: R - Router, T - Trans Bridge, B - Source Route Bridge
                  S - Switch, H - Host, I - IGMP, r - Repeater

Device ID        Local Intrfce     Holdtme    Capability  Platform  Port ID
C2950-1          Fas 0/0            148         S I       WS-C2950T-Fas 0/15
RX-SWV.cisco.com Fas 0/1            167         T S       WS-C3524-XFas 0/13

Router# show cdp neighbors detail

-------------------------
Device ID: C2950-1
Entry address(es):
```

```
Platform: Cisco WS-C2950T-24,  Capabilities: Switch IGMP
Interface: FastEthernet0/0,  Port ID (outgoing port): FastEthernet0/15
Holdtime : 139 sec

Version :
Cisco IOS C2950 Software (C2950-I6Q4L2-M), Version 12.1(9)EA1, RELEASE SOFTWARE
 .
 .
 .

Router# show cdp traffic

CDP counters :
        Total packets output: 81684, Input: 81790
        Hdr syntax: 0, Chksum error: 0, Encaps failed: 0
        No memory: 0, Invalid packet: 0, Fragmented: 0
        CDP version 1 advertisements output: 0, Input: 0
        CDP version 2 advertisements output: 81684, Input: 81790
```

Additional References

The following sections provide references related to Using Cisco Discovery Protocol.

Related Documents

Related Topic	Document Title
Cisco IOS Cisco Discovery Protocol commands	*Cisco IOS Network Management Command Reference*
Configuring SNMP Support	"Configuring SNMP Support" module
Debugging commands	*Cisco IOS Debug Command Reference*
Configuring On-Demand Routing	"Configuring On-Demand Routing" module

Standards

Standard	Title
IEEE 802.1Q	*Virtual LANS*

MIBs

MIB	MIBs Link
• CISCO-CDP MIB	To locate and download MIBs for selected platforms, Cisco IOS releases, and feature sets, use Cisco MIB Locator found at the following URL: http://www.cisco.com/go/mibs

RFCs

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

Technical Assistance

Description	Link
The Cisco Support website provides extensive online resources, including documentation and tools for troubleshooting and resolving technical issues with Cisco products and technologies. To receive security and technical information about your products, you can subscribe to various services, such as the Product Alert Tool (accessed from Field Notices), the Cisco Technical Services Newsletter, and Really Simple Syndication (RSS) Feeds. Access to most tools on the Cisco Support website requires a Cisco.com user ID and password.	http://www.cisco.com/techsupport

Feature Information for Using Cisco Discovery Protocol

Table 2 lists the features in this module and provides links to specific configuration information. Only features that were introduced or modified in Cisco IOS Release 10.3 or a later release appear in the table.

Not all commands may be available in your Cisco IOS software release. For release information about a specific command, see the command reference documentation.

Use Cisco Feature Navigator to find information about platform support and software image support. Cisco Feature Navigator enables you to determine which Cisco IOS and Catalyst OS software images support a specific software release, feature set, or platform. To access Cisco Feature Navigator, go to http://www.cisco.com/go/cfn. An account on Cisco.com is not required.

Note Table 2 lists only the Cisco IOS software release that introduced support for a given feature in a given Cisco IOS software release train. Unless noted otherwise, subsequent releases of that Cisco IOS software release train also support that feature.

Table 2 Feature Information for Configuring Cisco Discovery Protocol

Feature Name	Releases	Feature Information
Configuring Cisco Discovery Protocol	10.3, 11.2(1) 12.2(8)T Cisco IOS XE Release 2.1	Cisco Discovery Protocol is a Layer 2 media independent and network independent protocol that runs on Cisco devices and enables networking applications to "learn" about nearby, directly connected devices. This protocol facilitates the management of Cisco devices and allows systems using different network-layer protocols to learn about one another. In Cisco IOS XE Release 2.1, this feature was introduced on Cisco ASR 1000 Series Routers. The following sections provide information about this feature: • VLAN Trunking Protocol, page 2 • Type-Length-Value Fields, page 2 • Cisco Discovery Protocol, page 3 • Using Cisco Discovery Protocol with SNMP, page 4 • Cisco Discovery Protocol and On-Demand Routing Support for ATM PVCs, page 4 • Cisco Discovery Protocol Support in IPv6, page 4 • Benefits of Cisco Discovery Protocol, page 4 • How to Use Cisco Discovery Protocol, page 5

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

RMON Support

Configuring RMON Support

First Published: July 27, 1999
Last Updated: May 2, 2008

Feature History

Release	Modification
12.0(5)T	This feature was introduced.
12.4(6)T	This feature was integrated into Cisco IOS Release 12.4(6)T
Cisco IOS XE Release 2.1	This feature was introduced on Cisco ASR 1000 Series Routers.

This chapter describes the Remote Monitoring (RMON) MIB agent specification, and how it can be used in conjunction with Simple Network Management Protocol (SNMP) to monitor traffic using alarms and events.

For a complete description of the RMON commands mentioned in this chapter, refer to the "RMON Commands" chapter in the "System Management" part of the Release 12.2 *Cisco IOS Configuration Fundamentals Command Reference*. To locate documentation of other commands that appear in this chapter, use the command reference master index or search online.

Configuring RMON Support

The RMON option identifies activity on individual nodes and allows you to monitor all nodes and their interaction on a LAN segment. Used in conjunction with the SNMP agent in a router, RMON allows you to view both traffic that flows through the router and segment traffic not necessarily destined for the router. Combining RMON alarms and events (classes of messages that indicate traffic violations and various unusual occurrences over a network) with existing MIBs allows you to choose where proactive monitoring will occur.

Americas Headquarters:
Cisco Systems, Inc., 170 West Tasman Drive, San Jose, CA 95134-1706 USA

Note Full RMON packet analysis (as described in RFC 1757) is supported only on an Ethernet interface of Cisco 2500 series routers and Cisco AS5200 series universal access servers. RMON requires that SNMP be configured (you must be running a version of SNMP on the server that contains the RMON MIB). A generic RMON console application is recommended in order to take advantage of the RMON network management capabilities. This feature supports RFCs 1757 and 2021.

RMON can be very data- and processor-intensive. Users should measure usage effects to ensure that router performance is not degraded by RMON and to minimize excessive management traffic overhead. Native mode in RMON is less intensive than promiscuous mode.

All Cisco IOS software images ordered without the explicit RMON option include limited RMON support (RMON alarms and event groups only). Images ordered with the RMON option include support for all nine management groups (statistics, history, alarms, hosts, hostTopN, matrix, filter, capture, and event). As a security precaution, support for the capture group allows capture of packet header information only; data payloads are not captured.

In Cisco IOS 12.1, the RMON agent was rewritten to improve performance and add some new features. Table 1 highlights some of the improvements implemented.

Table 1 RMON MIB Updates

Prior to the RMON MIB Update in Cisco IOS Release 12.1	New functionality in Cisco IOS Release 12.1
RMON configurations do not persist across reboots. Information is lost after a new session on the RMON server.	RMON configurations persist across reboots. Information is preserved after a new session on the RMON server.
Packet analysis applies only on the MAC header of the packet.	Complete packet capture is performed with analysis applied to all frames in packet.
Only RMON I MIB objects are used for network monitoring.	RMON I and selected RMON II objects are used for network monitoring.

RMON MIB features include the following:

- usrHistory group. This MIB group is similar to the RMON etherHistory group except that the group enables the user to specify the MIB objects that are collected at each interval.
- partial probeConfig group. This MIB group is a subset of the probeConfig group implemented in read-only mode. These objects implement the simple scalars from this group. Table 2 details new partial probeConfig group objects.

Table 2 partial probeConfig Group Objects

Object	Description
probeCapabilities	The RMON software groups implemented.
probeSoftwareRev	The current version of Cisco IOS running on the device.
probeHardwareRev	The current version of the Cisco device.
probeDateTime	The current date and time.
probeResetControl	Initiates a reset.
probeDownloadFile	The source of the image running on the device.

Table 2 partial probeConfig Group Objects (continued)

Object	Description
probeDownloadTFTPServer	The address of the server that contains the Trivial File Transfer Protocol (TFTP) file that is used by the device to download new versions of Cisco IOS software.
probeDownloadAction	Specifies the action of the commands that cause the device to reboot.
probeDownloadStatus	The state of a reboot.
netDefaultGateway	The router mapped to the device as the default gateway.
hcRMONCapabilities	Specifies the features mapped to this version of RMON.

Configuring RMON Alarm and Event Notifications

To enable RMON on an Ethernet interface, use the following command in interface configuration mode:

Command	Purpose
`Router(config-if)# rmon {native | promiscuous}`	Enables RMON.

In native mode, RMON monitors only the packets normally received by the interface. In promiscuous mode, RMON monitors all packets on the LAN segment.

The default size of the queue that holds packets for analysis by the RMON process is 64 packets. To change the size of the queue, use the following command in global configuration mode:

Command	Purpose
`Router(config)# rmon queuesize size`	Changes the size of the RMON queue.

To set an RMON alarm or event, us the following commands in global configuration mode, as needed:

Command	Purpose
`Router(config)# rmon alarm number variable interval {delta | absolute} rising-threshold value [event-number] falling-threshold value [event-number] [owner string]`	Sets an alarm on a MIB object.
`Router(config)# rmon event number [log] [trap community] [description string] [owner string]`	Adds or removes an event in the RMON event table.

You can set an alarm on any MIB object in the access server. To disable an alarm, you must enable the **no** form of this command on each alarm you configure. You cannot disable all the alarms you configure at once. Refer to RFC 1757 to learn more about alarms and events and how they interact with each other.

The RMON MIB defines two traps, the risingAlarm and fallingAlarm traps generated when an RMON alarmEntry risingThreshold or fallingThreshold event occurs. Thresholds allow you to minimize the number of notifications sent on the network. Alarms are triggered when a problem exceeds a set rising threshold value. No more alarm notifications are sent until the agent recovers, as defined by the falling threshold value. This means that notifications are not sent each time a minor failure or recovery occurs.

Configuring RMON Groups

RMON tables can be created for buffer capture, filter, hosts, and matrix information. The buffer capture table details a list of packets captured off a channel or a logical data or events stream. The filter table details a list of packet filter entries that screen packets for specified conditions as they travel between interfaces. The hosts table details a list of host entries. The matrix table details a list of traffic matrix entries indexed by source and destination MAC addresses.

To gather RMON statistics for these data types, use the following commands in interface configuration mode, as needed:

Command	Purpose
Router(config-if)# **rmon collection history** {**controlEntry** *integer*} [**owner** *ownername*] [**buckets** *bucket-number*] [**interval** *seconds*]	Enables an RMON history group of statistics on an interface.
Router(config-if)# **rmon collection host** {**controlEntry** *integer*} [**owner** *ownername*]	Enables an RMON host collection group of statistics on an interface.
Router(config-if)# **rmon collection matrix** {**controlEntry** *integer*} [**owner** *ownername*]	Enables an RMON matrix group of statistics on an interface.
Router(config-if)# **rmon collection rmon1** {**controlEntry** *integer*} [**owner** *ownername*]	Enables all possible autoconfigurable RMON statistic collections on an interface.

To specifically monitor these commands, use the **show rmon capture**, **show rmon filter**, **show rmon hosts**, and **show rmon matrix** EXEC commands listed in the following table.

Monitoring and Verifying RMON Configuration

To display the current RMON status, use one or more of the following commands in EXEC mode:

Command	Function
Router> **show rmon** or Router> **show rmon task**	Displays general RMON statistics.
Router> **show rmon alarms**	Displays the RMON alarm table.
Router> **show rmon capture**	Displays the RMON buffer capture table and current configuration. Available only on Cisco 2500 series routers and Cisco AS5200 access servers.
Router> **show rmon events**	Displays the RMON event table.

Command	Function
`Router> show rmon filter`	Displays the RMON filter table. Available only on Cisco 2500 series routers and Cisco AS5200 access servers.
`Router> show rmon history`	Displays the RMON history table. Available only on Cisco 2500 series routers and Cisco AS5200 access servers.
`Router> show rmon hosts`	Displays the RMON hosts table. Available only on Cisco 2500 series routers and Cisco AS5200 access serverss.
`Router> show rmon matrix`	Display the RMON matrix table and values associated with RMON variables. Available only on Cisco 2500 series routers and Cisco AS5200 access servers.
`Router> show rmon statistics`	Display the RMON statistics table. Available only on Cisco 2500 series routers and Cisco AS5200 access servers.
`Router> show rmon topn`	Display the RMON top-n hosts table. Available only on Cisco 2500 series routers and Cisco AS5200 access servers.

RMON Configuration Examples

This section provides the following examples:

- Alarm and Event Example
- show rmon Command Example

Alarm and Event Example

The following example enables the **rmon event** global configuration command:

```
Router(config)# rmon event 1 log trap eventtrap description "High ifOutErrors" owner
owner_a
```

This example creates RMON event number 1, which is defined as "High ifOutErrors", and generates a log entry when the event is triggered by an alarm. The user "owner_a" owns the row that is created in the event table by this command. This example also generates an SNMP trap when the event is triggered.

The following example configures an RMON alarm using the **rmon alarm** global configuration command:

```
Router(config)# rmon alarm 10 ifEntry.20.1 20 delta rising-threshold 15 1
falling-threshold 0 owner owner_a
```

This example configures RMON alarm number 10. The alarm monitors the MIB variable ifEntry.20.1 once every 20 seconds until the alarm is disabled, and checks the change in the rise or fall of the variable. If the ifEntry.20.1 value shows a MIB counter increase of 15 or more, such as from 100000 to 100015, the alarm is triggered. The alarm in turn triggers event number 1, which is configured with the **rmon event** command. Possible events include a log entry or an SNMP trap. If the ifEntry.20.1 value changes by 0, the alarm is reset and can be triggered again.

show rmon Command Example

To display the RMON buffer capture table and current configuration, enter the **show rmon capture** EXEC command (Cisco 2500 series routers and Cisco AS5200 access servers only). A sample configuration follows:

```
Router# show rmon capture

Buffer 1 is active, owned by John Smith
Captured data is from channel 1
Slice size is 128, download size is 128
Download offset is 0
Full Status is full, full action is wrapWhenFull
Granted -1 octets out of -1 requested
Buffer has been on since 18:59:48, and has captured 522 packets
Current capture buffer entries:
Packet 3271 was captured 2018256 ms since buffer was turned on
Its length is 184 octets and has a status type of 0
Packet ID is 3721, and contains the following data:
03 00 00 00 00 0100 A0CC 3C9D DF00 A6F0 03
Packet 3722 was captured 2018452 ms since buffer was turned on
Its length is 64 octets and has a status type of 0
Packet ID is 3722, and contains the following data:
01 80C2 0000 0000 6009 FDFE D300 2642 03
```

To view values associated with RMON variables, enter the **show rmon matrix** EXEC command (Cisco 2500 series routers and Cisco AS5200 access servers only). The following is a sample output:

```
Router# show rmon matrix

Matrix 1 is active and owned by
Monitors ifEntry.1.1
Table size is 42, last time an entry was deleted was at 11:18:09
Source addr is 0000.0c47.007b, dest addr is ffff.ffff.ffff
Transmitted 2 pkts, 128 octets, 0 errors
Source addr is 0000.92a8.319e, dest addr is 0060.5c86.5b82
Transmitted 2 pkts, 384 octets, 1 error
```

Command Reference

The following commands are introduced or modified in the feature or features documented in this module. For information about these commands, see the *Cisco IOS Network Management Command Reference* at http://www.cisco.com/en/US/docs/ios/netmgmt/command/reference/nm_book.html. For information about all Cisco IOS commands, go to the Command Lookup Tool at http://tools.cisco.com/Support/CLILookup or to the Cisco IOS Master Commands List.

- **rmon capture userdata**
- **rmon collection history**
- **rmon collection host**
- **rmon collection matrix**
- **rmon collection rmon1**
- **show rmon capture**
- **show rmon filter**
- **show rmon hosts**
- **show rmon matrix**

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

Network Analysis Module (NM-NAM)

The Network Analysis Module (NM-NAM) feature is a network module that monitors and analyzes network traffic for a system using extended Remote Monitoring (RMON) standards, RMON2, and other Management Information Bases (MIBs).

Note The Network Analysis Module (NAM) is available in multiple hardware forms for some Cisco routers and Catalyst switches. This document applies only to the NAM for branch routers, also known as modular access, multiservice, or integrated services routers.

NAM provides Layer 2 to Layer 7 visibility into network traffic for remote troubleshooting, real-time traffic analysis, application performance monitoring, capacity planning, and managing network-based services, including quality of service (QoS) and Voice over IP (VoIP). The NAM Traffic Analyzer is software that is embedded in the NM-NAM that gives you browser-based access to the RMON1, RMON2, DSMON, and voice monitoring features of the NAM.

Feature History for NM-NAM

Release	Modification
12.3(4)XD	This feature was introduced on the following platforms: Cisco 2600XM series, Cisco 2691, Cisco 3660, Cisco 3725, and Cisco 3745.
12.3(7)T	This feature was integrated into Cisco IOS Release 12.3(7)T.
12.3(8)T4	This feature was implemented on the following platforms: Cisco 2811, Cisco 2821, and Cisco 2851.
12.3(11)T	This feature was implemented on the Cisco 3800 series.

Finding Support Information for Platforms and Cisco IOS Software Images

Use Cisco Feature Navigator to find information about platform support and Cisco IOS software image support. Access Cisco Feature Navigator at http://www.cisco.com/go/fn. You must have an account on Cisco.com. If you do not have an account or have forgotten your username or password, click **Cancel** at the login dialog box and follow the instructions that appear.

Americas Headquarters:
Cisco Systems, Inc., 170 West Tasman Drive, San Jose, CA 95134-1706 USA

Contents

- Prerequisites for the Network Analysis Module (NM-NAM), page 2
- Restrictions for the Network Analysis Module (NM-NAM), page 2
- Information About the Network Analysis Module (NM-NAM), page 3
- How to Configure and Manage the Network Analysis Module (NM-NAM), page 12
- Configuration Examples for the Network Analysis Module (NM-NAM), page 47
- Additional References, page 53
- Command Reference, page 55
- Glossary, page 56

Prerequisites for the Network Analysis Module (NM-NAM)

- Install Cisco IOS Release 12.3(4)XD, Cisco IOS Release 12.3(7)T, or a later release.
- Install the NM-NAM network module. Make sure that the network module is properly seated and that the EN (enable) and PWR (power) LEDs come on. Refer to the *Cisco Network Modules Hardware Installation Guide*.
- For Cisco 2691, Cisco 3725, and Cisco 3745 routers only, make sure that the router runs ROM Monitor (ROMMON) Version 12.2(8r)T2 or a later version. This ROMMON version contains a fix that prevents the router from resetting all the network modules when it is reloaded. Refer to the *ROM Monitor Download Procedures for Cisco 2691, Cisco, 3631, Cisco 3725, and Cisco 3745 Routers*.

Restrictions for the Network Analysis Module (NM-NAM)

General Restrictions

- Cisco IOS Release 12.3(4)XD, Cisco IOS Release 12.3(7)T, or a later release is required.
- Network Analysis Module Release 3.2 or a later release is required.
- Only one NM-NAM can be installed in the router at any time.
- SNMPv3 is not supported.
- Online insertion and removal (OIR), or hot swapping network modules, is supported on some platforms. To find out if your router supports hot swapping, refer to the *Network Modules Quick Start Guide*.

Traffic Monitoring Restrictions for the Internal NAM Interface

The following restrictions apply only to traffic that is monitored through the internal NAM interface:

- Only IP traffic can be monitored.
- The NAM Traffic Analyzer (web GUI) provides Layer 3 and higher layer information about the original packets. The Layer 2 header is modified by the router when it forwards the packets to the NAM, so the Layer 2 information that the NAM records is not applicable to the original packets.
- When Network Address Translation (NAT) is used, the router forwards packets containing the NAT "inside" network addresses to the NAM.

- When access control lists are used:
  - Packets dropped by an inbound access list are not forwarded to the NAM.
  - Packets dropped by an outbound access list are forwarded to the NAM for analysis.
- The NAM does *not* monitor the following:
  - Packets that are dropped by the Cisco IOS because of errors
  - Outbound IP multicast, IP broadcast, and User Datagram Protocol (UDP) flooding packets
  - Packets in generic routing encapsulation (GRE) tunnels

Note The previous restrictions (in the "Traffic Monitoring Restrictions for the Internal NAM Interface" section) do not apply to traffic monitored through the external NAM interface.

Information About the Network Analysis Module (NM-NAM)

To configure and manage the NM-NAM, you should understand the following concepts:

- NM-NAM Hardware, page 3
- NAM User Interfaces, page 4
- NAM Network Interfaces, page 5
- NM-NAM Operating Topologies and IP Address Assignments, page 6
- NAM CLI, page 11

Note For NM-NAM features and benefits, supported hardware and software, and other product information, refer to the *Cisco Branch Router Network Analysis Module Data Sheet*.

NM-NAM Hardware

For information on hardware installation and cable connections, refer to the *Cisco Network Modules Hardware Installation Guide*.

Specifications

Table 1 NM-NAM Specifications

Specification	Description
Processor	500 Mhz Intel Mobile Pentium III
SDRAM	256 MB
Internal disk storage	NM-NAM 20 GB IDE
Dimensions (H x W x D)	1.55 x 7.10 x 7.2 in. (3.9 x 18.0 x 19.3 cm)
Weight	1.5 lb (0.7 kg) (maximum)
Operating temperature	3° to 104°F (0° to 40°C)

Table 1 NM-NAM Specifications (continued)

Specification	Description
Nonoperating temperature	–40° to 185°F (–40° to 85°C)
Humidity	5 to 95% noncondensing
Operating altitude	0 to 10,000 ft (0 to 3,000 m)

Faceplate and LEDs

Figure 1 NM-NAM Faceplate and LEDs

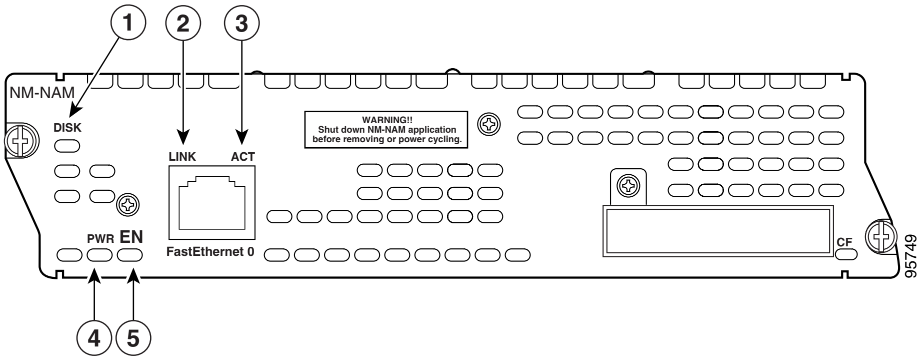

1
2
3
NM-NAM
DISK
WARNING!!
Shut down NM-NAM application before removing or power cycling.
LINK
ACT
PWR
EN
FastEthernet 0
CF
95749
4
5

Figure 1 Callout	LED	Indicates
1	DISK	There is activity on the hard drive.
2	LINK	The Fast Ethernet connection is available to the network module.
3	ACT	There is activity on the Fast Ethernet connection.
4	PWR	Power is available to the network module.
5	EN	The module has passed self-test and is available to the router.

NAM User Interfaces

The NAM has three user interfaces:

- Web GUI—The NAM Traffic Analyzer provides a browser-based GUI to configure and monitor the NAM.
- CLI—A NAM-specific command-line interface is used to configure NAM. It can be accessed through a NAM console session from the router or through Telnet or Secure Shell Protocol (SSH) over the network.
- SNMP—The NAM supports SNMPv1 and SNMPv2c access to the RMON MIBs. Note that the NAM Simple Network Management Protocol (SNMP) agent is separate from the SNMP agent in the router; the agents use different IP addresses and have independent communities.

NAM Network Interfaces

The NAM uses three interfaces for communication (see Figure 2):

- Analysis-Module Interface
- Internal NAM Interface
- External NAM Interface

Note The NM-NAM does not have an external console port. To access the NAM console, open a NAM console session from the router or use Telnet or SSH over the network. The lack of an external console port on the NM-NAM means that the initial boot configuration is possible only through the router.

Figure 2 NAM Network Interfaces

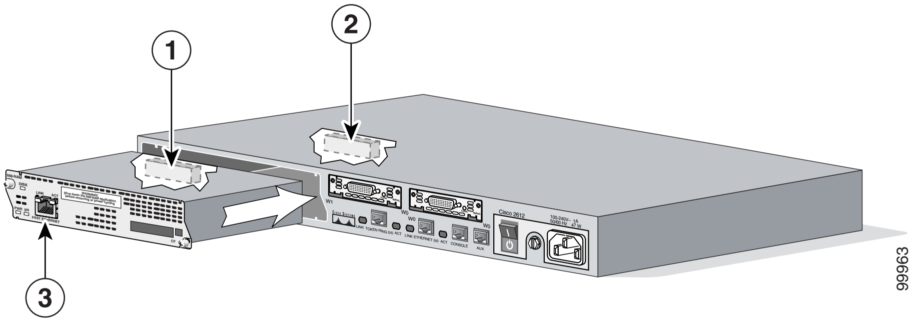


Figure 2 Callout	Interface	Location	Configure and Manage From
1	Internal NAM interface	NM-NAM internal	NAM CLI
2	Analysis-Module interface	Router internal	Cisco IOS CLI
3	External NAM interface	NM-NAM faceplate	NAM CLI

Analysis-Module Interface

The Analysis-Module interface is used to access the NAM console for the initial configuration. After configuring the NAM IP parameters, the Analysis-Module interface is typically used only during NAM software upgrades and while troubleshooting if the NAM Traffic Analyzer is inaccessible.

Visible only to the Cisco IOS software on the router, the Analysis-Module interface is an internal Fast Ethernet interface on the router that connects to the internal NAM interface. The Analysis-Module interface is connected to the router's Peripheral Component Interconnect (PCI) backplane, and all configuration and management of the Analysis-Module interface must be performed from the Cisco IOS CLI.

Internal NAM Interface

The internal NAM interface is used for monitoring traffic that passes through router interfaces. You can also select the internal NAM interface as the management interface for the NAM.

Visible only to the NAM software on the NM-NAM, the internal NAM interface is the Fast Ethernet interface on the NM-NAM that connects to the Analysis-Module interface on the router. The internal NAM interface is connected to the PCI bus on the NM-NAM, and all configuration and management of the internal NAM interface must be performed from the NAM software.

External NAM Interface

The external NAM interface can be used to monitor LAN traffic. You can also select the external NAM interface as the management interface for the NAM.

Visible only to the NAM software on the NM-NAM, the external NAM interface is the Fast Ethernet interface on the NM-NAM faceplate (see Figure 1 on page 4). The external NAM interface supports data requests and data transfers from outside sources, and it provides direct connectivity to the LAN through an RJ-45 connector. All configuration and management of the external NAM interface must be performed from the NAM software.

NM-NAM Operating Topologies and IP Address Assignments

This section includes the following topics:

- Management Traffic—Choose One of the NM-NAM Interfaces, page 6
- Monitored Traffic—Use One or Both of the NM-NAM Interfaces, page 7
- Sample Operating Topologies, page 8

Management Traffic—Choose One of the NM-NAM Interfaces

Select either the internal or external NAM interface to handle management traffic such as IP, HTTP, SNMP, Telnet, and SSH. You cannot send management traffic through both NAM interfaces at the same time.

How you assign IP addresses on the NAM network interfaces depends on which NAM interface, internal or external, you use for management traffic. See the following sections:

- Internal NAM Interface for Management Traffic—How to Assign IP Addresses, page 6
- External NAM Interface for Management Traffic—How to Assign IP Addresses, page 7

Internal NAM Interface for Management Traffic—How to Assign IP Addresses

If you select the internal NAM interface to handle management traffic:

- For the Analysis-Module interface (in Cisco IOS CLI), assign an IP address from a routable subnet. To conserve IP address space, you can configure the Analysis-Module as an IP unnumbered interface and borrow the IP address of another router interface, such as a Fast Ethernet or loopback interface. The borrowed IP address must come from a routable subnet.
- For the NAM system (in NAM CLI), assign an IP address from the same subnet that is assigned to the Analysis-Module interface.

External NAM Interface for Management Traffic—How to Assign IP Addresses

If you select the external NAM interface to handle management traffic:

- For the Analysis-Module interface (in Cisco IOS CLI), we recommend that you use the IP unnumbered interface configuration to borrow the IP address of another router interface. The subnet does not need to be routable.
- For the NAM system (in NAM CLI), assign an IP address from the subnet that is connected to the external NAM interface.

Monitored Traffic—Use One or Both of the NM-NAM Interfaces

You can use either or both the internal and external NAM interfaces for monitoring traffic:

- Internal NAM Interface—Monitor LAN and WAN Traffic, page 7
- External NAM Interface—Monitor LAN Traffic, page 7

The same interface can be used for both management traffic and monitored traffic simultaneously.

Internal NAM Interface—Monitor LAN and WAN Traffic

When you monitor traffic through the internal NAM interface, you must enable NAM packet monitoring on each router interface that you want to monitor. NAM packet monitoring uses Cisco Express Forwarding (CEF) to send a copy of each packet that is received or sent out of the router interface to the NAM.

Note Some restrictions apply when monitoring traffic through the internal NAM interface. See the "Traffic Monitoring Restrictions for the Internal NAM Interface" section on page 2.

Monitoring traffic through the internal NAM interface enables the NAM to see any encrypted traffic after it has already been decrypted by the router.

Note Traffic sent through the internal NAM interface—and the router's Analysis-Module interface—uses router resources such as CPU, SDRAM bandwidth, and backplane PCI bandwidth. Therefore, we recommend that you use the internal NAM interface to monitor WAN interfaces, and use the external NAM interface to monitor LAN interfaces.

External NAM Interface—Monitor LAN Traffic

Monitoring traffic through the external NAM interface does not impact router resources. Therefore, we recommend that you use the external NAM interface to monitor LAN traffic.

To monitor ports on Ethernet switching cards or modules (NM-16ESW-*x*, NMD-36ESW-*x*, HWIC-4ESW, or HWIC-D-9ESW), configure a Switched Port Analyzer (SPAN) session whose destination is the Ethernet switch port that connects to the external NAM interface. For more information about configuring SPAN for these cards and modules, refer to the following documents:

- *16- and 36-Port Ethernet Switch Module for Cisco 2600 Series, Cisco 3600 Series, and Cisco 3700 Series*, Cisco IOS feature module
- *Cisco HWIC-4ESW and HWIC-D-9ESW EtherSwitch Interface Cards*, Cisco IOS feature module

Sample Operating Topologies

In each of the following topologies, the router's LAN interface is monitored through the external NAM interface, and the router's WAN interface is monitored through the internal NAM interface:

- NAM Management Interface Is Internal and Analysis-Module Interface Is Assigned an IP Address, page 8
- NAM Management Interface Is Internal and Analysis-Module Interface Is IP Unnumbered, page 9
- NAM Management Interface Is External and Analysis-Module Interface Is IP Unnumbered, page 10

To see sample configurations for the following topologies, see the "Configuration Examples for the Network Analysis Module (NM-NAM)" section on page 47.

NAM Management Interface Is Internal and Analysis-Module Interface Is Assigned an IP Address

Figure 3 shows a sample topology, in which:

- The internal NAM interface is used for management traffic.
- IP addresses from the same routable subnet are assigned to the Analysis-Module interface and the NAM system.

Figure 3 ***Sample Topology: NAM Management Interface Is Internal and Analysis-Module Interface Is Assigned an IP Address***

Figure 3 Callout	Interface	Location
1	Analysis-Module interface	Router internal
2	Internal NAM interface (**management**)	NM-NAM internal
3	External NAM interface	NM-NAM faceplate

Figure 3 Callout	Interface	Location
4	Serial interface	WAN interface card (WIC)
5	Fast Ethernet interface	Router rear panel

NAM Management Interface Is Internal and Analysis-Module Interface Is IP Unnumbered

Figure 4 shows a sample topology, in which:

- The internal NAM interface is used for management traffic.
- IP addresses from the same routable subnet are assigned to the Analysis-Module interface and the NAM system.
- To conserve IP address space, the Analysis-Module interface is configured as IP unnumbered to borrow the IP address of the Fast Ethernet interface.

Figure 4 Sample Topology: NAM Management Interface Is Internal and Analysis-Module Interface Is IP Unnumbered

Figure 4 Callout	Interface	Location
1	Analysis-Module interface	Router internal
2	Internal NAM interface (**management**)	NM-NAM internal
3	External NAM interface	NM-NAM faceplate
4	Serial interface	WAN interface card (WIC)
5	Fast Ethernet interface	Router rear panel

NAM Management Interface Is External and Analysis-Module Interface Is IP Unnumbered

Figure 5 shows a sample topology where:

- The external NAM interface is used for management traffic.
- The Analysis-Module interface is configured as IP unnumbered to borrow an IP address from the loopback interface.
- The borrowed loopback interface IP address is not routable.
- The NAM system is configured with an IP address from the LAN subnet that is connected to the external NAM interface.

Figure 5 Sample Topology: NAM Management Interface Is External and Analysis-Module Interface Is IP Unnumbered

Figure 5 Callout	Interface	Location
1	Analysis-Module interface	Router internal
2	Internal NAM interface	NM-NAM internal
3	External NAM interface (**management**)	NM-NAM faceplate
4	Loopback interface	Router internal
5	Serial interface	WAN interface card (WIC)
6	Fast Ethernet interface	Router rear panel

NAM CLI

This section includes the following topics:

- NAM CLI Access
- NAM CLI Prompt
- Basic NAM CLI Commands
- NAM CLI Context-Sensitive Help

NAM CLI Access

There are three ways to access the NAM CLI:

- Open a NAM console session from the router in which the NM-NAM is installed—See the "Opening and Closing a NAM Console Session from the Router" section on page 18.
- Telnet—See the "Opening and Closing a Telnet or SSH Session to the NAM" section on page 38.
- SSH—See the "Opening and Closing a Telnet or SSH Session to the NAM" section on page 38.

Until you properly configure the NAM IP parameters, the only way to access the NAM CLI is by opening a NAM console session from the router.

NAM CLI Prompt

The NAM CLI prompt is `root@`*`nam-system-hostname`*`#`. For example, if the NAM system hostname is configured as "nam1," then the NAM CLI prompt appears as `root@nam1#`.

If the NAM system hostname has not yet been configured, the NAM CLI prompt is `root@localhost#`.

Basic NAM CLI Commands

Table 2 briefly describes the basic NAM CLI commands that are used for initial configuration and maintenance of the NM-NAM. For a complete description of all NAM CLI commands, refer to the *Network Analysis Module Command Reference* for your NAM software release.

Note Although NAM CLI commands appear similar to Cisco IOS commands, the commands described in Table 2 operate in the NAM CLI only.

Table 2 Basic NAM CLI Commands

NAM CLI Command	Purpose
exsession on	Enables outside logins (Telnet).
exsession on ssh	Enables outside logins (SSH).
ip address	Sets the system IP address.
ip broadcast	Sets the system broadcast address.
ip domain	Sets the system domain name.
ip gateway	Sets the system default gateway address.
ip host	Sets the system hostname.

Table 2 ***Basic NAM CLI Commands (continued)***

NAM CLI Command	Purpose
ip http secure server enable	Enables the secure HTTP server.
ip http server enable	Enables the HTTP server.
ip interface external	Selects the external NAM interface for management traffic.
ip interface internal	Selects the internal NAM interface for management traffic.
ip nameserver	Sets the system name server address.
password root	Sets a new password to access the root (read/write) level of NAM.
patch	Downloads and installs a software patch.
ping	Checks connectivity to a network device.
show ip	Displays the NAM IP parameters.

NAM CLI Context-Sensitive Help

Table 3 shows how to use the NAM CLI context-sensitive help.

Table 3 ***NAM CLI Context-Sensitive Help Commands***

NAM CLI Command	Purpose

`(prompt)# ?`
or
`(prompt)# help`
	Displays a list of commands available for the command mode.

`(prompt)# abbreviated-command-entry<Tab>`	Lists commands in the current mode that begin with a particular character string.
`(prompt)# command ?`	Lists the available syntax options (arguments and keywords) for the command.
`(prompt)# command keyword ?`	Lists the next available syntax option for the command.

How to Configure and Manage the Network Analysis Module (NM-NAM)

This section contains the following procedures:

- Configuring the Analysis-Module Interface on the Router, page 13 (required)
- Disabling AAA Login Authentication on the NAM Console Line, page 16 (optional)
- Opening and Closing a NAM Console Session from the Router, page 18 (required for initial configuration)
- Configuring the NM-NAM, page 21 (required for initial configuration)

- Configuring a Static Route to the NAM Through the Analysis-Module Interface, page 25 (required for using the internal NAM interface for management traffic)
- Enabling NAM Packet Monitoring, page 26 (required for monitoring traffic through the internal NAM interface)
- Enabling and Accessing the NAM Traffic Analyzer, page 28 (required)
- Changing the NAM Root Password, page 31 (optional)
- Resetting the NAM Root Password to the Default Value, page 34 (optional)
- Opening and Closing a Telnet or SSH Session to the NAM, page 38 (optional)
- Upgrading the NAM Software, page 41 (optional)

Configuring the Analysis-Module Interface on the Router

This section describes how to configure the Analysis-Module interface on the router. For general information on the Analysis-Module interface, see the "Analysis-Module Interface" section on page 5.

For information on assigning the IP address of the Analysis-Module interface, see the "NM-NAM Operating Topologies and IP Address Assignments" section on page 6.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **interface** *type number*
4. **ip address** *ip-address mask*
5. **interface analysis-module** *slot*/**0**
6. **ip unnumbered** *interface number*
   or
   ip address *ip-address mask*
7. **no shutdown**
8. **end**
9. **show ip interface brief**
   or
   show running-config

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. - Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**interface** *type number* **Example:** `Router(config)# interface loopback 0`	(Optional) Configures an interface, and enters interface configuration mode. - Perform this step if you plan to configure the Analysis-Module interface as an IP unnumbered interface. - This step configures the router interface (such as a loopback or Fast Ethernet interface) whose IP address you plan to borrow for the IP unnumbered Analysis-Module interface.
Step 4	**ip address** *ip-address mask* **Example:** `Router(config-if)# ip address 10.20.30.40 255.255.255.0`	(Optional) Sets an IP address and mask for the interface. - Perform this step if you plan to configure the Analysis-Module interface as an IP unnumbered interface. - If you plan to use the internal NAM interface for management traffic, this IP address must come from a routable subnet.
Step 5	**interface analysis-module** *slot*/**0** **Example:** `Router(config)# interface analysis-module 1/0`	Configures the Analysis-Module interface. - This is the Fast Ethernet interface on the router that is connected to the internal NM-NAM interface.

Step 6	**ip unnumbered** *interface number*
	or
	ip address *ip-address mask*
	Example:
	`Router(config-if)# ip unnumbered loopback 0`
	Example:
	`Router(config-if)# ip address 10.20.30.40 255.255.255.0`
		Configures the Analysis-Module interface as IP unnumbered and specifies the interface whose IP address is borrowed by the Analysis-Module interface.
		or
		Sets an IP address and mask on the Analysis-Module interface.
		- Use the **ip unnumbered** command if you performed Step 3 and Step 4.

	Command or Action	Purpose
Step 7	**no shutdown** **Example:** `Router(config-if)# no shutdown`	Activates the Analysis-Module interface.

Step 8	**end**
	Example:
	`Router(config-if)# end`
	`Router#`
		Returns to privileged EXEC mode.

Step 9	**show ip interface brief**
	or
	show running-config
	Example:
	`Router# show ip interface brief`
	Example:
	`Router# show running-config`
		Displays the IP addresses and summary status of the interfaces.
		or
		Displays the contents of the currently running configuration file.
		• Verify that you properly configured the Analysis-Module interface.
		• If you configured the Analysis-Module interface as IP unnumbered, then use the **show running-config** command to verify proper configuration of both the Analysis-Module interface and the interface whose IP address you borrowed for the Analysis-Module interface.

Tip To avoid losing your configuration at the next system reload or power cycle, save the running configuration to the startup configuration by entering the **copy run start** command in privileged EXEC mode.

Examples

This section provides the following examples:

- Configuring the Analysis-Module Interface—Routable Subnet: Example, page 15
- Configuring the Analysis-Module Interface—IP Unnumbered with Routable Subnet: Example, page 16
- Configuring the Analysis-Module Interface—IP Unnumbered with Subnet That Is Not Routable: Example, page 16
- Sample Output for the show ip interface brief Command, page 16

Configuring the Analysis-Module Interface—Routable Subnet: Example

In the following example, the Analysis-Module interface is configured with a routable IP address. The NM-NAM is installed in router slot 2.

```
!
interface Analysis-Module 2/0
 ip address 209.165.200.230 255.255.255.224
 no shutdown
```

Configuring the Analysis-Module Interface—IP Unnumbered with Routable Subnet: Example

In the following example, the Analysis-Module interface is IP unnumbered and borrows the IP address of the Fast Ethernet interface. The IP address is from a routable subnet, and the NM-NAM is installed in router slot 1.

```
!
interface FastEthernet 0/0
 ip address 209.165.202.129 255.255.255.224
 no shutdown
!
interface Analysis-Module 1/0
 ip unnumbered FastEthernet 0/0
 no shutdown
!
```

Configuring the Analysis-Module Interface—IP Unnumbered with Subnet That Is Not Routable: Example

In the following example, the Analysis-Module interface is IP unnumbered and borrows a loopback interface IP address that is not routable. The NM-NAM is installed in router slot 3.

```
!
interface loopback 0
 ip address 10.20.30.40 255.255.255.0
!
interface Analysis-Module 3/0
 ip unnumbered loopback 0
 no shutdown
!
```

Sample Output for the show ip interface brief Command

```
Router# show ip interface brief

Interface                  IP-Address      OK?   Method      Status         Protocol
FastEthernet0/0            172.20.105.213  YES   NVRAM       up             up
FastEthernet0/1            172.20.105.53   YES   NVRAM       up             up
Analysis-Module2/0         10.1.1.1        YES   manual      up             up
Router#
```

What to Do Next

If you configured authentication, authorization, and accounting (AAA) on your router, then proceed to the "Disabling AAA Login Authentication on the NAM Console Line" section on page 16.

Otherwise, proceed to the "Opening and Closing a NAM Console Session from the Router" section on page 18.

Disabling AAA Login Authentication on the NAM Console Line

If you configured authentication, authorization, and accounting (AAA) on your router, then you may have to log in twice to open a NAM console session from the router: first with your AAA username and password, and second with the NAM login and password.

If you do not want to log in twice to open a NAM console session from the router, then disable AAA login authentication on the router's NAM console line by performing the steps in this section.

Note, however, that if your router contains both the NM-NAM and the NM-CIDS, the Cisco intrusion detection system network module, then AAA can be a useful tool for centrally controlling access to both network modules. For information about AAA, refer to the *Cisco IOS Security Configuration Guide*.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **aaa authentication login** *list-name* **none**
4. **line** *number*
5. **login authentication** *list-name*
6. **end**
7. **show running-config**

DETAILED STEPS

	Command or Action	Purpose

Step 1	**`enable`**
	Example:
	`Router> enable`
		Enables privileged EXEC mode.
		- Enter your password if prompted.

Step 2	**`configure terminal`**
	Example:
	`Router# configure terminal`
		Enters global configuration mode.

Step 3	**`aaa authentication login`** *`list-name`* **`none`**
	Example:
	`Router(config)# aaa authentication login nam none`
		Creates a local authentication list.
		- The **none** keyword specifies no authentication for this list.

Step 4	**`line`** *`number`*
	Example:
	`Router(config)# line 33`
		Enters line configuration mode for the line to which you want to apply the authentication list.
		- The *number* value is determined by the slot number in which the NM-NAM is installed:
		number = (32 x *slot*) + 1

Step 5	**`login authentication`** *`list-name`*
	Example:
	`Router(config-line)# login authentication nam`
		Applies the authentication list to the line.
		- Specify the list name that you configured in Step 3.

Step 6	**`end`**
	Example:
	`Router(config-line)# end`
	`Router#`
		Returns to privileged EXEC mode.

Step 7	**`show running-config`**
	Example:
	`Router# show running-config`
		Displays the contents of the currently running configuration file.
		- Verify that you configured the local authentication list and applied it to the line associated with the NM-NAM.

What to Do Next

Proceed to the "Opening and Closing a NAM Console Session from the Router" section on page 18.

Opening and Closing a NAM Console Session from the Router

This section describes how to open and close a NAM console session from the router.

SUMMARY STEPS

1. **enable**
2. **service-module analysis-module** *slot*/**0 session**
3. Press **Return**.
   or
   If a username prompt appears, then log in with your AAA username and password.
4. At the login prompt, enter **root**.
5. At the password prompt, enter your password.
   or
   If you have not changed the password from the factory-set default, enter **root** as the root password.
6. Perform the tasks that you need to perform in the NAM CLI. When you want to end the NAM console session and return to the Cisco IOS CLI, complete Step 7 through Step 10.
7. **exit**
8. Hold **Ctrl-Shift** and press **6**. Release all keys, and then press **x**.
9. **disconnect**
10. Press **Enter**.

DETAILED STEPS

	Command or Action	Purpose

Step 1	**`enable`**
	Example:
	`Router> enable`
		Enables privileged EXEC mode.
		• Enter your password if prompted.

Step 2	**`service-module analysis-module`** *`slot`*`/`**`0 session`**
	Example:
	`Router# service-module analysis-module 1/0 session`
	Example:
	`Router# service-module analysis-module 1/0 session clear`
	`[confirm]`
	` [OK]`
	`Router# service-module analysis-module 1/0 session`
		Establishes a console session with the NAM.
		• If you cannot open a NAM console session, make sure that the NAM console line is clear by first entering the **service-module analysis-module** *slot*/**0 session clear** command in privileged EXEC mode.

	Command or Action	Purpose

Step 3	Press **Return**.
	or
	If a username prompt appears, then log in with your AAA username and password.
	Example:
	`Trying 10.1.1.1, 2065 ... Open`
	`<Press Return>`
	
	`Cisco Network Analysis Module (NM-NAM)`
	
	`nam1.cisco.com login:`
	Example:
	`Trying 10.1.1.1, 2065... Open`
	`User Access Verification`
	
	`Username: myaaausername`
	`Password: <myaaapassword>`
	`Cisco Network Analysis Module (NM-NAM)`
	
	`nam1.cisco.com login:`
		Activates the NAM console line.
		or
		Completes AAA login authentication and activates the NAM console line.
		• If AAA is configured on your router and you do not want to log in twice to access the NAM console, then complete the steps in the "Disabling AAA Login Authentication on the NAM Console Line" section on page 16.

Step 4	At the login prompt, enter **root**. **Example:** `login: root`	Accesses the root (read/write) level of NAM.
Step 5	At the password prompt, enter your password. or If you have not changed the password from the factory-set default, enter **root** as the root password. **Example:** `Password: <root>`	—
Step 6	Perform the tasks that you need to perform in the NAM CLI. When you want to end the NAM console session and return to the Cisco IOS CLI, complete Step 7 through Step 10.	For initial configuration tasks, see the "Configuring the NM-NAM" section on page 21. For help using NAM CLI commands, see the "NAM CLI Context-Sensitive Help" section on page 12.

Step 7	**`exit`**
	Example:
	`root@localhost(sub-custom-filter-capture)# exit`
	`root@localhost# exit`
	
	`login:`
		Logs out of the NAM system or leaves a subcommand mode.
		• If you are in a subcommand mode, continue to enter the **exit** command until you see the NAM login prompt.

	Command or Action	Purpose

Step 8	Hold **Ctrl-Shift** and press **6**. Release all keys, and then press **x**.
	Example:
	`login: <suspend keystroke>`
	`Router#`
		Suspends and closes the Telnet session.

Step 9	**`disconnect`**
	Example:
	`Router# disconnect`
		Disconnects a line.

Step 10	Press **Enter**.
	Example:
	`Closing connection to 10.20.30.40 [confirm]`
	`<Enter>`
		Confirms that you want to disconnect the line.

Examples

This section provides the following examples:

- Opening and Closing a NAM Console Session When AAA Authentication Is Not Configured or Is Disabled on the NAM Console Line: Example, page 20
- Opening and Closing a NAM Console Session When AAA Authentication Is Configured and Enabled on the NAM Console Line: Example, page 21

Opening and Closing a NAM Console Session When AAA Authentication Is Not Configured or Is Disabled on the NAM Console Line: Example

In the following example, a NAM console session is opened and closed from the router. The NM-NAM is installed in router slot 2.

```
Router# service-module analysis-module 2/0 session
Trying 10.1.1.1, 2065 ... Open


Cisco Network Analysis Module (NM-NAM)

nam1.cisco.com login: root
Password: <password>
Terminal type: vt100

Cisco Network Analysis Module (NM-NAM) Console, 3.2
Copyright (c) 1999-2003 by cisco Systems, Inc.

WARNING! Default password has not been changed!
root@nam1.cisco.com#
root@nam1.cisco.com# exit


Cisco Network Analysis Module (NM-NAM)

nam1.cisco.com login: <suspend keystroke>
Router# disconnect
Closing connection to 10.1.1.1 [confirm] <Enter>
Deleting login session
```

Opening and Closing a NAM Console Session When AAA Authentication Is Configured and Enabled on the NAM Console Line: Example

In the following example, a NAM console session is opened and closed from the router. The NM-NAM is installed in router slot 2.

```
Router# service-module analysis-module 2/0 session
Trying 10.1.1.1, 2065 ... Open
User Access Verification

Username: myaaausername
Password: <myaaapassword>
Cisco Network Analysis Module (NM-NAM)

nam1.cisco.com login: root
Password: <nampassword>
Terminal type: vt100

Cisco Network Analysis Module (NM-NAM) Console, 3.2
Copyright (c) 1999-2003 by cisco Systems, Inc.

WARNING! Default password has not been changed!
root@nam1.cisco.com#
root@nam1.cisco.com# exit



Cisco Network Analysis Module (NM-NAM)

nam1.cisco.com login: <suspend keystroke>
Router# disconnect
Closing connection to 10.1.1.1 [confirm] <Enter>
Deleting login session
```

Troubleshooting Tips

Make sure that the NAM console line is clear by entering the **service-module analysis-module** *slot*/**0 session clear** command in privileged EXEC mode.

What to Do Next

Proceed to the "Configuring the NM-NAM" section.

Configuring the NM-NAM

This section describes how to configure the NM-NAM to establish network connectivity and configure IP parameters. This task must be performed from the NAM CLI. For more advanced NAM configuration, use the NAM Traffic Analyzer (web GUI) or refer to the *Network Analysis Module Command Reference* for your NAM software release.

For information on assigning IP addresses, see the "NM-NAM Operating Topologies and IP Address Assignments" section on page 6.

Prerequisites

Before performing this task, access the NAM console by performing Step 1 through Step 5 in the "Opening and Closing a NAM Console Session from the Router" section on page 18.

SUMMARY STEPS

1. **ip interface** {**internal** | **external**}
2. **ip address** *ip-address subnet-mask*
3. **ip broadcast** *broadcast-address*
4. **ip gateway** *ip-address*
5. **exsession on**
   or
   exsession on ssh
6. **ip domain** *name*
7. **ip host** *name*
8. **ip nameserver** *ip-address* [*ip-address*][*ip-address*]
9. **ping** {*host* | *ip-address*}
10. **show ip**

DETAILED STEPS

	Command or Action	Purpose

Step 1	`ip interface {internal | external}`
	Example:
	`root@localhost# ip interface internal`
	Example:
	`root@localhost# ip interface external`
		Specifies which NAM interface will handle management traffic.

Step 2	`ip address ip-address subnet-mask`
	Example:
	`root@localhost# ip address 172.20.104.126 255.255.255.248`
		Configures the NAM system IP address.
		• For information on assigning the IP address, see the "Management Traffic—Choose One of the NM-NAM Interfaces" section on page 6.

Step 3	`ip broadcast broadcast-address`
	Example:
	`root@localhost# ip broadcast 10.255.255.255`
		(Optional) Configures the NAM system broadcast address.

Step 4	`ip gateway ip-address`
	Example:
	`root@localhost# ip gateway 172.20.104.125`
		Configures the NAM system default gateway address.

	Command or Action	Purpose

Step 5	**`exsession on`**
	or
	`exsession on ssh`
	Example:
	`root@localhost# exsession on`
	Example:
	`root@localhost# exsession on ssh`
		(Optional) Enables outside logins.
		• **exsession on** enables Telnet access.
		• **exsession on ssh** enables SSH access.
		Note The NAM software K9 crypto patch is required to configure the **ssh** option. You can download the patch from Cisco.com.

Step 6	**`ip domain`** *`name`*
	Example:
	`root@localhost# ip domain cisco.com`
		(Optional) Sets the NAM system domain name.

Step 7	**`ip host`** *`name`*
	Example:
	`root@localhost# ip host nam1`
		(Optional) Sets the NAM system hostname.

Step 8	**`ip nameserver`** *`ip-address`* `[`*`ip-address`*`][`*`ip-address`*`]`
	Example:
	`root@nam1# ip nameserver 209.165.201.1`
		(Optional) Sets one or more NAM system name servers.
		• We recommend that you configure a name server for the NAM system to resolve Domain Name System (DNS) requests.

Step 9	**`ping`** `{`*`host`* `|` *`ip-address`*`}`
	Example:
	`root@nam1# ping 10.20.30.40`
		Checks connectivity to a network device.
		• Verify connectivity to the router or another known host.

Step 10	**`show ip`**
	Example:
	`root@nam1# show ip`
		Displays the NAM IP parameters.
		• Verify that you properly configured the NM-NAM.

Examples

This section provides the following examples:

- Configuring the NM-NAM: Example, page 23
- Checking Network Connectivity with Ping: Example, page 24
- Sample Output for the show ip NAM CLI Command, page 24

Configuring the NM-NAM: Example

In the following example, the external NAM interface is used for management traffic. The HTTP server and Telnet access are enabled. The resulting NAM CLI prompt is `root@nam1.cisco.com#`.

```
!
ip address 172.20.105.215 255.255.255.192
!
```

```
ip host "nam1"
!
ip domain "cisco.com"
!
ip gateway 172.20.105.210
!
ip broadcast 10.255.255.255
!
ip nameserver 209.165.201.29
!
ip interface external
!
ip http server enable
!
exsession on
!
```

Checking Network Connectivity with Ping: Example

```
root@nam1.cisco.com# ping 172.20.105.213

PING 172.20.105.213 (172.20.105.213) from 172.20.105.215 : 56(84) bytes of data.
64 bytes from 172.20.105.213: icmp_seq=0 ttl=255 time=353 usec
64 bytes from 172.20.105.213: icmp_seq=1 ttl=255 time=289 usec
64 bytes from 172.20.105.213: icmp_seq=2 ttl=255 time=284 usec
64 bytes from 172.20.105.213: icmp_seq=3 ttl=255 time=283 usec
64 bytes from 172.20.105.213: icmp_seq=4 ttl=255 time=297 usec

--- 172.20.105.213 ping statistics ---
5 packets transmitted, 5 packets received, 0% packet loss
round-trip min/avg/max/mdev = 0.283/0.301/0.353/0.028 ms
root@nam1.cisco.com#
```

Sample Output for the show ip NAM CLI Command

```
root@nam1.cisco.com# show ip

IP address:               172.20.105.215
Subnet mask:              255.255.255.192
IP Broadcast:             10.255.255.255
IP Interface:             External
DNS Name:                 nam1.cisco.com
Default Gateway:          172.20.105.210
Nameserver(s):            209.165.201.29
HTTP server:              Enabled
HTTP secure server:       Disabled
HTTP port:                80
HTTP secure port:         443
TACACS+ configured:       No
Telnet:                   Enabled
SSH:                      Disabled
root@nam1.cisco.com#
```

What to Do Next

If you selected the internal NAM interface to handle management traffic in Step 1, then proceed to the "Configuring a Static Route to the NAM Through the Analysis-Module Interface" section on page 25.

If you plan to monitor traffic through the internal NAM interface, then proceed to the "Enabling NAM Packet Monitoring" section on page 26.

If you do not plan to monitor traffic through the internal NAM interface, then proceed to the "Enabling and Accessing the NAM Traffic Analyzer" section on page 28.

Configuring a Static Route to the NAM Through the Analysis-Module Interface

This section describes how to ensure that the router can route packets to the NAM by configuring a static route through the Analysis-Module interface.

If you select the internal NAM interface to handle management traffic, then configuring a static route to the NAM through the Analysis-Module interface is:

- Required when the Analysis-Module interface is IP unnumbered.
- Recommended when the Analysis-Module interface is assigned a unique IP address.

If you select the external NAM interface to handle management traffic, then you do not need to perform this task. Proceed to the "What to Do Next" section on page 26.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ip route** *nam-ip-address mask* **analysis-module** *slot/unit*
4. **end**
5. **ping** {*nam-ip-address* | *nam-hostname*}

DETAILED STEPS

	Command or Action	Purpose

Step 1	**`enable`**
	Example:
	`Router> enable`
		Enables privileged EXEC mode.
		• Enter your password if prompted.

Step 2	**`configure terminal`**
	Example:
	`Router# configure terminal`
		Enters global configuration mode.

Step 3	**`ip route`** *`nam-ip-address mask`* **`analysis-module`** *`slot/unit`*
	Example:
	`Router(config)# ip route 172.20.105.215 255.255.255.192 analysis-module 1/0`
		Establishes a static route to the NAM.

Step 4	**`end`**
	Example:
	`Router(config-if)# end`
	`Router#`
		Returns to privileged EXEC mode.

Step 5	**`ping`** {*`nam-ip-address`* | *`nam-hostname`*}
	Example:
	`Router# ping 172.20.105.215`
		Verifies network connectivity to the NAM.

Examples

This section provides the following examples:

- Configuring a Static Route to the NAM Through the Analysis-Module Interface: Example, page 26
- Verifying Network Connectivity with Ping: Example, page 26

Configuring a Static Route to the NAM Through the Analysis-Module Interface: Example

In the following example, a static route is configured to the NAM whose system IP address is 172.20.105.215. The NM-NAM is installed in router slot 1.

```
!
ip route 172.20.105.215 255.255.255.192 analysis-module 1/0
!
interface FastEthernet 0/0
 ip address 209.165.202.129 255.255.255.224
 no shutdown
!
interface Analysis-Module 1/0
 ip unnumbered FastEthernet 0/0
 no shutdown
!
```

Verifying Network Connectivity with Ping: Example

In the following example, entering the **ping** command verifies network connectivity to the NAM with IP address 172.20.105.215.

```
Router# ping 172.20.105.215

Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 172.20.105.215, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 1/1/1 ms
Router#
```

What to Do Next

If you plan to monitor traffic through the internal NAM interface, then proceed to the "Enabling NAM Packet Monitoring" section on page 26.

If you do not plan to monitor traffic through the internal NAM interface, then proceed to the "Enabling and Accessing the NAM Traffic Analyzer" section on page 28.

Enabling NAM Packet Monitoring

This section describes how to enable NAM packet monitoring on router interfaces that you want to monitor through the internal NAM interface.

When you enable NAM packet monitoring on an interface, CEF sends an extra copy of each IP packet that is received or sent out on that interface to the NAM through the Analysis-Module interface on the router and the internal NAM interface.

SUMMARY STEPS

1. **enable**
2. **configure terminal**

3. **ip cef**
4. **interface** *type slot/port*
   or
   interface *type slot/wic-slot/port*
5. **analysis-module monitoring**
6. Repeat Step 4 and Step 5 for each interface that you want the NAM to monitor.
7. **end**
8. **show running-config**

DETAILED STEPS

	Command or Action	Purpose

Step 1	**`enable`**
	Example:
	`Router> enable`
		Enables privileged EXEC mode.
		- Enter your password if prompted.

Step 2	**`configure terminal`**
	Example:
	`Router# configure terminal`
		Enters global configuration mode.

Step 3	**`ip cef`**
	Example:
	`Router(config)# ip cef`
		Enables the CEF switching path.

Step 4	**`interface`** *`type slot/port`*
	or
	`interface` *`type slot/wic-slot/port`*
	Example:
	`Router(config)# interface serial 0/0`
		Selects an interface for configuration.

Step 5	**`analysis-module monitoring`**
	Example:
	`Router(config-if)# analysis-module monitoring`
		Enables NAM packet monitoring on the interface.

Step 6	Repeat Step 4 and Step 5 for each interface that you want the NAM to monitor through the internal NAM interface.	—

	Command or Action	Purpose

Step 7	**end**
	Example:
	`Router(config-if)# end`
	`Router#`
		Returns to privileged EXEC mode.

Step 8	**show running-config** **Example:** `Router# show running-config`	Displays the contents of the currently running configuration file. • Verify that you enabled the CEF switching path and enabled packet monitoring on the correct interfaces.

Example

This section provides the following example:

- Enabling NAM Packet Monitoring: Example, page 28

Enabling NAM Packet Monitoring: Example

In the following example, NAM packet monitoring is enabled on the serial interfaces:

```
interface Serial 0/0
 ip address 172.20.105.213 255.255.255.240
 ip route-cache flow
 speed auto
 full-duplex
 analysis-module monitoring
 no mop enabled
!
interface Serial 0/1
 ip address 172.20.105.53 255.255.255.252
 ip route-cache flow
 duplex auto
 speed auto
 analysis-module monitoring
!
interface Analysis-Module 2/0
 ip address 10.1.1.1 255.255.255.0
 hold-queue 60 out
!
```

What to Do Next

Proceed to the "Enabling and Accessing the NAM Traffic Analyzer" section on page 28.

Enabling and Accessing the NAM Traffic Analyzer

This section describes how to enable and access the NAM Traffic Analyzer (web GUI).

Prerequisites

- Make sure that your web browser supports your NAM software release. For a list of supported browsers, refer to the NAM software release notes.

- If you plan to use the HTTP secure server (HTTPs), then you must first download and install the NAM software K9 crypto patch. Until you install the patch, the **ip http secure** commands are disabled. You can download the NAM software K9 crypto patch from Cisco.com.

Restrictions

You can use the HTTP server or the HTTP secure server, but you cannot use both simultaneously.

SUMMARY STEPS

1. Open a NAM console session from the router. See the "Opening and Closing a NAM Console Session from the Router" section on page 18.
   or
   Open a Telnet or SSH session to the NAM. See the "Opening and Closing a Telnet or SSH Session to the NAM" section on page 38.
2. **ip http server enable**
   or
   ip http secure server enable
3. Enter a web username.
   or
   Press **Return** to enter the default web username "admin".
4. Enter a password.
5. Enter the password again.
6. On your PC, open a web browser.
7. In the web browser, enter the NAM system IP address or hostname as the URL.

DETAILED STEPS

	Command or Action	Purpose
Step 1	Open a NAM console session from the router. See the "Opening and Closing a NAM Console Session from the Router" section on page 18. or Open a Telnet or SSH session to the NAM. See the "Opening and Closing a Telnet or SSH Session to the NAM" section on page 38.	Accesses the NAM CLI.

Step 2	**`ip http server enable`**
	or
	`ip http secure server enable`
	Example:
	`root@localhost# ip http server enable`
	Example:
	`root@localhost# ip http secure server enable`
		Enables the HTTP server.
		or
		Enables the HTTP secure server (HTTPs).

	Command or Action	Purpose

Step 3	Enter a web username.
	or
	Press **Return** to enter the default web username "admin".
	Example:
	`Please enter a web administrator user name [admin]: joeadmin`
	Example:
	`Please enter a web administrator user name [admin]: <cr>`
		Configures a web username.
		• The NAM requires at least one web username and password configuration.
		• If NAM does not prompt you for a web username and password, then at least one web username and password combination was previously configured.

Step 4	Enter a password. **Example:** `New password: <adminpswd>`	Configures a password for the web username.
Step 5	Enter the password again. **Example:** `Confirm password: <adminpswd>`	Confirms the password for the web username.
Step 6	On your PC, open a web browser.	—

Step 7	In the web browser, enter the NAM system IP address or hostname as the URL.
	Example:
	`http://172.20.105.215/`
	Example:
	`https://172.20.105.215/`
	Example:
	`http://nam1/`
		Opens the NAM Traffic Analyzer in your web browser.
		• You are automatically redirected to the NAM Traffic Analyzer login page.

Examples

This section provides the following examples:

- Enabling the NAM Traffic Analyzer: Example, page 30
- Accessing the NAM Traffic Analyzer: Example, page 31

Enabling the NAM Traffic Analyzer: Example

```
root@nam1# ip http server enable
Enabling HTTP server...

No web users are configured.
Please enter a web administrator user name [admin]: <cr>
```

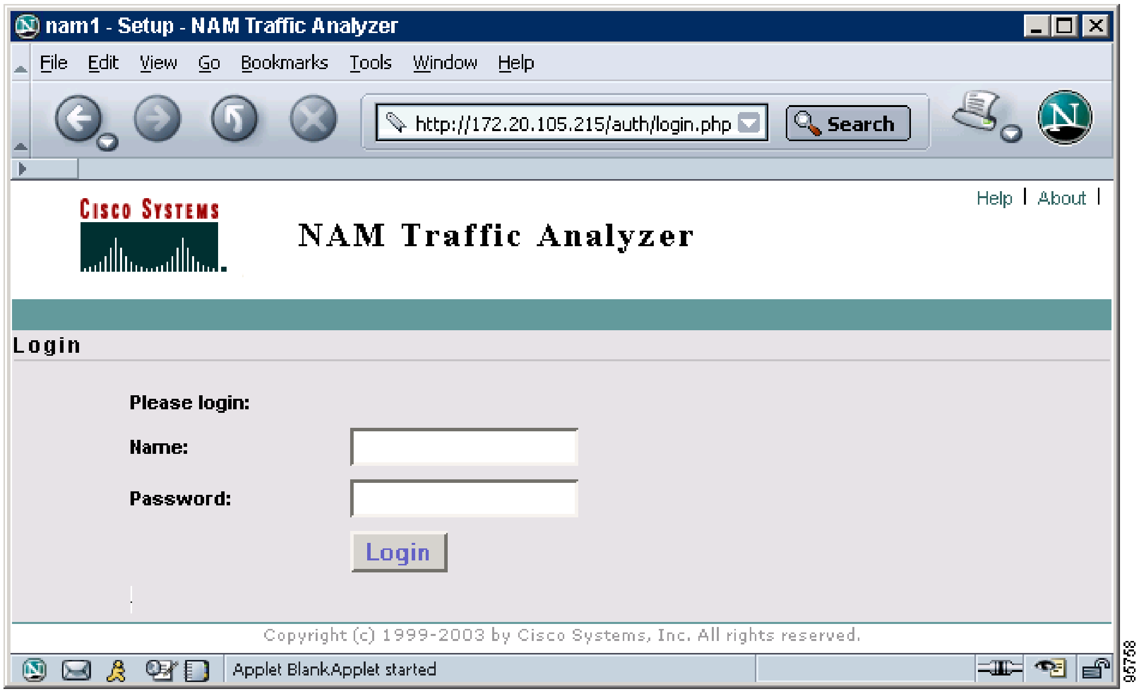
```
New password: <pswd>
Confirm password: <pswd>

User admin added.
Successfully enabled HTTP server.
root@nam1#
```

Accessing the NAM Traffic Analyzer: Example

Figure 6 shows the NAM Traffic Analyzer login page that appears when you enter the NAM system IP address or hostname as the URL in a web browser.

Figure 6 Sample NAM Traffic Analyzer Login Page



Please login:

Name:

Password:

Copyright (c) 1999-2003 by Cisco Systems, Inc. All rights reserved.

Applet BlankApplet started

What to Do Next

For information on the NAM Traffic Analyzer, refer to the *User Guide for the Network Analysis Module Traffic Analyzer* for your NAM software release. This document is available on Cisco.com and as online help within the NAM Traffic Analyzer application.

Changing the NAM Root Password

This section describes how to set a new password to access the root (read/write) level of NAM, where you can enter NAM CLI commands. The factory-set default root password is “root”.

Prerequisites

Before performing this task, access the NAM console by performing Step 1 through Step 5 in the “Opening and Closing a NAM Console Session from the Router” section on page 18.

SUMMARY STEPS

1. **password root**
2. Enter the new password.
3. Enter the new password again.
4. **exit**
5. At the login prompt, enter **root**.
6. At the password prompt, enter your password.

DETAILED STEPS

	Command or Action	Purpose

Step 1	**`password root`**
	Example:
	`root@localhost.cisco.com# password root`
		Starts the process of changing the NAM's root (read/write) level password.

Step 2	Enter the new password. **Example:** `New UNIX password: <password>`	Enters the new password.
Step 3	Enter the new password again. **Example:** `Retype new UNIX password: <password>`	Confirms the new password.

Step 4	**`exit`**
	Example:
	`root@localhost# exit`
		Logs out of the NAM system.

Step 5	At the login prompt, enter **root**. **Example:** `login: root`	Accesses the root (read/write) level of NAM.
Step 6	At the password prompt, enter your password. **Example:** `Password: <password>`	Verifies that the new password is accepted.

Examples

This section provides the following examples:

- Changing the NAM Root Password: Example, page 33
- Verifying the NAM Root Password: Example, page 34

Changing the NAM Root Password: Example

```
root@nam1.cisco.com# password root
Changing password for user root
New UNIX password: <rtpswd>
Retype new UNIX password: <rtpswd>
passwd:all authentication tokens updated successfully
root@nam1.cisco.com#
root@nam1.cisco.com# exit
```

Verifying the NAM Root Password: Example

```
nam1.cisco.com login: root
Password: <rtpswd>
Terminal type: vt100

Cisco Network Analysis Module (NM-NAM) Console, 3.2
Copyright (c) 1999-2003 by cisco Systems, Inc.

root@nam1.cisco.com#
root@nam1.cisco.com# exit
```

Troubleshooting Tips

If you forget the NAM root password, see the "Resetting the NAM Root Password to the Default Value" section on page 34.

Resetting the NAM Root Password to the Default Value

This section describes how to reset the NAM root password to the default value of "root". Use this procedure when you cannot remember the NAM root password but need to access the NAM CLI.

Note This procedure requires that you reload the NAM software.

SUMMARY STEPS

1. **enable**
2. **service-module analysis-module** *slot*/**0 reload**
3. **y**
4. **service-module analysis-module** *slot*/**0 session**
5. When prompted, enter ******* to change the boot configuration.
6. **boot flash**
7. When prompted to select from the helper menu, enter **6**.
8. When prompted to select from the helper menu, enter **r**.
9. **y**
10. Hold **Ctrl**-**Shift** and press **6**. Release all keys, and then press **x**.
11. **disconnect**
12. Press **Enter**.

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**service-module analysis-module** *slot*/**0 reload** **Example:** `Router# service-module analysis-module 1/0 reload`	Reloads the software on the NM-NAM.
Step 3	**y** **Example:** `Do you want to proceed with reload?[confirm] y`	Confirms that you want to proceed with the NAM software reload.

Step 4	**service-module analysis-module** *slot*/**0 session**
	Example:
	`Router# service-module analysis-module 1/0 session`
	Example:
	`Router# service-module analysis-module 1/0 session clear`
	`[confirm]`
	` [OK]`
	`Router# service-module analysis-module 1/0 session`
		Establishes a console session with the NAM.
		• Perform this step immediately after reloading the NAM software.
		• If you cannot open a NAM console session, make sure that the NAM console line is clear by first entering the **service-module analysis-module** *slot*/**0 session clear** command in privileged EXEC mode.

Step 5	When prompted, enter *** to change the boot configuration. **Example:** `Please enter '***' to change boot configuration: ***`	Interrupts the boot loader. • Enter *** immediately after the prompt appears. • If you do not enter *** in time to interrupt the boot loader, then the NAM login prompt eventually appears. Complete Step 10 through Step 12 to return to the Cisco IOS CLI on the router, and then retry this task, starting with Step 2.
Step 6	**boot flash** **Example:** `ServicesEngine boot-loader> boot flash`	Loads the NAM helper image. • This command is entered in the boot loader CLI, which is separate from the NAM CLI and Cisco IOS CLI.
Step 7	When prompted to select from the helper menu, enter **6**. **Example:** `Selection [12345678rh]: 6`	Selects the menu option to reset the root password to the default value of "root".

	Command or Action	Purpose
Step 8	When prompted to select from the helper menu, enter **r**. **Example:** `Selection [12345678rh]:r`	Selects the menu option to exit the helper and reset the NAM.

Step 9	**`y`**
	Example:
	`About to exit and reset Services Engine.`
	`Are you sure? [y/N] y`
		Confirms that you want to exit the helper and reset the NAM.
		• This time, ignore the prompt to enter ***.

Step 10	Hold **Ctrl-Shift** and press **6**. Release all keys, and then press **x**.
	Example:
	`login: <suspend keystroke>`
	`Router#`
		Suspends and closes the Telnet session.

Step 11	**`disconnect`**
	Example:
	`Router# disconnect`
		Disconnects a line.

Step 12	Press **Enter**.
	Example:
	`Closing connection to 10.20.30.40 [confirm]`
	`<Enter>`
		Confirms that you want to disconnect the line.

Example

This section provides the following example:

- Resetting the NAM Root Password to the Default Value: Example, page 36

Resetting the NAM Root Password to the Default Value: Example

```
Router# service-module analysis-module 1/0 reload
Do you want to proceed with reload?[confirm] y
Trying to reload Service Module Analysis-Module1/0.

Router# service-module analysis-module 1/0 session
Trying 172.20.104.87, 2033 ... Open
.
<debug output omitted>
.
Booting from flash..., please wait.

[BOOT-ASM]
7

Please enter '***' to change boot configuration: ***

 ServicesEngine Bootloader Version :1.0.6aN
ServicesEngine boot-loader> boot flash
.
<debug output omitted>
.
==============================================================================
Cisco Systems, Inc.
Services engine helper utility for NM-NAM
Version 1.1(1) [200311111641]

-----
Main menu
1 - Download application image and write to HDD
2 - Download application image and reformat HDD
3 - Download bootloader and write to flash
4 - Download helper and write to flash
5 - Display software versions
6 - Reset application image CLI passwords to default
7 - Change file transfer method (currently ftp/http)
8 - Show upgrade log
9 - Send Ping
r - Exit and reset Services Engine
h - Exit and shutdown Services Engine

Selection [123456789rh]: 6
Restored default CLI passwords of application image.
==============================================================================
Cisco Systems, Inc.
Services engine helper utility for NM-NAM
Version 1.1(1) [200311111641]

-----
Main menu
1 - Download application image and write to HDD
2 - Download application image and reformat HDD
3 - Download bootloader and write to flash
4 - Download helper and write to flash
5 - Display software versions
6 - Reset application image CLI passwords to default
7 - Change file transfer method (currently ftp/http)
8 - Show upgrade log
9 - Send Ping
r - Exit and reset Services Engine
h - Exit and shutdown Services Engine

Selection [123456789rh]: r
About to exit and reset Services Engine.
Are you sure? [y/N] y
INITSending all processes the TERM signal...
Sending all processes the KILL signal...
Unmounting file systems:
Please stand by while rebooting the system...
Restarting system.
.
<debug output omitted>
.
Cisco Network Analysis Module (NM-NAM)

nam1.cisco.com login: <suspend keystroke>
Router#
Router# disconnect
Closing connection to 10.1.1.1 [confirm] <Enter>
Deleting login session
```

Troubleshooting Tips

If you have trouble opening a NAM console session from the router, make sure that the NAM console line is clear by entering the **service-module analysis-module** *slot*/**0 session clear** command in privileged EXEC mode.

What to Do Next

Verify that the default root password of "root" is accepted by performing Step 1 through Step 5 in the "Opening and Closing a NAM Console Session from the Router" section on page 18.

To change the NAM root password, see the "Changing the NAM Root Password" section on page 31.

Opening and Closing a Telnet or SSH Session to the NAM

This section describes how to open and close a Telnet or SSH session to the NAM. This task is not commonly performed, because you would typically use the NAM Traffic Analyzer (web GUI) to monitor and maintain the NAM. If, however, you cannot access the NAM Traffic Analyzer, then you might want to use Telnet or SSH to troubleshoot from the NAM CLI.

If your NM-NAM is not properly configured for Telnet or SSH access (see the following Prerequisites section), then you can open a Telnet session to the router in which the NM-NAM is installed, and then open a NAM console session from the router. See the "Opening and Closing a NAM Console Session from the Router" section on page 18.

Prerequisites

- Configure the NAM system IP address. Optionally, set the NAM system hostname. See the "Configuring the NM-NAM" section on page 21.
- Verify NAM network connectivity by performing one of the following ping tests:
  - From a host beyond the gateway, ping the NAM system IP address.
  - From the NAM CLI, ping the NAM system default gateway.

Telnet Prerequisites

- Enter the **exsession on** NAM CLI command. See Step 5 of the "Configuring the NM-NAM" section on page 21.

SSH Prerequisites

- Install the NAM software K9 crypto patch, which you can download from Cisco.com.
- Enter the **exsession on ssh** NAM CLI command. See Step 5 of the "Configuring the NM-NAM" section on page 21.

SUMMARY STEPS

1. **telnet** {*ip-address* | *hostname*}
   or
   ssh {*ip-address* | *hostname*}
2. At the login prompt, enter **root**.

3. At the password prompt, enter your password.
   or
   If you have not changed the password from the factory-set default, enter **root** as the root password.
4. Perform the tasks that you need to perform in the NAM CLI. When you want to end the Telnet or SSH session to the NAM and return to the Cisco IOS CLI, complete Step 5 and Step 6.
5. **exit**
6. **logout**

DETAILED STEPS

	Command or Action	Purpose

Step 1	`telnet {ip-address | hostname}`
	or
	`ssh {ip-address | hostname}`
	Example:
	`Router# telnet 10.20.30.40`
	Example:
	`Router# ssh 10.20.30.40`
		Logs in to a host that supports Telnet.
		or
		Starts an encrypted session with a remote networking device.
		• Use the NAM system IP address or NAM system hostname.

Step 2	At the login prompt, enter **root**. **Example:** `login: root`	Accesses the root (read/write) level of NAM.
Step 3	At the password prompt, enter your password. or If you have not changed the password from the factory-set default, enter **root** as the root password. **Example:** `Password: root`	—
Step 4	Perform the tasks that you need to perform in the NAM CLI. When you want to end the Telnet or SSH session to the NAM and return to the Cisco IOS CLI, complete Step 5 and Step 6.	For help using NAM CLI commands, see the "NAM CLI Context-Sensitive Help" section on page 12.

	Command or Action	Purpose

Step 5	**exit**
	Example:
	`root@localhost(sub-custom-filter-capture)# exit`
	`root@localhost#`
		Leaves a subcommand mode.
		• Return to command mode.

Step 6	**logout**
	Example:
	`root@localhost# logout`
	`Connection closed by foreign host.`
		Logs out of the NAM system.

Examples

This section provides the following examples:

- Opening and Closing a Telnet Session to the NAM Using the NAM System IP Address: Example, page 40
- Opening and Closing an SSH Session to the NAM Using the NAM System Hostname: Example, page 40

Opening and Closing a Telnet Session to the NAM Using the NAM System IP Address: Example

```
Router> telnet 172.20.105.215
Trying 172.20.105.215 ... Open

Cisco Network Analysis Module (NM-NAM)

login: root
Password: <password>
Terminal type: vt100

Cisco Network Analysis Module (NM-NAM) Console, 3.2
Copyright (c) 1999-2003 by cisco Systems, Inc.

WARNING! Default password has not been changed!
root@nam.cisco.com#
root@nam.cisco.com# logout

[Connection to 172.20.105.215 closed by foreign host]
Router>
```

Opening and Closing an SSH Session to the NAM Using the NAM System Hostname: Example

```
host [/home/user] ssh -l root nmnam2
root@nmnam2’s password: <password>
Terminal type: vt100

Cisco Network Analysis Module (NM-NAM) Console, 3.2
Copyright (c) 1999-2003 by cisco Systems, Inc.

WARNING! Default password has not been changed!
root@nmnam2.cisco.com#
root@nmnam2.cisco.com# logout

Connection to nmnam2 closed.
host [/home/user]
```

Upgrading the NAM Software

This section describes how to upgrade the NAM software. This task is performed from the NAM CLI.

NAM Software Images

The NM-NAM contains three NAM software images:

- NAM application image on the hard drive—Source of the NAM Traffic Analyzer and NAM CLI
- Helper image in flash memory—Used to recover or upgrade NAM software images
- Bootloader image in flash memory—Used to specify whether to boot the NAM application image or the helper image

Types of NAM Software Upgrades

NAM software upgrades are available in two forms:

- Patches—Incremental updates to software releases that are installed with the **patch** NAM CLI command. Patches are available only for the NAM application image.
- Images—Full image releases that are installed from the helper image. Full image upgrades are typically used to update the NAM application image, but if necessary and recommended by technical support, you can also use the helper image to upgrade the bootloader image or helper image.

Prerequisites

- Download the NAM software image from Cisco.com, and copy the image to an FTP server.
- Before performing this task, access the NAM console by completing Step 1 through Step 5 in the "Opening and Closing a NAM Console Session from the Router" section on page 18.

Perform one of the following tasks in this section, depending on whether you are adding a patch to your NAM application or are performing a full software image upgrade:

- Upgrading the NAM Software—Patch, page 41
- Upgrading the NAM Software—Full Image, page 42

Upgrading the NAM Software—Patch

Perform this task to add a patch to your NAM application image. This task is performed from the NAM CLI.

SUMMARY STEPS

1. **patch** *ftp://user:passwd@host/full-path/filename*
   or
   patch *ftp://user@host/full-path/filename*
2. **show patches**

DETAILED STEPS

	Command or Action	Purpose

Step 1	**patch** *ftp://user:password@host/full-path/filename*
	or
	patch *ftp://user@host/full-path/filename*
	Example:
	`root@nam1.cisco.com# patch ftp://person:mypwd@examplehost/dir/subdir/nam-app.3-2.cryptoK9.patch.1-0.bin`
	Example:
	`root@nam1.cisco.com# patch ftp://person@examplehost/dir/subdir/nam-app.3-2.cryptoK9.patch.1-0.bin`
	`Proceeding with installation. Please do not interrupt.`
	`If installation is interrupted, please try again.`
	`Downloading nam-app.3-2.cryptoK9.patch.1-0.bin. Please wait...`
	`Password for person@examplehost: <mypwd>`
		Downloads and installs a software patch.
		• Use the first option, which includes the password, if the FTP server does not allow anonymous users.
		• If you use the second option, enter your password when prompted.
		• Remember to perform this task in the NAM CLI.

Step 2	**show patches** **Example:** `root@nam1.cisco.com# show patches`	Displays all installed patches. • Verify that your patch was successfully installed.

Upgrading the NAM Software—Full Image

Perform this task to upgrade one of your NAM software images to a new release. This task is performed from the NAM CLI.

SUMMARY STEPS

1. **reboot**
2. **y**
3. When prompted, enter *** to change the boot configuration.
4. **boot flash**
5. When prompted to select from the helper menu, enter **1**.
6. **ftp://***ip-address***/***path***/***nam-image-file*
7. **y**
8. **r**
9. **y**

DETAILED STEPS

	Command or Action	Purpose

Step 1	**`reboot`**
	Example:
	`root@nam1.cisco.com# reboot`
		Shuts down and restarts the NAM.
		• Remember to perform this task in the NAM CLI.

Step 2	**`y`**
	Example:
	`Reboot the NAM? (Y/N) [N]: y`
		Confirms that you want to reboot the NAM.
		• After you confirm the reboot, the NAM displays a series of messages as it stops processes, shuts down, and then restarts.

Step 3	When prompted, enter *** to change the boot configuration. **Example:** `Please enter '***' to change boot configuration: ***`	Interrupts the boot loader. • Enter *** immediately after the prompt appears. • If you do not enter the *** in time to interrupt the boot loader, then return to Step 1 and try again.

Step 4	**`boot flash`**
	Example:
	`ServicesEngine boot-loader> boot flash`
		Loads the NAM helper image.
		• This command is entered in the boot loader CLI, which is separate from the NAM CLI and Cisco IOS CLI.

Step 5	When prompted to select from the helper menu, enter **1** or **2**.
	Example:
	`Selection [12345678rh]: 1`
	Example:
	`Selection [12345678rh]: 2`
		Selects the menu option to download the NAM software image onto the NM-NAM internal memory.
		• Option 1 preserves all configuration and report data while installing the NAM software image.
		• Option 2 reformats the NM-NAM hard drive, deleting all report data and NAM software configurations, except the basic IP configuration. Although useful for recovering a corrupted hard drive, Option 2 should be used with caution or when recommended by technical support.
		• The helper menu also has an option (7) to change the file transfer method from the default FTP method. Before performing Step 5, you may enter 7 to select the TFTP transfer method. Because many TFTP servers have problems transferring files as large as the NAM application image, we recommend that you use the default FTP method.

Step 6	**`ftp://`***`ip-address`***`/`***`path`***`/`***`nam-image-file`*
	Example:
	`Download NAM application image via ftp and write to HDD`
	`URL of application image []: ftp://172.20.98.136/dir1/dir2/nam-image.bin.gz`
		Specifies the FTP location and filename of the NAM software image.

	Command or Action	Purpose
Step 7	**y** **Example:** `Do you want to proceed installing it? [y/N] y`	Confirms that you want to install the specified NAM software image.
Step 8	**r** **Example:** `Selection [12345678rh]:r`	Selects the menu option to exit the helper and reset the NAM.

Step 9	**y**
	Example:
	`About to exit and reset Services Engine.`
	`Are you sure? [y/N] y`
		Confirms that you want to exit the helper and reset the NAM.
		• This time, ignore the prompt to enter ***.

Examples

This section provides the following examples:

- Upgrading the NAM Software—Patch: Example, page 44
- Upgrading the NAM Software—Full Image: Example, page 45

Upgrading the NAM Software—Patch: Example

```
Router> enable
Password: <password>
Router#
Router# service-module analysis-Module 1/0 session
Trying 172.20.104.86, 2033 ... Open

Cisco Network Analysis Module (NM-NAM)

nam1.cisco.com login: root
Password: <password>
Terminal type:vt100

Cisco Network Analysis Module (NM-NAM) Console, 3.2(0.10)
Copyright (c) 1999-2003 by cisco Systems, Inc.

WARNING! Default password has not been changed!

root@nam1.cisco.com# patch
ftp://person@examplehost/dir/subdir/nam-app.3-2.cryptoK9.patch.1-0.bin

Proceeding with installation. Please do not interrupt.
If installation is interrupted, please try again.

Downloading nam-app.3-2.cryptoK9.patch.1-0.bin. Please wait...
Password for person@examplehost: <mypwd>
ftp://person@examplehost/dir/subdir/nam-app.3-2.cryptoK9.patch.1-0.bin
(1K)
/usr/local/nam/patch/wor  [########################]       1K |  104.43K/s
1894 bytes transferred in 0.02 sec (102.35k/sec)

Verifying nam-app.3-2.cryptoK9.patch.1-0.bin. Please wait...
Patch nam-app.3-2.cryptoK9.patch.1-0.bin verified.
```

```
Applying /usr/local/nam/patch/workdir/nam-app.3-2.cryptoK9.patch.1-0.bin.
Please wait...
########################################### [100%]
########################################### [100%]

Patch applied successfully.
root@nam1.cisco.com# show patches

Tue Aug 31 21:04:28 2004 Patch:nam-app.3-2.strong-crypto-patchK9-1-0
Description:Strong Crypto Patch for NAM.

root@nam1.cisco.com#
```

Upgrading the NAM Software—Full Image: Example

```
Router> enable
Password: <password>
Router#
Router# service-module analysis-Module 1/0 session
Trying 172.20.104.86, 2033 ... Open

Cisco Network Analysis Module (NM-NAM)

nam1.cisco.com login: root
Password: <password>
Terminal type:vt100

Cisco Network Analysis Module (NM-NAM) Console, 3.2(0.10)
Copyright (c) 1999-2003 by cisco Systems, Inc.

WARNING! Default password has not been changed!

root@nam1.cisco.com#
root@nam1.cisco.com# reboot
Reboot the NAM? (Y/N) [N]: y

System reboot in process...
.
<debug output omitted>
.
Booting from flash..., please wait.

[BOOT-ASM]
7

Please enter '***' to change boot configuration: ***

 ServicesEngine Bootloader Version :1.0.6-NAM

ServicesEngine boot-loader>
ServicesEngine boot-loader> boot flash
.
<debug output omitted>
.
==============================================================================
Cisco Systems, Inc.
Services engine helper utility for NM-NAM
Version 1.1(1) [200311111641]

-----
Main menu
1 - Download application image and write to HDD
2 - Download application image and reformat HDD
3 - Download bootloader and write to flash
```

```
4 - Download helper and write to flash
5 - Display software versions
6 - Reset application image CLI passwords to default
7 - Change file transfer method (currently ftp/http)
8 - Show upgrade log
9 - Send Ping
r - Exit and reset Services Engine
h - Exit and shutdown Services Engine

Selection [123456789rh]: 1


-----
Download NAM application image via ftp and write to HDD
URL of application image []: ftp://172.20.98.136/dir1/dir2/nam-image.bin.gz
Getting c6svc-nam.mainline-DAILY_20030825.bin.gz from 171.69.17.19 via ftp.
ftp://172.20.98.136/dir1/dir2/nam-image.bin.gz
(46389K)
-                         [########################]   46389K | 7421.38K/s
47502347 bytes transferred in 6.25 sec (7421.14k/sec)
upgrade.bin size:48241545
File transfer successful.
Checking upgrade.bin
Do you want to proceed installing it? [y/N] y
.
<debug output omitted>
.
Application image upgrade complete. You can boot the image now.
==============================================================================
Cisco Systems, Inc.
Services engine helper utility for NM-NAM
Version 1.1(1) [200311111641]

-----
Main menu
1 - Download application image and write to HDD
2 - Download application image and reformat HDD
3 - Download bootloader and write to flash
4 - Download helper and write to flash
5 - Display software versions
6 - Reset application image CLI passwords to default
7 - Change file transfer method (currently ftp/http)
8 - Show upgrade log
9 - Send Ping
r - Exit and reset Services Engine
h - Exit and shutdown Services Engine

Selection [123456789rh]: r
About to exit and reset Services Engine.
Are you sure? [y/N] y
```

Troubleshooting Tips

If you have trouble opening a NAM console session from the router, make sure that the NAM console line is clear by entering the **service-module analysis-module** *slot*/**0 session clear** command in privileged EXEC mode.

Configuration Examples for the Network Analysis Module (NM-NAM)

This section provides the following configuration examples:

- NAM Management Interface Is Internal and Analysis-Module Interface Is Assigned an IP Address: Example, page 47
- NAM Management Interface Is Internal and Analysis-Module Interface Is IP Unnumbered: Example, page 49
- NAM Management Interface Is External and Analysis-Module Interface Is IP Unnumbered: Example, page 51

NAM Management Interface Is Internal and Analysis-Module Interface Is Assigned an IP Address: Example

In this configuration example:

- The internal NAM interface is used for management traffic.
- IP addresses from the same routable subnet are assigned to the Analysis-Module interface and the NAM system.
- A static route to the NAM through the Analysis-Module interface is configured.
- The internal NAM interface is used to monitor WAN traffic on interface Serial 0/0, and the external NAM interface is used to monitor LAN traffic on interface Fast Ethernet 0/0.
- The NM-NAM is installed in router slot 2.

Figure 7 shows the topology used in the example, and the following sections show the router and NAM configurations:

- Router Configuration (Cisco IOS Software), page 48
- NAM Configuration (NAM Software), page 49

Figure 7 NAM Management Interface Is Internal and Analysis-Module Interface Is Assigned an IP Address

Figure 7 Callout	Interface	Location
1	Analysis-Module interface	Router internal
2	Internal NAM interface (**management**)	NM-NAM internal
3	External NAM interface	NM-NAM faceplate
4	Serial interface	WAN interface card (WIC)
5	Fast Ethernet interface	Router rear panel

Router Configuration (Cisco IOS Software)

```
!
ip cef
!
ip route 209.165.200.226 255.255.255.224 analysis-module 2/0
!
interface FastEthernet0/0
 ip address 209.165.202.129 255.255.255.224
 ip route-cache flow
 speed auto
 full-duplex
 no mop enabled
 no shutdown
!
interface Serial 0/0
 encapsulation ppp
 ip address 209.165.201.1 255.255.255.224
 analysis-module monitoring
 no shutdown
```

```
!
interface analysis-module 2/0
 ip address 209.165.200.225 255.255.255.224
 hold-queue 60 out
 no shutdown
!
```

NAM Configuration (NAM Software)

```
!
ip address 209.165.200.226 255.255.255.224
!
ip host "nam1"
!
ip domain "cisco.com"
!
ip gateway 209.165.200.225
!
ip broadcast 10.255.255.255
!
ip nameserver 172.16.201.29
!
ip interface internal
!
ip http server enable
!
exsession on
!
```

NAM Management Interface Is Internal and Analysis-Module Interface Is IP Unnumbered: Example

In this configuration example:

- The internal NAM interface is used for management traffic.
- IP addresses from the same routable subnet are assigned to the Analysis-Module interface and the NAM system.
- To conserve IP address space, the Analysis-Module interface is configured as IP unnumbered to borrow the IP address of the Fast Ethernet interface.
- A static route to the NAM through the Analysis-Module interface is configured.
- The internal NAM interface is used to monitor WAN traffic on interface Serial 0/0, and the external NAM interface is used to monitor LAN traffic on interface Fast Ethernet 0/0.
- The NM-NAM is installed in router slot 2.

Figure 8 shows the topology used in the example, and the following sections show the router and NAM configurations:

- Router Configuration (Cisco IOS Software), page 50
- NAM Configuration (NAM Software), page 51

Figure 8 Sample Topology: NAM Management Interface Is Internal and Analysis-Module Interface Is IP Unnumbered

Figure 8 Callout	Interface	Location
1	Analysis-Module interface	Router internal
2	Internal NAM interface (**management**)	NM-NAM internal
3	External NAM interface	NM-NAM faceplate
4	Serial interface	WAN interface card (WIC)
5	Fast Ethernet interface	Router rear panel

Router Configuration (Cisco IOS Software)

```
!
ip cef
!
ip route 209.165.200.226 255.255.255.224 analysis-module 2/0
!
interface FastEthernet0/0
 ip address 209.165.200.225 255.255.255.224
 ip route-cache flow
 speed auto
 full-duplex
 no mop enabled
 no shutdown
!
interface Serial 0/0
 encapsulation ppp
 ip address 209.165.201.1 255.255.255.224
 analysis-module monitoring
 no shutdown
```

```
!
interface analysis-module 2/0
 ip unnumbered FastEthernet0/0
 no shutdown
 hold-queue 60 out
!
```

NAM Configuration (NAM Software)

```
!
ip address 209.165.200.226 255.255.255.224
!
ip host "nam1"
!
ip domain "cisco.com"
!
ip gateway 209.165.200.225
!
ip broadcast 10.255.255.255
!
ip nameserver 172.16.201.29
!
ip interface internal
!
ip http server enable
!
exsession on
!
```

NAM Management Interface Is External and Analysis-Module Interface Is IP Unnumbered: Example

In this configuration example:

- The external NAM interface is used for management traffic.
- The Analysis-Module interface is configured as IP unnumbered to borrow the IP address of the loopback interface.
- The borrowed loopback interface IP address is not routable.
- The NAM system is configured with an IP address from the LAN subnet that is connected to the external NAM interface.
- The internal NAM interface is used to monitor WAN traffic on interface Serial 0/0, and the external NAM interface is used to monitor LAN traffic on interface Fast Ethernet 0/0.
- The NM-NAM is installed in router slot 3.

Figure 9 shows the topology used in the example, and the following sections show the router and NAM configurations:

- Router Configuration (Cisco IOS Software), page 52
- NAM Configuration (NAM software), page 53

Figure 9 Sample Topology: NAM Management Interface Is External and Analysis-Module Interface Is IP Unnumbered

Figure 9 Callout	Interface	Location
1	Analysis-Module interface	Router internal
2	Internal NAM interface	NM-NAM internal
3	External NAM interface (**management**)	NM-NAM faceplate
4	Loopback interface	Router internal
5	Serial interface	WAN interface card (WIC)
6	Fast Ethernet interface	Router rear panel

Router Configuration (Cisco IOS Software)

```
!
ip cef
!
interface loopback 0
 ip address 10.1.1.1 255.255.255.0
!
interface FastEthernet0/0
 ip address 209.165.201.1 255.255.255.224
 ip route-cache flow
 speed auto
 full-duplex
 no mop enabled
 no shutdown
!
interface Serial 0/0
 encapsulation ppp
 ip address 209.165.202.129 255.255.255.224
 analysis-module monitoring
 no shutdown
!
interface analysis-module 3/0
 ip unnumbered loopback 0
 hold-queue 60 out
 no shutdown
!
```

NAM Configuration (NAM software)

```
!
ip address 209.165.201.2 255.255.255.224
!
ip host "nam1"
!
ip domain "cisco.com"
!
ip gateway 209.165.201.1
!
ip broadcast 10.255.255.255
!
ip nameserver 209.165.201.29
!
ip interface external
!
ip http server enable
!
exsession on
!
```

Additional References

The following sections provide references related to the Network Analysis Module (NM-NAM) feature.

Related Documents

Related Topic	Document Title
Compatibility matrixes for NAM software releases, Cisco IOS releases, and platforms Links to software downloads, product documentation, and technical documentation, including NAM software release notes, user guide, and command reference	Cisco Network Analysis Module (NAM)
Installing and cabling network modules	*Cisco Network Modules Hardware Installation Guide*
Safety and compliance	*Cisco Network Modules and Interface Cards Regulatory Compliance and Safety Information*
Cisco IOS interface commands: complete command syntax, command mode, command history, defaults, usage guidelines, and examples	*Cisco IOS Interface and Hardware Component Command Reference*
Router documentation	Modular Access Routers
IP unnumbered interfaces	*Understanding and Configuring the ip unnumbered Command*
Authentication, authorization, and accounting (AAA)	*Cisco IOS Security Configuration Guide*

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
Router MIBs: • CISCO-ENTITY-VENDORTYPE-OID-MIB Network Analysis Module (NAM) MIBs: • ART-MIB • DSMON-MIB • HC-RMON-MIB • MIB-II • RMON-MIB • RMON2-MIB • SMON-MIB	To locate and download MIBs for selected platforms, Cisco IOS releases, and feature sets, use Cisco MIB Locator found at the following URL: http://www.cisco.com/go/mibs

RFCs

RFCs	Title
RFC 2021	*Remote Network Monitoring Management Information Base Version 2 using SMIv2*
RFC 2074	*Remote Network Monitoring MIB Protocol Identifiers*
RFC 2613	*Remote Network Monitoring MIB Extensions for Switch Networks Version 1.0*
RFC 2819	*Remote Network Monitoring Management Information Base*
RFC 3273	*Remote Network Monitoring Management Information Base for High Capacity Networks*
RFC 3287	*Remote Monitoring MIB Extensions for Differentiated Services*

Technical Assistance

Description	Link
Technical Assistance Center (TAC) home page, containing 30,000 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/public/support/tac/home.shtml

Command Reference

The following new commands are pertinent to this feature. To see the command pages for these commands and other commands used with this feature, go to the *Cisco IOS Master Commands List*, Release 12.4, at http://www.cisco.com/univercd/cc/td/doc/product/software/ios124/124mindx/124index.htm.

- **analysis-module monitoring**
- **interface analysis-module**
- **service-module analysis-module reload**
- **service-module analysis-module reset**
- **service-module analysis-module session**
- **service-module analysis-module shutdown**
- **service-module analysis-module status**
- **show controllers analysis-module**
- **show interfaces analysis-module**

Glossary

AAA—authentication, authorization, and accounting. Pronounced “triple a.”

access list—A list kept by routers to control access to or from the router for a number of services (for example, to prevent packets with a certain IP address from leaving a particular interface on the router).

CEF—Cisco Express Forwarding.

DSMON—Differentiated Services Monitoring.

flooding—Traffic passing technique used by switches and bridges in which traffic received on an interface is sent out all the interfaces of that device except the interface on which the information was received originally.

GRE—generic routing encapsulation. Tunneling protocol developed by Cisco that can encapsulate a wide variety of protocol packet types inside IP tunnels, creating a virtual point-to-point link to Cisco routers at remote points over an IP internetwork. By connecting multiprotocol subnetworks in a single-protocol backbone environment, IP tunneling using GRE allows network expansion across a single-protocol backbone environment.

GUI—graphical user interface. A user environment that uses pictorial as well as textual representations of the input and the output of applications and the hierarchical or other data structure in which information is stored. Such conventions as buttons, icons, and windows are typical, and many actions are performed using a pointing device (such as a mouse). Microsoft Windows and the Apple Macintosh are prominent examples of platforms using a GUI.

IP multicast—Routing technique that allows IP traffic to be propagated from one source to a number of destinations or from many sources to many destinations. Rather than sending one packet to each destination, one packet is sent to a multicast group identified by a single IP destination group address.

MIB—Management Information Base. Database of network management information that is used and maintained by a network management protocol, such as SNMP or Common Management Information Protocol (CMIP). The value of a MIB object can be changed or retrieved using SNMP or CMIP commands, usually through a GUI network management system. MIB objects are organized in a tree structure that includes public (standard) and private (proprietary) branches.

NAT—Network Address Translation. Mechanism for reducing the need for globally unique IP addresses. NAT allows an organization with addresses that are not globally unique to connect to the Internet by translating those addresses into globally routable address space. Also known as *Network Address Translator.*

NetFlow—A feature of some routers that allows them to categorize incoming packets into flows. Because packets in a flow often can be treated in the same way, this classification can be used to bypass some of the work of the router and accelerate its switching operation.

PCI—Peripheral Component Interconnect. An industry local bus standard.

QoS—quality of service. Cisco IOS QoS technology lets complex networks control and predictably service a variety of networked applications and traffic types.

RMON—remote monitoring. MIB agent specification described in RFC 1271 that defines functions for the remote monitoring of networked devices. The RMON specification provides numerous monitoring, problem detection, and reporting capabilities.

SNMP—Simple Network Management Protocol. Network management protocol used almost exclusively in TCP/IP networks. SNMP provides a means to monitor and control network devices, and to manage configurations, statistics collection, performance, and security. SNMPv2c supports centralized and distributed network management strategies and includes improvements in the Structure of Management Information (SMI), protocol operations, management architecture, and security. SNMPv3 provides secure access to devices by a combination of authenticating and encrypting packets over the network.

SSH—Secure Shell Protocol. A protocol that provides a secure remote connection to a router through a Transmission Control Protocol (TCP) application.

UDP—User Datagram Protocol. Connectionless transport layer protocol in the TCP/IP protocol stack. UDP is a simple protocol that exchanges datagrams without acknowledgments or guaranteed delivery, requiring that error processing and retransmission be handled by other protocols. UDP is defined in RFC 768.

VoIP—Voice over IP. The capability to carry normal telephony-style voice over an IP-based Internet with POTS-like functionality, reliability, and voice quality. VoIP enables a router to carry voice traffic (for example, telephone calls and faxes) over an IP network. In VoIP, the digital signal processor (DSP) segments the voice signal into frames, which then are coupled in groups of two and stored in voice packets. These voice packets are transported using IP in compliance with ITU-T specification H.323.

Note Refer to *Internetworking Terms and Acronyms* for terms not included in this glossary.

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

HTTP Services

HTTP 1.1 Web Server and Client

First Published: November 20, 2006
Last Updated: April 18, 2008

The HTTP 1.1 Web Server and Client feature provides a consistent interface for users and applications by implementing support for HTTP 1.1 in Cisco IOS software-based devices. When combined with the HTTPS feature, the HTTP 1.1 Web Server and Client feature provides a complete, secure solution for HTTP services between Cisco devices.

This module describes the concepts and the tasks related to configuring the HTTP 1.1 Web Server and Client feature.

Finding Feature Information in This Module

Your Cisco IOS software release may not support all of the features documented in this module. For the latest feature information and caveats, see the release notes for your platform and software release. To reach links to specific feature documentation in this module and to see a list of the releases in which each feature is supported, use the "Feature Information for the HTTP 1.1 Web Server and Client" section on page 11.

Finding Support Information for Platforms and Cisco IOS and Catalyst OS Software Images

Use Cisco Feature Navigator to find information about platform support and Cisco IOS and Catalyst OS software image support. To access Cisco Feature Navigator, go to http://www.cisco.com/go/cfn. An account on Cisco.com is not required.

Contents

- Information About the HTTP 1.1 Web Server and Client, page 2
- How to Configure HTTP 1.1 Web Server and Client, page 3
- Configuration Examples for HTTP 1.1 Web Server, page 7
- Where to Go Next, page 8
- Additional References, page 9
- Feature Information for the HTTP 1.1 Web Server and Client, page 11

Americas Headquarters:
Cisco Systems, Inc., 170 West Tasman Drive, San Jose, CA 95134-1706 USA

Information About the HTTP 1.1 Web Server and Client

This feature updates the Cisco implementation of the Hypertext Transfer Protocol (HTTP) from 1.0 to 1.1. The HTTP server allows features and applications, such as the Cisco web browser user interface, to be run on your routing device.

The Cisco implementation of HTTP 1.1 is backward-compatible with previous Cisco IOS releases. If you are currently using configurations that enable the HTTP server, no configuration changes are needed, as all defaults remain the same.

The process of enabling and configuring the HTTP server also remains the same as in previous releases. Support for Server Side Includes (SSIs) and HTML forms has not changed. Additional configuration options, in the form of the **ip http timeout-policy** command and the **ip http max-connections** command, have been added. These options allow configurable resource limits for the HTTP server. If you do not use these optional commands, the default policies are used.

Remote applications may require that you enable the HTTP server before using them. Applications that use the HTTP server include:

- Cisco web browser user interface, which uses the Cisco IOS Homepage Server, HTTP-based EXEC Server, and HTTP IOS File System (IFS) Server
- VPN Device Manager (VDM) application, which uses the VDM Server and the XML Session Manager (XSM)
- QoS Device Manager (QDM) application, which uses the QDM Server
- IP Phone and Cisco IOS Telephony Service applications, which use the ITS Local Directory Search and IOS Telephony Server (ITS)

No Cisco applications use the HTTP Client in Cisco IOS Release 12.2(15)T.

About HTTP Server General Access Policies

The **ip http timeout-policy** command allows you to specify general access characteristics for the server by configuring a value for idle time, connection life, and request maximum. By adjusting these values you can configure a general policy; for example, if you want to maximize throughput for HTTP connections, you should configure a policy that minimizes connection overhead. You can configure this type of policy by specifying large values for the **life** and **request** options so that each connection stays open longer and more requests are processed for each connection.

Another example would be to configure a policy that minimizes the response time for new connections. You can configure this type of policy by specifying small values for the **life** and **request** options so that the connections are quickly released to serve new clients.

A throughput policy would be better for HTTP sessions with dedicated management applications, as it would allow the application to send more requests before the connection is closed, while a response time policy would be better for interactive HTTP sessions, as it would allow more people to connect to the server at the same time without having to wait for connections to become available.

In general, you should configure these options as appropriate for your environment. The value for the **idle** option should be balanced so that it is large enough not to cause an unwanted request or response timeout on the connection, but small enough that it does not hold a connection open longer than necessary.

Access security policies for the HTTP server are configured using the **ip http authentication** command, which allows only selective users to access the server, the **ip http access-class** command, which allows only selective IP hosts to access the server, and the **ip http accounting commands** command, which specifies a particular command accounting method for HTTP server users.

How to Configure HTTP 1.1 Web Server and Client

This section contains the following tasks:

- Configuring the HTTP 1.1 Web Server, page 3
- Configuring the HTTP Client, page 6

Configuring the HTTP 1.1 Web Server

Perform this task to enable the HTTP server and configure optional server characteristics. The HTTP server is disabled by default.

Note If you want to configure authentication (step 4), you must configure the authentication type before you begin configuring the HTTP 1.1 web server.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ip http server**
4. **ip http authentication** {**aaa** | **enable** | **local** | **tacacs**} (optional)
5. **ip http accounting commands** *level* {**default** | *named-accounting-method-list*} (optional)
6. **ip http port** *port-number* (optional)
7. **ip http path** *url* (optional)
8. **ip http access-class** *access-list-number* (optional)
9. **ip http max-connections** *value* (optional)
10. **ip http timeout-policy idle** *seconds* **life** *seconds* **requests** *value* (optional)

DETAILED STEPS

	Command or Action	Purpose

Step 1	`enable`
	Example:
	`Router> enable`
		Enables privileged EXEC mode.
		• Enter your password if prompted.

Step 2	`configure terminal`
	Example:
	`Router# configure terminal`
		Enters global configuration mode.

	Command or Action	Purpose
Step 3	**ip http server** **Example:** `Router(config)# ip http server`	Enables the HTTP 1.1 server, including the Cisco web browser user interface. **Note** If you are enabling the HTTP over Secure Socket Layer (HTTPS) server using the **ip http secure-server** command, you should disable the standard HTTP server using the **no ip http server** command. This command is required to ensure only secure connections to the server.

Step 4	**ip http authentication** {**aaa** | **enable** | **local** | **tacacs**}
	Example:
	`Router(config)# ip http authentication local`
		(Optional) Specifies the authentication method to be used for login when a client connects to the HTTP server. The methods for authentication are:
		aaa—Indicates that the authentication method used for the AAA login service (specified by the **aaa authentication login default** command) should be used for authentication.
		enable—Indicates that the “enable” password should be used for authentication. (This is the default method.)
		local —Indicates that the login user name, password and privilege level access combination specified in the local system configuration (by the **username** global configuration command) should be used for authentication and authorization.
		tacacs—Indicates that the TACACS (or XTACACS) server should be used for authentication.

Step 5	**ip http accounting commands** *level* {**default** | *named-accounting-method-list*} **Example:** `Router(config)# ip http accounting commands 15 default`	(Optional) Specifies a particular command accounting method for HTTP server users. Command accounting for HTTP and HTTPS is automatically enabled when authentication, authorization, and accounting (AAA) is configured on the device. It is not possible to disable accounting for HTTP and HTTPS. HTTP and HTTPS will default to using the global AAA default method list for accounting. The CLI can be used to configure HTTP and HTTPS to use any predefined AAA method list. *level*—Valid privilege level entries are integers from 0 to 15. **default**—Indicates the **default** accounting method list configured by the aaa accounting commands CLI. *named-accounting-method-list*—Indicates the name of the predefined command accounting method list.
Step 6	**ip http port** *port-number* **Example:** `Router(config)# ip http port 8080`	(Optional) Specifies the server port that should be used for HTTP communication (for example, for the Cisco web browser user interface).

	Command or Action	Purpose
Step 7	**ip http path** *url* **Example:** `Router(config)# ip http path slot1:`	(Optional) Sets the base HTTP path for HTML files. The base path is used to specify the location of the HTTP server files (HTML files) on the local system. Generally, the HTML files are located in system flash memory.
Step 8	**ip http access-class** *access-list-number* **Example:** `Router(config)# ip http access-class 20`	(Optional) Specifies the access list that should be used to allow access to the HTTP server.
Step 9	**ip http max-connections** *value* **Example:** `Router(config)# ip http max-connections 10`	(Optional) Sets the maximum number of concurrent connections to the HTTP sever that will be allowed. The default value is 5.
Step 10	**ip http timeout-policy idle** *seconds* **life** *seconds* **requests** *value* **Example:** `Router(config)# ip http timeout-policy idle 30 life 120 requests 100`	(Optional) Sets the characteristics that determine how long a connection to the HTTP server should remain open. The characteristics are: **idle**—The maximum number of seconds the connection will be kept open if no data is received or response data cannot be sent out on the connection. Note that a new value may not take effect on any already existing connections. If the server is too busy or the limit on the **life** time or the number of **requests** is reached, the connection may be closed sooner. The default value is 180 seconds (3 minutes). **life**—The maximum number of seconds the connection will be kept open, from the time the connection is established. Note that the new value may not take effect on any already existing connections. If the server is too busy or the limit on the idle time or the number of requests is reached, it may close the connection sooner. Also, since the server will not close the connection while actively processing a request, the connection may remain open longer than the specified **life** time if processing is occurring when the life maximum is reached. In this case, the connection will be closed when processing finishes. The default value is 180 seconds (3 minutes). The maximum value is 86400 seconds (24 hours). **requests**—The maximum limit on the number of requests processed on a persistent connection before it is closed. Note that the new value may not take effect on already existing connections. If the server is too busy or the limit on the idle time or the life time is reached, the connection may be closed before the maximum number of requests are processed. The default value is 1. The maximum value is 86400.

Configuring the HTTP Client

Perform this task to enable the HTTP client and configure optional client characteristics.

The standard HTTP 1.1 client and the secure HTTP client are always enabled. No commands exist to disable the HTTP client. For information about configuring optional characteristics for the HTTPS client, see the *HTTPS-HTTP Server and Client with SSL 3.0*, Release 12.2(15)T, feature module.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ip http client cache** {**ager interval** *minutes* | **memory** {**file** *file-size-limit* | **pool** *pool-size-limit*}
4. **ip http client connection** {**forceclose** | **idle timeout** *seconds* | **retry** *count* | **timeout** *seconds*}
5. **ip http client password** *password*
6. **ip http client proxy-server** *proxy-name* **proxy-port** *port-number*
7. **ip http client response timeout** *seconds*
8. **ip http client source-interface** *type number*
9. **ip http client username** *username*

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**ip http client cache** {**ager interval** *minutes* | **memory** {**file** *file-size-limit* | **pool** *pool-size-limit*} **Example:** `Router(config)# ip http client cache memory file 5`	Configures HTTP client cache.
Step 4	**ip http client connection** {**forceclose** | **idle timeout** *seconds* | **retry** *count* | **timeout** *seconds*} **Example:** `Router(config)# ip http client connection timeout 10`	Configures an HTTP client connection.

	Command or Action	Purpose
Step 5	**ip http client password** *password* **Example:** `Router(config)# ip http client password pswd1`	Configures the default password used for connections to remote HTTP servers.
Step 6	**ip http client proxy-server** *proxy-name* **proxy-port** *port-number* **Example:** `Router(config)# ip http client proxy-server server1 proxy-port 52`	Configures an HTTP proxy server.
Step 7	**ip http client response timeout** *seconds* **Example:** `Router(config)# ip http client response timeout 60`	Specifies the timeout value, in seconds, that the HTTP client waits for a response from the server.
Step 8	**ip http client source-interface** *type number* **Example:** `Router(config)# ip http client source-interface ethernet1/0`	Configures a source interface for the HTTP client.
Step 9	**ip http client username** *username* **Example:** `Router(config)# ip http client user1`	Configures the default username used for connections to remote HTTP servers.

Configuration Examples for HTTP 1.1 Web Server

This section provides the following configuration examples:

- Configuring the HTTP 1.1 Web Server: Example, page 7
- Verifying HTTP Connectivity, page 8

Configuring the HTTP 1.1 Web Server: Example

The following example shows a typical configuration that enables the server and sets some of the characteristics:

```
ip http server
ip http authentication aaa
ip http accounting commands 15 default
ip http path flash:
ip access-list standard 20
 permit 209.165.202.130 0.0.0.255
 permit 209.165.201.1 0.0.255.255
 permit 209.165.200.225 0.255.255.255
! (Note: all other access implicitly denied)
 end
ip http access-class 10
ip http max-connections 10
ip http accounting commands 1 oneacct
```

In the following example, a Throughput timeout policy is applied. This configuration would allow each connection to be idle a maximum of 30 seconds (approximately). Each connection will remain open (be “alive”) until either the HTTP server has been busy processing requests for approximately 2 minutes (120 seconds) or until approximately100 requests have been processed.

```
ip http timeout-policy idle 30 life 120 requests 100
```

In the following example, a Response Time timeout policy is applied. This configuration would allow each connection to be idle a maximum of 30 seconds (approximately). Each connection will be closed as soon as the first request has been processed.

```
ip http timeout-policy idle 30 life 30 requests 1
```

Verifying HTTP Connectivity

To verify remote connectivity to the HTTP server, enter the system IP address in a web browser, followed by a colon and the appropriate port number (80 is the default port number).

For example, if the system IP address is 209.165.202.129 and the port number is 8080, enter **http://209.165.202.129:8080** as the URL in a web browser.

If HTTP authentication is configured, a login dialog box will appear. Enter the appropriate username and password. If the default login authentication method of “enable” is configured, you may leave the username field blank, and use the “enable” password to log in.

The system home page should appear in your browser.

Where to Go Next

For information about secure HTTP connections using Secure Sockets Layer (SSL) 3.0, refer to the *HTTPS - HTTP with SSL 3.0*, Release 12.2(15)T, feature module at:
http://www.cisco.com/en/US/docs/ios/12_2t/12_2t15/feature/guide/ftsslsht.html

Additional References

The following sections provide references related to the HTTP 1.1 Web Server and Client.

Related Documents

Related Topic	Document Title
HTTPS	- *HTTPS—HTTP with SSL 3.0*, Release 12.2(15)T, feature module - *Firewall Support of HTTPS Authentication Proxy*, Release 12.2(15)T, feature module
HTTP commands	*Cisco IOS Network Management Command Reference*, Release 12.4T

Standards

No specific standards are supported by this feature. Note that HTTP 1.1, as defined in RFC 2616, is currently classified as a "Standards Track" document by the IETF.

MIBs

MIB	MIBs Link
- No specific MIBs are supported for this feature.	To locate and download MIBs for selected platforms, Cisco IOS releases, and feature sets, use Cisco MIB Locator found at the following URL: http://www.cisco.com/go/mibs

RFCs

RFC[1]	Title
RFC 2616	*Hypertext Transfer Protocol -- HTTP/1.1*

[1] Not all supported RFCs are listed.

The Cisco implementation of the HTTP version 1.1 supports a subset of elements defined in RFC 2616. Following is a list of supported RFC 2616 headers:

- Allow (Only GET, HEAD, and POST methods are supported)
- Authorization, WWW-Authenticate - Basic authentication only
- Cache-control
- Chunked Transfer Encoding
- Connection close
- Content-Encoding
- Content-Language

- Content-Length
- Content-Type
- Date, Expires
- Location

Technical Assistance

Description	Link
The Cisco Support website provides extensive online resources, including documentation and tools for troubleshooting and resolving technical issues with Cisco products and technologies. To receive security and technical information about your products, you can subscribe to various services, such as the Product Alert Tool (accessed from Field Notices), the Cisco Technical Services Newsletter, and Really Simple Syndication (RSS) Feeds. Access to most tools on the Cisco Support website requires a Cisco.com user ID and password.	http://www.cisco.com/techsupport

Feature Information for the HTTP 1.1 Web Server and Client

Table 1 lists the features in this module and provides links to specific configuration information. Only features that were introduced or modified in Cisco IOS Release 12.4T or Cisco IOS Release 12.2SR or a later release appear in the table.

Not all commands may be available in your Cisco IOS software release. For release information about a specific command, see the command reference documentation.

Use Cisco Feature Navigator to find information about platform support and software image support. Cisco Feature Navigator enables you to determine which Cisco IOS and Catalyst OS software images support a specific software release, feature set, or platform. To access Cisco Feature Navigator, go to http://www.cisco.com/go/cfn. An account on Cisco.com is not required.

Note Table 1 lists only the Cisco IOS software release that introduced support for a given feature in a given Cisco IOS software release train. Unless noted otherwise, subsequent releases of that Cisco IOS software release train also support that feature.

Table 1 Feature Information for HTTP 1.1 Web Server and Client

Feature Name	Releases	Feature Information
HTTP 1.1 Web Server and Client	12.2(15)T 12.4(15)T 12.2(33)SRC 12.2(33)SB	The HTTP 1.1 Web Server and Client feature provides a consistent interface for users and applications by implementing support for HTTP 1.1 in Cisco IOS software-based devices. When combined with the HTTPS feature, the HTTP 1.1 Web Server and Client feature provides a complete, secure solution for HTTP services between Cisco devices. This entire module provides information about this feature. The following commands were introduced or modified by this feature: **debug ip http all**, **debug ip http client**, **ip http access-class**, **ip http authentication**, **ip http client cache**, **ip http client connection**, **ip http client password**, **ip http client proxy-server**, **ip http client response timeout**, **ip http client source-interface**, **ip http client username**, **ip http max-connections**, **ip http path**, **ip http port**, **ip http server**, **ip http timeout-policy**, **show ip http client**, **show ip http client connection**, **show ip http client history**, **show ip http client session-module**, **show ip http server**, **show ip http server secure status**.
HTTP TACAC+ Accounting Support	12.4(15)T 12.2(33)SRC 12.2(33)SB Cisco IOS XE Release 2.1	The HTTP TACAC+ Accounting Support feature introduces the **ip http accounting commands** command. This command is used to specify a particular command accounting method for HTTP server users. Command accounting provides information about the commands for a specified privilege level that are being executed on a device. Each command accounting record corresponds to one IOS command executed at its respective privilege level, as well as the date and time the command was executed, and the user who executed it. The following sections provide information about this feature: • Configuring the HTTP 1.1 Web Server, page 3 The following commands were introduced or modified by this feature: **ip http accounting commands**.
HTTP Security	Cisco IOS XE Release 2.1	This feature was introduced on Cisco ASR 1000 Series Routers.

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

HTTPS—HTTP Server and Client with SSL 3.0

First Published: March 31, 2003
Last Updated: March 31, 2008

The HTTPS—HTTP Server and Client with SSL 3.0 feature provides Secure Socket Layer (SSL) version 3.0 support for the HTTP 1.1 server and HTTP 1.1 client within Cisco IOS software. SSL provides server authentication, encryption, and message integrity to allow secure HTTP communications. SSL also provides HTTP client authentication. HTTP over SSL is abbreviated as HTTPS.

Finding Feature Information in This Module

Your Cisco IOS software release may not support all of the features documented in this module. For the latest feature information and caveats, see the release notes for your platform and software release. To reach links to specific feature documentation in this module and to see a list of the releases in which each feature is supported, use the "Feature Information for HTTPS—HTTP Server and Client with SSL 3.0" section on page 17.

Finding Support Information for Platforms and Cisco IOS and Catalyst OS Software Images

Use Cisco Feature Navigator to find information about platform support and Cisco IOS and Catalyst OS software image support. To access Cisco Feature Navigator, go to http://www.cisco.com/go/cfn. An account on Cisco.com is not required.

Contents

- Prerequisites for HTTPS—HTTP Server and Client with SSL 3.0, page 2
- Restrictions for HTTPS—HTTP Server and Client with SSL 3.0, page 2
- Information About HTTPS—HTTP Server and Client with SSL 3.0, page 2
- How to Configure the HTTPS—HTTP Server and Client with SSL 3.0, page 4
- Configuration Examples for the HTTPS—HTTP Server and Client with SSL 3.0 feature, page 13
- Additional References, page 14
- Command Reference, page 15
- Feature Information for HTTPS—HTTP Server and Client with SSL 3.0, page 17

Americas Headquarters:
Cisco Systems, Inc., 170 West Tasman Drive, San Jose, CA 95134-1706 USA

- Glossary, page 18

Prerequisites for HTTPS—HTTP Server and Client with SSL 3.0

To enable secure HTTP connections (encryption) without a configured certificate authority trustpoint, you must first ensure that each device has the key (such as a Rivest, Shamir, and Adleman [RSA] public key or a shared key) of the other device. In most cases, an RSA key pair will be generated automatically. The RSA key pair is used for creating a self-signed certificate (which is also generated automatically).

Restrictions for HTTPS—HTTP Server and Client with SSL 3.0

The HTTPS—HTTP Server and Client with SSL 3.0 feature is available only in Cisco IOS software images that support SSL. SSL is supported in “IPSec 56” (contains “k8” in the image name) and “IPSec 3DES” images (contains “k9” in the image name). “IPSec 56” images provide up to 64-bit encryption, “IPSec 3 DES” images provide greater than 64-bit encryption. The following CipherSuites are supported in IPSec Data Encryption Standard (DES) images:

- SSL_RSA_WITH_RC4_128_MD5—RSA key exchange (RSA Public Key Cryptography) with RC4 128-bit encryption and message digest algorithm 5 (MD5) for message digest
- SSL_RSA_WITH_RC4_128_SHA—RSA key exchange with RC4 128-bit encryption and Secure Hash Algorithm (SHA) for message digest
- SSL_RSA_WITH_3DES_EDE_CBC_SHA—RSA key exchange with 3DES and DES-EDE3-CBC for message encryption and SHA for message digest
- SSL_RSA_WITH_DES_CBC_SHA—RSA key exchange with DES-CBC for message encryption and SHA for message digest

For IPSec 56 images, only the SSL_RSA_WITH_DES_CBC_SHA CipherSuite is supported. For further details on these CipherSuites, see the *SSL Protocol Version 3.0* Internet-Draft document (see the “Related Documents” section on page 14).

RSA (in conjunction with the specified encryption and digest algorithm combinations) is used for both key generation and authentication on SSL connections. This usage is independent of whether a certificate authority (CA) trustpoint is configured.

Information About HTTPS—HTTP Server and Client with SSL 3.0

To configure the HTTP with SSL 3.0 (HTTPS) feature, you should understand the following concepts:

- Secure HTTP Server and Secure HTTP Client, page 2
- Certificate Authority Trustpoints, page 3
- CipherSuites, page 3

Secure HTTP Server and Secure HTTP Client

A secure HTTP connection means that data sent to and received from an HTTP server are encrypted before being sent out over the Internet. HTTP with SSL encryption provides a secure connection to allow such functions as configuring a router from a web browser. Cisco’s implementation of the secure HTTP server and secure HTTP client uses an implementation of the SSL version 3.0. Application layer encryption provides an alternative to older methods such as having to set up a tunnel to the HTTP server for remote management. HTTP over SSL is abbreviated as HTTPS; the URL of a secure connection will begin with https:// instead of http://.

The Cisco IOS HTTP secure server's primary role is to listen for HTTPS requests on a designated port (the default HTTPS port is 443) and to pass the request to the HTTP 1.1 web server. The HTTP 1.1 server processes requests and passes responses (served pages) back to the HTTP secure server, which, in turn, responds to the original request.

The Cisco IOS HTTP secure client's primary role is to respond to Cisco IOS application requests for HTTPS User Agent services, perform HTTPS User Agent services on the application's behalf, and pass the response back to the application.

Certificate Authority Trustpoints

Certificate authorities (CAs) are responsible for managing certificate requests and issuing certificates to participating IPSec network devices. These services provide centralized security key and certificate management for the participating devices. Specific CA servers are referred to as "trustpoints."

The HTTPS server provides a secure connection by providing a certified X.509v3 certificate to the client when a connection attempt is made. The certified X.509v3 certificate is obtained from a specified CA trustpoint. The client (usually a web browser), in turn, has a public key that allows it to authenticate the certificate.

Configuring a CA trustpoint is highly recommended for secure HTTP connections. However, if a CA trustpoint is not configured for the routing device running the HTTPS server, the server will certify itself and generate the needed RSA key pair. Because a self-certified (self-signed) certificate does not provide adequate security, the connecting client will generate a notification that the certificate is self-certified, and the user will have the opportunity to accept or reject the connection. This option is available for internal network topologies (such as testing).

The HTTPS—HTTP Server and Client with SSL 3.0 feature also provides an optional command (**ip http secure-client-auth**) that, when enabled, has the HTTPS server request an X.509v3 certificate from the client. Authenticating the client provides more security than server authentication by itself.

For additional information on certificate authorities, see the "Configuring Certification Authority Interoperability" chapter in the *Cisco IOS Security Configuration Guide*.

CipherSuites

A CipherSuite specifies the encryption algorithm and digest algorithm to use on an SSL connection. Web browsers offer a list of supported CipherSuites when connecting to the HTTPS server, and the client and server will negotiate the best encryption algorithm to use from those that are supported by both. For example, Netscape Communicator 4.76 supports U.S. security with RSA Public Key Cryptography, MD2, MD5, RC2-CBC, RC4, DES-CBC, and DES-EDE3-CBC.

For the best possible encryption, you should use a browser that supports 128-bit encryption, such as Microsoft Internet Explorer version 5.5 (or later), or Netscape Communicator version 4.76 (or later). The SSL_RSA_WITH_DES_CBC_SHA CipherSuite provides less security than the other CipherSuites, because it does not offer 128-bit encryption.

In terms of router processing load (speed), the following list ranks the CipherSuites from fastest to slowest (slightly more processing time is required for the more secure and more complex CipherSuites):

1. SSL_RSA_WITH_DES_CBC_SHA

2. SSL_RSA_WITH_RC4_128_MD5
3. SSL_RSA_WITH_RC4_128_SHA
4. SSL_RSA_WITH_3DES_EDE_CBC_SHA

How to Configure the HTTPS—HTTP Server and Client with SSL 3.0

To configure the HTTPS—HTTP Server and Client with SSL 3.0 feature, complete the procedures in the following sections:

- Declaring a Certificate Authority Trustpoint, page 4
- Configuring the HTTPS Server with SSL 3.0, page 7
- Configuring Standard and Secure HTTP Server Options, page 9
- Configuring the HTTPS Client with SSL 3.0, page 11

Declaring a Certificate Authority Trustpoint

Configuring a CA trustpoint is highly recommended for secure HTTP connections. The certified X.509v3 certificate for the secure HTTP server (or client) is obtained from the specified CA trustpoint. If you do not declare a CA trustpoint, then a self-signed certificate will be used for secure HTTP connections. The self-signed certificate is generated automatically.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **hostname** *name*
4. **ip domain-name** *name*
5. **crypto key generate rsa usage-keys**
6. **crypto ca trustpoint** *name*
7. **enrollment url** *url*
8. **enrollment http-proxy** *host-name port-number*
9. **crl** {**query** *url* | **optional** | **best-effort**}
10. **primary**
11. **exit**
12. **crypto ca authenticate** *name*
13. **crypto ca enrollment** *name*
14. **copy running-config startup-config**
    or
    copy system:running-config nvram:startup-config

DETAILED STEPS

	Command or Action	Purpose

Step 1	**`enable`**
	Example:
	`Router> enable`
		Enables privileged EXEC mode.
		• Enter your password if prompted.

Step 2	**`configure terminal`**
	Example:
	`Router# configure terminal`
		Enters global configuration mode.

Step 3	**`hostname`** *`name`*
	Example:
	`Router(config)# hostname Router`
		Specifies the hostname of the router.
		• This step is needed only if you have not previously configured a hostname for your router. The hostname is required because a fully qualified domain name is needed for security keys and certificates.

Step 4	**`ip domain-name`** *`name`*
	Example:
	`Router(config)# ip domain-name example.com`
		Specifies the IP domain name of the router.
		• This step is needed only if you have not previously configured an IP domain name for your router. The domain name is required because a fully qualified domain name is needed for security keys and certificates.

Step 5	**`crypto key generate rsa usage-keys`**
	Example:
	`Router(config)# crypto key generate rsa usage-keys`
		(Optional) Generates an RSA key pair.
		• The **usage-keys** keyword specifies that two RSA special-usage key pairs should be generated (that is, one encryption pair and one signature pair) instead of one general-purpose key pair.
		• RSA key pairs are used to sign and encrypt Internet key exchange (IKE) key management messages and are required before you can obtain a certificate for your router.
		• RSA key pairs are generated automatically. This command can be used to regenerate the keys, if needed.
		Note There are other keywords and arguments for this command, but they do not pertain to this feature.

Step 6	**`crypto ca trustpoint`** *`name`*
	Example:
	`Router(config)# crypto ca trustpoint TP1`
		Specifies a local configuration name for the CA trustpoint and enters CA trustpoint configuration mode.
		Note The **crypto ca identity** command was replaced by the **crypto ca trustpoint** command in Cisco IOS Release 12.2(8)T.

Step 7	**`enrollment url`** *`url`*
	Example:
	`Router(ca-trustpoint)# enrollment url http://example.com`
		Specifies a URL of the CA where your router should send certificate requests.
		• If you are using Simple Certificate Enrollment Protocol (SCEP) for enrollment, the URL argument must be in the form **http://***CA-name*, where *CA-name* is the host Domain Name System (DNS) name or IP address of the CA trustpoint.

	Command or Action	Purpose
Step 8	**enrollment http-proxy** *host-name port-number* **Example:** `Router(ca-trustpoint)# enrollment http-proxy example.com 8080`	(Optional) Configures the router to obtain certificates from the CA through an HTTP proxy server.

Step 9	**crl** {**query** *url* | **optional** | **best-effort**}
	Example:
	`Router(ca-trustpoint)# crl query ldap://example.com`
		Configures the router to request a certificate revocation list (CRL), make CRL checking optional, or perform CRL checking on a “best-effort” basis.
		• CRLs ensure that the certificate of the peer has not been revoked.
		• The **crl optional** command configures the router to accept certificates even if the appropriate CRL cannot be downloaded.
		• Use the **crl query** *url* command to specify the Lightweight Directory Access Protocol (LDAP) URL of the CA server; for example, **ldap://another-server**.

Step 10	**primary** **Example:** `Router(ca-trustpoint)# primary`	(Optional) Specifies that this trustpoint should be used as the primary (default) trustpoint for CA requests. • Use this command if more than one CA trustpoint will be configured on this router.
Step 11	**exit** **Example:** `Router(ca-trustpoint)# exit`	Exits CA trustpoint configuration mode and returns to global configuration mode.
Step 12	**crypto ca authenticate** *name* **Example:** `Router(config)# crypto ca authenticate TP1`	Authenticates the CA by getting the public key of the CA. • Use the same name that you used when declaring the CA in the **crypto ca trustpoint** command.
Step 13	**crypto ca enrollment** *name* **Example:** `Router(config)# crypto ca enrollment TP1`	Obtains the certificate from the specified CA trustpoint. • This command requests a signed certificate from the CA for each RSA key pair.
Step 14	**copy running-config startup-config** or **copy system:running-config nvram:startup-config** **Example:** `Router(config)# copy running-config startup-config`	Saves the configuration to NVRAM. • This command is required to save the certificates into NVRAM. If not used, the certificates would be lost at router reload. **Note** To execute EXEC mode commands in global configuration mode, you can add the **do** keyword before the command. For example, instead of **copy running-config startup-config**, you could enter **do copy running-config startup-config**.

Configuring the HTTPS Server with SSL 3.0

To disable the standard HTTP server and configure the HTTPS server with SSL 3.0, complete the procedure in this section.

Prerequisites

If a certificate authority is to be used for certification, you should declare the CA trustpoint on the routing device before enabling the secure HTTP server.

SUMMARY STEPS

1. **enable**
2. **show ip http server status**
3. **configure terminal**
4. **no ip http server**
5. **ip http secure-server**
6. **ip http secure-port** *port-number*
7. **ip http secure-ciphersuite** [**3des-ede-cbc-sha**] [**rc4-128-sha**] [**rc4-128-md5**] [**des-cbc-sha**]
8. **ip http secure-client-auth**
9. **ip http secure-trustpoint** *name*
10. **end**
11. **show ip http server secure status**

DETAILED STEPS

	Command or Action	Purpose

Step 1	`enable`
	Example:
	`Router> enable`
		Enables privileged EXEC mode.
		• Enter your password if prompted.

Step 2	`Router# show ip http server status` **Example:** `Router# show ip http server status`	(Optional) Displays the status of the HTTP server. • If you are unsure whether the secure HTTP server is supported in the software image you are running, enter this command and look for the line "HTTP secure server capability: {Present | Not present}". • This command displays the status of the standard HTTP server (enabled or disabled).

Step 3	`configure terminal`
	Example:
	`Router# configure terminal`
		Enters global configuration mode.

	Command or Action	Purpose
Step 4	**no ip http server** **Example:** `Router(config)# no ip http server`	Disables the standard HTTP server. **Note** When enabling the HTTPS server you should always disable the standard HTTP server to prevent insecure connections to the same services. This is a precautionary step (typically, the HTTP server is disabled by default).
Step 5	**ip http secure-server** **Example:** `Router(config)# ip http secure-server`	Enables the HTTPS server.
Step 6	**ip http secure-port** *port-number* **Example:** `Router(config)# ip http secure-port 1025`	(Optional) Specifies the port number that should be used for the HTTPS server. The default port number is 443. Valid options are 443 or any number in the range 1025 to 65535.
Step 7	**ip http secure-ciphersuite** [**3des-ede-cbc-sha**] [**rc4-128-sha**] [**rc4-128-md5**] [**des-cbc-sha**] **Example:** `Router(config)# ip http secure-ciphersuite rc4-128-sha rc4-128-md5`	(Optional) Specifies the CipherSuites (encryption algorithms) that should be used for encryption over the HTTPS connection. • This command allows you to restrict the list of CipherSuites that the server offers the connecting clients. For example, you may want to allow only the most secure CipherSuite to be used. • Unless you have a reason to specify the CipherSuites that should be used, or you are unfamiliar with the details of these CipherSuites, you should leave this command unconfigured and let the server and client negotiate the CipherSuite that they both support (this is the default).
Step 8	**ip http secure-client-auth** **Example:** `Router(config)# ip http secure-client-auth`	(Optional) Configures the HTTP server to request an X.509v3 certificate from the client in order to authenticate the client during the connection process. • In the default connection and authentication process, the client requests a certificate from the HTTP server, but the server does not attempt to authenticate the client. Authenticating the client provides more security than server authentication by itself, but not all clients may be configured for CA authentication.
Step 9	**ip http secure-trustpoint** *name* **Example:** `Router(config)# ip http secure-trustpoint trustpoint-01`	Specifies the CA trustpoint that should be used to obtain an X.509v3 security certificate and to authenticate the connecting client's certificate. • Use of this command assumes you have already declared a CA trustpoint using the **crypto ca trustpoint** command and associated submode commands. • Use the same trustpoint name that you used in the associated **crypto ca trustpoint** command.

	Command or Action	Purpose

Step 10	**end**
	Example:
	`Router(config)# end`
		Ends the current configuration session and returns you to privileged EXEC mode.

Step 11	**show ip http server secure status** **Example:** `Router# show ip http server secure status`	Displays the status of the HTTP secure server configuration.

Verifying the Configuration of the HTTPS Server

To verify the configuration of the HTTPS server, connect to the router running the HTTPS server with a web browser by entering **https://***url*, where *url* is the IP address or hostname of the router. Successful connection using the **https** prefix (instead of the standard **http**) indicates that the HTTPS server is configured properly. If a port other than the default port is configured (using the **ip http secure-port** command), you must also specify the port number after the URL. For example:

```
https://209.165.202.129:1026
```

or

```
https://host.domain.com:1026
```

Generally, you can verify that the HTTPS server is configured and that you have a secure connection by locating an image of a padlock at the bottom of your browser window. Also note that secure HTTP connections have a URL that starts with “https:” instead of “http:”.

Providing Additional Security and Efficiency

The configuration of the standard HTTP server applies to the secure HTTP server as well. To provide additional security and efficiency to both the standard HTTP server and the HTTPS server, complete the procedure in this section.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ip http path** *path-name*
4. **ip http access-class** *access-list-number*
5. **ip http max-connections** *value*
6. **ip http timeout-policy idle** *seconds* **life** *seconds* **requests** *value*

DETAILED STEPS

	Command or Action	Purpose

Step 1	**`enable`**
	Example:
	`Router> enable`
		Enables privileged EXEC mode.
		• Enter your password if prompted.

Step 2	**`configure terminal`**
	Example:
	`Router# configure terminal`
		Enters global configuration mode.

Step 3	**`ip http path`** *`path-name`*
	Example:
	`Router(config)# ip http path slot1:`
		(Optional) Sets the base HTTP path for HTML files.
		• The base path is used to specify the location of the HTTP server files (HTML files) on the local system. Generally, the HTML files are located in system flash memory.

Step 4	**`ip http access-class`** *`access-list-number`*
	Example:
	`Router(config)# ip http access-class 20`
		(Optional) Specifies the access list that should be used to allow access to the HTTP server.

Step 5	**`ip http max-connections`** *`value`*
	Example:
	`Router(config)# ip http max-connections 10`
		(Optional) Sets the maximum number of concurrent connections to the HTTP server that will be allowed. The default value is 5.

	Command or Action	Purpose

Step 6	**`ip http timeout-policy idle`** *`seconds`* **`life`** *`seconds`* **`requests`** *`value`*
	Example:
	`Router(config)# ip http timeout-policy idle 30 life 120 requests 100`
		(Optional) Sets the characteristics that determine how long a connection to the HTTP server should remain open. The characteristics are:
		• **idle**—The maximum number of seconds the connection will be kept open if no data is received or response data cannot be sent out on the connection. Note that a new value may not take effect on any already existing connections. If the server is too busy or the limit on the **life** time or the number of **requests** is reached, the connection may be closed sooner. The default value is 180 seconds (3 minutes).
		• **life**—The maximum number of seconds the connection will be kept open, from the time the connection is established. Note that the new value may not take effect on any already existing connections. If the server is too busy or the limit on the idle time or the number of requests is reached, it may close the connection sooner. Also, because the server will not close the connection while actively processing a request, the connection may remain open longer than the specified **life** time if processing is occurring when the life maximum is reached. In this case, the connection will be closed when processing finishes. The default value is 180 seconds (3 minutes). The maximum value is 86,400 seconds (24 hours).
		• **requests**—The maximum limit on the number of requests processed on a persistent connection before it is closed. Note that the new value may not take effect on any already existing connections. If the server is too busy or the limit on the idle time or the life time is reached, the connection may be closed before the maximum number of requests are processed. The default value is 1. The maximum value is 86,400.

Configuring the HTTPS Client with SSL 3.0

To configure the HTTPS client with SSL 3.0, complete the procedure in this section.

Prerequisites

The standard HTTP client and the secure HTTP client are always enabled.

A certificate authority is required for secure HTTP client certification; the following steps assume that you have previously declared a CA trustpoint on the routing device. If a CA trustpoint is not configured, and the remote HTTPS server requires client authentication, connections to the secure HTTP client will fail.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ip http client secure-trustpoint** *trustpoint-name*

4. **ip http client secure-ciphersuite** [**3des-ede-cbc-sha**] [**rc4-128-sha**] [**rc4-128-md5**] [**des-cbc-sha**]
5. **end**
6. **show ip http client secure status**

DETAILED STEPS

	Command or Action	Purpose

Step 1	**`enable`**
	Example:
	`Router> enable`
		Enables privileged EXEC mode.
		• Enter your password if prompted.

Step 2	**`configure terminal`**
	Example:
	`Router# configure terminal`
		Enters global configuration mode.

Step 3	**`ip http client secure-trustpoint`** *`trustpoint-name`*
	Example:
	`Router(config)# ip http client secure-trustpoint trustpoint01`
		(Optional) Specifies the CA trustpoint that should be used if the remote HTTP server requests client authentication.
		• Use of this command assumes you have already declared a CA trustpoint using the **crypto ca trustpoint** command and associated submode commands.
		• Use the same trustpoint name that you used in the associated **crypto ca trustpoint** command.
		• This command is optional if client authentication is not needed, or if a primary trustpoint has been configured. If the **ip http client secure-trustpoint** command is not used, the router will use the primary trustpoint, as specified by the **primary** CA trustpoint configuration mode command.

Step 4	**`ip http client secure-ciphersuite`** `[`**`3des-ede-cbc-sha`**`] [`**`rc4-128-sha`**`] [`**`rc4-128-md5`**`] [`**`des-cbc-sha`**`]`
	Example:
	`Router(config)# ip http client secure-ciphersuite rc4-128-sha rc4-128-md5`
		(Optional) Specifies the CipherSuites (encryption algorithms) that should be used for encryption over the HTTPS connection.
		• This command allows you to restrict the list of CipherSuites that the client offers when connecting to a secure HTTP server. For example, you may want to allow only the most secure CipherSuites to be used.
		• Unless you have a reason to specify the CipherSuites that should be used, or you are unfamiliar with the details of these CipherSuites, you should leave this command unconfigured and let the server and client negotiate the CipherSuite that they both support (this is the default).

	Command or Action	Purpose

Step 5	`end`
	Example:
	`Router(config)# end`
		Ends the current configuration session and returns to privileged EXEC mode.

Step 6	`show ip http client secure status`
	Example:
	`Router# show ip http client secure status`
		Displays the status of the HTTP secure server configuration.

Configuration Examples for the HTTPS—HTTP Server and Client with SSL 3.0 feature

The following example shows a configuration session in which the secure HTTP server is enabled, the port for the secure HTTP server is configured as 1025, and the remote CA trustpoint server "CA-trust-local" is used for certification.

```
Router# show ip http server status

HTTP server status: Disabled
HTTP server port: 80
HTTP server authentication method: enable
HTTP server access class: 0
HTTP server base path:
Maximum number of concurrent server connections allowed: 5
Server idle time-out: 600 seconds
Server life time-out: 600 seconds
Maximum number of requests allowed on a connection: 1
HTTP secure server capability: Present
HTTP secure server status: Disabled
HTTP secure server port: 443
HTTP secure server ciphersuite: 3des-ede-cbc-sha des-cbc-sha rc4-128-md5 rc4-12a
HTTP secure server client authentication: Disabled
HTTP secure server trustpoint:

Router# configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
Router(config)# ip http secure-server
Router(config)# ip http client secure-trustpoint CA-trust-local
Router(config)# ip http secure-port 1024
Invalid secure port value.

Router(config)# ip http secure-port 1025
Router(config)# ip http secure-ciphersuite rc4-128-sha rc4-128-md5
Router(config)# end
Router# show ip http server secure status
HTTP secure server status: Enabled
HTTP secure server port: 1025
HTTP secure server ciphersuite: rc4-128-md5 rc4-128-sha
HTTP secure server client authentication: Disabled
HTTP secure server trustpoint: CA-trust-local
```

In the following example, the CA trustpoint CA-trust-local is specified, and the HTTPS client is configured to use this trustpoint for client authentication requests:

```
Router# config terminal
Enter configuration commands, one per line.  End with CNTL/Z.
Router(config)# crypto ca trustpoint CA-trust-local
Router(ca-trustpoint)# enrollment url http://example.com
Router(ca-trustpoint)# crl query ldap://example.com
Router(ca-trustpoint)# primary
Router(ca-trustpoint)# exit
Router(config)# ip http client secure-trustpoint CA-trust-local
Router(config)# end
Router# copy running-config startup-config
```

Additional References

The following sections provide references related to the HTTPS—HTTP Server and Client with SSL 3.0 feature.

Related Documents

Related Topic	Document Title
SSL 3.0	*The SSL Protocol Version 3.0* This document is available from various sources online.
Standard Cisco Web Client	*HTTP 1.1 Web Client*
Standard Cisco Web Server	*HTTP 1.1 Web Server*
Certification Authority Interoperability	• *Configuring Certification Authority Interoperability* • *Certificate Autoenrollment* • *Certificate Enrollment Enhancements, Release 12.2(8)T feature document* • *Trustpoint CLI* • *Source Interface Selection for Outgoing Traffic with Certificate Authority*

Standards

Standard	Title
No new or modified standards are supported by this feature.	—

Related MIBs

MIBs	MIBs Link
No new or modified MIBs are supported by this feature.	To locate and download MIBs for selected platforms, Cisco IOS releases, and feature sets, use Cisco MIB Locator found at the following URL: http://www.cisco.com/go/mibs

Related RFCs

RFCs	Description
RFC 2616	Cisco's implementation of HTTP is based on *RFC 2616: Hypertext Transfer Protocol -- HTTP/1.1*.

Technical Assistance

Description	Link
The Cisco Support website provides extensive online resources, including documentation and tools for troubleshooting and resolving technical issues with Cisco products and technologies. To receive security and technical information about your products, you can subscribe to various services, such as the Product Alert Tool (accessed from Field Notices), the Cisco Technical Services Newsletter, and Really Simple Syndication (RSS) Feeds. Access to most tools on the Cisco Support website requires a Cisco.com user ID and password.	http://www.cisco.com/techsupport

Command Reference

The following commands are introduced or modified in the feature or features documented in this module. For information about these commands, see the *Cisco IOS Network Management Command Reference* at http://www.cisco.com/en/US/docs/ios/netmgmt/command/reference/sbnm_r.html. For information about all Cisco IOS commands, use the Command Lookup Tool at http://tools.cisco.com/Support/CLILookup or a Cisco IOS master commands list.

- **debug ip http ssl error**
- **ip http client secure-ciphersuite**
- **ip http client secure-trustpoint**
- **ip http secure-ciphersuite**
- **ip http secure-client-auth**
- **ip http secure-port**

- **ip http secure-server**
- **ip http secure-trustpoint**
- **show ip http client secure status**
- **show ip http server secure status**

Feature Information for HTTPS—HTTP Server and Client with SSL 3.0

Table 1 lists the release history for this feature.

Not all commands may be available in your Cisco IOS software release. For release information about a specific command, see the command reference documentation.

Use Cisco Feature Navigator to find information about platform support and software image support. Cisco Feature Navigator enables you to determine which Cisco IOS and Catalyst OS software images support a specific software release, feature set, or platform. To access Cisco Feature Navigator, go to http://www.cisco.com/go/cfn. An account on Cisco.com is not required.

Note Table 1 lists only the Cisco IOS software release that introduced support for a given feature in a given Cisco IOS software release train. Unless noted otherwise, subsequent releases of that Cisco IOS software release train also support that feature.

Table 1 Feature Information for HTTPS—HTTP Server and Client with SSL 3.0

Feature Name	Releases	Feature Information
HTTPS—HTTP Server and Client with SSL 3.0	12.2(15)T 12.2(33)SRA 12.2(33)SXH 12.2(33)SB Cisco IOS XE Release 2.1	This feature provides Secure Socket Layer (SSL) version 3.0 support for the HTTP 1.1 server and HTTP 1.1 client within Cisco IOS software. SSL provides server authentication, encryption, and message integrity to allow secure HTTP communications. SSL also provides HTTP client authentication. This feature is supported only in Cisco software images that support SSL. Specifically, SSL is supported in “IPSec 56” and “IPSec 3DES” images (contains “k8” or “k9” in the image name).

Glossary

RSA—RSA is a widely used Internet encryption and authentication system that uses public and private keys for encryption and decryption. The RSA algorithm was invented in 1978 by Ron Rivest, Adi Shamir, and Leonard Adleman. The abbreviation RSA comes from the first letter of the last names of the three original developers. The RSA algorithm is included in many applications, such as the web browsers from Microsoft and Netscape. The RSA encryption system is owned by RSA Security.

SHA—The Secure Hash Algorithm. SHA was developed by NIST and is specified in the Secure Hash Standard (SHS, FIPS 180). Often used as an alternative to Digest 5 algorithm.

signatures, digital—In the context of SSL, “signing” means to encrypt with a private key. In digital signing, one-way hash functions are used as input for a signing algorithm. In RSA signing, a 36-byte structure of two hashes (one SHA and one MD5) is signed (encrypted with the private key).

SSL 3.0—Secure Socket Layer version 3.0. SSL is a security protocol that provides communications privacy over the Internet. The protocol allows client and server applications to communicate in a way that is designed to prevent eavesdropping, tampering, or message forgery. SSL uses a program layer located between the Internet’s HTTP and TCP layers. SSL is included as part of most web server products and as part of most Internet browsers. The SSL 3.0 specification can be found at *http://home.netscape.com/eng/ssl3/*.

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

HTTP Inspection Engine

The HTTP Inspection Engine feature allows users to configure their Cisco IOS Firewall to detect and prohibit HTTP connections—such as tunneling over port 80, unauthorized request methods, and non-HTTP compliant file transfers—that are not authorized within the scope of the security policy configuration. Tunneling unauthorized protocols through port 80 and over HTTP exposes a network to significant security risks.

The Cisco IOS Firewall can now be configured with a security policy that adheres to the following tasks:

- Allowing specific traffic targeted for port 80 to traverse the firewall. The traffic is inspected for protocol conformance and for the types of HTTP commands that are allowed or disallowed.
- Denying specific traffic targeted for port 80 that does not comply to HTTP traffic standards. The firewall is enabled to drop the packet, reset the connection, and send a syslog message, as appropriate.

Feature History for HTTP Inspection Engine

Release	Modification
12.3(14)T	This feature was introduced.

Finding Support Information for Platforms and Cisco IOS Software Images

Use Cisco Feature Navigator to find information about platform support and Cisco IOS software image support. Access Cisco Feature Navigator at http://www.cisco.com/go/fn. You must have an account on Cisco.com. If you do not have an account or have forgotten your username or password, click **Cancel** at the login dialog box and follow the instructions that appear.

Contents

- Restrictions for HTTP Inspection Engine, page 2
- Information About HTTP Inspection Engine, page 2
- How to Define and Apply an HTTP Application Policy to a Firewall for Inspection, page 2
- Configuration Examples for Setting Up an HTTP Inspection Engine, page 10
- Additional References, page 11

Americas Headquarters:
Cisco Systems, Inc., 170 West Tasman Drive, San Jose, CA 95134-1706 USA

- Command Reference, page 12

Restrictions for HTTP Inspection Engine

The Cisco 831 router with 48M RAM does not have enough memory to support this feature.

Information About HTTP Inspection Engine

Before configuring an application firewall to detect and police specific traffic targeted for port 80, you should understand the following concepts:

- What Is a Security Policy?, page 2
- Cisco IOS HTTP Application Policy Overview, page 2

What Is a Security Policy?

The application firewall uses a security policy, which consists of a collection of static signatures, to detect security violations. A static signature is a collection of parameters that specify protocol conditions that must be met before an action is taken. (For example, a signature may specify that an HTTP data stream containing the POST method must reset the connection.) These protocol conditions and reactions are defined by the end user via the command-line interface (CLI) to form a security policy.

Cisco IOS HTTP Application Policy Overview

HTTP uses port 80 to transport Internet web services, which are commonly used on the network and rarely challenged with regards to their legitimacy and conformance to standards. Because port 80 traffic is typically allowed through the network without being challenged, many application developers are leveraging HTTP traffic as an alternative transport protocol in which to enable their application to travel through or even bypass the firewall.

Most firewalls provide only packet filtering capabilities that simply permit or deny port 80 traffic without inspecting the data stream; the Cisco IOS application firewall for HTTP performs packet inspection as follows:

- Detects HTTP connections that are not authorized within the scope of the security policy configuration.
- Detects users who are tunneling applications through port 80.

If the packet is not in compliance with the HTTP protocol, it will be dropped, the connection will be reset, and a syslog message will be generated, as appropriate.

How to Define and Apply an HTTP Application Policy to a Firewall for Inspection

This section contains the following procedures:

- Defining an HTTP Application Policy, page 3

- Applying an HTTP Application Policy to a Firewall for Inspection, page 7

Defining an HTTP Application Policy

Use this task to create an HTTP application firewall policy.

Restrictions

Although application firewall policies are defined in global configuration mode, only one global policy for a given protocol is allowed per interface.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **appfw policy-name** *policy-name*
4. **application** *protocol*
5. **strict-http action** {**reset** | **allow**} [**alarm**]
6. **content-length** {**min** *bytes* **max** *bytes* | **min** *bytes* | **max** *bytes*} **action** {**reset** | **allow**} [**alarm**]
7. **content-type-verification** [**match-req-resp**] **action** {**reset** | **allow**} [**alarm**]
8. **max-header-length** {**request** *bytes* **response** *bytes*} **action** {**reset** | **allow**} [**alarm**]
9. **max-uri-length** *bytes* **action** {**reset** | **allow**} [**alarm**]
10. **request-method** {**rfc** *rfc-method* | **extension** *extension-method*} **action** {**reset** | **allow**} [**alarm**]
11. **port-misuse** {**p2p** | **tunneling** | **im** | **default**} **action** {**reset** | **allow**} [**alarm**
12. **transfer-encoding type** {**chunked** | **compress** | **deflate** | **gzip** | **identity** | **default**} **action** {**reset** | **allow**} [**alarm**]
13. **timeout** *seconds*
14. **audit-trail** {**on** | **off**}
15. **exit**
16. **exit**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**appfw policy-name** *policy-name* **Example:** `Router(config)# appfw policy-name mypolicy`	Defines an application firewall policy and puts the router in application firewall policy configuration mode.
Step 4	**application** *protocol* **Example:** `Router(cfg-appfw-policy)# application http`	Allows you to configure inspection parameters for a given protocol. Currently, only HTTP traffic can be inspected. • *protocol* —Specify the **http** keyword. This command puts you in appfw-policy-*protocol* configuration mode, where "*protocol*" is dependent upon the specified protocol. Because only HTTP can be specified, the configuration mode is appfw-policy-http.
Step 5	**strict-http action** {**reset** | **allow**} [**alarm**] **Example:** `Router(cfg-appfw-policy-http)# strict-http action allow alarm`	(Optional) Allows HTTP messages to pass through the firewall or resets the TCP connection when HTTP noncompliant traffic is detected.
Step 6	**content-length** {**min** *bytes* **max** *bytes* | **min** *bytes* | **max** *bytes*} **action** {**reset** | **allow**} [**alarm**] **Example:** `Router(cfg-appfw-policy-http)# content-length max 1 action allow alarm`	(Optional) Permits or denies HTTP traffic through the firewall on the basis of message size. • **min** | **max** *bytes*—Minimum or maximum content length, in bytes, allowed per message. Number of bytes range: 0 to 65535.
Step 7	**content-type-verification** [**match-req-resp**] **action** {**reset** | **allow**} [**alarm**] **Example:** `Router(cfg-appfw-policy-http)# content-type-verification match-req-resp action allow alarm`	(Optional) Permits or denies HTTP traffic through the firewall on the basis of content message type.
Step 8	**max-header-length** {**request** *bytes* **response** *bytes*} **action** {**reset** | **allow**} [**alarm**] **Example:** `Router(cfg-appfw-policy-http)# max-header-length request 1 response 1 action allow alarm`	(Optional) Permits or denies HTTP traffic on the basis of the message header length. • *bytes*—Number of bytes ranging from 0 to 65535.

	Command or Action	Purpose
Step 9	**max-uri-length** *bytes* **action** {**reset** | **allow**} [**alarm**] **Example:** `Router(cfg-appfw-policy-http)# max-uri-length 1 action allow alarm`	(Optional) Permits or denies HTTP traffic on the basis of the URI length in the request message.
Step 10	**request method** {**rfc** *rfc-method* | **extension** *extension-method*} **action** {**reset** | **allow**} [**alarm**] **Example:** `Router(cfg-appfw-policy-http)# request-method rfc default action allow alarm`	(Optional) Permits or denies HTTP traffic according to either the request methods or the extension methods. • **rfc**—Specifies that the supported methods of RFC 2616, *Hypertext Transfer Protocol—HTTP/1.1*, are to be used for traffic inspection. • *rfc-method*—Any one of the following RFC 2616 methods can be specified: **connect**, **default, delete**, **get**, **head**, **options**, **post**, **put**, **trace**. • **extension**—Specifies that the extension methods are to be used for traffic inspection. • *extension-method*—Any one of the following extension methods can be specified: **copy**, **default,** **edit**, **getattribute**, **getproperties**, **index**, **lock**, **mkdir**, **move**, **revadd**, **revlabel**, **revlog**, **save**, **setattribute**, **startrev**, **stoprev**, **unedit**, **unlock**.
Step 11	**port-misuse** {**p2p** | **tunneling** | **im** | **default**} **action** {**reset** | **allow**} [**alarm**] **Example:** `Router(cfg-appfw-policy-http)# port-misuse default action allow alarm`	(Optional) Permits or denies HTTP traffic through the firewall on the basis of specified applications in the HTTP message. • **p2p**—Peer-to-peer protocol applications subject to inspection: Kazaa and Gnutella. • **tunneling**—Tunneling applications subject to inspection: HTTPPort/HTTPHost, GNU Httptunnel, GotoMyPC, Firethru, Http-tunnel.com Client • **im**—Instant messaging protocol applications subject to inspection: Yahoo Messenger. • **default**—All applications are subject to inspection.

	Command or Action	Purpose

Step 12	**`transfer-encoding type`** `{`**`chunked`** `|` **`compress`** `|` **`deflate`** `|` **`gzip`** `|` **`identity`** `|` **`default`**`}` **`action`** `{`**`reset`** `|` **`allow`**`}` `[`**`alarm`**`]`
	Example:
	`Router(cfg-appfw-policy-http)# transfer-encoding type default action allow alarm`
		(Optional) Permits or denies HTTP traffic according to the specified transfer-encoding of the message.
		• **chunked**—Encoding format (specified in RFC 2616, *Hypertext Transfer Protocol—HTTP/1*) in which the body of the message is transferred in a series of chunks; each chunk contains its own size indicator.
		• **compress**—Encoding format produced by the UNIX "compress" utility.
		• **deflate**—"ZLIB" format defined in RFC 1950, *ZLIB Compressed Data Format Specification version 3.3*, combined with the "deflate" compression mechanism described in RFC 1951, *DEFLATE Compressed Data Format Specification version 1.3*.
		• **gzip**—Encoding format produced by the "gzip" (GNU zip) program.
		• **identity**—Default encoding, which indicates that no encoding has been performed.
		• **default**—All of the transfer encoding types.

Step 13	**`timeout`** *`seconds`*
	Example:
	`Router(cfg-appfw-policy-http)# timeout 60`
		(Optional) Overrides the global TCP idle timeout value for HTTP traffic.
		Note If this command is not issued, the default value specified via the **ip inspect tcp idle-time** command will be used.

Step 14	**`audit-trail`** `{`**`on`** `|` **`off`**`}`
	Example:
	`Router(cfg-appfw-policy-http)# audit-trail on`
		(Optional) Turns audit trail messages on or off.
		Note If this command is not issued, the default value specified via the **ip inspect audit-trail** command will be used.

Step 15	**`exit`**
	Example:
	`Router(cfg-appfw-policy-http)# exit`
		Exits cfg-appfw-policy-http configuration mode.

Step 16	**`exit`**
	Example:
	`Router(cfg-appfw-policy)# exit`
		Exits cfg-appfw-policy configuration mode.

What to Do Next

After you have successfully defined an application policy for HTTP traffic inspection, you must apply the policy to an inspection rule. Thereafter, the inspection rule must be applied to an interface. For information on completing this task, see the section "Applying an HTTP Application Policy to a Firewall for Inspection."

Applying an HTTP Application Policy to a Firewall for Inspection

Use this task to apply an HTTP application policy to an inspection rule, followed by applying the inspection rule to an interface.

Note An application policy can coexist with other inspection protocols (for example, an HTTP policy and an FTP policy can coexist).

Prerequisites

You must have already defined an application policy (as shown in the section "Defining an HTTP Application Policy").

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ip inspect name** *inspection-name* **appfw** *policy-name*
4. **ip inspect name** *inspection-name* **http** [**alert** {**on** | **off**}] [**audit-trail** {**on** | **off**}] [**timeout** *seconds*]
5. **interface** *type number*
6. **ip inspect** *inspection-name* {**in** | **out**}
7. **exit**
8. **exit**
9. **show appfw configuration** [*name*]

   or

   show ip inspect {**name** *inspection-name* | **config** | **interfaces** | **session** [**detail**] | **statistics** | **all**}

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**ip inspect name** *inspection-name* **appfw** *policy-name* **Example:** `Router(config)# ip inspect name firewall appfw mypolicy`	Defines a set of inspection rules for the application policy. • *policy-name*—Must match the policy name specified via the **appfw policy-name** command.
Step 4	**ip inspect name** *inspection-name* **http** [**alert** {**on** | **off**}] [**audit-trail** {**on** | **off**}] [**timeout** *seconds*] **Example:** `Router(config)# ip inspect name firewall http`	Defines a set of inspection rules that is to be applied to all HTTP traffic. • The *inspection-name* argument must match the *inspection-name* argument specified in Step 3.
Step 5	**interface** *type number* **Example:** `Router#(config)# interface FastEthernet0/0`	Configures an interface type and enters interface configuration mode.

Step 6	**ip inspect** *inspection-name* {**in** | **out**}
	Example:
	`Router#(config-if)# ip inspect firewall in`
		Applies the inspection rules (defined in Step 3 and Step 4) to all traffic entering the specified interface.
		• The *inspection-name* argument must match the inspection name defined via the **ip inspect name** command.

Step 7	**exit** **Example:** `Router#(config-if)# exit`	`Exits interface configuration mode.`

	Command or Action	Purpose
Step 8	**exit** **Example:** Router(config)# exit	Exits global configuration mode.
Step 9	**show appfw configuration** [*name*] **Example:** Router# show appfw configuration or **show ip inspect** {**name** *inspection-name* | **config** | **interfaces** | **session** [**detail**] | **statistics** | **all**} **Example:** Router# show ip inspect config	(Optional) Displays application firewall policy configuration information. (Optional) Displays firewall-related configuration information.

Troubleshooting Tips

To help troubleshoot the application firewall configuration, issue the following application-firewall specific debug command: **debug appfw** {**application** *protocol* | **function-trace** | **object-creation** | **object-deletion** | **events** | **timers** | **detailed**}.

The following sample configuration shows how to configure an HTTP policy with application firewall debugging enabled:

```
Router(config)# appfw policy-name myPolicyAPPFW  FUNC:appfw_policy_find
APPFW  FUNC:appfw_policy_find -- Policy myPolicy is not found
APPFW  FUNC:appfw_policy_alloc
APPFW  FUNC:appfw_policy_alloc -- policy_alloc 0x65727278
APPFW  FUNC:appfw_policy_alloc -- Policy 0x65727278 is set to valid
APPFW  FUNC:appfw_policy_alloc -- Policy myPolicy has been created
APPFW  FUNC:appfw_policy_command -- memlock policy 0x65727278

! Debugging sample for application (HTTP) creation

Router(cfg-appfw-policy)# application httpAPPFW  FUNC:appfw_http_command
APPFW  FUNC:appfw_http_appl_find
APPFW  FUNC:appfw_http_appl_find -- Application not found
APPFW  FUNC:appfw_http_appl_alloc
APPFW  FUNC:appfw_http_appl_alloc -- appl_http 0x64D7A25C
APPFW  FUNC:appfw_http_appl_alloc -- Application HTTP parser structure 64D7A25C created

! Debugging sample for HTTP-specific application inspection
Router(cfg-appfw-policy-http)#
Router(cfg-appfw-policy-http)# strict-http action reset alarm
APPFW  FUNC:appfw_http_subcommand
APPFW  FUNC:appfw_http_subcommand -- strict-http cmd turned on

Router# debug appfw detailed

APPFW Detailed Debug debugging is on
fw7-7206a#debug appfw object-creation
APPFW Object Creations debugging is on
fw7-7206a#debug appfw object-deletion
APPFW Object Deletions debugging is on
```

Configuration Examples for Setting Up an HTTP Inspection Engine

This section contains the following configuration example:

- Setting Up and Verifying an HTTP Inspection Engine: Example, page 10

Setting Up and Verifying an HTTP Inspection Engine: Example

The following example show how to define the HTTP application firewall policy “mypolicy.” This policy includes all supported HTTP policy rules. This example also includes sample output from the **show appfw configuration** and **show ip inspect config** commands, which allow you to verify the configured setting for the application policy.

```
! Define the HTTP policy.
appfw policy-name mypolicy
 application http
  strict-http action allow alarm
  content-length maximum 1 action allow alarm
  content-type-verification match-req-rsp action allow alarm
  max-header-length request 1 response 1 action allow alarm
  max-uri-length 1 action allow alarm
  port-misuse default action allow alarm
  request-method rfc put action allow alarm
  transfer-encoding type default action allow alarm
!
!
! Apply the policy to an inspection rule.
ip inspect name firewall appfw mypolicy
ip inspect name firewall http
!
!
! Apply the inspection rule to all HTTP traffic entering the FastEthernet0/0 interface.
interface FastEthernet0/0
 ip inspect firewall in
!
!
! Issue the show appfw configuration command and the show ip inspect config command after
the inspection rule “mypolicy” is applied to all incoming HTTP traffic on the
FastEthernet0/0 interface.
!
Router# show appfw configuration

Application Firewall Rule configuration
  Application Policy name mypolicy
    Application http
      strict-http action allow alarm
      content-length minimum 0 maximum 1 action allow alarm
      content-type-verification match-req-rsp action allow alarm
      max-header-length request length 1 response length 1 action allow alarm
      max-uri-length 1 action allow alarm
      port-misuse default action allow alarm
      request-method rfc put action allow alarm
      transfer-encoding default action allow alarm

Router# show ip inspect config

Session audit trail is disabled
Session alert is enabled
```

```
one-minute (sampling period) thresholds are [400:500] connections
max-incomplete sessions thresholds are [400:500]
max-incomplete tcp connections per host is 50. Block-time 0 minute.
tcp synwait-time is 30 sec -- tcp finwait-time is 5 sec
tcp idle-time is 3600 sec -- udp idle-time is 30 sec
dns-timeout is 5 sec
Inspection Rule Configuration
Inspection name firewall
http alert is on audit-trail is off timeout 3600
```

Additional References

The following sections provide references related to the HTTP Inspection Engine feature.

Related Documents

Related Topic	Document Title
Firewall commands: complete command syntax, command mode, defaults, usage guidelines, and examples	*Cisco IOS Security Command Reference,* Release 12.3T

Standards

Standards	Title
No new or modified standards are supported by this feature.	—

MIBs

MIBs	MIBs Link
No new or modified MIBs are supported by this feature.	To locate and download MIBs for selected platforms, Cisco IOS releases, and feature sets, use Cisco MIB Locator found at the following URL: http://www.cisco.com/go/mibs

RFCs

RFCs	Title
RFC 2616	*Hypertext Transfer Protocol -- HTTP/1.1*

Technical Assistance

Description	Link
Technical Assistance Center (TAC) home page, containing 30,000 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/public/support/tac/home.shtml

Command Reference

The following new and modified commands are pertinent to this feature. To see the command pages for these commands and other commands used with this feature, go to the *Cisco IOS Master Commands List*, Release 12.4, at http://www.cisco.com/univercd/cc/td/doc/product/software/ios124/124mindx/124index.htm.

New Commands

- **appfw policy-name**
- **application**
- **audit-trail**
- **content-length**
- **content-type-verification**
- **debug appfw**
- **max-header-length**
- **max-uri-length**
- **port-misuse**
- **request-method**
- **show appfw**
- **strict-http**
- **timeout**
- **transfer-encoding type**

Modified Command

- **ip inspect name**

Selective Enabling of Applications Using an HTTP or HTTPS Server

HTTP Server - Enabling of Applications

The Selective Enabling of Applications Using an HTTP or HTTPS Server feature eliminates a potential security vulnerability by providing a facility to enable selected HTTP and HTTP over Secure Socket Layer (HTTPS) services on both the Cisco IOS HTTP and HTTPS server infrastructure. This feature also provides the capability to view the current state of the HTTP and HTTPS services, including which services are enabled or disabled.

Feature History for the Selective Enabling of Applications Using an HTTP or HTTPS Server Feature

Release	Modification
12.3(14)T	This feature was introduced.

Finding Support Information for Platforms and Cisco IOS Software Images

Use Cisco Feature Navigator to find information about platform support and Cisco IOS software image support. Access Cisco Feature Navigator at http://www.cisco.com/go/fn. You must have an account on Cisco.com. If you do not have an account or have forgotten your username or password, click **Cancel** at the login dialog box and follow the instructions that appear.

Contents

- Information About Selective Enabling of Applications Using an HTTP or HTTPS Server, page 2
- How to Enable Selected Applications Using an HTTP or HTTPS Server, page 2
- Configuration Examples for Selective Enabling of Applications Using an HTTP or HTTPS Server, page 5
- Additional References, page 5
- Command Reference, page 7

Americas Headquarters:
Cisco Systems, Inc., 170 West Tasman Drive, San Jose, CA 95134-1706 USA

Information About Selective Enabling of Applications Using an HTTP or HTTPS Server

To use the Selective Enabling of Applications Using an HTTP or HTTPS Server feature, you should understand the following concept:

- Selective Enabling of Applications Within the HTTP and HTTPS Infrastructure, page 2

Selective Enabling of Applications Within the HTTP and HTTPS Infrastructure

The Selective Enabling of Applications Using an HTTP or HTTPS Server feature eliminates a potential security vulnerability by providing a facility to enable selected HTTP and HTTPS services on both the Cisco IOS HTTP and HTTPS server infrastructure. This feature also provides the capability to view the current state of the HTTP and HTTPS services, including which services are enabled or disabled.

Prior to this feature, HTTP or HTTPS applications running on a router or a switch, were either all enabled or all disabled when the HTTP server or HTTPS server was enabled or disabled, respectively (using the **ip http server** and **ip http secure-server** commands). In the situation where all HTTP or HTTPS applications were enabled, remote end-users were given potential access to services that could allow them to pose a potential security threat to service providers.

With this new feature, the Cisco IOS HTTP and HTTPS infrastructure provides a way to enable only selected HTTP and HTTPS applications to run on a router or a switch, thereby bypassing a potential security vulnerability. Selected HTTP and HTTPS applications can be enabled using the new **ip http active-session-modules** and **ip http secure-active-session-modules** configuration commands, respectively.

Note The maximum number of sessions that can be registered with the Cisco IOS HTTP or HTTPS server is 32.

How to Enable Selected Applications Using an HTTP or HTTPS Server

This section contains the following procedures:

- Enabling Selected HTTP Applications, page 2
- Enabling Selected HTTPS Applications, page 3

Enabling Selected HTTP Applications

Perform this task to selectively enable the HTTP applications that will service incoming HTTP requests from remote clients.

SUMMARY STEPS

1. **enable**
2. **configure terminal**

3. **ip http session-module-list** *listname prefix1* [*prefix2,..., prefixn*]
4. **ip http active-session-modules** {*listname* | **none** | **all**}
5. **end**
6. **show ip http server session-module**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**ip http session-module-list** *listname prefix1* [*prefix2,...,prefixn*] **Example:** `Router(config)# ip http session-module-list list1 SCEP,HOME_PAGE`	Defines a list of HTTP or HTTPS application names.

Step 4	**ip http active-session-modules** {*listname* | **none** | **all**}
	Example:
	`Router(config)# ip http active-session-modules list1`
		Selectively enables HTTP applications that will service incoming HTTP requests from remote clients.
		• The *listname* argument enables only those HTTP services configured in the list identified by the **ip http session-module-list** command to serve HTTP requests.
		• The keyword **none** disables all HTTP services from serving HTTP requests.
		• The keyword **all** enables all HTTP services to serve HTTP requests.

Step 5	**end**
	Example:
	`Router(config)# end`
		Ends your configuration session and returns the CLI to Privileged Exec mode.

Step 6	**show ip http server session-module** **Example:** `Router# show ip http server session-module`	(Optional) Displays information about all HTTP and HTTPS services available on the router or switch, including their current state of service, such as whether they are enabled or disabled.

Enabling Selected HTTPS Applications

Perform this task to selectively enable the HTTPS applications that will service incoming HTTPS requests from remote clients.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ip http session-module-list** *listname prefix1* [*prefix2*,..., *prefixn*]
4. **ip http secure-active-session-modules** {*listname* | **none** | **all**}
5. **end**
6. **show ip http server session-module**

DETAILED STEPS

	Command or Action	Purpose

Step 1	**`enable`**
	Example:
	`Router> enable`
		Enables privileged EXEC mode.
		• Enter your password if prompted.

Step 2	**`configure terminal`**
	Example:
	`Router# configure terminal`
		Enters global configuration mode.

Step 3	**`ip http session-module-list`** *`listname prefix1`* `[`*`prefix2,...,prefixn`*`]`
	Example:
	`Router(config)# ip http session-module-list list1 SCEP,HOME_PAGE`
		Defines a list of HTTP or HTTPS application names.

Step 4	**`ip http secure-active-session-modules`** `{`*`listname`* `|` **`none`** `|` **`all`**`}`
	Example:
	`Router(config)# ip http secure-active-session-modules list1`
		Selectively enables HTTPS applications that will service incoming HTTPS requests from remote clients.
		• The *listname* argument enables only those HTTPS services configured in the list identified by the **ip http session-module-list** command to serve HTTPS requests.
		• The keyword **none** disables all HTTPS services from serving HTTPS requests.
		• The keyword **all** enables all HTTPS services to serve HTTPS requests.

Step 5	**`end`**
	Example:
	`Router(config)# end`
		Ends your configuration session and returns the CLI to Privileged Exec mode.

Step 6	**`show ip http server session-module`**
	Example:
	`Router# show ip http server session-module`
		(Optional) Displays information about all HTTP and HTTPS services available on the router or switch, including their current state of service, such as whether they are enabled or disabled.

Configuration Examples for Selective Enabling of Applications Using an HTTP or HTTPS Server

This section provides the following configuration example:

- Enabling Selected HTTP and HTTPS Applications: Example, page 5

Enabling Selected HTTP and HTTPS Applications: Example

The following configuration sample shows a configuration with different set of services available for HTTP and HTTPS requests. In this example, all HTTP applications are enabled for providing services to remote clients, but for HTTPS services, only the HTTPS applications defined in list1 (Simple Certificate Enrollment Protocol [SCEP] and HOME_PAGE) are enabled.

```
ip http session-module-list list1 SCEP,HOME_PAGE
ip http active-session-modules all
ip http server
ip http secure-server
ip http secure-active-session-modules list1
```

Additional References

The following sections provide references related to the Selective Enabling of Applications Using an HTTP or HTTPS Server feature.

Related Documents

Related Topic	Document Title
Additional HTTP configuration information	"Using the Cisco Web Browser User Interface" *chapter in the section "Cisco IOS User Interfaces" in the Cisco IOS Configuration Fundamentals and Network Management Configuration Guide, Release 12.3T*
Additional HTTPS configuration information	*HTTPS - HTTP Server and Client with SSL 3.0*, Cisco IOS Release 12.2(15)T feature module.
Additional HTTP and HTTPS commands	*Cisco IOS Configuration Fundamentals and Network Management Command Reference*, Release 12.3T

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
None	To locate and download MIBs for selected platforms, Cisco IOS releases, and feature sets, use Cisco MIB Locator found at the following URL: http://www.cisco.com/go/mibs

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
Technical Assistance Center (TAC) home page, containing 30,000 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/public/support/tac/home.shtml

Command Reference

The following commands are introduced or modified in the feature or features documented in this module. For information about these commands, see the *Cisco IOS Network Management Command Reference* at http://www.cisco.com/en/US/docs/ios/netmgmt/command/reference/br_book.html. For information about all Cisco IOS commands, go to the Command Lookup Tool at http://tools.cisco.com/Support/CLILookup or to the *Cisco IOS Master Commands List*.

- **ip http active-session-modules**
- **ip http secure-active-session-modules**
- **ip http session-module-list**
- **show ip http server**

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

HTTP Client API for Tcl IVR

The HTTP Client API for Tcl IVR feature provides support for Tcl IVR applications to retrieve data from or post data to an HTTP server. Also introduced with this feature is a new command-line interface structure for configuring voice applications and support for additional Tcl 8.3.4 commands.

Feature History for HTTP Client API for Tcl IVR

Release	Modification
12.3(14)T	This feature was introduced.

Finding Support Information for Platforms and Cisco IOS Software Images

Use Cisco Feature Navigator to find information about platform support and Cisco IOS software image support. Access Cisco Feature Navigator at http://www.cisco.com/go/fn. You must have an account on Cisco.com. If you do not have an account or have forgotten your username or password, click **Cancel** at the login dialog box and follow the instructions that appear.

Contents

- Prerequisites for HTTP Client API for Tcl IVR and New Cisco Voice Application Command-Line Interface Structure, page 1
- Restrictions for HTTP Client API for Tcl IVR and New Cisco Voice Application Command-Line Interface Structure, page 2
- Information About HTTP Client API for Tcl IVR and New Cisco Voice Application Command-Line Interface Structure, page 3

Prerequisites for HTTP Client API for Tcl IVR and New Cisco Voice Application Command-Line Interface Structure

- Familiarity with Tcl IVR, VoiceXML, and Cisco IOS commands.
- Required hardware:

Americas Headquarters:
Cisco Systems, Inc., 170 West Tasman Drive, San Jose, CA 95134-1706 USA

  - Cisco 3600 series
  - Cisco AS5300
  - Cisco AS5350
  - Cisco AS5400
  - Cisco AS5800
  - Cisco AS58550
- Required software:
  - Cisco IOS Release 12.3(14)T or later
  - Tcl 8.3.4
  - VoiceXML 2.0

Restrictions for HTTP Client API for Tcl IVR and New Cisco Voice Application Command-Line Interface Structure

If Cisco IOS configuration commands are used within the Tcl scripts, submode commands must be entered as quoted arguments on the same line as the configuration command.

Information About HTTP Client API for Tcl IVR and New Cisco Voice Application Command-Line Interface Structure

- HTTP API for Tcl IVR 2.0, page 3
- Newly-Supported Tcl 8.3.4 Commands, page 3
- New Cisco Voice Application Command-Line Interface Structure, page 4

HTTP API for Tcl IVR 2.0

An HTTP application programming interface to the IOS HTTP client is provided. The HTTP package is accessed using the **package require httpios 1.0** Tcl command. Additional commands are provided to configure HTTP. See the *Tcl IVR API Version 2.0 Programming Guide* for more information.

Newly-Supported Tcl 8.3.4 Commands

The following Tcl 8.3.4 commands are now supported:

- cd
- close
- eof
- fconfigure
- file
- fileevent
- flush
- glob
- namespace
- open
- package
- pwd
- read
- seek

The following command is modified:

- puts

See the *Tcl IVR API Version 2.0 Programming Guide* for more information.

New Cisco Voice Application Command-Line Interface Structure

The **call application voice** command structure for configuring Tcl and IVR applications has been restructured to provide easier configuration of application parameters than the earlier CLI structure.

For more information, see the "Cisco IOS Release 12.3(14)T and Later Voice Application Command-Line Interface Structure Changes" section in Configuring Basic Functionality for Tcl IVR and VoiceXML Applications in the *Cisco IOS Tcl IVR and VoiceXML Application Guide*.

Cisco Networking Services (CNS)

Cisco Networking Services

First Published: November 20, 2006
Last Updated: April 18, 2008

The Cisco Networking Services (CNS) feature is a collection of services that can provide remote event-driven configuring of Cisco IOS networking devices and remote execution of some command-line interface (CLI) commands.

Finding Feature Information in This Module

Your Cisco IOS software release may not support all of the features documented in this module. To reach links to specific feature documentation in this module and to see a list of the releases in which each feature is supported, use the "Feature Information for CNS" section on page 48.

Finding Support Information for Platforms and Cisco IOS and Catalyst OS Software Images

Use Cisco Feature Navigator to find information about platform support and Cisco IOS and Catalyst OS software image support. To access Cisco Feature Navigator, go to http://www.cisco.com/go/cfn. An account on Cisco.com is not required.

Contents

- Prerequisites for CNS, page 2
- Restrictions for CNS, page 2
- Information About CNS, page 3
- How to Configure CNS, page 17
- Configuration Examples for CNS, page 36
- Additional References, page 45
- Feature Information for CNS, page 48

Americas Headquarters:
Cisco Systems, Inc., 170 West Tasman Drive, San Jose, CA 95134-1706 USA

Prerequisites for CNS

- Configure the remote router to support the CNS configuration agent and the CNS event agent.
- Configure a transport protocol on the remote router that is compatible with the remote router's external interface. Table 1 lists the supported transport protocols that can be used depending on the router interface.
- Create the configuration template in the CNS configuration-engine provisioning database. (This task is best done by a senior network designer.)

Table 1 Router Interface and Transport Protocols Required by CNS Services

	Transport Protocol		
Router Interface	**SLARP**	**ATM InARP**	**PPP (IPCP)**
T1	Yes	Yes	Yes
ADSL	No	Yes	Yes
Serial	Yes	No	Yes

CNS Image Agent

- Determine where to store the Cisco IOS images on a file server to make the image available to many other networking devices. If the CNS Event Bus is to be used to store and distribute the images, the CNS event agent must be configured.
- Set up a file server to enable the networking devices to download the new images. Protocols such as TFTP, HTTP, HTTPS, and rcp can be used.
- Determine how to handle error messages generated by CNS image agent operations. Error messages can be sent to the CNS Event Bus or an HTTP or HTTPS URL.

Restrictions for CNS

CNS Configuration Engine

- The CNS configuration engine must be the Cisco Intelligence Engine 2100 (Cisco IE2100) series and must be running software version 1.3.
- The configuration engine must have access to an information database of attributes for building a configuration. This database can reside on the Cisco IE2100 itself.
- Configuration templates must be prepared on the CNS configuration engine before installation of the remote router.
- The user of CNS Flow-Through Provisioning and the CNS configuration engine must be familiar with designing network topologies, designing configuration templates, and using the CNS configuration engine.

CNS Image Agent

During automated image loading operations you must try to prevent the Cisco IOS device from losing connectivity with the file server that is providing the image. Image reloading is subject to memory issues and connection issues. Boot options must also be configured to allow the Cisco IOS device to boot another image if the first image reload fails. For more details see the "Managing Configuration Files" module of the *Cisco IOS Configuration Fundamentals Configuration Guide*, Release 12.4.

CNS Frame Relay Zero Touch

The CNS Frame Relay Zero Touch solution does not support switched virtual circuits (SVCs).

The Frame Relay zero touch solution does not support IP over PPP over Frame Relay because routing to an interface (or subinterface) that supports IP over PPP over Frame Relay is not possible.

Command Scheduler

The EXEC CLI specified in a Command Scheduler policy list must neither generate a prompt nor can it be terminated using keystrokes. Command Scheduler is designed as a fully automated facility, and no manual intervention is permitted.

Remote Router

- The remote router must run a Cisco IOS image that supports the CNS configuration agent and CNS event agent.
- Ports must be prepared on the remote router for connection to the network.
- You must ensure that the remote router is configured using Cisco Configuration Express.

Information About CNS

To configure CNS, you should understand the following concepts:

- CNS, page 4
- CNS Configuration Agent, page 4
- Initial CNS Configuration, page 4
- Incremental CNS Configuration, page 5
- Synchronized Configuration, page 5
- CNS Config Retrieve Enhancement with Retry and Interval, page 5
- CNS EXEC Agent, page 5
- CNS Event Agent, page 5
- CNS Image Agent, page 5
- CNS Results Messages, page 6
- CNS Message Formats, page 6
- CNS Security Enhancement, page 9
- CNS Interactive CLI, page 10
- CNS IDs, page 10
- Command Scheduler, page 10
- CNS Flow-Through Provisioning, page 11
- CNS Zero Touch, page 15
- CNS Frame Relay Zero Touch, page 15

CNS

CNS is a foundation technology for linking users to networking services and provides the infrastructure for the automated configuration of large numbers of network devices. Many IP networks are complex with many devices, and each device must currently be configured individually. When standard configurations do not exist or have been modified, the time involved in initial installation and subsequent upgrading is considerable. The volume of smaller, more standardized, customer networks is also growing faster than the number of available network engineers. Internet service providers (ISPs) now need a method for sending out partial configurations to introduce new services. To address all these issues, CNS has been designed to provide "plug-and-play" network services using a central directory service and distributed agents. CNS features include CNS configuration and event agents and a Flow-Through Provisioning structure. The configuration and event agents use a CNS configuration engine to provide methods for automating initial Cisco IOS device configurations, incremental configurations, and synchronized configuration updates, and the configuration engine reports the status of the configuration load as an event to which a network monitoring or workflow application can subscribe. The CNS Flow-Through Provisioning uses the CNS configuration and event agents to provide an automated workflow, eliminating the need for an on-site technician.

CNS Configuration Agent

The CNS configuration agent is involved in the initial configuration and subsequent partial configurations on a Cisco IOS device. To activate the CNS configuration agent, enter any of the **cns config** CLI commands.

Initial CNS Configuration

When a routing device first comes up, it connects to the configuration server component of the CNS configuration agent by establishing a TCP connection through the use of the **cns config initial** command, a standard CLI command. The device issues a request and identifies itself by providing a unique configuration ID to the configuration server.

When the CNS web server receives a request for a configuration file, it invokes the Java servlet and executes the corresponding embedded code. The embedded code directs the CNS web server to access the directory server and file system to read the configuration reference for this device (configuration ID) and template. The Configuration Agent prepares an instantiated configuration file by substituting all the parameter values specified in the template with valid values for this device. The configuration server forwards the configuration file to the CNS web server for transmission to the routing device.

The CNS configuration agent accepts the configuration file from the CNS web server, performs XML parsing, checks syntax (optional), and loads the configuration file. The routing device reports the status of the configuration load as an event to which a network monitoring or workflow application can subscribe.

For more details on using the Cisco CNS configuration engine to automatically install the initial CNS configuration, see the *Cisco CNS Configuration Engine Administrator's Guide* at http://www.cisco.com/univercd/cc/td/doc/product/rtrmgmt/cns/ce/rel13/ag13/index.htm.

Incremental CNS Configuration

Once the network is up and running, new services can be added using the CNS configuration agent. Incremental (partial) configurations can be sent to routing devices. The actual configuration can be sent as an event payload by way of the event gateway (push operation) or as a signal event that triggers the device to initiate a pull operation.

The routing device can check the syntax of the configuration before applying it. If the syntax is correct, the routing device applies the incremental configuration and publishes an event that signals success to the configuration server. If the device fails to apply the incremental configuration, it publishes an event that indicates an error.

Once the routing device has applied the incremental configuration, it can write the configuration to NVRAM or wait until signaled to do so.

Synchronized Configuration

When a routing device receives a configuration, the device has the option to defer application of the configuration upon receipt of a write-signal event. The CNS Configuration Agent feature allows the device configuration to be synchronized with other dependent network activities.

CNS Config Retrieve Enhancement with Retry and Interval

The Cisco Networking Services (CNS) Config Retrieve Enhancement with Retry and Interval feature adds new functionality to the **cns config retrieve** command enabling you to specify the retry interval and an amount of time in seconds to wait before attempting to retrieve a configuration from a trusted server.

CNS EXEC Agent

The CNS EXEC agent allows a remote application to execute an EXEC mode CLI command on a Cisco IOS device by sending an event message that contains the command. A restricted set of EXEC **show** commands is supported.

CNS Event Agent

Although other CNS agents may be configured, no other CNS agents are operational until the **cns event** command is entered because the CNS event agent provides a transport connection to the CNS event bus for all other CNS agents. The other CNS agents use the connection to the CNS event bus to send and receive messages. The CNS event agent does not read or modify the messages.

CNS Image Agent

Administrators maintaining large networks of Cisco IOS devices need an automated mechanism to load image files onto large numbers of remote devices. Existing network management applications are useful to determine which images to run and how to manage images received from the Cisco online software center. Other image distribution solutions do not scale to cover thousands of devices and cannot distribute images to devices behind a firewall or using Network Address Translation (NAT). The CNS image agent enables the managed device to initiate a network connection and request an image download allowing devices using NAT, or behind firewalls, to access the image server.

The CNS image agent can be configured to use the CNS Event Bus. To use the CNS Event Bus, the CNS event agent must be enabled and connected to the CNS event gateway in the CNS Configuration Engine. The CNS image agent can also use an HTTP server that understands the CNS image agent protocol. Deployment of CNS image agent operations can use both the CNS Event Bus and an HTTP server.

CNS Results Messages

When a partial configuration has been received by the router, each line of the configuration will be applied in the same order as it was received. If the Cisco IOS parser has an error with one of the lines of the configuration, then all the configuration up to this point will be applied to the router, but none of the configuration beyond the error will be applied. If an error occurs, the **cns config partial** command will retry until the configuration successfully completes. In the pull mode, the command will not retry after an error. By default, NVRAM will be updated except when the **no-persist** keyword is configured.

A message will be published on the CNS event bus after the partial configuration is complete. The CNS event bus will display one of the following status messages:

- cisco.mgmt.cns.config.complete—CNS configuration agent successfully applied the partial configuration.
- cisco.mgmt.cns.config.warning—CNS configuration agent fully applied the partial configuration, but encountered possible semantic errors.
- cisco.mgmt.cns.config.failure(CLI syntax)—CNS configuration agent encountered a command line interface (CLI) syntax error and was not able to apply the partial configuration.
- cisco.mgmt.cns.config.failure(CLI semantic)—CNS configuration agent encountered a CLI semantic error and was not able to apply the partial configuration.

In Cisco IOS Releases 12.4(4)T, 12.2 (33)SRA, and later releases, a second message is sent to the subject “cisco.cns.config.results” in addition to the appropriate message above. The second message contains both overall and line-by-line information about the configuration that was sent and the result of the action requested in the original message. If the action requested was to apply the configuration, then the information in the results message is semantic in nature. If the action requested was to check syntax only, then the information in the results message is syntactical in nature.

CNS Message Formats

SOAP Message Format

Using the Service-Oriented Access Protocol (SOAP) protocol provides a way to format the layout of CNS messages in a consistent manner. SOAP is a lightweight protocol intended for exchanging structured information in a decentralized, distributed environment. SOAP uses extensible markup language (XML) technologies to define an extensible messaging framework that provides a message format that can be exchanged over a variety of underlying protocols.

Within the SOAP message structure, there is a security header that enables CNS notification messages to authenticate user credentials.

CNS messages are classified into three message types: request, response and notification. The formats of these three message types are defined below.

Request Message

The following is the format of a CNS request message to the Cisco IOS device:

```
<?xml version="1.0" encoding="UTF-8"?>
<SOAP:Envelope xmlns:SOAP="http://www.w3.org/2003/05/soap-envelope">
  <SOAP:Header>
    <wsse:Security xmlns:wsse="http://schemas.xmlsoap.org/ws/2002/04/secext"
SOAP:mustUnderstand="0">
      <wsse:usernameToken>
        <wsse:Username>john</wsse:Username>
        <wsse:Password>cisco</wsse:Password>
      </wsse:usernameToken>
    </wsse:Security>
    <cns:cnsHeader version="1.0" xmlns:cns="http://www.cisco.com/management/cns/envelope">
      <cns:Agent>CNS_CONFIG</cns:Agent>
      <cns:Request>
        <cns:correlationID>IDENTIFIER</cns:correlationID>
        <cns:ReplyTo>
          <cns:URL>http://10.1.36.9:80/cns/ResToServer</cns:URL>
        </cns:ReplyTo>
      </cns:Request>
      <cns:Time>2003-04-23T20:27:19.847Z</cns:Time>
    </cns:cnsHeader>
  </SOAP:Header>
  <SOAP:Body xmlns="http://www.cisco.com/management/cns/config">
    <config-event config-action="read" no-syntax-check="TRUE">
      <config-data>
        <config-id>AAA</config-id>
        <cli>access-list 1 permit any</cli>
      </config-data>
    </config-event>
  </SOAP:Body>
</SOAP:Envelope>
```

Note The ReplyTo field is optional. In the absence of the ReplyTo field, the response to the request will be sent to the destination where the request originated. The body portion of this message contains the payload and is processed by the CNS agent mentioned in the Agent field.

Response Message

The following is the format of a CNS response message from the Cisco IOS device as a response to a request:

```
?xml version="1.0" encoding="UTF-8"?
SOAP:Envelope xmlns:SOAP="http://www.w3.org/2003/05/soap-envelope"
SOAP:Header
wsse:Security xmlns:wsse="http://schemas.xmlsoap.org/ws/2002/04/secext"
SOAP:mustUnderstand="true"
wsse:UsernameToken
wsse:Username infysj-7204-8 /wsse:Username
wsse:Password NTM3NTg2NzIzOTg2MTk2MjgzNQ==/wsse:Password
/wsse:UsernameToken /wsse:Security
CNS:cnsHeader Version="2.0" xmlns:CNS="http://www.cisco.com/management/cns/envelope"
CNS:Agent CNS_CONFIG /CNS:Agent
CNS:Response
CNS:correlationID IDENTIFIER /CNS:correlationID
/CNS:Response
CNS:Time 2005-06-23T16:27:36.185Z /CNS:Time
/CNS:cnsHeader
/SOAP:Header
SOAP:Body xmlns="http://www.cisco.com/management/cns/config"
```

```
config-success config-id AAA /config-id /config-success
/SOAP:Body
/SOAP:Envelope
```

Note The value of CorrelationId is echoed from the corresponding request message.

The body portion of this message contains the response from the Cisco IOS device to a request. If the request is successfully processed, the body portion contains the value of the response put in by the agent that processed the request. If the request cannot be successfully processed, then the body portion will contain an error response.

Notification Message

The following is the format of a CNS notification message sent from the Cisco IOS device:

```
?xml version="1.0" encoding="UTF-8"?
SOAP:Envelope xmlns:SOAP="http://www.w3.org/2003/05/soap-envelope"
SOAP:Header
wsse:Security xmlns:wsse="http://schemas.xmlsoap.org/ws/2002/04/secext"
SOAP:mustUnderstand="true"
wsse:UsernameToken
wsse:Username dvlpr-7200-2 /wsse:Username
wsse:Password /wsse:Password
/wsse:UsernameToken
/wsse:Security
CNS:cnsHeader version="2.0" xmlns:CNS="http://www.cisco.com/management/cns/envelope"
CNS:Agent CNS_CONFIG_CHANGE/CNS:Agent
CNS:Notify /CNS:Notify
CNS:Time 2006-01-09T18:57:08.441Z/CNS:Time
/CNS:cnsHeader
/SOAP:Header
SOAP:Body xmlns="http://www.cisco.com/management/cns/config-change"
configChanged version="1.1" sessionData="complete"
sequence lastReset="2005-12-11T20:18:39.673Z" 7 /sequence
changeInfo
user/user
async port con_0 /port /async
when
absoluteTime 2006-01-09T18:57:07.973Z /absoluteTime
/when
/changeInfo
changeData
changeItem
context /context
enteredCommand
cli access-list 2 permit any /cli
/enteredCommand
oldConfigState
cli access-list 1 permit any /cli
/oldConfigState
newConfigState
cli access-list 1 permit any /cli
cli access-list 2 permit any /cli
/newConfigState
/changeItem
/changeData
/configChanged
/SOAP:Body
/SOAP:Envelope
```

A notification message is sent from the Cisco IOS device without a corresponding request message when a configuration change is made. The body of the message contains the payload of the notification and it may also contain error information. If the request message sent to the Cisco IOS device fails in XML parsing and the CorrelationId field cannot be parsed, then an error notification message will be sent instead of an error response.

Error Reporting

Error is reported in the body of the response or a notification message in the SOAP Fault element. The following is the format for reporting errors.

```
?xml version="1.0" encoding="UTF-8"?
SOAP:Envelope xmlns:SOAP="http://www.w3.org/2003/05/soap-envelope"
SOAP:Header
wsse:Security xmlns:wsse="http://schemas.xmlsoap.org/ws/2002/04/secext"
SOAP:mustUnderstand="true"
wsse:UsernameToken
wsse:Username dvlpr-7200-2 /wsse:Username
wsse:Password /wsse:Password
/wsse:UsernameToken
/wsse:Security
CNS:cnsHeader version="2.0" xmlns:CNS="http://www.cisco.com/management/cns/envelope"
CNS:Agent CNS_CONFIG /CNS:Agent
CNS:Response
CNS:correlationID SOAP_IDENTIFIER /CNS:correlationID
/CNS:Response
CNS:Time 2006-01-09T19:10:10.009Z /CNS:Time
/CNS:cnsHeader
/SOAP:Header
SOAP:Body xmlns="http://www.cisco.com/management/cns/config"
SOAP:Detail
config-failure
config-id AAA /config-id
error-info
line-number 1 /line-number
error-message CNS_INVALID_CLI_CMD /error-message
/error-info
/config-failure
/SOAP:Detail
/SOAP:Fault
/SOAP:Body
/SOAP:Envelope
```

CNS Security Enhancement

Before the introduction of the CNS Security Enhancement feature, the CNS message format did not support security. Using the new CNS SOAP message structure, the username and password are authenticated.

If authentication, authorization, and accounting (AAA) is configured, then CNS SOAP messages will be authenticated with AAA. If AAA is not configured, there will be no authentication. For backward compatibility, CNS will support the existing non-SOAP message format and will respond accordingly without security.

The **cns aaa authentication** command is required to turn on CNS Security Enhancement. This command determines whether the CNS messages are using AAA security or not. If the **cns aaa authentication** command is configured, then all incoming SOAP messages into the device are authenticated by AAA.

CNS Interactive CLI

The CNS Interactive CLI feature provides a XML interface that allows you to send interactive commands to a router, such as commands that generate prompts for user input. A benefit of this feature is that interactive commands can be aborted before they have been fully processed. For example, for commands that generate a significant amount of output, the XML interface can be customized to limit the size of the output or the length of time allowed for the output to accumulate. The capability to use a programmable interface to abort a command before its normal termination (similar to manually aborting a command) can greatly increase the efficiency of diagnostic applications that might use this functionality. The new XML interface also allows for multiple commands to be processed in a single session. The response for each command is packaged together and sent in a single response event.

CNS IDs

The CNS ID is a text string that is used exclusively with a particular CNS agent. The CNS ID is used by the CNS agent to identify itself to the server application with which it communicates. For example, the CNS configuration agent will include the configuration ID when communicating between the networking device and the configuration server. The configuration server uses the CNS configuration ID as a key to locate the attribute containing the Cisco IOS CLI configuration intended for the device that originated the configuration pull.

The network administrator must ensure a match between the CNS agent ID as defined on the routing device and the CNS agent ID contained in the directory attribute that corresponds to the configuration intended for the routing device. Within the routing device, the default value of the CNS agent ID is always set to the hostname. If the hostname changes, the CNS agent ID also changes. If the CNS agent ID is set using the CLI, any change will be followed by a message sent to syslog or an event message will be sent.

The CNS agent ID does not address security issues.

Command Scheduler

The Command Scheduler (KRON) Policy for System Startup feature enables support for the Command Scheduler upon system startup.

The Command Scheduler allows customers to schedule fully-qualified EXEC mode CLI commands to run once, at specified intervals, at specified calendar dates and times, or upon system startup. Originally designed to work with CNS commands, Command Scheduler now has a broader application. Using the CNS image agent feature, remote routers residing outside a firewall or using Network Address Translation (NAT) addresses can use Command Scheduler to launch CLI at intervals, to update the image running in the router.

Command Scheduler has two basic processes. A policy list is configured containing lines of fully-qualified EXEC CLI commands to be run at the same time or same interval. One or more policy lists are then scheduled to run after a specified interval of time, at a specified calendar date and time, or upon system startup. Each scheduled occurrence can be set to run either once only or on a recurring basis.

CNS Flow-Through Provisioning

Cisco Networking Services (CNS) Flow-Through Provisioning provides the infrastructure for automated configuration of large numbers of network devices. Based on CNS event and configuration agents, it eliminates the need for an onsite technician to initialize the device. The result is an automated workflow from initial subscriber-order entry through Cisco manufacturing and shipping to final device provisioning and subscriber billing. This functionality focuses on a root problem of today's service-provider and other similar business models: use of human labor in activating service.

To achieve such automation, CNS Flow-Through Provisioning relies on standardized configuration templates that you create. However, the use of such templates requires a known fixed hardware configuration, uniform for all subscribers. There is no way to achieve this without manually prestaging each line card or module within each chassis. While the inventory within a chassis is known at time of manufacture, controlling which line cards or modules are in which slots thereafter is labor-intensive and error-prone.

To overcome these difficulties, CNS Flow-Through Provisioning defines a new set of Cisco IOS commands—the **cns** commands. When a remote router is first powered on, these commands do the following:

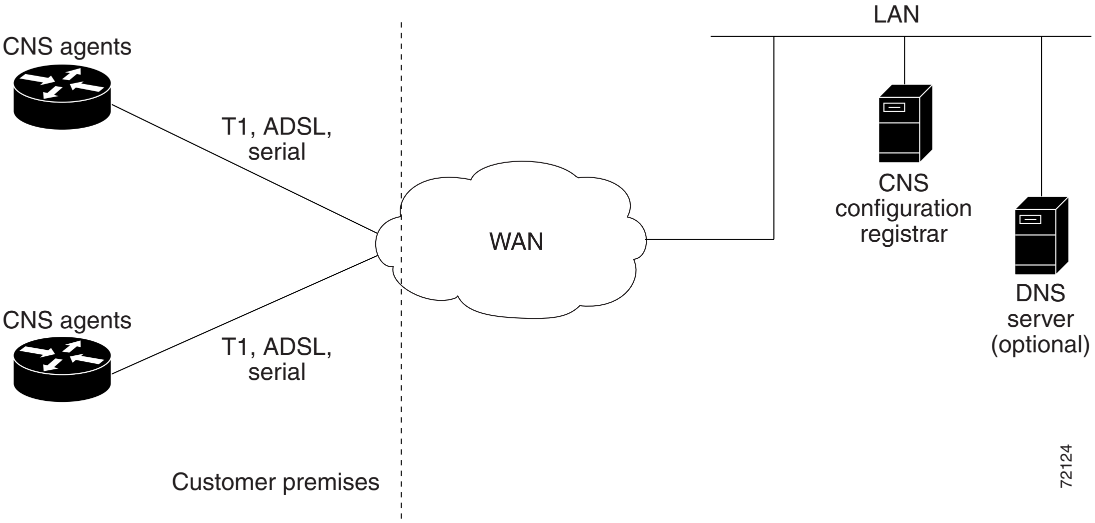
1. To each router interface in turn, applies a preset temporary bootstrap configuration that tries to contact the CNS configuration engine. A successful connection determines the connecting interface.
2. Connects, by way of software called a CNS agent, to a CNS configuration engine housed in a Cisco IE2100 device.
3. Passes to the CNS configuration engine a device-unique ID, along with a human-readable description of the router's line-card or module inventory by product number and location, in XML format.

In turn, the configuration engine does the following:

1. Locates in a Lightweight Directory Access Protocol (LDAP) directory, based on the device IDs, a predefined configuration template for the main chassis and subconfiguration template for each line card or module.
2. Substitutes actual slot numbers from the chassis inventory for the template's slot-number parameters, thus resolving the templates into subscriber-specific configurations that match the true line-card or module slot configuration.
3. Downloads this initial configuration to the target router. The CNS agent directly applies the configuration to the router.

Figure 1 shows the CNS Flow-Through Provisioning architecture.

Figure 1 CNS Flow-Through Provisioning Architecture



Configurations

CNS Flow-Through Provisioning involves three different types of configuration on the remote router:

- Bootstrap configuration

  You specify the preset bootstrap configuration on which this solution depends as part of your order from Cisco using Cisco Configuration Express, an existing service integrated with the Cisco.com order-entry tool. You specify a general-subscriber nonspecific bootstrap configuration that provides connectivity to the CNS configuration engine. Cisco then applies this configuration to all the devices of that order in a totally automated manufacturing step. This configuration runs automatically on power-on.

- Initial configuration

  The CNS configuration engine downloads an initial configuration, once only, to replace the temporary bootstrap configuration. You can either save or not save it in the router's nonvolatile NVRAM memory:

  - If you save the configuration, the bootstrap configuration is overwritten.
  - If you do not save the configuration, the download procedure repeats each time that the router powers off and then back on. Repeating the download procedure enables the router to update to the current Cisco IOS configuration without intervention.

- Incremental (partial) configuration

  On subsequent reboot, incremental or partial configurations are performed to update the configuration without the network having to shut down. Such configurations can be delivered either in a push operation that you initiate or a pull operation on request from the router.

Unique IDs

Key to this solution is the capability to associate, with each device, a simple, manageable, and unique ID that is compatible with your systems for order entry, billing, provisioning, and shipping and can also link your order-entry system to the Cisco order-fulfillment system. Such an ID must have the following characteristics:

- Be available from manufacturing as part of order fulfillment
- Be recordable on the shipping carton and chassis
- Be available to the device's Cisco IOS software
- Be modifiable after the device is first powered up
- Be representative of both a specific chassis and a specific entry point into your network

To define such an ID, CNS Flow-Through Provisioning equips the CNS agent with a new set of commands—the **cns** commands—with which you specify how configurations should be done and, in particular, how the system defines unique IDs. You enable the Cisco IOS software to auto-discover the unique ID according to directions that you specify and information that you provide, such as chassis serial number, MAC address, IP address, and several other possibilities. The **cns** commands are part of the bootstrap configuration of the manufactured device, specified to Cisco Configuration Express at time of order.

Within this scope, Configuration Express and the **cns** commands also allow you to define custom asset tags to your own specifications, which are serialized during manufacture and automatically substituted into the unit's bootstrap configuration.

Cisco appends tags to the carton for all the various types of IDs supported by the **cns** commands, so that these values can be bar-code read at shipping time and fed back into your systems. Alternatively, these IDs are also available through a direct XML-software interface between your system and the Cisco order-status engine, eliminating the need for bar-code reading. The CNS agent also provides a feedback mechanism whereby the remote device can receive XML events or commands to modify the device's ID, in turn causing that same device to broadcast an event indicating the old/new IDs.

Management Point

On most networks, a small percentage of individual remote routers get configured locally. This can potentially be a serious problem, not only causing loss of synchronization across your network but also opening your system to the possibility that an automatic reconfiguration might conflict with an existing configuration and cause a router to become unusable or even to lose contact with the network.

To address this problem, you can designate a management point in your network, typically on the Cisco IE2100 CNS configuration engine, and configure it to keep track of the configurations on all remote routers.

To enable this solution, configure the CNS agent to publish an event on the CNS event bus whenever any change occurs to the running configuration. This event indicates exactly what has changed (old/new), eliminating the need for the management point to perform a highly unscalable set of operations such as telnetting into the device, applying a script, reading back the entire running configuration, and determining the difference between old and new configurations. Additionally, you can arrange for Simple Network Management Protocol (SNMP) notification traps of configuration changes occurring through the SNMP MIB set.

Point-to-Point Event Bus

Today's business environment requires that you be able to ensure your customers a level of service not less than what they are actually paying for. Toward this end, you activate service-assurance applications that broadcast small poll/queries to the entire network while expecting large responses from a typically small subset of devices according to the criteria of the query.

For these queries to be scalable, it is necessary for the replying device to bypass the normal broadcast properties of the event bus and instead reply on a direct point-to-point channel. While all devices need the benefit of the broadcasted poll so that they can all be aware of the query to which they may need to reply, the devices do not have to be aware of each others' replies. Massive copying and retransmission of device query replies, as part of the unnecessary reply broadcast, is a serious scalability restriction.

To address this scalability problem, the CNS event bus has a point-to-point connection feature that communicates directly back to the poller station.

CNS Flow-Through Provisioning provides the following benefits.

Automated Configuration

CNS Flow-Through Provisioning simplifies installation by moving configuration requirements to the CNS configuration engine and allowing the Cisco IOS configuration to update automatically. The registrar uses popular industry standards and technologies such as XML, Active Directory Services Interface (ADSI)/Active Directory, HTTP/Web Server, ATM Switch Processor (ASP), and Publish-Subscribe Event Bus. The CNS configuration agent enables the CNS configuration engine to configure remote routers in a plug-and-play manner.

Unique IP Addresses and Hostname

CNS Flow-Through Provisioning uses DNS reverse lookup to retrieve the hostname by passing the IP address, then assigns the IP address and optionally the hostname to the remote router. Both IP address and hostname are thus guaranteed to be unique.

Reduced Technical Personnel Requirements

CNS Flow-Through Provisioning permits remote routers to be installed by a person with limited or no technical experience. Because configuration occurs automatically on connection to the network, a network engineer or technician is not required for installation.

Rapid Deployment

Because a person with limited or no technical experience can install a remote router immediately without any knowledge or use of Cisco IOS software, the router can be sent directly to its final premises and be brought up without technician deployment.

Direct Shipping

Routers can be shipped directly to the remote end-user site, eliminating warehousing and manual handling. Configuration occurs automatically on connection to the network.

Remote Updates

CNS Flow-Through Provisioning automatically handles configuration updates, service additions, and deletions. The CNS configuration engine performs a push operation to send the information to the remote router.

Security

Event traffic to and from the remote router is opaque to unauthorized listeners or intruders to your network. CNS agents leverage the latest security features in Cisco IOS software.

CNS Zero Touch

The CNS Zero Touch feature provides a zero touch deployment solution where the router contacts a CNS configuration engine to retrieve its full configuration automatically. This capability is made possible through a single generic bootstrap configuration file common across all service provider end customers subscribing to the services. Within the CNS framework, customers can create this generic bootstrap configuration without device-specific or network-specific information such as interface type, line type, or controller type (if applicable).

The CNS connect functionality is configured with a set of CNS connect templates. A CNS connect profile is created for connecting to the CNS configuration engine and to implement the CNS connect templates on a Customer Premise Equipment (CPE) router. CNS connect variables can be used as placeholders within a CNS connect template configuration. These variables, such as the active DLCI, are substituted with real values before the CNS connect templates are sent to the router's parser.

To use the zero touch functionality, the router that is to be initialized must have a generic bootstrap configuration. This configuration includes CNS connect templates, CNS connect profiles, and the **cns config initial** command. This command initiates the CNS connect function.

The CNS connect functionality performs multiple ping iterations through the router's interfaces and lines, as well as any available controllers. For each iteration, the CNS connect function attempts to ping the CNS configuration engine. If the ping is successful, the pertinent configuration information can be downloaded from the CNS configuration engine. If connectivity to the CNS configuration engine is unsuccessful, the CNS connect function removes the configuration applied to the selected interface, and the CNS connect process restarts with the next available interface specified by the CNS connect profile.

The CNS Zero Touch feature provides the following benefits:

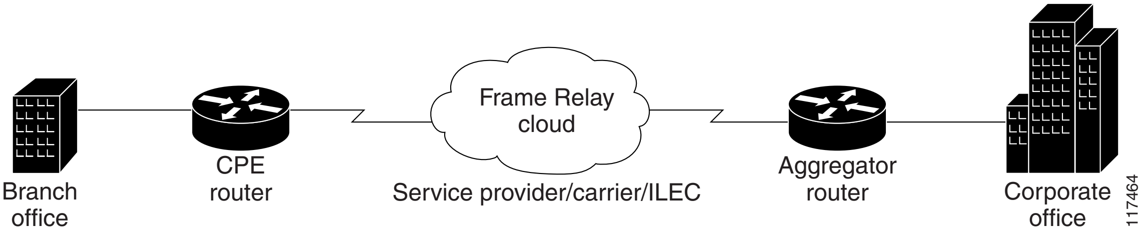
- Ensures consistent CNS commands between Cisco IOS Release 12.3 and 12.3T.
- Use of a channel service unit (E1 or T1 controller) is allowed.

CNS Frame Relay Zero Touch

The CNS Frame Relay Zero Touch feature provides a CNS zero touch deployment solution over Frame Relay where the CPE router discovers its data-link connection identifier (DLCI) and IP address dynamically, and then contacts a CNS engine to retrieve its full configuration automatically. This capability is made possible through a single generic bootstrap configuration file common across all service provider end customers subscribing to the services. Within the CNS framework, customers who deploy Frame Relay can create this generic bootstrap configuration without device-specific or network-specific information such as the DLCI, IP address, interface type, controller type (if applicable), or the next hop interface used for the static default route.

Figure 2 illustrates a typical customer network architecture using Frame Relay.

Figure 2 Connectivity in a Frame Relay Customer Network



The CPE router is deployed at multiple sites. Each site connects to a Frame Relay cloud through a point-to-point permanent virtual circuit (PVC). Connectivity from the Frame Relay cloud to the corporate office is through a PVC that terminates at the corporate office. IP traffic sent to the CNS configuration engine is routed through the corporate office. The PVC is identified by its DLCI. The DLCI can vary between branch offices. In order to support zero touch deployment, the CPE router must be able to learn which DLCI to use to connect to the CNS configuration engine.

To support the zero touch capability, the Frame Relay functionality has been modified in the following two ways:

- A new Cisco IOS command, the **ip address dynamic** command has been introduced to discover the CPE router's IP address dynamically based on the aggregator router's IP address. To configure IP over Frame Relay, the local IP address must be configured on the interface.
- The CPE router can now read Local Management Interface (LMI) messages from a Frame Relay switch and determine the list of available DLCIs.

The CNS connect functionality is configured with a set of CNS connect templates. A CNS connect profile is created for connecting to the CNS configuration engine and to implement the CNS connect templates on a CPE router. CNS connect variables can be used as placeholders within a CNS connect template configuration. These variables, such as the active DLCI, are substituted with real values before the CNS connect templates are sent to the router's parser.

When a CPE router is placed in a Frame Relay network, it contains a generic bootstrap configuration. This configuration includes customer-specific Frame Relay configuration (including the LMI type), CNS connect templates, CNS connect profiles, and the **cns config initial** command. This command initiates the CNS connect function.

The CNS connect functionality begins by selecting the first available controller or interface specified by the CNS connect profile and then performs multiple ping iterations through all the associated active DLCIs. For each iteration, the CNS connect function attempts to ping the CNS configuration engine. If the ping is successful, the pertinent configuration information can be downloaded from the CNS configuration engine.

When iterating over the active DLCIs on a Frame Relay interface, the router must be able to automatically go through a list of active DLCIs returned by the LMI messages for that interface and select an active DLCI to use. When more than one of the active DLCIs allow IP connectivity to the CNS configuration engine, the DLCI used will be the first one tried by the CNS connect functionality. If the ping attempt is unsuccessful, the next active DLCI is tried and so on. If connectivity to the CNS configuration engine is unsuccessful for all active DLCIs, the CNS connect function removes the configuration applied to the selected controller or interface, and the CNS connect process restarts with the next available controller or interface specified by the CNS connect profile.

The CNS Frame Relay Zero Touch feature provides the following benefits:

- A service provider can have a single common bootstrap configuration.
- The generic bootstrap configuration does not require the IP address to be hard-wired.
- The point-to-point DLCI does not need to be known in advance.
- IP directly over Frame Relay is allowed.
- Use of a channel service unit (E1 or T1 controller) is allowed.

How to Configure CNS

This section contains the following tasks:

- Deploying the CNS Router, page 17
- Configuring the CNS Event and EXEC Agents, page 20
- Configuring the CNS Image Agent, page 23
- Configuring CNS Security Features, page 25
- Retrieving a CNS Image from a Server, page 26
- Retrieving a CNS Configuration from a Server, page 27
- Configuring Command Scheduler Policy Lists and Occurrences, page 28
- Configuring Advanced CNS Features, page 31
- Troubleshooting CNS Agents, page 33

Deploying the CNS Router

Perform this task to manually install an initial CNS configuration.

Your remote router arrives from the factory with a bootstrap configuration. Upon initial power-on, the router automatically pulls a full initial configuration from the CNS configuration engine, although you can optionally arrange for this manually as well. After initial configuration, you can optionally arrange for periodic incremental (partial) configurations for synchronization purposes.

For more details on using the Cisco CNS configuration engine to automatically install the initial CNS configuration, see the *Cisco CNS Configuration Engine Administrator's Guide* at http://www.cisco.com/univercd/cc/td/doc/product/rtrmgmt/cns/ce/rel13/ag13/index.htm.

Initial CNS Configuration

Initial configuration of the remote router occurs automatically when the router is initialized on the network. Optionally, you can perform this configuration manually.

CNS assigns the remote router a unique IP address or hostname. After resolving the IP address (using Serial Line Address Resolution Protocol (SLARP), ATM Inverse ARP (ATM InARP), or PPP protocols), the system optionally uses Domain Name System (DNS) reverse lookup to assign a hostname to the router and invokes the CNS agent to download the initial configuration from the CNS configuration engine.

Incremental Configuration

Incremental or partial configuration allows the remote router to be incrementally configured after its initial configuration. You must perform these configurations manually through the CNS configuration engine. The registrar allows you to change the configuration templates, edit parameters, and submit the new configuration to the router without a software or hardware restart.

Prerequisites

Before you can configure an incremental configuration, CNS must be operational and the required CNS agents configured.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **cns template connect** *name*
4. **cli** *config-text*
5. Repeat Step 4 to add all required CLI commands.
6. **exit**
7. **cns connect** *name* [**retry-interval** *interval-seconds*] [**retries** *number-retries*] [**timeout** *timeout-seconds*] [**sleep** *sleep-seconds*]
8. **discover** {**line** *line-type* | **controller** *controller-type* | **interface** [*interface-type*]}
or
template *name*
9. **exit**
10. **cns config initial** {*host-name* | *ip-address*} [**encrypt**] [*port-number*] [**page** *page*] [**syntax-check**] [**no-persist**] [**source** *ip-address*] [**status** *url*] [**event**] [**inventory**]
11. **exit**

DETAILED STEPS

	Command or Action	Purpose

Step 1	`enable`
	Example:
	`Router enable`
		Enables privileged EXEC mode.
		• Enter your password if prompted.

Step 2	`configure terminal`
	Example:
	`Router# configure terminal`
		Enters global configuration mode.

Step 3	`cns template connect` *name* **Example:** `Router(config)# cns template connect template 1`	Enters CNS template connect configuration mode and defines the name of a CNS connect template.
Step 4	`cli` *config-text* **Example:** `Router(config-templ-conn)# cli encapsulation ppp`	Specifies commands to configure the interface.

	Command or Action	Purpose
Step 5	Repeat Step 4 to add all required CLI commands. **Example:** `Router(config-templ-conn)# cli ip directed-broadcast`	Repeat Step 4 to add other CLI commands to configure the interface or to configure the modem lines.

Step 6	**`exit`**
	Example:
	`Router(config-templ-conn)# exit`
		Exits CNS template connect configuration mode and completes the configuration of a CNS connect template.
		Note Entering the **exit** command is required. This requirement was implemented to prevent accidentally entering a command without the **cli** command.

Step 7	**cns connect** *name* [**retry-interval** *interval-seconds*] [**retries** *number-retries*] [**timeout** *timeout-seconds*] [**sleep** *sleep-seconds*] **Example:** `Router(config)# cns connect profile-1 retry-interval 15 timeout 90`	Enters CNS connect configuration mode and defines the parameters of a CNS connect profile for connecting to the CNS configuration engine.

Step 8	**discover** {**line** *line-type* | **controller** *controller-type* | **interface** [*interface-type*]}
	or
	template *name*
	Example:
	`Router(config-cns-conn)# discover interface serial`
	or
	Example:
	`Router(config-cns-conn)# template template-1`
		(Optional) Configures a generic bootstrap configuration.
		• **discover**—Defines the interface parameters within a CNS connect profile for connecting to the CNS configuration engine.
		or
		• **template**—Specifies a list of CNS connect templates within a CNS connect profile to be applied to a router’s configuration.

Step 9	**`exit`**
	Example:
	`Router(config-cns-conn)# exit`
		Exits CNS connect configuration mode and returns to global configuration mode.

Step 10	**cns config initial** {*host-name* | *ip-address*} [**encrypt**] [*port-number*] [**page** *page*] [**syntax-check**] [**no-persist**] [**source** *interface name*] [**status** *url*] [**event**] [**inventory**] **Example:** `Router(config)# cns config initial 10.1.1.1 no-persist`	Starts the CNS configuration agent, connects to the CNS configuration engine, and initiates an initial configuration. You can use this command only before the system boots for the first time. **Note** The optional **encrypt** keyword is available only in images that support Secure Socket Layer (SSL). **Caution** If you write the new configuration to NVRAM by omitting the **no-persist** keyword, the original bootstrap configuration is overwritten.

	Command or Action	Purpose

Step 11	`exit`
	Example:
	`Router(config)# exit`
		Exits global configuration mode and returns to privileged EXEC mode.

Configuring the CNS Event and EXEC Agents

Perform this task to enable and configure the CNS Event and EXEC agents.

CNS Event Agent Parameters

The CNS event agent command—**cns event**—has several parameters that can be configured. The **failover-time** keyword is useful if you have a backup CNS event gateway configured. If the CNS event agent is trying to connect to the gateway and it discovers that the route to the backup gateway is available before the route to the primary gateway, the *seconds* argument specifies how long the CNS event agent will continue to search for a route to the primary gateway before attempting to link to the backup gateway.

Unless you are using a bandwidth-constrained link, you should set a keepalive timeout and retry count. Doing so allows the management network to recover gracefully should a Cisco IE2100 configuration engine ever fail. Without the keepalive data, such a failure requires manual intervention on every device. The *seconds* value multiplied by the *retry-count* value determines the length of idle time before the CNS event agent will disconnect and attempt to reconnect to the gateway. We recommend a minimum *retry-count* value of 2.

If the optional **source** keyword is used, the source IP address might be a secondary IP address of a specific interface to allow a management network to run on top of a production network.

Note Although other CNS agents may be configured, no other CNS agents are operational until the **cns event** command is entered because the CNS event agent provides a transport connection to the CNS event bus for all other CNS agents.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **cns config partial** {*host-name* | *ip-address*} [**encrypt**] [*port-number*] [**source** *ip-address*] [**inventory**]
4. **logging cns-events** [*severity-level*]
5. **cns exec** [*host-name* | *ip-address*] [**encrypt** [*enc-port-number*]] [*port-number*] [**source** *ip-address*]
6. **cns event** {*host-name* | *ip-address*} [**encrypt**] [*port-number*] [**backup**] [**failover-time** *seconds*] [**keepalive** *seconds retry-count*] [**source** *ip-address*] [**clock-timeout** *time*] [**reconnect** *time*]
7. **exit**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**cns config partial** {*host-name* | *ip-address*} [**encrypt**] [*port-number*] [**source** *interface name*] [**inventory**] **Example:** `Router(config)# cns config partial 172.28.129.22 80`	(Optional) Starts the CNS configuration agent, which provides CNS configuration services to Cisco IOS clients, and initiates an incremental (partial) configuration. • Use the optional *port-number* argument to specify the port number for the configuration server. The default is 80. • Use the optional **source** keyword and *ip-address* argument to specify the use of an IP address as the source for CNS configuration agent communications. • Use the optional **inventory** keyword to send an inventory of the line cards and modules in the router to the CNS configuration engine as part of the HTTP request. **Note** The optional **encrypt** keyword is available only in images that support SSL.
Step 4	**logging cns-events** [*severity-level*] **Example:** `Router(config)# logging cns-events 2`	(Optional) Enables XML-formatted system event message logging to be sent through the CNS event bus. • Use the optional *severity-level* argument to specify the number or name of the desired severity level at which messages should be logged. The default is level 7 (debugging).
Step 5	**cns exec** [*host-name* | *ip-address*] [**encrypt** [enc-*port-number*]] [*port-number*] [**source** *ip-address*] **Example:** `Router(config)# cns exec 10.1.2.3 93 source 172.17.2.2`	(Optional) Enables and configures the CNS EXEC agent, which provides CNS EXEC services to Cisco IOS clients. • Use the optional *port-number* argument to specify the port number for the EXEC server. The default is 80. • Use the optional **source** keyword and *ip-address* argument to specify the use of an IP address as the source for CNS EXEC agent communications. **Note** The optional **encrypt keyword** is available only in images that support SSL.

	Command or Action	Purpose

Step 6	`cns event {hostname | ip-address} [encrypt] [port-number] [backup] [failover-time seconds] [keepalive seconds retry-count] [source ip-address] [clock-timeout time] [reconnect time]`
	Example:
	`Router(config)# cns event 172.28.129.22 source 172.22.2.1`
		Configures the CNS event gateway, which provides CNS event services to Cisco IOS clients.
		• The optional **encrypt** keyword is available only in images that support SSL.
		• Use the optional *port-number* argument to specify the port number for the event server. The default is 11011 with no encryption and 11012 with encryption.
		• Use the optional **backup** keyword to indicate that this is the backup gateway. Before configuring a backup gateway, ensure that a primary gateway is configured.
		• Use the optional **failover-time** keyword and *seconds* argument to specify a time interval in seconds to wait for the primary gateway route after the route to the backup gateway is established.
		• Use the optional **keepalive** keyword with the *seconds* and *retry-count* arguments to specify the keepalive timeout in seconds and the retry count.
		• Use the optional **source** keyword and *ip-address* argument to specify the use of an IP address as the source for CNS event agent communications.
		• Use the optional **clock-timeout** keyword to specify the maximum time, in minutes, that the CNS event agent will wait for the clock to be set for transports (such as SSL) that require an accurate clock.
		• Use the optional **reconnect** keyword to specify the configurable upper limit of the maximum retry timeout.
		Note Until the **cns event** command is entered, no transport connections to the CNS event bus are made and therefore no other CNS agents are operational.

Step 7	`exit`
	Example:
	`Router(config)# exit`
		Exits global configuration mode and returns to privileged EXEC mode.

Troubleshooting Tips

- Use the **show cns event connections** command to check that the CNS event agent is connected to the CNS event gateway.
- Use the **show cns event subject** command to check that the image agent subject names are registered. Subject names for the CNS image agent begin with cisco.mgmt.cns.image.

Configuring the CNS Image Agent

Perform this task to configure CNS image agent parameters using CLI commands.

CNS Image Agent ID

CNS uses a unique identifier to identify an image agent associated with that Cisco IOS device. Using the same process as CNS event and configuration agents, the configuration of the **cns id** command determines whether an IP address or MAC address of a specified interface, the hardware serial hardware number of the device, an arbitrary text string, or the hostname of the device is used as the image ID. By default, the system uses the hostname of the device.

The CNS image ID is sent in the content of the messages sent by the image agent and allows an application to know the unique image ID of the Cisco IOS device that generated the message. A password can be configured and associated with the image ID in the image agent messages.

Prerequisites

- To configure the CNS image agent to use HTTP or HTTP over SSL (HTTPS) to communicate with an image server, you need to know the URL for the image server and the URL to which status messages can be sent.
- If you are using HTTPS to communicate with the image server, you must set up security certificates to allow the server to be authenticated by the image agent when the connection is established.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **cns id** *type number* {**dns-reverse** | **ipaddress** | **mac-address**} [**event** | **image**]
   or
   cns id {**hardware-serial** | **hostname** | **string** *text*} [**event** | **image**]
4. **cns image** [**server** *server-url* [**status s***tatus-url*]]
5. **cns image password** *image-password*
6. **cns image retry** *seconds*
7. **exit**

DETAILED STEPS

	Command or Action	Purpose

Step 1	`enable`
	Example:
	`Router enable`
		Enables privileged EXEC mode.
		• Enter your password if prompted.

Step 2	`configure terminal`
	Example:
	`Router# configure terminal`
		Enters global configuration mode.

	Command or Action	Purpose

Step 3	**cns id** *type number* {**dns-reverse** | **ipaddress** | **mac-address**} [**event** | **image**]
	or
	cns id {**hardware-serial** | **hostname** | **string** *text*} [**event** | **image**]
	Example:
	`Router(config)# cns id fastethernet 0/1 ipaddress image`
	or
	Example:
	`Router(config)# cns id hardware-serial image`
		Specifies a unique CNS ID and interface type and number from which to retrieve the unique ID.
		or
		Specifies a unique CNS ID assigned from the hardware serial number, device hostname, or an arbitrary text string.
		The following information applies to either version of the syntax.
		• Use the **event** keyword to specify an event agent ID.
		• Use the **image** keyword to specify an image agent ID.
		• If no keywords are used, the configuration agent ID is configured.
		Note The **dns-reverse** keyword is not supported in Cisco IOS Release 12.2(33)SRA.

Step 4	**cns image** [**server** *server-url* [**status** *status-url*]] **Example:** `Router(config)# cns image server https://10.21.2.3/cns/imgsvr status https://10.21.2.3/cns/status/`	Enables CNS image agent services and specifies the URL of the image distribution server. • Use the optional **status** keyword and *status-url* argument to specify the URL of a web server to which error messages are written. • If the **status** keyword and *status-url* argument are not specified, status messages are sent as events on the CNS Event Bus. To view the status messages on the CNS Event Bus, the CNS event agent must be configured.
Step 5	**cns image password** *image-password* **Example:** `Router(config)# cns image password abctext`	(Optional) Specifies a password for CNS image agent services. • If a password is configured, the password is included with the image ID in CNS image agent messages sent out by the image agent. The receiver of these messages can use this information to authenticate the sending device.
Step 6	**cns image retry** *seconds* **Example:** `Router(config)# cns image retry 240`	(Optional) Specifies an image upgrade retry interval in seconds. • The default interval is 60 seconds.
Step 7	**exit** **Example:** `Router(config)# exit`	Exits global configuration mode and returns the router to privileged EXEC mode.

What to Do Next

Proceed to the "Retrieving a CNS Image from a Server" section to connect to the web server and download an image.

If any of the commands in the task fail, proceed to the "Troubleshooting CNS Agents" section to try to determine the problem.

Configuring CNS Security Features

Perform this task to configure CNS security features.

CNS Trusted Servers

Use the **cns trusted-server** command to specify a trusted server for an individual CNS agent or for all the CNS agents. To avoid security violations, you can build a list of trusted servers from which CNS agents can receive messages. An attempt to connect to a server not on the list will result in an error message being displayed.

Configure a CNS trusted server when a CNS agent will redirect its response to a server address that is not explicitly configured on the command line for the specific CNS agent. For example, the CNS exec agent may have one server configured but receive a message from the CNS event bus that overrides the configured server. The new server address has not been explicitly configured, so the new server address is not a trusted server. An error will be generated when the CNS exec agent tries to respond to this new server address unless the **cns trusted-server** command has been configured for the new server address.

CNS Security Enhancement

CNS messages can be configured to use the CNS SOAP message structure, in which the username and password are authenticated. If AAA is configured, then CNS SOAP messages will be authenticated with AAA. If AAA is not configured, there will be no authentication.

Use the **cns aaa authentication** command to determine whether the CNS messages are using AAA security or not. If the **cns aaa authentication** command is configured, then all incoming SOAP messages into the device are authenticated by AAA.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **cns trusted-server** {**all-agents** | **config** | **event** | **exec** | **image**} *name*
4. **cns message format notification** [**version 1** | **version 2**]
5. **cns aaa authentication** *authentication-method*

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. - Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**cns trusted-server** {**all-agents** | **config** | **event** | **exec** | **image**} *name* **Example:** `Router(config)# cns trusted-server event 10.19.2.5`	Configures a CNS trusted server for the specified hostname or IP address.
Step 4	**cns message format notification** [**version 1** | **version 2**] **Example:** `Router(config)# cns message format notification version 1`	Configures the message format for notification messages from a CNS device. Received messages which do not conform to the configured message format are rejected. Use version 1 to configure the non-SOAP message format. Use version 2 for SOAP message format.
Step 5	**cns aaa authentication** *authentication-method* **Example:** `Router(config)# cns aaa authentication method1`	Enables CNS AAA options. **Note** The authentication methods must be configured within AAA.

Retrieving a CNS Image from a Server

Perform this task to poll the image distribution server using HTTP or HTTPS.

Prerequisites

This task assumes that you have already configured the CNS image agent using the tasks in the "Configuring the CNS Image Agent" section.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **cns image retrieve** [**server** *server-url* [**status** *status-url*]]

DETAILED STEPS

	Command or Action	Purpose

Step 1	**`enable`**
	Example:
	`Router> enable`
		Enables privileged EXEC mode.
		- Enter your password if prompted.

Step 2	**`configure terminal`**
	Example:
	`Router# configure terminal`
		Enters global configuration mode.

Step 3	**`cns image retrieve`** `[`**`server`** *`server-url`* `[`**`status`** *`status-url`*`]]`
	Example:
	`Router(config)# cns image retrieve server https://10.19.2.3/imgsvr/ status https://10.19.2.3/imgsvr/status/`
		Contacts a Cisco CNS image distribution server and downloads a new image if a new image exists.
		- Use the optional **status** keyword and *status-url* argument to specify the URL of a web server to which status messages are written.
		- If the **server** and **status** keywords are not specified, the server and status URLs configured with the **cns image** command are used.
		Note We recommend using the **cns trusted-server** command to specify the host part of the server or status URL as a trusted server.

Troubleshooting Tips

- If the web server appears to be down, use the **ping** command to check connectivity.
- If using HTTP, use the **show ip http client all** command to display information about HTTP clients and connections.

Retrieving a CNS Configuration from a Server

Use this task to request the configuration of a device from a configuration server. Use the **cns trusted-server** command to specify which configuration server can be used (trusted).

Prerequisites

This task assumes that you have specified a trusted server using tasks in the "CNS Security Enhancement" section.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **cns config retrieve** {*host-name* | *ip-address*} [**encrypt**] [*port-number*] [**page** *page*] [**overwrite-startup**] [**retry** *retries* **interval** *seconds*] [**syntax-check**] [**no-persist**] [**source** *interface name*] [**status** *url*] [**event**] [**inventory**]

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**cns config retrieve** {*host-name* | *ip-address*} [**encrypt**] [*port-number*] [**page** *page*] [**overwrite-startup**] [**retry** *retries* **interval** *seconds*] [**syntax-check**] [**no-persist**] [**source** *interface name*] [**status** *url*] [**event**] [**inventory**] **Example:** `Router(config)# cns config retrieve server1 retry 5 interval 45`	Allows the router to retrieve configuration data from a web server. • The **retry** keyword is a number in the range 1 to 100, and will prompt for an **interval** in the range 1 to 3600 seconds.

Troubleshooting Tips

If you need to stop the retrieval process, enter the Ctrl+Shift+6 key sequence.

Configuring Command Scheduler Policy Lists and Occurrences

Perform this task to set up Command Scheduler policy lists of EXEC CNS commands and configure a Command Scheduler occurrence to specify the time or interval after which the CNS commands will run.

Command Scheduler Policy Lists

Policy lists consist of one or more lines of fully-qualified EXEC CLI commands. All commands in a policy list are executed when the policy list is run by Command Scheduler using the **kron occurrence** command. Use separate policy lists for CLI commands that are run at different times. No editor function is available, and the policy list is run in the order in which it was configured. To delete an entry, use the **no** form of the **cli** command followed by the appropriate EXEC command. If an existing policy list name is used, new entries are added to the end of the policy list. To view entries in a policy list, use the **show running-config** command. If a policy list is scheduled to run only once, it will not be displayed by the **show running-config** command after it has run.

Policy lists can be configured after the policy list has been scheduled, but each policy list must be configured before it is scheduled to run.

Command Scheduler Occurrences

An occurrence for Command Scheduler is defined as a scheduled event. Policy lists are configured to run after a specified interval of time, at a specified calendar date and time, or upon system startup. Policy lists can be run once, as a one-time event, or as recurring events over time.

Command Scheduler occurrences can be scheduled before the associated policy list has been configured, but a warning will advise you to configure the policy list before it is scheduled to run.

Prerequisites

The clock time must be set on the routing device before a Command Scheduler occurrence is scheduled to run. If the clock time is not set, a warning message will appear on the console screen after the **kron occurrence** command has been entered. Use the **clock** command or Network Time Protocol (NTP) to set the clock time.

The EXEC CLI to be run by Command Scheduler must be tested on the routing device to determine if it will run without generating a prompt or allowing execution interruption by keystrokes. Initial testing is important because Command Scheduler will delete the entire policy list if any CLI syntax fails. Removing the policy list ensures that any CLI dependencies will not generate more errors.

If you use the **conditional** keyword with the **kron policy-list** command, execution of the commands will stop when an error is encountered.

Restrictions

- No more than 31 policy lists can be scheduled to run at the same time.
- If a one-time occurrence is scheduled, the occurrence will not be displayed by the **show running-config** command after the occurrence has run.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **kron policy-list** *list-name* [**conditional**]
4. **cli** *command*
5. **exit**
6. **kron occurrence** *occurrence-name* [**user** *username*] {**in** [[*numdays*:]*numhours*:]*nummin* | **at** *hours*:*min* [[*month*] *day-of-month*] [*day-of-week*]} {**oneshot** | **recurring** | **system-startup**}
7. **policy-list** *list-name*
8. **exit**
9. **show kron schedule**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** Router> enable	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** Router# configure terminal	Enters global configuration mode.
Step 3	**kron policy-list** *list-name* [**conditional**] **Example:** Router(config)# kron policy-list cns-weekly	Specifies a name for a new or existing Command Scheduler policy list and enters kron-policy configuration mode. • If the *list-name* is new, a new policy list structure is created. • If the *list-name* exists, the existing policy list structure is accessed. The policy list is run in configured order with no editor function. • If the optional **conditional** keyword is used, execution of the commands stops when an error is encountered.
Step 4	**cli** *command* **Example:** Router(config-kron-policy)# cli cns image retrieve server https://10.19.2.3/cnsweek/ status https://10.19.2.3/cnsstatus/week/	Specifies the fully-qualified EXEC command and associated syntax to be added as an entry in the specified Command Scheduler policy list. • Each entry is added to the policy list in the order in which it is configured. • Repeat this step to add other EXEC CLI commands to a policy list to be executed at the same time or interval. **Note** EXEC commands that generate a prompt or can be terminated using keystrokes will cause an error.
Step 5	**exit** **Example:** Router(config-kron-policy)# exit	Exits kron-policy configuration mode and returns the router to global configuration mode.
Step 6	**kron occurrence** *occurrence-name* [**user** *username*] {**in** [[*numdays***:**]*numhours***:**]*nummin* | **at** *hours***:***min* [[*month*] *day-of-month*] [*day-of-week*]} {**oneshot** | **recurring** | **system-startup**} **Example:** Router(config)# kron occurrence may user sales at 6:30 may 20 oneshot	Specifies a name and schedule for a new or existing Command Scheduler occurrence and enters kron-occurrence configuration mode. • Use the **in** keyword to specify a delta time interval with a timer that starts when this command is configured. • Use the **at** keyword to specify a calendar date and time. • Choose either the **oneshot** or **recurring** keyword to schedule Command Scheduler occurrence once or repeatedly. Add the optional **system-startup** keyword for the occurrence to be at system startup.

	Command or Action	Purpose
Step 7	**policy-list** *list-name* **Example:** `Router(config-kron-occurrence)# policy-list sales-may`	Specifies a Command Scheduler policy list. • Each entry is added to the occurrence list in the order in which it is configured. **Note** If the CLI commands in a policy list generate a prompt or can be terminated using keystrokes, an error will be generated and the policy list will be deleted.
Step 8	**exit** **Example:** `Router(config-kron-occurrence)# exit`	Exits kron-occurrence configuration mode and returns the router to global configuration mode. • Repeat this step to exit global configuration mode.
Step 9	**show kron schedule** **Example:** `Router# show kron schedule`	(Optional) Displays the status and schedule information of Command Scheduler occurrences.

Examples

In the following example, output information is displayed about the status and schedule of all configured Command Scheduler occurrences:

```
Router# show kron schedule

Kron Occurrence Schedule
cns-weekly inactive, will run again in 7 days 01:02:33
may inactive, will run once in 32 days 20:43:31 at 6:30 on May 20
```

Troubleshooting Tips

Use the **debug kron** command in privileged EXEC mode to troubleshoot Command Scheduler command operations. Use any debugging command with caution because the volume of output generated can slow or stop the router operations.

Configuring Advanced CNS Features

Perform this task to configure more advanced CNS features. After the CNS agents are operational, you can configure some other features. You can enable the CNS inventory agent—that is, send an inventory of the router’s line cards and modules to the CNS configuration engine—and enter CNS inventory mode.

Some other advanced features allow you to use the Software Developer’s Toolkit (SDK) to specify how CNS notifications should be sent or how to access MIB information. Two encapsulation methods can be used: either nongranular (SNMP) encapsulation or granular (XML) encapsulation.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **cns mib-access encapsulation** {**snmp** | **xml** [**size** *bytes*]}

4. **cns notification encapsulation** {**snmp** | **xml**}
5. **cns inventory**
6. **transport event**
7. **exit**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.

Step 3	**cns mib-access encapsulation** {**snmp** | **xml** [**size** *bytes*]}
	Example:
	`Router(config)# cns mib-access encapsulation snmp`
		(Optional) Specifies the type of encapsulation to use when accessing MIB information.
		• Use the **snmp** keyword to specify that nongranular encapsulation is used to access MIB information.
		• Use the **xml** keyword to specify that granular encapsulation is used to access MIB information. The optional **size** keyword specifies the maximum size for response events, in bytes. The default byte value is 3072.

Step 4	**cns notifications encapsulation** {**snmp** | **xml**}
	Example:
	`Router(config)# cns notifications encapsulation xml`
		(Optional) Specifies the type of encapsulation to use when sending CNS notifications.
		• Use the **snmp** keyword to specify that nongranular encapsulation is used when CNS notifications are sent.
		• Use the **xml** keyword to specify that granular encapsulation is used when CNS notifications are sent.

Step 5	**cns inventory** **Example:** `Router(config)# cns inventory`	Enables the CNS inventory agent and enters CNS inventory mode. • An inventory of the router's line cards and modules is sent to the CNS configuration engine.
Step 6	**transport event** **Example:** `Router(cns-inv)# transport event`	Specifies that inventory requests are sent out with each CNS inventory agent message.
Step 7	**exit** **Example:** `Router(cns-inv)# exit`	Exits CNS inventory mode and returns to global configuration mode. • Repeat this command to return to privileged EXEC mode.

Troubleshooting CNS Agents

This section explains how to troubleshoot CNS agent issues.

The **show** commands created for the CNS image agent display information that is reset to zero after a successful reload of the device. Depending on the configuration of the image distribution process, the new image may not reload immediately. When a reload is not immediate or has failed, use the CNS image agent **show** commands to determine whether the image agent has connected to the image distribution server over HTTP or whether the image agent is receiving events from an application over the CNS Event Bus.

SUMMARY STEPS

1. **enable**
2. **show cns image status**
3. **clear cns image status**
4. **show cns image connections**
5. **show cns image inventory**
6. **debug cns image** [**agent** | **all** | **connection** | **error**]
7. **show cns event connections**
8. **show cns event subject** [*name*]

DETAILED STEPS

	Command or Action	Purpose

Step 1	`enable`
	Example:
	`Router> enable`
		Enables higher privilege levels, such as privileged EXEC mode.
		• Enter your password if prompted.

Step 2	`show cns image status`
	Example:
	`Router# show cns image status`
		(Optional) Displays information about the CNS image agent status.

Step 3	`clear cns image status`
	Example:
	`Router# clear cns image status`
		(Optional) Clears CNS image agent status statistics.

Step 4	`show cns image connections`
	Example:
	`Router# show cns image connections`
		(Optional) Displays information about CNS image management server HTTP or HTTPS connections.

	Command or Action	Purpose
Step 5	**show cns image inventory** **Example:** `Router# show cns image inventory`	(Optional) Displays inventory information about the CNS image agent. • This command displays a dump of XML that would be sent out in response to an image agent inventory request message. The XML output can be used to determine the information requested by an application.
Step 6	**debug cns image** [**agent** | **all** | **connection** | **error**] **Example:** `Router# debug cns image all`	(Optional) Displays debugging messages for CNS image agent services.
Step 7	**show cns event connections** **Example:** `Router# show cns event connections`	(Optional) Displays the status of the CNS event agent connection—such as whether it is connecting to the gateway, connected, or active—and to display the gateway used by the event agent and its IP address and port number.
Step 8	**show cns event subject** [*name*] **Example:** `Router# show cns event subject subject1`	(Optional) Displays a list of subjects of the CNS event agent that are subscribed to by applications.

Examples

This section provides the following output examples:

- Sample Output for the show cns image status Command
- Sample Output for the show cns image connections Command
- Sample Output for the show cns image inventory Command
- Sample Output for the debug cns image Command

Sample Output for the show cns image status Command

In the following example, status information about the CNS image agent is displayed using the **show cns image status** privileged EXEC command:

```
Router# show cns image status

Last upgrade started at 11:45:02.000 UTC Mon May 6 2003
Last upgrade ended at 11:56:04.000 UTC Mon May 6 2003 status SUCCESS

Last successful upgrade ended at 11:56:04.000 UTC Mon May 6 2003
Last failed upgrade ended at 06:32:15.000 UTC Wed Apr 16 2003
Number of failed upgrades: 2
Number of successful upgrades: 6
 messages received: 12
 receive errors: 5
Transmit Status
  TX Attempts:4
    Successes:3         Failures 2
```

Sample Output for the show cns image connections Command

In the following example, information about the status of the CNS image management HTTP connections is displayed using the **show cns image connections** privileged EXEC command:

```
show cns image connections

CNS Image Agent:  HTTP connections
Connection attempts 1
never connected:0   Abrupt disconnect:0
Last successful connection at 11:45:02.000 UTC Mon May 6 2003
```

Sample Output for the show cns image inventory Command

In the following example, information about the CNS image agent inventory is displayed using the **show cns image inventory** privileged EXEC command:

```
show cns image inventory

Inventory Report
imageInventoryReport deviceName imageID Router /imageID hostName Router /ho
IOS (tm) C2600 Software (C2600-I-M), Experimental Version 12.3(20030414:081500)]
Copyright (c) 1986-2003 by cisco Systems, Inc.
Compiled Mon 14-Apr-03 02:03 by engineer /versionString imageFile tftp://10.25.2.1.
```

Sample Output for the debug cns image Command

In the following example, debugging messages for all CNS image agent services are displayed using the **debug cns image** privileged EXEC command. The CNS image agent in this example is connecting to an image server over HTTP. After connecting, the image server asks for an inventory of the Cisco IOS device.

```
Router# debug cns image all

    All cns image debug flags are on

Router# cns image retrieve

May  7 06:11:42.175: CNS Image Agent: set EXEC lock
May  7 06:11:42.175: CNS Image Agent: received message from EXEC
May  7 06:11:42.175: CNS Image Agent: set session lock 1
May  7 06:11:42.175: CNS Image Agent: attempting to send to
destination(http://10.1.36.8:8080/imgsrv/xgate):
?xml version="1.0" encoding="UTF-8"? cnsMessageversion="1.0" senderCredentials userName
dvlpr-7200-6 /userName /senderCredentials
messageID dvlpr-7200-6_2 /messageID sessionControl imageSessionStart version="1.0"
initiatorInfotrigger EXEC/trigger initiatorCredentials userName dvlpr-7200-6/userName
/initiatorCredentials /initiatorInfo /imageSessionStart /sessionControl /cnsMessage
May  7 06:11:42.175: CNS Image Agent: clear EXEC lock
May  7 06:11:42.175: CNS Image Agent: HTTP message sent
url:http://10.1.36.8:8080/imgsrv/xgate

May  7 06:11:42.191: CNS Image Agent: response data alloc 4096 bytes
May  7 06:11:42.191: CNS Image Agent: HTTP req data free
May  7 06:11:42.191: CNS Image Agent: response data freed
May  7 06:11:42.191: CNS Image Agent: receive message
?xml version="1.0" encoding="UTF-8"?
cnsMessage version="1.0"
senderCredentials
userName myImageServer.cisco.com/userName
passWord R0lGODlhcgGSALMAAAQCAEMmCZtuMFQxDS8b/passWord
/senderCredentials
messageID dvlpr-c2600-2-476456/messageID
request
replyTo
```

```
serverReply http://10.1.36.8:8080/imgsrv/xgate /serverReply
/replyTo
imageInventory
inventoryItemList
all/
/inventoryItemList
/imageInventory
/request
/cnsMessage
```

Sample Output for the show cns event Commands

The following example displays the IP address and port number of the primary and backup gateways:

```
Router# show cns event connections

The currently configured primary event gateway:
        hostname is 10.1.1.1.
        port number is 11011.
Event-Id is Internal test1
Keepalive setting:
        none.
Connection status:
        Connection Established.
The currently configured backup event gateway:
        none.
The currently connected event gateway:
        hostname is 10.1.1.1.
        port number is 11011.
```

The following sample displays a list of subjects of the CNS event agent that are subscribed to by applications:

```
Router# show cns event subject

The list of subjects subscribed by applications.
   cisco.cns.mibaccess:request
   cisco.cns.config.load
   cisco.cns.config.reboot
   cisco.cns.exec.cmd
```

Configuration Examples for CNS

This section provides the following configuration examples:

- Deploying the CNS Router: Example, page 37
- Configuring a Partial Configuration: Example, page 37
- Enabling and Configuring CNS Agents: Example, page 37
- CNS Flow-Through Provisioning: Examples, page 38
- Command Scheduler Policy Lists and Occurrences: Examples, page 41
- Retrieving a CNS Image from a Server: Example, page 42
- Retrieving a CNS Configuration from a Server: Examples, page 42
- Using the CNS Zero Touch Solution: Examples, page 43

Deploying the CNS Router: Example

The following example shows an initial configuration on a remote router. The hostname of the remote router is the unique ID. The CNS configuration engine IP address is 172.28.129.22.

```
cns template connect template1
 cli ip address negotiated
 cli encapsulation ppp
 cli ip directed-broadcast
 cli no keepalive
 cli no shutdown
 exit
cns connect host1 retry-interval 30 retries 3
exit
 hostname RemoteRouter
 ip route 172.28.129.22 255.255.255.0 10.11.11.1
 cns id Ethernet 0 ipaddress
 cns config initial 10.1.1.1 no-persist
 exit
```

Configuring a Partial Configuration: Example

Incremental or partial configuration allows the remote router to be incrementally configured after its initial configuration. You must perform these configurations manually through the CNS configuration engine. The registrar allows you to change the configuration templates, edit parameters, and submit the new configuration to the router without a software or hardware restart.

The following example shows incremental (partial) configuration on a remote router. The CNS configuration engine IP address is 172.28.129.22, and the port number is 80.

```
cns config partial 172.28.129.22 80
```

Enabling and Configuring CNS Agents: Example

The following example shows various CNS agents being enabled and configured starting with the configuration agent being enabled with the **cns config partial** command to configure an incremental (partial) configuration on a remote router. The CNS configuration engine IP address is 172.28.129.22, and the port number is 80. The CNS exec agent is enabled with an IP address of 172.28.129.23, and the CNS event agent is enabled with an IP address of 172.28.129.24. Until the CNS event agent is enabled, no other CNS agents are operational.

```
 cns config partial 172.28.129.22 80
 cns exec 172.28.129.23 source 172.22.2.2
 cns event 172.28.129.24 source 172.22.2.1
 exit
```

In the following example, the CNS image agent parameters are configured using the CLI. An image ID is specified to use the IP address of the FastEthernet interface 0/1, a password is configured for the CNS image agent services, the CNS image upgrade retry interval is set to four minutes, and image management and status servers are configured.

```
cns id FastEthernet0/1 ipaddress image
cns image retry 240
cns image password abctext
cns image server https://10.21.2.3/cns/imgsvr status https://10.21.2.3/cns/status/
```

In the following example, the CNS image agent is configured to use the CNS Event Bus. An image ID is specified as the hardware serial number of the networking device, the CNS event agent is enabled with a number of parameters, and the CNS image agent is enabled without any keywords or options. The CNS image agent will listen for events on the CNS Event Bus.

```
cns id hardware-serial image
cns event 10.21.9.7 11011 keepalive 240 120 failover-time 5
cns image
cns image password abctext
```

CNS Flow-Through Provisioning: Examples

Cisco Configuration Express File Using T1 over HDLC Protocol Example

The following example shows use of the Cisco Configuration Express file to configure the remote router before delivery to its final premises. In the example, 172.28.129.22 is the IP address of the CNS configuration engine.

```
cns config initial 172.28.129.22 no-persist
!cns configure and event agents
cns event 172.28.129.22
controller t1 0
!T1 configuration
framing esf
linecode b8zs
channel-group 0 timeslots 1-24 speed 64
exit
cns id s0:0 ipaddress
interface s0:0
!Assigns IP address to s0:0
ip address slarp retry 2
exit
ip route 10.0.0.0 0.0.0.0 s0:0
!IP static route
end
```

T1 Configuration Template Example

The following example shows use of the T1 configuration template to build the configuration for use on T1:

```
hostname ${LDAP://this:attrName=IOShostname}
enable password ${LDAP://this:attrName=IOSpassword}
controller T1 0
clock source ${LDAP://this:attrName=IOST1-clocksource}
linecode ${LDAP://this:attrName=IOST1-line}
framing ${LDAP://this:attrName=IOST1-framing}
channel-group ${LDAP://this:attrName=IOST1-channel-group}
timeslots ${LDAP://this:attrName=IOST1-timeslots}
speed ${LDAP://this:attrName=IOST1-speed}
```

Voice Configuration Template Example

The following example shows use of the voice configuration template to build the configuration for using voice:

```
voice-port 1/1
codec ${LDAP://this:attrName=IOSvoice-port1}
exit
dial-peer voice 1 pots
application ${LDAP://this:attrName=IOSdial-peer1}
port 1/1
```

Remote Router Example

The following example shows a remote router configuration:

```
Router# show running-config

    Current configuration: 1659 bytes
    !
    version 12.2
    no service pad
    service timestamps debug uptime
    service timestamps log uptime
    no service password-encryption
    !
    hostname tira-24V
    !
    !
    network-clock base-rate 64k
    ip subnet-zero
    ip cef
    !
    ip audit notify log
    ip audit po max-events 100
    !
    class-map match-any voice
    match access-group 100
    !
    !
    policy-map qos
    class voice
    priority percent 70
    voice service voip
    h323
    !
    no voice confirmation-tone
    voice-card 0
    !
    !
    controller T1 0
    framing sf
    linecode ami
    !
    controller T1 1
    mode cas
    framing esf
    linecode b8zs
    ds0-group 0 timeslots 1 type e&m-immediate-start
    ds0-group 1 timeslots 2 type e&m-immediate-start
    !
    !
    interface Ethernet0
    ip address 10.1.1.2 255.255.0.0
    !
    interface Serial0
    bandwidth 1536
    ip address 10.11.11.1 255.255.255.0
    no ip mroute-cache
    load-interval 30
    clockrate 148000
    !
    ip classless
    ip route 223.255.254.254 255.255.255.0 10.3.0.1
    !
    no ip http server
    ip pim bidir-enable
    !
    access-list 100 permit udp any range 16384 32767 any
    access-list 100 permit tcp any any eq 1720
    call rsvp-sync
    !
    voice-port 1:0
    timeouts wait-release 3
    !
    voice-port 1:1
    timeouts wait-release 3
    !
    !
    mgcp profile default
    !
    dial-peer cor custom
    !
    dial-peer voice 1000 pots
    destination-pattern 1000
    port 1:0
    forward-digits 0
    !
    dial-peer voice 1001 pots
    destination-pattern 1001
    no digit-strip
    port 1:1
    forward-digits 0
    !
    dial-peer voice 2000 voip
    destination-pattern 2000
    session target ipv4:10.11.11.2
    codec g711ulaw
    !
    dial-peer voice 2001 voip
    destination-pattern 2001
    session target ipv4:10.11.11.2
    signal-type ext-signal
    codec g711ulaw
    !
    !
    line con 0
    line aux 0
    line 2 3
    line vty 0 4
```

The following example shows configuration of a serial interface to connect to and download a configuration from a Cisco IE2100 CNS configuration engine. The IE2100 IP address is 10.1.1.1. The gateway IP address to reach the 10.1.1.0 network is 10.11.11.1. The CNS default ID is the hostname, so that **cns id** command is not needed. However, the **hostname** command is key to retrieving the configuration file on the CNS configuration engine.

This configuration auto-tries ever serial interface on the remote router in turn, applies the **config-cli** commands to that interface, and tries to ping the address in the **cns config initial** command. When it succeeds, it performs a normal initial configuration.

```
! Initial basic configuration (serial interface) PPP
    cns connect serial retry-interval 1 retries 1
    config-cli ip address negotiated
    config-cli encapsulation ppp
    config-cli ip directed-broadcast
    config-cli no keepalive
    config-cli no shutdown
    exit
    hostname 26ML
    ip route 10.1.1.1 255.255.255.0 10.11.11.1
    cns config initial 10.1.1.1 no-persist
    cns inventory config
! Initial basic configuration (serial interface) HDLC
    cns config connect serial retry-interval 1 retries 1
    config-cli ip address slarp retry 1
    config-cli no shutdown
    exit
    hostname tira-36V
    ip route 10.1.1.1 255.255.255.0 10.11.11.1
    cns config initial 10.1.1.1 no-persist
    cns inventory config
    Incremental configuration (serial interface)
    cns config partial 10.1.1.1
    cns event 10.1.1.1
```

Command Scheduler Policy Lists and Occurrences: Examples

In the following example, a Command Scheduler policy named cns-weekly is configured to run two sets of EXEC CLI involving CNS commands. The policy is then scheduled with two other policies to run every seven days, one hour and thirty minutes.

```
kron policy-list cns-weekly
    cli cns image retrieve server http://10.19.2.3/week/ status
    http://10.19.2.5/status/week/
    cli cns config retrieve page /testconfig/config.asp no-persist
    exit
kron occurrence week in 7:1:30 recurring
    policy-list cns-weekly
    policy-list itd-weekly
    policy-list mkt-weekly
```

In the following example, a Command Scheduler policy named sales-may is configured to run a CNS command to retrieve a specified image from a remote server. The policy is then scheduled to run only once on May 20, at 6:30 a.m.

```
kron policy-list sales-may
cli cns image retrieve server 10.19.2.3 status 10.19.2.3
exit
kron occurrence may at 6:30 May 20 oneshot
policy-list sales-may
```

In the following example, a Command Scheduler policy named image-sunday is configured to run a CNS command to retrieve a specified image from a remote server. The policy is then scheduled to run every Sunday at 7:30 a.m.

```
kron policy-list image-sunday
    cli cns image retrieve server 10.19.2.3 status 10.19.2.3
    exit
kron occurrence sunday user sales at 7:30 sunday recurring
    policy-list image-sunday
```

In the following example, a Command Scheduler policy named file-retrieval is configured to run a CNS command to retrieve a specific file from a remote server. The policy is then scheduled to run on system startup.

```
kron policy-list file-retrieval
cli cns image retrieve server 10.19.2.3 status 10.19.2.3
exit
kron occurrence system-startup
policy-list file-retrieval
```

Retrieving a CNS Image from a Server: Example

In the following example, the CNS image agent polls a file server using the **cns image retrieve** command. Assuming that the CNS image agent is already enabled, the file server and status server paths specified here will overwrite any existing image agent server and status configuration. The new file server will be polled and a new image, if it exists, will be downloaded to the networking device.

```
 cns image retrieve server https://10.19.2.3/cns/ status https://10.19.2.3/cnsstatus/
```

Retrieving a CNS Configuration from a Server: Examples

Retrieving Configuration Data from the CNS Trusted Server

The following example shows how to request a configuration from a trusted server at 10.1.1.1:

```
cns trusted-server all 10.1.1.1
exit
cns config retrieve 10.1.1.1
```

The following example shows how to request a configuration from a trusted server at 10.1.1.1 and to configure a CNS configuration retrieve interval using the **cns config retrieve** command:

```
cns trusted-server all 10.1.1.1
exit
cns config retrieve 10.1.1.1 retry 50 interval 1500
CNS Config Retrieve Attempt 1 out of 50 is in progress
Next cns config retrieve retry is in 1499 seconds (Ctrl-Shft-6 to abort this command).
..
00:26:40: %CNS-3-TRANSPORT: CNS_HTTP_CONNECTION_FAILED:10.1.1.1 -Process= "CNS config
retv", ipl= 0, pid= 43
00:26:40: %CNS-3-TRANSPORT: CNS_HTTP_CONNECTION_FAILED -Process= "CNS config retv", ipl=
0, pid= 43......

cns config retrieve 10.1.1.1
```

Applying the Retrieved Data to the Running Configuration File

The following example shows how to check and apply configuration data retrieved from the server to running configuration file only. The CNS Configuration Agent will attempt to retrieve configuration data at 30-second intervals until the attempt is successful, or is unsuccessful five times in these attempts.

```
cns config retrieve 10.1.1.1 syntax-check no-persist retry 5 interval 30
```

Overwriting the Startup Configuration File with the Retrieved Data

The following example shows how to overwrite the startup configuration file with the configuration data retrieved from the server. The configuration data will not be applied to the running configuration.

```
cns config retrieve 10.1.1.1 syntax-check no-persist retry 5 interval 30
cns config retrieve 10.1.1.1 overwrite-startup
```

Using the CNS Zero Touch Solution: Examples

Configuring PPP on a Serial Interface

The following example shows the bootstrap configuration for configuring PPP on a serial interface:

```
cns template connect ppp-serial
cli ip address negotiated
cli encapsulation ppp
cli ip directed-broadcast
cli no keepalive
exit
cns template connect ip-route
cli ip route 10.0.0.0 0.0.0.0 ${next-hop}
exit
cns connect serial-ppp ping-interval 1 retries 1
discover interface serial
template ppp-serial
template ip-route
exit
hostname 26ML
cns config initial 10.1.1.1 no-persist inventory
```

Configuring PPP on an Asynchronous Interface

The following example shows the bootstrap configuration for configuring PPP on an asynchronous interface:

```
cns template connect async
cli modem InOut
 .
 .
 .
exit
cns template connect async-interface
cli encapsulation ppp
cli ip unnumbered FastEthernet0/0
cli dialer rotary-group 0
exit
cns template connect ip-route
cli ip route 10.0.0.0 0.0.0.0 ${next-hop}
exit

cns connect async
discover line Async
template async
discover interface
template async-interface
template ip-route
exit
hostname async-example
cns config initial 10.1.1.1 no-persist inventory
```

Configuring HDLC on a Serial Interface

The following example shows the bootstrap configuration for configuring High-Level Data Link Control (HDLC) on a serial interface:

```
cns template connect hdlc-serial
cli ip address slarp retry 1
exit
cns template connect ip-route
cli ip route 0.0.0.0 0.0.0.0 ${next-hop}
exit
```

```
cns connect hdlc-serial ping-interval 1 retries 1
discover interface serial
template hdlc-serial
template ip-route
exit
hostname host1
cns config initial 10.1.1.1 no-persist inventory
```

Configuring Aggregator Router Interfaces

The following examples show how to configure a standard serial interface and a serial interface bound to a controller on an aggregator router (also known as the DCE). In order for connectivity to be established, the aggregator router must have a point-to-point subinterface configured.

Standard Serial Interface

```
interface Serial0/1
 no ip address
 encapsulation frame-relay
 frame-relay intf-type dce
exit
interface Serial0/1.1 point-to-point
 10.0.0.0 255.255.255.0
 frame-relay interface-dlci 8
```

Serial Interface Bound to a Controller

```
controller T1 0
 framing sf
 linecode ami
 channel-group 0 timeslots 1-24
exit
interface Serial0:0
 no ip address
 encapsulation frame-relay
 frame-relay intf-type dce
exit
interface Serial0:0.1 point-to-point
 ip address ip-address mask
 frame-relay interface-dlci dlci
```

Configuring IP over Frame Relay

The following example shows the bootstrap configuration for configuring IP over Frame Relay on a CPE router:

```
cns template connect setup-frame
 cli encapsulation frame-relay
 exit
cns template connect ip-over-frame
 cli frame-relay interface-dlci ${dlci}
 cli ip address dynamic
 exit
cns template connect ip-route
 cli ip route 10.0.0.0 0.0.0.0 ${next-hop}
 exit
cns connect ip-over-frame
 discover interface Serial
 template setup-frame
 discover dlci
 template ip-over-frame
 template ip-route
exit
cns config initial 10.1.1.1
```

Configuring IP over Frame Relay over T1

The following example shows the bootstrap configuration for configuring IP over Frame Relay over T1 on a CPE router:

```
cns template connect setup-frame
 cli encapsulation frame-relay
 exit
cns template connect ip-over-frame
 cli frame-relay interface-dlci ${dlci}
 cli ip address dynamic
 exit
cns template connect ip-route
 cli ip route 0.0.0.0 0.0.0.0 ${next-hop}
 exit
cns template connect t1-controller
 cli framing esf
 cli linecode b8zs
 cli channel-group 0 timeslots 1-24 speed 56
 exit
cns connect ip-over-frame-over-t1
 discover controller T1
 template t1-controller
 discover interface
 template setup-frame
 discover dlci
 template ip-over-frame
 template ip-route
exit
cns config initial 10.1.1.1
```

Additional References

The following sections provide references related to CNS.

Related Documents

Related Topic	Document Title
CNS commands: complete command syntax, command mode, command history, defaults, usage guidelines, and examples	*Cisco IOS Network Management Command Reference*, Release 12.4T

Related Topic	Document Title
CNS Configuration Engine	*Cisco Intelligence Engine 2100 Configuration Registrar Manual, Release 1.1 or later* *Cisco CNS Configuration Engine Administrator's Guide*
IAD and Router Hardware and Software	• Cisco IAD2420 series hardware and software documents, available online at http://www.cisco.com/univercd/cc/td/doc/product/access/iad/iad2420/index.htm • Cisco 2600 series hardware and software documents, available online at http://www.cisco.com/univercd/cc/td/doc/product/access/acs_mod/cis2600/hw_inst/index.htm • Cisco 3600 series hardware and software documents, available online at http://www.cisco.com/univercd/cc/td/doc/product/access/acs_mod/cis3600/hw_inst/index.htm

Standards

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

MIBs

MIB	MIBs Link
The CNS Flow-Through Provisioning feature provides two mechanisms for accessing MIBs: a nongranular mechanism using SNMP encapsulation and a granular mechanism using XML encapsulation. These mechanisms enable you to access the MIBS currently available in the remote router. The MIBS currently available depend on the router platform and Cisco IOS release.	To locate and download MIBs for selected platforms, Cisco IOS releases, and feature sets, use Cisco MIB Locator found at the following URL: http://www.cisco.com/go/mibs

RFCs

RFC	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 Support website provides extensive online resources, including documentation and tools for troubleshooting and resolving technical issues with Cisco products and technologies. To receive security and technical information about your products, you can subscribe to various services, such as the Product Alert Tool (accessed from Field Notices), the Cisco Technical Services Newsletter, and Really Simple Syndication (RSS) Feeds. Access to most tools on the Cisco Support website requires a Cisco.com user ID and password.	http://www.cisco.com/techsupport

Feature Information for CNS

Table 2 lists the features in this module and provides links to specific configuration information. Only features that were introduced or modified in Cisco IOS Release 12.2(1), 12.0(3)S or a later release appear in the table.

Not all commands may be available in your Cisco IOS software release. For release information about a specific command, see the command reference documentation.

Use Cisco Feature Navigator to find information about platform support and software image support. Cisco Feature Navigator enables you to determine which Cisco IOS and Catalyst OS software images support a specific software release, feature set, or platform. To access Cisco Feature Navigator, go to http://www.cisco.com/go/cfn. An account on Cisco.com is not required.

Note Table 2 lists only the Cisco IOS software release that introduced support for a given feature in a given Cisco IOS software release train. Unless noted otherwise, subsequent releases of that Cisco IOS software release train also support that feature.

Table 2 Feature Information for CNS

Feature Name	Releases	Feature Information
CNS	12.2(25)S 12.2(33) SRA 12.2(33)SB	The CNS feature is a collection of services that can provide remote event-driven configuring of Cisco IOS networking devices and remote execution of some command-line interface (CLI) commands. The following sections provide information about this feature: • Prerequisites for CNS, page 2 • Restrictions for CNS, page 2 • Information About CNS, page 3 • CNS, page 4 • CNS Configuration Agent, page 4 • How to Configure CNS, page 17 • Configuration Examples for CNS, page 36 The following commands were introduced or modified by this feature: **clear cns config stats**, **clear cns counters**, **clear cns event stats**, **cli (cns)**, **cns config cancel**, **cns config initial**, **cns config notify**, **cns config partial**, **cns config retrieve**, **cns connect**, **cns event**, **cns exec**, **cns id**, **cns template connect**, **cns trusted-server**, **debug cns config**, **debug cns exec**, **debug cns xml-parser**, **logging cns-events**, **show cns config stats**, **show cns event connections**, **show cns event stats**, **show cns event subject**.

Table 2 Feature Information for CNS (continued)

Feature Name	Releases	Feature Information
CNS Configuration Agent	12.0(18)ST 12.0(22)S 12.2(2)T 12.2(33)SRA 12.2(33)SB	The CNS Configuration Agent feature supports routing devices by providing the following: • Initial configurations • Incremental (partial) configurations • Synchronized configuration updates The following sections provide information about this feature: • CNS Configuration Agent, page 4 • Initial CNS Configuration, page 4 • Incremental CNS Configuration, page 5 • Synchronized Configuration, page 5 • Configuring the CNS Event and EXEC Agents, page 20 • Troubleshooting CNS Agents, page 33 The following commands were introduced or modified by this feature: **cns config cancel**, **cns config initial**, **cns config partial**, **cns config retrieve**, **debug cns config**, **debug cns xml-parser**, **show cns config outstanding**, **show cns config stats**, **show cns config status**.
CNS Config Retrieve Enhancement with Retry and Interval	12.4(15)T 12.2(33)SRC 12.2(33)SB	The Cisco Networking Services (CNS) Config Retrieve Enhancement with Retry and Interval feature adds two options to the **cns config retrieve** command enabling you to specify an amount of time in seconds to wait before attempting to retrieve a configuration from a trusted server. The number of retries is restricted to 100 to prevent the configuration agent from indefinitely attempting to reach an unreachable server. Use the keyboard combination **Ctrl-Shift-6** to abort the **cns config retrieve** command. The following sections provide information about this feature: • CNS Config Retrieve Enhancement with Retry and Interval, page 4 • Retrieving a CNS Configuration from a Server, page 27 • Retrieving a CNS Configuration from a Server: Example, page 43 The following command was modified by this feature: **cns config retrieve**.

Table 2 Feature Information for CNS (continued)

Feature Name	Releases	Feature Information
CNS Enhanced Results Message	12.2(33)SRA 12.4(4)T	The CNS Enhanced Results Message feature sends a second CNS result message to the subject "cisco.cns.config.results" in addition to the CNS results messages sent to the CNS Event bus after a partial configuration is complete. The following sections provide information about this feature: • CNS Results Messages, page 6 • Configuring the CNS Event and EXEC Agents, page 20 • Configuring a Partial Configuration: Example, page 37 The following command was modified by this feature: **cns config partial**.
CNS Event Agent	12.0(18)ST 12.0(22)S 12.2(2)T 12.2(33)SRA 12.2(33)SB	The CNS Event Agent is part of the Cisco IOS infrastructure that allows Cisco IOS applications to publish and subscribe to events on a CNS Event Bus. CNS Event Agent works in conjunction with the CNS Configuration Agent feature. The following sections provide information about this feature: • CNS Event Agent, page 5 • Configuring the CNS Event and EXEC Agents, page 20 • Troubleshooting CNS Agents, page 33 The following commands were introduced or modified by this feature: **cns event**, **show cns event connections**, **show cns event stats**, **show cnsevent subject**.

Table 2 Feature Information for CNS (continued)

Feature Name	Releases	Feature Information
CNS Flow-Through Provisioning	12.2(2)T 12.2(2)XB 12.2(11)YT 12.2(11)YV	Cisco Networking Services (CNS) Flow-Through Provisioning provides the infrastructure for automated configuration of large numbers of network devices. Based on CNS event and config agents, it eliminates the need for an onsite technician to initialize the device. The result is an automated workflow from initial subscriber-order entry through Cisco manufacturing and shipping to final device provisioning and subscriber billing. This focuses on a root problem of today's service-provider and other similar business models: use of human labor in activating service. The following sections provide information about this feature: • Prerequisites for CNS, page 2 • CNS Flow-Through Provisioning, page 11 • Configuring the CNS Event and EXEC Agents, page 20 • CNS Flow-Through Provisioning: Examples, page 38 The following commands were introduced or modified by this feature: **cns config cancel**, **cns config connect-intf**, **cns config initial**, **cns config partial**, **cns config notify**, **cns event**, **cns id**, **cns inventory**, **cns mib-access encapsulation**, **cns notifications encapsulation**, **config-cli**, **debug cns config**, **debug cns event**, **debug cns management**, **debug cns xml-parser**, **line-cli**, **show cns config connections**, **show cns config outstanding**, **show cns event stats**, **show cns event subject**. **Note** The **cns config connect-intf** command was replaced by the **cns connect** and **cns template connect** commands in Cisco IOS Release 12.3(2)XF, 12.3(8)T, 12.3(9), 12.2(33)SRA, 12.2(31)SB2, and later releases. **Note** The **config-cli** and **line-cli** commands were replaced by the **cli (cns)** command in Cisco IOS Release 12.3(2)XF, 12.3(8)T, 12.3(9), 12.2(33)SRA, 12.2(31)SB2, and later releases.

Table 2 Feature Information for CNS (continued)

Feature Name	Releases	Feature Information
CNS Frame-Relay Zero Touch	12.3(2)XF 12.3(8)T	The CNS Frame Relay Zero Touch feature provides a CNS zero touch deployment solution over Frame Relay where the CPE router discovers its DLCI and IP address dynamically and then contacts a CNS engine to retrieve its full configuration automatically. The following sections provide information about this feature: • Restrictions for CNS, page 2 • CNS Frame Relay Zero Touch, page 15 • Deploying the CNS Router, page 17 • Using the CNS Zero Touch Solution: Examples, page 43 The following commands were introduced or modified by this feature: **cli (cns)**, **cns config connect-intf**, **cns connect,** **cns template connect**, **config-cli**, **discover (cns)**, **line-cli**, **template (cns)**. **Note** The **cns config connect-intf** command was replaced by the **cns connect** and **cns template connect** commands in Cisco IOS Releases 12.3(2)XF, 12.3(8)T, 12.3(9), 12.2(33)SRA, 12.2(31)SB2, and later releases. **Note** The **config-cli** and **line-cli** commands were replaced by the **cli (cns)** command in Cisco IOS Releases 12.3(2)XF, 12.3(8)T, 12.3(9), 12.2(33)SRA, 12.2(31)SB2, and later releases.

Table 2 Feature Information for CNS (continued)

Feature Name	Releases	Feature Information
CNS Image Agent	12.2(33)SEE 12.3(1) 12.2(31)SB2 12.2(33)SRB 12.2(33)SB	The CNS Image Agent feature is an infrastructure in Cisco IOS software to enable automated installation and activation of Cisco IOS images on Cisco IOS networking devices. The following sections provide information about this feature: • Prerequisites for CNS, page 2 • Restrictions for CNS, page 2 • CNS Image Agent, page 5 • Configuring the CNS Image Agent, page 23 • Retrieving a CNS Image from a Server, page 26 • Troubleshooting CNS Agents, page 33 • Enabling and Configuring CNS Agents: Example, page 37 • Retrieving a CNS Image from a Server: Example, page 42 The following commands were introduced or modified by this feature: **clear cns image connections**, **clear cns image status**, **cns id**, **cns image**, **cns image password**, **cns image retrieve**, **cns image retry**, **debug cns image**, **show cns image connections**, **show cns image inventory**, **show cns image status**.
CNS Interactive CLI	12.0(28)S 12.2(18)SXE 12.2(18)SXF2 12.2(33)SRC	The CNS Interactive CLI feature introduces a new XML interface that allows you to send interactive commands to a router, such as commands that generate prompts for user input. The following section provides information about this feature: • CNS Interactive CLI, page 10
CNS Security Enhancement	12.4(9)T 12.2(33)SRA	The CNS Security Enhancement feature improves the security of Cisco Networking Services (CNS) messages by authenticating sender credentials through the use of the Service-Oriented Access Protocol (SOAP) message format. The following sections provide information about this feature: • CNS Message Formats, page 6 • CNS Security Enhancement, page 9 • Configuring CNS Security Features, page 25 The following commands were introduced or modified by this feature: **cns aaa authentication**, **cns message format notification**.

Table 2 Feature Information for CNS (continued)

Feature Name	Releases	Feature Information
CNS Zero Touch	12.3(9)	The CNS Zero Touch feature provides a zero touch deployment solution where the router contacts a CNS configuration engine to retrieve its full configuration automatically. The following sections provide information about this feature: • Prerequisites for CNS, page 2 • Restrictions for CNS, page 2 • CNS Zero Touch, page 15 • Deploying the CNS Router, page 17 • Using the CNS Zero Touch Solution: Examples, page 43 The following commands were introduced or modified by this feature: **cli (cns)**, **cns config connect-intf**, **cns connect**, **cns template connect**, **config-cli**, **discover (cns)**, **line-cli, template (cns)**. **Note** The **cns config connect-intf** command was replaced by the **cns connect** and **cns template connect** commands in Cisco IOS Release 12.3(2)XF, 12.3(8)T, 12.3(9), 12.2(33)SRA, 12.2(31)SB2, and later releases. **Note** The **config-cli** and **line-cli** commands were replaced by the **cli (cns)** command in Cisco IOS Release 12.3(2)XF, 12.3(8)T, 12.3(9), 12.2(33)SRA, 12.2(31)SB2, and later releases.
Command Scheduler	12.3(1) 12.2(33)SRA 12.2(33)SRC 12.2(33)SB	The Command Scheduler feature provides the ability to schedule some EXEC command-line interface (CLI) commands to run at specific times or at specified intervals. The following sections provide information about this feature: • Restrictions for CNS, page 2 • Command Scheduler, page 10 • Configuring Command Scheduler Policy Lists and Occurrences, page 28 • Command Scheduler Policy Lists and Occurrences: Examples, page 41 The following commands were introduced or modified by this feature: **cli**, **debug kron**, **kron occurrence**, **kron policy-list**, **policy-list**, **show kron schedule**.
CNS	Cisco IOS XE Release 2.1	This feature was introduced on Cisco ASR 1000 Series Routers.

Table 2 Feature Information for CNS (continued)

Feature Name	Releases	Feature Information
CNS - Configuration Agent	Cisco IOS XE Release 2.1	This feature was introduced on Cisco ASR 1000 Series Routers.
CNS - Interactive CLI	Cisco IOS XE Release 2.1	This feature was introduced on Cisco ASR 1000 Series Routers.
CNS Config Retrieve Enhancement with Retry and Interval	Cisco IOS XE Release 2.1	This feature was introduced on Cisco ASR 1000 Series Routers.
Command Scheduler (Kron)	Cisco IOS XE Release 2.1	This feature was introduced on Cisco ASR 1000 Series Routers.
Command Scheduler (Kron) Policy for System Startup	Cisco IOS XE Release 2.1	This feature was introduced on Cisco ASR 1000 Series Routers.

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

Distributed Director

DistributedDirector Enhancements for Cisco IOS Release 12.1(5)T

This feature module describes the DistributedDirector Enhancements for Cisco IOS Release 12.1(5)T feature and includes the following sections:

- Feature Overview, page 1
- Supported Platforms, page 3
- Supported Standards, MIBs, and RFCs, page 3
- Configuration Tasks, page 4
- Configuration Examples, page 5
- Command Reference, page 6

Feature Overview

The DistributedDirector Enhancements for Cisco IOS Release 12.1(5)T feature consists of the following modified features for the DistributedDirector, which were introduced in Cisco IOS Release 11.1(28)IA:

- Enhanced Fault Tolerance with Multiple Resource Records
- Event Recording with Syslog
- Enhanced Server Verification with Multiple Port Connect Tests

The DistributedDirector Enhancements for Cisco IOS Release 12.1(5)T feature also consists of several unrelated new commands. These commands can be found in the Command Reference section.

Enhanced Fault Tolerance with Multiple Resource Records

Before this enhancement, DistributedDirector would return a single Resource Record (RR) in each Domain Name System (DNS) response. A single RR is normally sufficient, but for some applications, server failover will occur more rapidly when applications are provided IP addresses of multiple servers.

The Enhanced Fault Tolerance with Multiple Resource Records feature enables DistributedDirector to return multiple RRs. The number of RRs returned in a single reply is configurable. The default number of RRs returned is one.

Americas Headquarters:
Cisco Systems, Inc., 170 West Tasman Drive, San Jose, CA 95134-1706 USA

Event Recording with Syslog

The Event Recording with Syslog feature enables DistributedDirector to log events by way of the industry-standard syslog system. Server state is logged, providing a useful log of when servers are considered up or down. The logging priority level is notification with priority level five. Additionally, the server selection process, DNS request, and DNS response may be logged. The logging priority level is informational with priority level six.

Enhanced Server Verification with Multiple Port Connect Tests

Before this enhancement, DistributedDirector could evaluate server status by performing a TCP connect test to a single server port. The Enhanced Server Verification with Multiple Port Connect Tests feature allows multiple port connect tests to be specified. If any one of the connect tests fails, the server is considered down.

Benefits

The features provided in Cisco IOS Release 12.1(5)T help make networks that use DistributedDirector more robust. These features ensure that applications have more useful information and perform better server verification, and they allow administrators to track DistributedDirector better. In particular:

- The Enhanced Fault Tolerance with Multiple Resource Records feature provides better fault tolerance for clients.
- The Event Recording with Syslog feature provides the ability to examine DNS traffic and the way in which servers are chosen.
- The Enhanced Server Verification with Multiple Port Connect Tests feature better reflects the reality that some services span several ports and require that all ports be up.

Restrictions

Enhanced Fault Tolerance with Multiple Resource Records

Configuring DistributedDirector to return a large number of records can reduce the benefit of using DistributedDirector to select the best server.

Event Recording with Syslog

Extensive syslog output is provided when logging server selection. Therefore, this feature should not be used when a heavy request load is expected.

Related Documents

For more information on the Cisco DistributedDirector, see the following documents, which are located on Cisco Connection Online (CCO) at http://www.cisco.com/public/sw-center/netmgmt/cmtk/mibs.shtml:

- Cisco DistributedDirector 2500 Series Installation and Configuration Guide
- Cisco DistributedDirector 4700-M Installation and Configuration Guide
- Release Notes for Cisco DistributedDirector System Software
- Cisco DistributedDirector Enhancements for Release 11.1(18)IA
- Cisco DistributedDirector Enhancements for Release 11.1(25)IA

- Cisco DistributedDirector Enhancements for Release 11.1(28)IA

Supported Platforms

- Cisco DistributedDirector 2501
- Cisco DistributedDirector 2502
- Cisco DistributedDirector 4700

Determining Platform Support Through Cisco Feature Navigator

Cisco IOS software is packaged in feature sets that support specific platforms. To get updated information regarding platform support for this feature, access Cisco Feature Navigator. Cisco Feature Navigator dynamically updates the list of supported platforms as new platform support is added for the feature.

Cisco Feature Navigator is a web-based tool that enables you to determine which Cisco IOS software images support a specific set of features and which features are supported in a specific Cisco IOS image. You can search by feature or release. Under the release section, you can compare releases side by side to display both the features unique to each software release and the features in common.

To access Feature Navigator, you must have an account on Cisco.com. If you have forgotten or lost your account information, send a blank e-mail to cco-locksmith@cisco.com. An automatic check will verify that your e-mail address is registered with Cisco.com. If the check is successful, account details with a new random password will be e-mailed to you. Qualified users can establish an account on Cisco.com by following the directions at http://www.cisco.com/register.

Cisco Feature Navigator is updated regularly when major Cisco IOS software releases and technology releases occur. For the most current information, go to the Cisco Feature Navigator home page at the following URL:

http://www.cisco.com/go/fn

Availability of Cisco IOS Software Images

Platform support for particular Cisco IOS software releases is dependent on the availability of the software images for those platforms. Software images for some platforms may be deferred, delayed, or changed without prior notice. For updated information about platform support and availability of software images for each Cisco IOS software release, refer to the online release notes or, if supported, Cisco Feature Navigator.

Supported Standards, MIBs, and RFCs

Standards

None

MIBs

None

To obtain lists of supported MIBs by platform and Cisco IOS release, and to download MIB modules, go to the Cisco MIB website on Cisco.com at the following URL:

http://www.cisco.com/public/sw-center/netmgmt/cmtk/mibs.shtml

RFCs

None

Configuration Tasks

See the following sections for configuration tasks for this feature. Each task is optional.

- Configuring Enhanced Fault Tolerance with Multiple Resource Records(Optional)
- Configuring Event Recording with Syslog (Optional)
- Configuring Enhanced Server Verification with Multiple Port Connect Tests (Optional)
- Verifying Enhanced Fault Tolerance with Multiple Resource Records (Optional)
- Verifying Event Recording with Syslog (Optional)
- Verifying Enhanced Server Verification with Multiple Port Connect Tests (Optional)

Configuring Enhanced Fault Tolerance with Multiple Resource Records

To configure the Enhanced Fault Tolerance with Multiple Resource Records feature on the DistributedDirector for a host name, use the following command in global configuration mode:

Command	Purpose
Router(config)# **ip director host** *host-name* [**a** | **mx**] **multiple** *integer*	Configures the number of RRs that the DistributedDirector returns for each DNS response.

Configuring Event Recording with Syslog

To configure the Event Recording with Syslog feature on the DistributedDirector for a host name, use the following commands in global configuration mode:

Command	Purpose
Router(config)# **logging** *host* Router(config)# **logging trap informational** Router(config)# **ip director host** *host-name* [**a** | **mx**] **logging**	Configures the DistributedDirector to log events to syslog.

Configuring Enhanced Server Verification with Multiple Port Connect Tests

To configure the Enhanced Server Verification with Multiple Port Connect Tests feature on the DistributedDirector, use the following commands in global configuration mode:

Command	Purpose
Router(config)# **ip director host** *host-name* [**a** | **mx**] **connect** *port-1* [**interval**] *connection-interval n* Router(config)# **ip director host** *host-name* [**a** | **mx**] **connect** *port-2* [**interval**] *connection-interval n* Router(config)# **ip director host** *host-name* [**a** | **mx**] **connect** *port-n* [**interval**] *connection-interval n*	Enables the DistributedDirector to verify that a server is available. When you configure multiple **ip director host connect** commands for the same host name but with different port numbers, the DistributedDirector verifies that all of the ports are accessible. If any of the ports is not accessible, the host is considered down.

Verifying Enhanced Fault Tolerance with Multiple Resource Records

To verify that the DistributedDirector is configured to return the best servers for RRs for each DNS, use the **show ip director hosts** command.

Verifying Event Recording with Syslog

To verify that the DistributedDirector is configured to send to syslog the DNS request and response information, use the **show ip director hosts** command.

Verifying Enhanced Server Verification with Multiple Port Connect Tests

To verify that the DistributedDirector is configured with a specific connection interval to specified distributed servers, use the **show ip director hosts** command.

Configuration Examples

This section provides the following configuration examples:

- Enhanced Fault Tolerance with Multiple Resource Records Example
- Event Recording with Syslog Example
- Enhanced Server Verification with Multiple Port Connect Tests Example

Enhanced Fault Tolerance with Multiple Resource Records Example

In the following examples, the DistributedDirector is configured to return the best three servers for A resource record on host name www.xyz.com, the best two servers for A resource record on host name alias.xyz.com, and the best two servers for MX resource mail.xyz.com, respectively:

```
ip director host www.xyz.com multiple 3
ip director host alias.xyz.com a multiple 2
ip director host mail.xyz.com mx multiple 2
```

Event Recording with Syslog Example

Before configuring the DistributedDirector to syslog events regarding DNS requests on a specific resource record, the following must be typed on the command line:

```
logging 172.21.34.2
logging trap informational
```

Note The IP address specified above is the IP address of the log server in which the syslog messages get recorded.

In the following examples, the DistributedDirector is configured to syslog events regarding DNS requests on A resource record for host name www.xyz.com, DNS requests on A resource record for host name alias.xyz.com, and DNS requests on MX host name mail.xyz.com, respectively:

```
ip director host www.xyz.com logging
ip director host alias.xyz.com a logging
ip director host mail.xyz.com mx logging
```

Enhanced Server Verification with Multiple Port Connect Tests Example

In the following example, the DistributedDirector is configured with a connection interval of 5 minutes to distributed servers on port 80 and port 90. The distributed servers will only be considered accessible if both port 80 and port 90 are accessible:

```
ip director host www.xyz.com connect 80 5
ip director host www.xyz.com connect 90 5
```

Command Reference

The following new and modified commands are pertinent to this feature. To see the command pages for these commands and other commands used with this feature, go to the *Cisco IOS Master Commands List*, Release 12.4, at http://www.cisco.com/univercd/cc/td/doc/product/software/ios124/124mindx/124index.htm.

New Commands

- **ip director access-group local**
- **ip director drp retries**
- **ip director drp timeout**
- **ip director drp timeout lookup**
- **ip director drp timeout measure**
- **ip director host active-close**
- **ip director host tolerance**
- **ip director host verify-url**
- **ip director server reinstatement**
- **ip director server route-map**
- **ip director server verify-url**

- **ip director server weights**
- **show ip director drp**

Modified Commands

- **ip director host connect**
- **ip director host logging**
- **ip director host multiple**

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

DNS Server Support for NS Records

This feature module describes the DNS Server Support for NS Records feature and includes the following sections:

- Feature Overview, page 1
- Supported Platforms, page 2
- Supported Standards, MIBs, and RFCs, page 2
- Configuration Tasks, page 2
- Configuration Examples, page 3
- Command Reference, page 4

Feature Overview

Domain Name System (DNS) is a client/server mechanism used to access a distributed database. The server portion of the DNS client/server mechanism is the name server (NS). An NS can be responsible for presenting information about a portion of the DNS distributed database or can be a forwarding/caching NS. In the latter case, the NS queries other NSs rather than maintaining a local portion of the DNS database.

DistributedDirector has improved server load-balancing capacity with the DNS Server Support for NS Records feature. This feature adds support for NS records to the Cisco IOS DNS server. With this feature, the DistributedDirector can distribute the server-selection process to multiple DistributedDirectors, improving overall server capacity.

Benefits

This feature allows an NS to delegate server responsibility for a domain by returning an NS record when queried. This function is useful to DistributedDirector because a computationally load can be distributed over a large number of DistributedDirectors, so each DistributedDirector can be free to perform computational expensive actions to select the best server.

Americas Headquarters:
Cisco Systems, Inc., 170 West Tasman Drive, San Jose, CA 95134-1706 USA

Related Documents

For more information on the Cisco DistributedDirector, see the following documents, which are located on Cisco.com at http://www.cisco.com/univercd/cc/td/doc/product/iaabu/distrdir/index.htm:

- Cisco DistributedDirector 4700-M Installation and Configuration Guide
- Release Notes for Cisco DistributedDirector System Software
- Cisco DistributedDirector Enhancements for Release 11.1(18)IA
- Cisco DistributedDirector Enhancements for Release 11.1(25)IA
- Cisco DistributedDirector Enhancements for Release 11.1(28)IA
- Cisco DistributedDirector Enhancements for Release 12.1(5)T
- Dynamic Feedback Protocol Support in DistributedDirector

Supported Platforms

- Cisco DistributedDirector 4500

Supported Standards, MIBs, and RFCs

Standards

None

MIBs

None

To obtain lists of supported MIBs by platform and Cisco IOS release, and to download MIB modules, go to the Cisco MIB website on Cisco.com at the following URL:

http://www.cisco.com/public/sw-center/netmgmt/cmtk/mibs.shtml

RFCs

DNS is defined in RFC 1035.

Configuration Tasks

See the following sections for configuration tasks for this feature. Each task in the list is identified as either required or optional.

- Configuring DNS Server Support for NS Records (required)
- Verifying DNS Server Support for NS Records (optional)

Configuring DNS Server Support for NS Records

To configure the DistributedDirector to create an NS resource record to be returned when the DNS server is queried for the associated domain, use the following commands in global configuration mode:

	Command	Purpose
Step 1	`Router(config)# ip host www.xyz.com ns ns.xyz.com`	Configures the DistributedDirector to create an NS resource record to be returned when the DNS server is queried for the associated domain.
Step 2	`Router(config)# ip host ns.xyz.com 10.0.0.1 10.0.0.2 10.0.0.3`	Defines a static host-name-to-address mapping in the host cache.
Step 3	`Router(config)# ip director host ns.xyz.com priority random 1`	Configures the order in which the DistributedDirector considers metrics when picking a server.
Step 4	`Router(config)# ip dns primary xyz.com soa ns.xyz.com`	Identifies the DistributedDirector as the primary DNS NS for a domain and as the SOA record source.

Verifying DNS Server Support for NS Records

To verify that the DistributedDirector is configured with NS record support, use the **show running-config** command or the **show host** command.

Configuration Examples

This section provides the following configuration example:

- DNS Server Support for NS Records Example

DNS Server Support for NS Records Example

The following example shows a top-level DistributedDirector using a low-cost metric, such as portion or random, to distribute load over second-level DistributedDirectors. Second-level DistributedDirectors then use more expensive metrics, such as drp-ext or drp-rtt, to perform more precise server selection. The relevant portions of this configuration are show below:

Top-Level DistributedDirector

```
ip host www.xyz.com ns ns.xyz.com
ip host ns2.xyz.com 10.0.0.1 10.0.0.2 10.0.0.3
ip director host ns.xyz.com priority random 1
ip dns primary www.xyz.com soa ns2.xyz.com
```

Second-Level DistributedDirector

```
ip host www.xyz.com 10.0.0.4 10.0.0.5 10.0.0.6
ip director host www.xyz.com priority drp-ext 1
ip director host www.xyz.com priority drp-rtt 2
ip director server 10.0.0.4 drp-association 10.0.0.7
ip director server 10.0.0.5 drp-association 10.0.0.8
ip director server 10.0.0.6 drp-association 10.0.0.9
```

Command Reference

The following new command is pertinent to this feature. To see the command pages for this command and other commands used with this feature, go to the *Cisco IOS Master Commands List*, Release 12.4, at http://www.cisco.com/univercd/cc/td/doc/product/software/ios124/124mindx/124index.htm.

- **ip host ns**

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

DistributedDirector Configurable Cache

Feature History

Release	Modification
12.2(8)T	This feature was introduced.

This document describes the DistributedDirector Configurable Cache feature in Cisco IOS Release 12.2(8)T. It includes the following sections:

- Feature Overview, page 1
- Supported Platforms, page 2
- Supported Standards, MIBs, and RFCs, page 3
- Prerequisites, page 3
- Configuration Tasks, page 3
- Configuration Examples, page 5
- Command Reference, page 6

Feature Overview

DistributedDirector maintains an internal cache of entries that is dynamically configurable. This internal configurable cache consists of sorting events that occur on a per-client basis. Users can configure both the size of this internal cache and the amount of time for which the DistributedDirector system will retain per-client sorting information.

The DistributedDirector Configurable Cache feature allows users to configure their systems in order to limit the amount of memory that DistributedDirector uses for Domain Name System (DNS) caching. When a query that is a duplicate of a previous query comes from the client within the cache timeout period, the same response can be produced without the use of any Director Response Protocol (DRP) queries or sorting.

Americas Headquarters:
Cisco Systems, Inc., 170 West Tasman Drive, San Jose, CA 95134-1706 USA

The DistributedDirector Cache Auto Refresh feature works in the background to continuously update all entries in the DistributedDirector cache. Once this background refresh feature is initiated, DistributedDirector periodically updates all expired cache entries. The DistributedDirector cache saves the latest answers to all past DNS queries received since cache auto refresh was initiated, and any repeat request is served directly from the cache when caching is enabled.

Benefits

- Use of this feature limits the amount of memory that DistributedDirector uses for DNS caching.
- This feature allows the user to configure how long an entry remains in the cache.

Related Features and Technologies

DistributedDirector Cache Auto Refresh

Related Documents

DistributedDirector Cache Auto Refresh, Cisco IOS Release 12.2(8)T feature module

Supported Platforms

- Cisco 2600 series
- Cisco 3620 series
- Cisco 3640 series
- Cisco 3660 series
- Cisco 3725 series
- Cisco 3745 series
- Cisco 7200 series

Determining Platform Support Through Cisco Feature Navigator

Cisco IOS software is packaged in feature sets that support specific platforms. To get updated information regarding platform support for this feature, access Cisco Feature Navigator. Cisco Feature Navigator dynamically updates the list of supported platforms as new platform support is added for the feature.

Cisco Feature Navigator is a web-based tool that enables you to quickly determine which Cisco IOS software images support a specific set of features and which features are supported in a specific Cisco IOS image. You can search by feature or release. Under the release section, you can compare releases side by side to display both the features unique to each software release and the features in common.

To access Feature Navigator, you must have an account on Cisco.com. If you have forgotten or lost your account information, send a blank e-mail to cco-locksmith@cisco.com. An automatic check will verify that your e-mail address is registered with Cisco.com. If the check is successful, account details with a new random password will be e-mailed to you. Qualified users can establish an account on Cisco.com by following the directions at http://www.cisco.com/register.

Cisco Feature Navigator is updated regularly when major Cisco IOS software releases and technology releases occur. For the most current information, go to the Cisco Feature Navigator home page at the following URL: http://www.cisco.com/go/fn.

Supported Standards, MIBs, and RFCs

Standards

No new standards are supported by this feature.

MIBs

No new MIBs are supported by this feature.

To obtain lists of supported MIBs by platform and Cisco IOS release, and to download MIB modules, go to the Cisco MIB website on Cisco.com at the following URL:

http://www.cisco.com/public/sw-center/netmgmt/cmtk/mibs.shtml

RFCs

No new RFCs are supported by this feature.

Prerequisites

The sorting cache must be enabled on DistributedDirector. To enable the sorting cache, use the **ip director cache** command.

Configuration Tasks

See the following sections for configuration tasks for this feature. Each task in the list is identified as either required or optional.

- Configuring the Size of the Cache (optional)
- Configuring How Long the System Retains Sorting Information (optional)

Configuring the Size of the Cache

To configure the variable size of the DistributedDirector cache, use the following commands in global configuration mode:

	Command	Purpose
Step 1	`Router(config)# `**`ip director cache`**	Enables the sorting cache on DistributedDirector.
Step 2	`Router(config)# `**`ip director cache size 1500`**	Configures the maximum number of cache entries, where *entries* equals 1500.

Configuring How Long the System Retains Sorting Information

To configure how long the DistributedDirector system will retain per-client sorting information, use the following command in global configuration mode:

	Command	Purpose
Step 1	`Router(config)# `**`ip director cache`**	Enables the sorting cache on DistributedDirector.
Step 2	`Router(config)# `**`ip director cache time 100`**	Configures how long the DistributedDirector system will retain per-client sorting information, where *seconds* equals 100.

Verifying DistributedDirector Cache Information

To show DistributedDirector cache information, use the **show ip director cache** command.

```
Router# show ip director cache

Director cache is on
Cache current size = 2 maximum size = 2000
Cache time for sort cache entries:60 secs
Director sort cache hits = 8
Entries:
www.myserver.org:for client 172.17.2.78, used 3 times, valid for:
00:00:42
server 172.21.34.10, rank 0, priority 0
              random incomplete:0
              DRP route lookup external to AS incomplete:0
              administrative preference incomplete:0
              DRP route lookup internal to AS complete:40
              DRP distance to associated server incomplete:0
              portion incomplete:0
              Round-trip time from DRP to client incomplete:0
              DFP originated weight incomplete:0
              Route-map evaluation incomplete:0
              Boomerang evaluation incomplete:0
server 172.21.34.10, rank 0, priority 0, best
              random incomplete:0
              DRP route lookup external to AS incomplete:0
              administrative preference incomplete:0
              DRP route lookup internal to AS complete:30
              DRP distance to associated server incomplete:0
              portion incomplete:0
              Round-trip time from DRP to client incomplete:0
              DFP originated weight incomplete:0
              Route-map evaluation incomplete:0
              Boomerang evaluation incomplete:0
www.boom1.com:for client 172.17.2.78, used 5 times, valid for:00:00:13
server 172.21.34.10, rank 0, priority 0
              random incomplete:0
```

```
                DRP route lookup external to AS incomplete:0
                administrative preference incomplete:0
                DRP route lookup internal to AS complete:40
                DRP distance to associated server incomplete:0
                portion incomplete:0
                Round-trip time from DRP to client incomplete:0
                DFP originated weight incomplete:0
                Route-map evaluation incomplete:0
                Boomerang evaluation incomplete:0
server 172.21.34.10, rank 0, priority 0, best
                random incomplete:0
                DRP route lookup external to AS incomplete:0
                administrative preference incomplete:0
                DRP route lookup internal to AS complete:30
                DRP distance to associated server incomplete:0
                portion incomplete:0
                Round-trip time from DRP to client incomplete:0
                DFP originated weight incomplete:0
                Route-map evaluation incomplete:0
                Boomerang evaluation incomplete:0
```

Configuration Examples

This section provides the following configuration examples:

- Configuring the Size of the Cache Example
- Configuring How Long the System Retains Sorting Information Example

Configuring the Size of the Cache Example

The following example configures the maximum number of cache entries:

```
Router(config)# ip director cache size 1500

Cache size shrinked to 1500

Router# show running-config

ip host myhost 172.18.18.10 172.18.18.20 172.18.18.30
.
.
.
ip director host myhost
ip dns primary myhost soa myhost myhost@com
no ip director drp synchronized
ip director cache size 1500
```

Configuring How Long the System Retains Sorting Information Example

The following example configures how long the DistributedDirector system will retain per-client sorting information:

```
Router(config)# ip director cache time 100

Router# show running-config

ip host myhost 172.18.18.10 172.18.18.20 172.18.18.30
```

```
.
.
.
ip director host myhost
ip dns primary myhost soa myhost myhost@com
no ip director drp synchronized
ip director cache time 100
```

Command Reference

The following new commands are pertinent to this feature. To see the command pages for these commands and other commands used with this feature, go to the *Cisco IOS Master Commands List*, Release 12.4, at http://www.cisco.com/univercd/cc/td/doc/product/software/ios124/124mindx/124index.htm.

- **ip director cache size**
- **ip director cache time**

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

DistributedDirector Cache Auto Refresh

Feature History

Release	Modification
12.2(8)T	This feature was introduced.

This document describes the DistributedDirector Cache Auto Refresh feature in Cisco IOS Release 12.2(8)T. It includes the following sections:

- Feature Overview, page 1
- Supported Platforms, page 2
- Supported Standards, MIBs, and RFCs, page 3
- Prerequisites, page 4
- Configuration Tasks, page 4
- Configuration Examples, page 5
- Command Reference, page 5

Feature Overview

The DistributedDirector Cache Auto Refresh feature works in the background to continuously update all entries in the DistributedDirector cache. Once this background refresh feature is initiated, DistributedDirector periodically updates all expired cache entries. The DistributedDirector cache saves the latest answers to all past Domain Name System (DNS) queries received since cache auto refresh was initiated, and any repeat request is served directly from the cache when caching is enabled.

The new **ip director cache refresh** command enables the automatic background refresh feature for the DistributedDirector cache.

Americas Headquarters:
Cisco Systems, Inc., 170 West Tasman Drive, San Jose, CA 95134-1706 USA

Benefits

Once the cache auto refresh feature is enabled, the cache will actively and continuously update every expired entry. This feature allows DistributedDirector to return answers to queries according to the latest and most accurate network information.

Related Technologies

DistributedDirector Configurable Cache

Related Documents

DistributedDirector Configurable Cache, Cisco IOS Release 12.2(8)T feature module

Supported Platforms

- Cisco 2600 series
- Cisco 3620 series
- Cisco 3640 series
- Cisco 3660 series
- Cisco 3725 series
- Cisco 3745 series
- Cisco 7200 series

Determining Platform Support Through Feature Navigator

Cisco IOS software is packaged in feature sets that support specific platforms. To get updated information regarding platform support for this feature, access Feature Navigator. Feature Navigator dynamically updates the list of supported platforms as new platform support is added for the feature.

Feature Navigator is a web-based tool that enables you to quickly determine which Cisco IOS software images support a specific set of features and which features are supported in a specific Cisco IOS image.

To access Feature Navigator, you must have an account on Cisco.com. If you have forgotten or lost your account information, send a blank e-mail to cco-locksmith@cisco.com. An automatic check will verify that your e-mail address is registered with Cisco.com. If the check is successful, account details with a new random password will be e-mailed to you. Qualified users can establish an account on Cisco.com by following the directions at http://www.cisco.com/register.

Feature Navigator is updated regularly when major Cisco IOS software releases and technology releases occur. For the most current information, go to the Feature Navigator home page at the following URL:

http://www.cisco.com/go/fn

Supported Standards, MIBs, and RFCs

Standards

No new standards are supported by this feature.

MIBs

No new MIBs are supported by this feature.

To obtain lists of supported MIBs by platform and Cisco IOS release, and to download MIB modules, go to the Cisco MIB website on Cisco.com at the following URL:

http://www.cisco.com/public/sw-center/netmgmt/cmtk/mibs.shtml

RFCs

No new RFCs are supported by this feature.

Prerequisites

The sorting cache must be enabled on DistributedDirector. To enable the sorting cache, use the **ip director cache** command.

Configuration Tasks

See the following section for configuration tasks for cache auto refresh. Each task in the list is identified as either required or optional.

- Enabling Cache Auto Refresh (required)

Enabling Cache Auto Refresh

To enable cache auto refresh for DistributedDirector, use the **ip director cache refresh** command.

	Command	Purpose
Step 1	`Router(config)#` **`ip director cache`**	Enables the sorting cache on DistributedDirector.
Step 2	`Router(config)#` **`ip director cache refresh`**	Enables the DistributedDirector Cache Auto Refresh feature.

To turn off cache auto refresh, use the **no ip director cache refresh** command.

	Command	Purpose
Step 1	`Router(config)#` **`no ip director cache refresh`**	Disables the DistributedDirector Cache Auto Refresh feature.

Verifying the Cache Auto Refresh Feature

To verify that the DistributedDirector Cache Auto Refresh feature is configured, enter the **show running-config** command.

```
Router(config)# ip director cache

Router(config)# ip director cache refresh

Router# show running-config

ip host myhost 172.22.2.10 172.22.2.20 172.22.2.30
.
.
.
ip director cache refresh
```

Configuration Examples

This section provides the following configuration example:

- Enabling the Cache Auto Refresh Feature Example

Enabling the Cache Auto Refresh Feature Example

In the following example, the cache auto refresh background feature for DistributedDirector is enabled:

```
Router(config)# ip director cache

Router(config)# ip director cache refresh

Router# show running-config

ip host myhost 172.22.2.10 172.22.2.20 172.22.2.30
.
.
.
ip director cache refresh
```

Command Reference

The following new commands are pertinent to this feature. To see the command pages for these commands and other commands used with this feature, go to the *Cisco IOS Master Commands List*, Release 12.4, at http://www.cisco.com/univercd/cc/td/doc/product/software/ios124/124mindx/124index.htm.

- **ip director cache refresh**

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

DistributedDirector Boomerang Support

Feature History

Release	Modification
12.2(8)T	Boomerang support was introduced.

This document describes boomerang support for DistributedDirector in Cisco IOS Release 12.2(8)T. It includes the following sections:

- Feature Overview, page 1
- Supported Platforms, page 3
- Supported Standards, MIBs, and RFCs, page 3
- Prerequisites, page 4
- Configuration Tasks, page 4
- Configuration Examples, page 6
- Command Reference, page 7

Feature Overview

Boomerang is a Director Response Protocol (DRP) metric for DistributedDirector. The boomerang server provides a way to select a content server with the fastest response time from a group of redundant content servers. Instead of relying on static maps, boomerang dynamically recognizes problems such as congestion and link failures and avoids them. The content server with the fastest response time, as determined by the priority of the configured metrics, is determined to be the best site.

When the boomerang metric is active, DistributedDirector instructs the DRP to send Domain Name System (DNS) responses directly back to the querying client. The DNS response contains the addresses of the sites associated with the particular DRP agent. All involved DRPs send back their DNS responses at the same time. The packet of the DRP that is closest to the client will arrive first. The client may take the first answer and ignore subsequent ones, a standard behavior of all local DNS server implementations. The DRP agent allows configuration for full boomerang support.

Americas Headquarters:
Cisco Systems, Inc., 170 West Tasman Drive, San Jose, CA 95134-1706 USA

The boomerang metric may or may not be used by DistributedDirector. Whether the boomerang metric is used depends on whether other metrics are specified at higher priority (and therefore have a lower priority number) than the boomerang metric. If a metric at higher priority successfully determines the best site, then that is what DistributedDirector uses. DistributedDirector reaches the boomerang metric only if all other metrics of higher priority than boomerang are unable to determine the best site.

If and when the boomerang metric is reached, all other metrics after it (that is, metrics that have a lower priority and a higher priority number) are effectively ignored. They are ignored because the actual resolution of the best site is determined not by DistributedDirector but by which boomerang reply reaches the DNS client first. DistributedDirector is not made aware of the best site as determined by the boomerang metric.

The boomerang metric can be used alone or along with other metrics at the same or different priority levels. If boomerang is specified at the same priority as other metrics, then boomerang decides the best site. If boomerang is specified with other metrics at different priorities, then the higher-priority metrics are examined in turn until there is no tie among sites, with the result that the best site can be determined. If the consideration extends to the boomerang metric, then boomerang is the deciding metric. All other metrics of a higher priority number (lower priority) than boomerang are ignored. The concept of weight does not apply to the boomerang metric.

The DRP is a simple User Datagram Protocol (UDP)-based application developed by Cisco Systems. It enables the Cisco DistributedDirector product to query routers (DRP Server Agents) in the field for Border Gateway Protocol (BGP) and Interior Gateway Protocol (IGP) routing table metrics between distributed servers and clients. DistributedDirector, a separate standalone product, uses DRP to transparently redirect end-user service requests to the topologically closest responsive server. DRP enables DistributedDirector to provide dynamic, scalable, and “network intelligent” Internet traffic load distribution among multiple geographically dispersed servers.

DRP Server Agents are border routers (or peers to border routers) that support the geographically distributed servers for which DistributedDirector service distribution is desired. Note that, because DistributedDirector makes decisions based on BGP and IGP information, all DRP Server Agents must have access to full BGP and IGP routing tables.

Benefits

The boomerang metric provides a way to select a site with the fastest response time. Instead of relying on static maps, it dynamically recognizes congestion and link failures and avoids them.

Restrictions

Both DistributedDirector and the DRP agents should be able to communicate with each other using the boomerang protocol. Therefore, when DistributedDirector is upgraded to include the boomerang functionality, the DRP agents must be made aware of the presence of the boomerang protocol.

Related Features and Technologies

- Director Response Protocol
- User Datagram Protocol
- Border Gateway Protocol
- Interior Gateway Protocol

Related Documents

- *Boomerang Support in the DRP Agent*, Cisco IOS Release 12.2(8)T feature module
- "Configuring IP Services" chapter of *Cisco IOS Configuration Fundamentals Configuration Guide,* Release 12.2

Supported Platforms

- Cisco 2600 series
- Cisco 3620 series
- Cisco 3640 series
- Cisco 3660 series
- Cisco 3725 series
- Cisco 3745 series
- Cisco 7200 series

Determining Platform Support Through Feature Navigator

Cisco IOS software is packaged in feature sets that support specific platforms. To get updated information regarding platform support for this feature, access Feature Navigator. Feature Navigator dynamically updates the list of supported platforms as new platform support is added for the feature.

Feature Navigator is a web-based tool that enables you to quickly determine which Cisco IOS software images support a specific set of features and which features are supported in a specific Cisco IOS image.

To access Feature Navigator, you must have an account on Cisco.com. If you have forgotten or lost your account information, send a blank e-mail to cco-locksmith@cisco.com. An automatic check will verify that your e-mail address is registered with Cisco.com. If the check is successful, account details with a new random password will be e-mailed to you. Qualified users can establish an account on Cisco.com by following the directions at http://www.cisco.com/register.

Feature Navigator is updated regularly when major Cisco IOS software releases and technology releases occur. For the most current information, go to the Feature Navigator home page at the following URL:

http://www.cisco.com/go/fn

Supported Standards, MIBs, and RFCs

Standards

No new or modified standards are supported by this feature.

MIBs

No new MIBs are supported by this feature.

To obtain lists of supported MIBs by platform and Cisco IOS release, and to download MIB modules, go to the Cisco MIB website on Cisco.com at the following URL:

http://www.cisco.com/public/sw-center/netmgmt/cmtk/mibs.shtml

RFCs

No new RFCs are supported by this feature.

Prerequisites

You must be using DistributedDirector with DRP metrics, including boomerang.

Configuration Tasks

See the following sections for configuration tasks for the boomerang metric feature. Each task in the list is identified as either required or optional.

- Setting DistributedDirector and DRP Clock Synchronization (optional)
- Configuring the Default Metric (optional)
- Specifying the Host Priority of the Boomerang Metric (required)

Setting DistributedDirector and DRP Clock Synchronization

To activate clock synchronization between of the DistributedDirector and the DRP clocks, use the **ip director drp synchronized** command.

Command	Purpose
`Router(config)# ip director drp synchronized`	Activates or deactivates clock synchronization between DistributedDirector and the DRP.

To deactivate clock synchronization between DistributedDirector and DRPs, use the **no ip director drp synchronized** command.

Command	Purpose
`Router(config)# no ip director drp synchronized`	Deactivates clock synchronization between DistributedDirector and the DRP.

Configuring the Default Metric

To set the boomerang metric as the default metric, use the **ip director default priorities boomerang** command.

Command	Purpose
`Router(config)# ip director default priorities boomerang 1`	Configures DistributedDirector to select a server using the boomerang metric at priority level *num*, where *num* is 1.

To remove boomerang as the default metric, use the **no ip director default priorities boomerang** command.

Command	Purpose
`Router(config)# `**`no ip director default priorities`**	Removes boomerang as the default metric.

Specifying the Host Priority of the Boomerang Metric

To configure the order in which DistributedDirector considers metrics when selecting a server, use the **ip director host priority** command.

Command	Purpose
`Router(config)# `**`ip director host boom1 priority boomerang`**	Configures DistributedDirector to select a server using the boomerang metric, where *hostname* is boom1.

To deactivate all priorities on all metrics associated with the defined hostname, use the **no** form of this command.

Command	Purpose
`Router(config)# `**`no ip director host boom1 priority`**	Deactivates all priorities on all metrics associated with the defined hostname, where *hostname* is boom1.

Verifying Boomerang Information

Step 1 To verify that the boomerang metric is configured, enter the **show running-config** command.

```
Router# show running-config

ip host boom1 172.22.2.10 172.22.2.20 172.22.2.30
ip director server 172.22.2.20 drp-association 172.24.4.2
ip director server 172.22.2.30 drp-association 172.24.4.3
ip director server 172.22.2.10 drp-association 172.24.4.1
ip director host boom1
no ip director cache
ip dns primary boom1 soa boom1 boom1@com
ip director host boom1 priority boomerang 1
no ip director drp synchronized
```

Step 2 To view information about all hosts, enter the **show ip director** command.

```
Router# show ip director

Distributed Director status:
Queries received: 0
Queries replied: 0
Queries received in the last second: 0
Queries received in the last minute: 0
Incomplete information selections: 0
TTL for reply RRs when sorted by DD: 0 secs
Queries awaiting processing by DD: 0
Queries awaiting metric info = 0
Director cache is on
Cache time for sort cache entries: 60 secs
Director sort cache hits = 0
Director Response Protocol:
  0 requests, 0 replies, 0 requeries, 0 bad replies
  Authentication key-chain "not defined"
  Output queue length = 0
  Maximum DRP query retry number = 2
  Timeout for each DRP lookup query = 1 secs
  Timeout for each DRP measurement query = 4 secs
```

Step 3 To view information about a specified host, enter the **show ip director host** command. The following command provides information about a host named boom1.

```
Router# show ip director host boom1

Host boom1 (A queries):
  Queries received: 0, queries replied: 0
  Servers:
    Server 172.22.2.10:
      Advertised 0 times as best server, last at never
      Server status: Untested, updated never
  Host specific priorities:
    Boomerang evaluation = 1
```

Configuration Examples

This section provides the following configuration examples:

- Setting DistributedDirector and DRP Clock Synchronization Example
- Configuring the Default Metric Example
- Specifying the Host Priority of the Boomerang Metric Example

Setting DistributedDirector and DRP Clock Synchronization Example

In the following example, DistributedDirector and DRP clock synchronization is activated:

```
Router(config)# ip director drp synchronized

Router# show running-config

ip host boom1 172.22.2.10 172.22.2.20 172.22.2.30
ip director server 172.22.2.20 drp-association 172.24.4.2
ip director server 172.22.2.30 drp-association 172.24.4.3
ip director server 172.22.2.10 drp-association 172.24.4.1
ip director host boom1
.
.
.
ip director drp synchronized
```

Configuring the Default Metric Example

In the following example, the boomerang metric is specified with a priority of 1:

```
Router(config)# ip director default priorities boomerang 1

Router# show running-config

ip host boom1 172.22.2.10 172.22.2.20 172.22.2.30
.
.
.
ip director host boom1
no ip director cache
ip dns primary boom1 soa boom1 boom1@com
ip director host boom1 priority boomerang 1
```

Specifying the Host Priority of the Boomerang Metric Example

The following example specifies the per-host priority of the metric, with a host named boom1, where the DRP internal metric is specified with a priority number of 1 and boomerang is specified with a priority number of 2:

```
Router(config)# ip director host boom1 priority drp-int 1 boomerang 2

Router# show running-config

ip host boom1 172.22.2.10 172.22.2.20 172.22.2.30
.
.
.
ip director host boom1
no ip director cache
ip dns primary boom1 soa boom1 boom1@com
ip director host boom1 priority drp-int 1 boomerang 2
```

Command Reference

The following new and modified commands are pertinent to this feature. To see the command pages for these commands and other commands used with this feature, go to the *Cisco IOS Master Commands List*, Release 12.4, at http://www.cisco.com/univercd/cc/td/doc/product/software/ios124/124mindx/124index.htm.

New Commands

- ip director drp synchronized

Modified Commands

- ip director default priorities
- ip director host priority

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

DistributedDirector MIB Support

First Published: February 25, 2002
Last Updated: May 2, 2008

Feature History

Release	Modification
12.2(8)T	This feature was introduced.
Cisco IOS XE Release 2.1	This feature was introduced on Cisco ASR 1000 Series Routers.

This document describes DistributedDirector MIB support and the enhancements and modifications made to the Cisco IOS Simple Network Management Protocol (SNMP) infrastructure in order to support DistributedDirector in Cisco IOS Release 12.2(8)T. It includes the following sections:

- Feature Overview, page 1
- Supported Platforms, page 3
- Supported Standards, MIBs, and RFCs, page 4
- Prerequisites, page 4
- Configuration Tasks, page 4
- Configuration Examples, page 6
- Command Reference, page 7

Feature Overview

Network management takes place between two major types of systems: those in control, called managing systems, and those observed and controlled, called managed systems. The most common type of managing system is called a *network management system* (NMS). Managed systems can include hosts, servers, or network components such as routers or intelligent repeaters.

To promote interoperability, cooperating systems must adhere to a common framework and a common language, called a *protocol*. In the Internet network management framework, that protocol is the SNMP.

Americas Headquarters:
Cisco Systems, Inc., 170 West Tasman Drive, San Jose, CA 95134-1706 USA

In a managed device, specialized low-impact software modules, called *agents*, access information about the device and make it available to the NMS. Managed devices maintain values for a number of variables and report those, as required, to the NMS. For example, an agent might report such data as the number of bytes and packets passing in and out of the device, or the number of broadcast messages sent and received. In the Internet network management framework, each variable is referred to as a *managed object*, which is anything that an agent can access and report back to the NMS.

All managed objects are contained in the Management Information Base (MIB), which is a database of the managed objects. The managed objects, or variables, can be set or read to provide information on network devices and interfaces. An NMS can control a managed device by sending a message to an agent of that managed device requiring the device to change the value of one or more of its variables.

The Cisco DistributedDirector MIB provides MIB support for DistributedDirector. This MIB contains DistributedDirector statistics, configurations, and status.

The DistributedDirector MIB contains five groups of object type definitions:

- ciscoDistDirGeneralGroup—A group of objects related to DistributedDirector general configurations, statistics, and status.
- ciscoDistDirHostGroup—A group of objects related to DistributedDirector host-specific configurations, statistics, and status.
- ciscoDistDirServerGroup—A group of objects related to DistributedDirector server-specific configurations, statistics, and status.
- ciscoDistDirMappingGroup—A group of objects related to associations between DistributedDirector host names and real servers.
- ciscoDistDirNotificatonGroup—A group of objects related to DistributedDirector significant events.

The DistributedDirector MIB defines the following tables:

- cddGeneralMetricProfTable—DistributedDirector metric profiles. A profile contains priority and weight values of DistributedDirector metrics, which can be applied to specific hosts or to the DistributedDirector default configuration.
- cddHostTable—DistributedDirector virtual host name or subdomain-specific configurations, statistics, and status entries.
- cddHostConnectCfgTable—DistributedDirector per-host server connect test configuration information entries.
- cddHostTolCfgTable—DistributedDirector per-host priority-level metric tolerance configuration information entries.
- cddServerTable—DistributedDirector server-specific information entries. This information includes the configuration parameters and statistics for each server.
- cddServerPortTable—DistributedDirector server port-specific information entries. This information includes the configuration parameters, statistics, and availability status for each service port on servers.
- cddServerPortMetricTable—DistributedDirector per-service per-metric weight entries.
- cddHostServerMappingTable—DistributedDirector associations of virtual host name to real server.

The DistributedDirector MIB defines the following notifications:

- ciscoDistDirEventServerUp—This trap is generated whenever a distributed server changes to the "up" state.
- ciscoDistDirEventServerDown—This trap is generated whenever a distributed server changes to the "down" state.

- ciscoDistDirEventHitRateHigh—This trap is generated whenever the incoming Domain Name system (DNS) HTTP query rate reaches a certain threshold. Use the Event MIB described in RFC 2981 to control the trigger of this notification.

The ciscoDistDirEventServerUp and ciscoDistDirEventServerDown notifications can be enabled or disabled using the Cisco IOS **snmp-server enable traps director** and **snmp-server host** commands.

The **snmp-server host** command is used in conjunction with the **snmp-server enable traps director** command. Use the **snmp-server enable traps director** command to specify which DistributedDirector SNMP notifications are sent globally. For a host to receive most notifications, at least one **snmp-server enable traps director** command and the **snmp-server host** command for that host must be enabled.

Benefits

The DistributedDirector MIB provides network management functionality to DistributedDirector.

Restrictions

The DistributedDirector MIB implementation for Cisco IOS Release 12.2(8)T supports read-only capability to the objects defined in the MIB.

Related Features and Technologies

- Event MIB
- SNMP
- Network management

Related Documents

- The “Configuring SNMP Support” chapter of *Cisco IOS Configuration Fundamentals Configuration Guide,* Release 12.2
- The “SNMP Commands” chapter of *Cisco IOS Configuration Fundamentals Command Reference,* Release 12.2
- RFC 1157, “Simple Network Management Protocol”
- Event MIB: RFC 2981, *Event MIB*

Supported Platforms

- Cisco 2600 series
- Cisco 3620 series
- Cisco 3640 series
- Cisco 3660 series
- Cisco 7200 series

Determining Platform Support Through Cisco Feature Navigator

Cisco IOS software is packaged in feature sets that support specific platforms. To get updated information regarding platform support for this feature, access Cisco Feature Navigator. Cisco Feature Navigator dynamically updates the list of supported platforms as new platform support is added for the feature.

Cisco Feature Navigator is a web-based tool that enables you to quickly determine which Cisco IOS software images support a specific set of features and which features are supported in a specific Cisco IOS image. You can search by feature or release. Under the release section, you can compare releases side by side to display both the features unique to each software release and the features in common.

To access Cisco Feature Navigator, you must have an account on Cisco.com. If you have forgotten or lost your account information, send a blank e-mail to cco-locksmith@cisco.com. An automatic check will verify that your e-mail address is registered with Cisco.com. If the check is successful, account details with a new random password will be e-mailed to you. Qualified users can establish an account on Cisco.com by following the directions at http://www.cisco.com/register.

Cisco Feature Navigator is updated regularly when major Cisco IOS software releases and technology releases occur. For the most current information, go to the Cisco Feature Navigator home page at the following URL:

http://www.cisco.com/go/fn

Supported Standards, MIBs, and RFCs

Standards

No new or modified standards are supported by this feature.

MIBs

- Cisco DistributedDirector MIB (CISCO-DIST-DIRECTOR-MIB.my)
- Event MIB (EVENT-MIB.my)

To obtain lists of supported MIBs by platform and Cisco IOS release, and to download MIB modules, go to the Cisco MIB website on Cisco.com at the following URL:

http://www.cisco.com/public/sw-center/netmgmt/cmtk/mibs.shtml

RFCs

- Event MIB: RFC 2981, *Event MIB*

Prerequisites

DistributedDirector must be running on the router.

Configuration Tasks

See the following sections for configuration tasks for the DistributedDirector MIB support feature. Each task in the list is identified as either required or optional.

- Enabling DistributedDirector SNMP Notifications (required)
- Specifying the Recipient of an SNMP Notification (required)

Enabling DistributedDirector SNMP Notifications

To enable DistributedDirector SNMP notifications, use the following command in global configuration mode:

Command	Purpose
`Router(config)#` **`snmp-server enable traps director`**	Enables DistributedDirector SNMP notifications.

To disable DistributedDirector SNMP notifications, use the following command in global configuration mode:

Command	Purpose
`Router(config)#` **`no snmp-server enable traps director`**	Disables DistributedDirector SNMP notifications.

Specifying the Recipient of an SNMP Notification

To specify the recipient of a DistributedDirector SNMP notification, use the following command in global configuration mode:

Command	Purpose
`Router(config)#` **`snmp-server host 10.0.0.1 public director`**	Specifies the recipient of a DistributedDirector SNMP notification, where the host 10.0.0.1 is using the community string defined as "public."

To remove the specified recipient, use the following command in global configuration mode:

Command	Purpose
`Router(config)#` **`no snmp-server host`** *`host-address`* **`director`**	Removes the recipient of a DistributedDirector SNMP notification.

Verifying DistributedDirector Notification Information

Enter the **show running-config** command to verify that DistributedDirector SNMP notification information is configured. Both server up and server down information is included, unless you specify one or the other.

```
Router# show running-config

ip host myhost 172.2.2.10 172.2.2.20 172.2.2.30
.
.
```

```
.
snmp-server enable traps director server-up server-down
```

Configuration Examples

This section provides the following configuration examples:

- Enabling DistributedDirector SNMP Notifications Example
- Specifying the Recipient of an SNMP Notification Example

Enabling DistributedDirector SNMP Notifications Example

In the following example, both ciscoDistDirEventServerUp and ciscoDistDirEventServerDown notifications are enabled:

```
Router(config)#  snmp-server enable traps director

Router# show running-config

ip host myhost 172.2.2.10 172.2.2.20 172.2.2.30
.
.
.
snmp-server enable traps director server-up server-down
```

Specifying the Recipient of an SNMP Notification Example

In the following example, the ciscoDistDirEventServerUp and ciscoDistDirEventServerDown notifications are to be sent to the host 10.0.0.1 using the community string defined as “public”:

```
Router(config)# snmp-server host 10.0.0.1 public director

Router# show snmp

Chassis:8768490
0 SNMP packets input
    0 Bad SNMP version errors
    0 Unknown community name
    0 Illegal operation for community name supplied
    0 Encoding errors
    0 Number of requested variables
    0 Number of altered variables
    0 Get-request PDUs
    0 Get-next PDUs
    0 Set-request PDUs
0 SNMP packets output
    0 Too big errors (Maximum packet size 1500)
    0 No such name errors
    0 Bad values errors
    0 General errors
    0 Response PDUs
    0 Trap PDUs

SNMP logging:enabled
    Logging to 10.0.0.1.162, 0/10, 0 sent, 0 dropped.
```

Command Reference

The following commands are introduced or modified in the feature or features documented in this module. For information about these commands, see the *Cisco IOS Network Management Command Reference* at http://www.cisco.com/en/US/docs/ios/netmgmt/command/reference/br_book.html. For information about all Cisco IOS commands, go to the Command Lookup Tool at http://tools.cisco.com/Support/CLILookup or to the *Cisco IOS Master Commands List*.

- **snmp-server enable traps director**
- **snmp-server host**

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

VPN Device Manager for XSM

VPN Device Manager Client for Cisco IOS Software (XSM Configuration)

Feature History

Release	Modification
12.1(6)E	This feature was introduced.
12.2(9)YE, 12.2(9)YO1	This feature was integrated into Cisco IOS Release 12.2YE and 12.2YO.
12.2(13)T	This feature was integrated into Cisco IOS Release 12.2T for inclusion in Release 12.3.
12.2(14)S	This feature was integrated into Cisco IOS Release 12.2S.

This document was written for Release 12.1(6)E, and last updated January 2003 for Release 12.2(14)S.

Note For the primary documentaiton of the latest version of the VPN Device Manager (version 1.2), see the "Installation Guide and Release Notes for VPN Device Manager 1.2" at http://www.cisco.com/univercd/cc/td/doc/product/rtrmgmt/vdm/vdm12rn.htm

This document describes the command-line interface (CLI) Cisco IOS commands required to activate the VPN Device Manager (VDM) client and includes the following sections:

- Feature Overview, page 2
- Supported Platforms, page 4
- Supported Standards, MIBs, and RFCs, page 5
- Prerequisites, page 5
- Configuring VDM, page 5
- Configuration Examples for VDM, page 8
- Command Reference, page 9
- Glossary, page 10

Americas Headquarters:
Cisco Systems, Inc., 170 West Tasman Drive, San Jose, CA 95134-1706 USA

Feature Overview

VDM software is installed directly onto Cisco VPN devices. It allows network administrators to use a web browser to manage and configure site-to-site VPNs on a single device. VDM implements a wizard-based GUI that allows simplified VPN configuration of the device on which it resides and peer-to-peer interfaces from that device to remote devices. VDM requires configuration of some Cisco IOS commands before it can be fully operational.

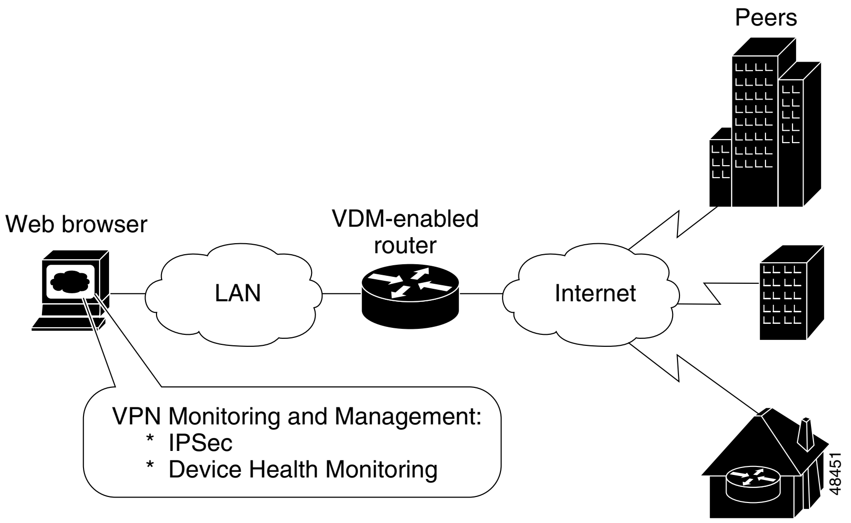
Note In addition to having the relevant Cisco IOS image installed on your device, make sure the VDM client software has been preinstalled in the device Flash memory. If it has not been, you must download it from Cisco.com. See the *Installation and Release Notes for VPN Device Manager* for the product version you are using for details on completing this task. See the VPN Device Manager index (http://www.cisco.com/warp/public/cc/pd/nemnsw/vpdvmn) for further information.

VDM also monitors general system statistics and VPN-specific information such as tunnel throughput and errors. The graphing capability allows comparison of such parameters as traffic volume, tunnel counts, and system utilization. VDM supports site-to-site VPNs. Its step-by-step wizards simplify the configuration of common VPN setups, interfaces, and policies, including:

- IPSec tunnels
- Preshared keys and Internet Key Exchange (IKE) policies

Figure 1 shows a simplified VDM deployment within a VPN.

Figure 1 Simplified VDM Deployment



XML Subscription Manager

XML Subscription Manager (XSM) is an HTTP-based service for retrieving information from a Cisco device. Once remote applications (such as VDM) are connected to the XSM server, they can subscribe to data sets called XML Request Descriptors (XRDs). These are XML-formatted messages describing configuration (access-control lists (ACLs), interfaces, crypto-maps, and others) and monitoring information (CPU, memory usage, interface statistics, and others).

XSM provides remote applications such as VDM with a constantly updated stream of data about Cisco device status by supplying real-time data without repeated device polling.

CLI Commands for VDM

This document gives details about Cisco IOS commands specific to VDM functionality. These commands are not related to general VPN functions but are designed to manage VDM itself via the XSM server. By using the Java-enabled VDM application, you can perform all VPN-related configuration and monitoring tasks within the application.

These commands are designed to complement VDM. The following tasks are performed by specific Cisco IOS XSM commands (command name in parentheses):

- Enabling VDM to receive data from the XSM feature set on the device (**xsm**)
- Enabling basic device monitoring, configuration, and data delivery for VDM (**xsm edm**)
- Enabling VPN-specific monitoring, configuration, and data delivery for VDM (**xsm vdm**)
- Enabling access to switch operations (for example, configuring switch ports and VLANs) when running VDM on a switch (**xsm dvdm**)
- Enabling collection of selected statistics generic to embedded devices on the XSM server (**xsm history edm**)
- Enabling collection of specific selected VPN statistics on the XSM server (**xsm history vdm**)
- Clearing VDM client sessions (**clear xsm**)
- Displaying information about the XSM server and VDM (**show xsm status**)
- Displaying all XRDs available to VDM (**show xsm xrd-list**)
- Setting user privilege levels for viewing VDM monitoring and configuration data (**xsm privilege monitor level** and **xsm privilege configuration level**)

For more information on VDM, the *Installation and Release Notes for VPN Device Manager* for the product version you are using or the Documentation CD-ROM that shipped with the product. See the VPN Device Manager index (http://www.cisco.com/warp/public/cc/pd/nemnsw/vpdvmn) for further information.

Related Features and Technologies

- Virtual Private Networks (VPNs)
- Security

Related Documents

- *Access VPN Solutions Using Tunneling Technology*
- *Access VPDN Dial-in Using L2TP*
- *Access VPDN Dial-in Using IPSec Over L2TP*
- Cisco IOS Dial Technologies Command Reference, Release 12.2
- Cisco IOS Security Configuration Guide, Release 12.2
- Cisco IOS Security Command Reference, Release 12.2

- "Configuring Virtual Private Networks" chapter in the Virtual Templates, Profiles, and Networks *part of the Cisco IOS Dial Technologies Configuration Guide,* Release 12.2
- Installation and Release Notes for VPN Device Manager
- VDM chapter in the *Cisco Enterprise VPN Configuration Guide*
- *VPN Device Manager*
- IPSec VPN Acceleration Services Module Installation and Configuration Note

Supported Platforms

The XSM Cisco IOS commands are available on the following VDM-enabled platforms:

- Cisco 1700 series routers
- Cisco 2600 series routers
- Cisco 3620, 3640, and 3660 routers
- Cisco 7100 series routers
- Cisco 7200 series routers
- Cisco 7400 series routers
- Cisco Catalyst 6500 series switches with IPSec VPN Acceleration Services Module installed
- Cisco 7600 series Internet routers with IPSec VPN Acceleration Services Module installed

This feature is supported on the following platforms in Cisco IOS Release 12.2(14)S:

- Cisco 7200 series
- Cisco 7400 series

Determining Platform Support Through Cisco Feature Navigator

Cisco IOS software is packaged in feature sets that support specific platforms. To get updated information regarding platform support for this feature, access Cisco Feature Navigator. Cisco Feature Navigator dynamically updates the list of supported platforms as new platform support is added for the feature.

Cisco Feature Navigator is a web-based tool that enables you to quickly determine which Cisco IOS software images support a specific set of features and which features are supported in a specific Cisco IOS image. You can search by feature or release. Under the release section, you can compare releases side by side to display both the features unique to each software release and the features in common.

To access Cisco Feature Navigator, you must have an account on Cisco.com. If you have forgotten or lost your account information, send a blank e-mail to cco-locksmith@cisco.com. An automatic check will verify that your e-mail address is registered with Cisco.com. If the check is successful, account details with a new random password will be e-mailed to you. Qualified users can establish an account on Cisco.com by following the directions at http://www.cisco.com/register.

Cisco Feature Navigator is updated regularly when major Cisco IOS software releases and technology releases occur. For the most current information, go to the Cisco Feature Navigator home page at the following URL:

http://www.cisco.com/go/fn

Availability of Cisco IOS Software Images

Platform support for particular Cisco IOS software releases is dependent on the availability of the software images for those platforms. Software images for some platforms may be deferred, delayed, or changed without prior notice. For updated information about platform support and availability of software images for each Cisco IOS software release, refer to the online release notes or, if supported, Cisco Feature Navigator.

Supported Standards, MIBs, and RFCs

Standards

No new or modified standards are supported by this feature.

MIBs

No new or modified MIBs are supported by this feature.

To obtain lists of supported MIBs by platform and Cisco IOS release, and to download MIB modules, go to the Cisco MIB website on Cisco.com at the following URL:

http://www.cisco.com/public/sw-center/netmgmt/cmtk/mibs.shtml

RFCs

No new or modified RFCs are supported by this feature.

Prerequisites

The VDM client software must be installed on your device. It might already have been installed if you chose the VPN option at the time of configuration.

Configuring VDM

See the following sections for configuration tasks for this feature. Each task in the list is identified as either required or optional.

- Enabling the XSM Server for VDM (required)
- Configuring XSM Privilege Levels for XRDs (optional)
- Disabling the XSM Server for VDM (optional)
- Verifying VDM Status on the XSM Server (optional)
- Clearing XSM Client Sessions (optional)
- Configuring XSM Statistics Collection (optional)

Enabling the XSM Server for VDM

Use the **xsm** command in global configuration mode to activate XSM clients (such as VDM) on your device. Enabling this command also enables the **xsm vdm** and **xsm edm** global configuration commands, so there is no need to enable them separately.

Command	Purpose
`Router(config)# xsm`	Enables XSM client access to the device.

Configuring XSM Privilege Levels for XRDs

To set the minimum required privilege levels and grant appropriate access to view, monitor, or configure the XSM client (such as VDM), use the following commands in global configuration mode. Privilege levels set on the device determine which access level users possess (configuration and monitoring, monitoring only, or neither).

Users with privilege levels lower than the required monitoring privilege level will not have access to either the configuration or monitoring data required for subscription to XML Request Descriptors (XRDs). The higher the number, the higher the privilege level. The privilege level for the **xsm privilege configuration level** command must be greater than or equal to that of the **xsm privilege monitor level** command.

Command	Purpose
`Router(config)# xsm privilege configuration level number`	Enables configuration privilege level to subscribe to XRDs. • *number*—Privilege level (1–15). Privilege level 15 is the default.
`Router(config)# xsm privilege monitor level number`	Enables monitor privilege level to subscribe to XRDs. • *number*—Privilege level (1–15). Privilege level 15 is the default.

Disabling the XSM Server for VDM

To disable the XSM server, use the command below in global configuration mode. Disabling this command also disables the **xsm vdm** and **xsm edm** global configuration commands.

Command	Purpose
`Router(config)# no xsm`	Disables XSM server.

Verifying VDM Status on the XSM Server

Use the **show xsm status** command to verify the status of clients (such as VDM) on the XSM server.

Command	Purpose
`Router# show xsm status`	Displays information and status about clients subscribed to the XSM server.

Use the **show xsm xrd-list** command to verify all XML Request Descriptors (XRDs) for XSM clients (such as VDM) made available by subscription to the XSM server.

Command	Purpose
`Router# show xsm xrd-list`	Displays all XRDs for clients subscribed to the XSM server.

Clearing XSM Client Sessions

Use the **clear xsm** command to clear data from XSM clients (such as VDM) on the XSM server. To disconnect a specific client, you must identify the session number. Use the **show xsm status** command to obtain specific session numbers.

Command	Purpose
`Router# clear xsm [session number]`	Clears XSM client sessions. • **session**—XSM session ID. • *number*—Number of the specific XSM client session you are clearing.

Configuring XSM Statistics Collection

To configure the XSM server and its related clients (such as VDM) for Embedded Device Manager (EDM) or VPN-specific statistics collection of up to 5 days of data, use the following commands in global configuration mode.

Command	Purpose
`Router(config)# xsm history edm`	Enables statistics collection for the EDM on the XSM server.
`Router(config)# xsm history vdm`	Enables specific VPN statistics collection on the XSM server.

Configuration Examples for VDM

This section provides the following configuration examples:

- Enabling the XSM Server for VDM Example
- Configuring XSM Privilege Levels for XRDs Example
- Disabling the XSM Server for VDM Example
- Configuring XSM Statistics Collection Example

Enabling the XSM Server for VDM Example

The following example shows how to enable the XSM client on the device:

```
xsm
```

Configuring XSM Privilege Levels for XRDs Example

The following example shows how to set a privilege level of 11, for subscription to XRDs:

```
xsm privilege monitor level 11
```

Disabling the XSM Server for VDM Example

The following example shows how to enable and then disable the XSM client on the device to troubleshoot VDM:

```
no xsm
xsm
```

Configuring XSM Statistics Collection Example

The following example shows how to configure the XSM server and its related clients (such as VDM) for Embedded Device Manager (EDM) or VPN-specific statistics collection of up to 5 days of data:

```
xsm history edm
xsm history vdm
```

Command Reference

The following modified commands are pertinent to this feature. To see the command pages for these commands and other commands used with this feature, go to the *Cisco IOS Master Commands List*, Release 12.4, at http://www.cisco.com/univercd/cc/td/doc/product/software/ios124/124mindx/124index.htm.

- **clear xsm**
- **crypto mib topn**
- **show xsm status**
- **show xsm xrd-list**
- **xsm**
- **xsm dvdm**
- **xsm edm**
- **xsm history edm**
- **xsm history vdm**
- **xsm privilege configuration level**
- **xsm privilege monitor level**
- **xsm vdm**

Glossary

Internet Key Exchange (IKE)—A key management protocol standard used in conjunction with IPSec and other standards. IPSec can be configured without IKE, but IKE enhances IPSec by providing additional features, flexibility, and ease of configuration for the IPSec standard. IKE authenticates the IPSec peers, negotiates IPSec keys, and negotiates IPSec security associations. Before any IPSec traffic can be passed, each router/firewall/host must be able to verify the identity of its peer. This can be done by manually entering preshared keys into both hosts or by a CA service.

IP security (IPSec)—A framework of open standards that provides data confidentiality, data integrity, and data authentication between participating peers. IPSec provides these security services at the IP layer.

Virtual Private Network (VPN)—A virtual network that uses advanced encryption and tunneling to permit organizations to establish secure, end-to-end, private network connections over public IP infrastructure networks, such as the Internet or extranets.

VPN Device Manager (VDM)—A browser-based tool for configuring and monitoring VPNs on a VPN-enabled device. VDM allows users to configure and monitor advanced VPN functionality within Cisco devices.

XML Subscription Manager (XSM)— A Cisco IOS subsystem that allows embedded device managers such as VDM to receive XML-based configuration and monitoring information for managing network devices.

XML Request Descriptor (XRD)—A specific requested type of data from XSM.

Embedded Device Manager (EDM)—An XSM adapter that publishes general network device configuration and monitoring information for device managers such as VDM.

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