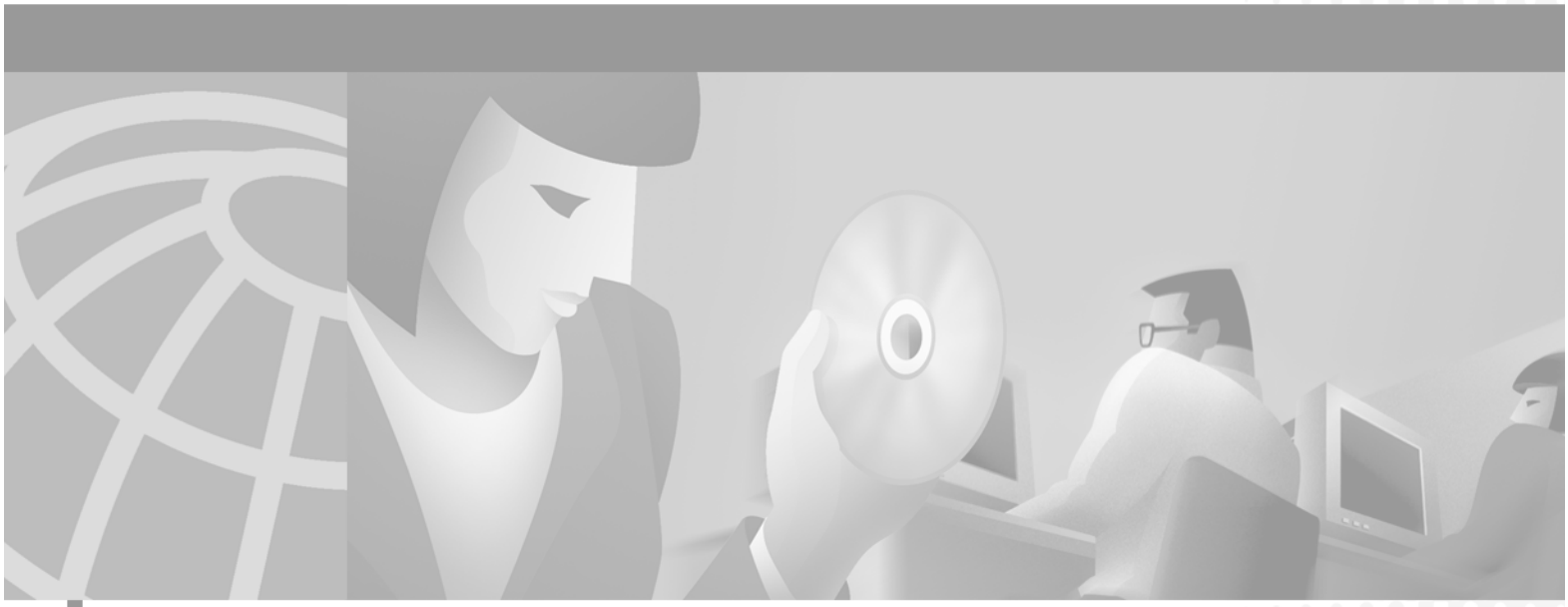
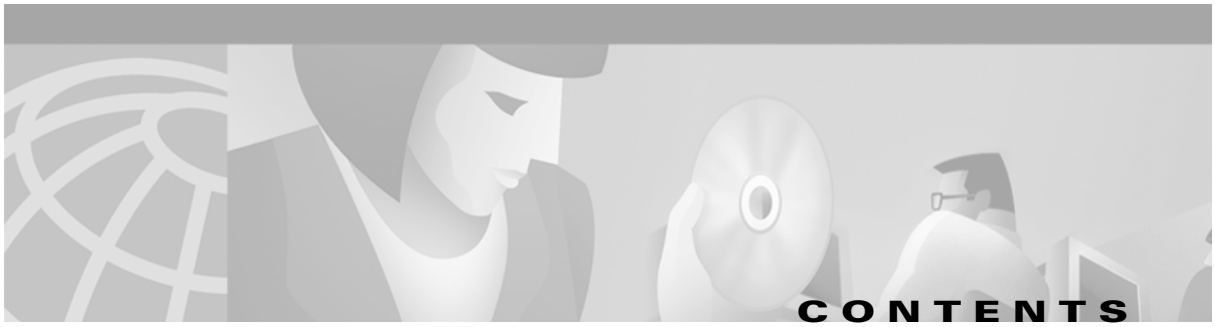
Cisco Systems



Cisco Application and Content Networking Software Command Reference

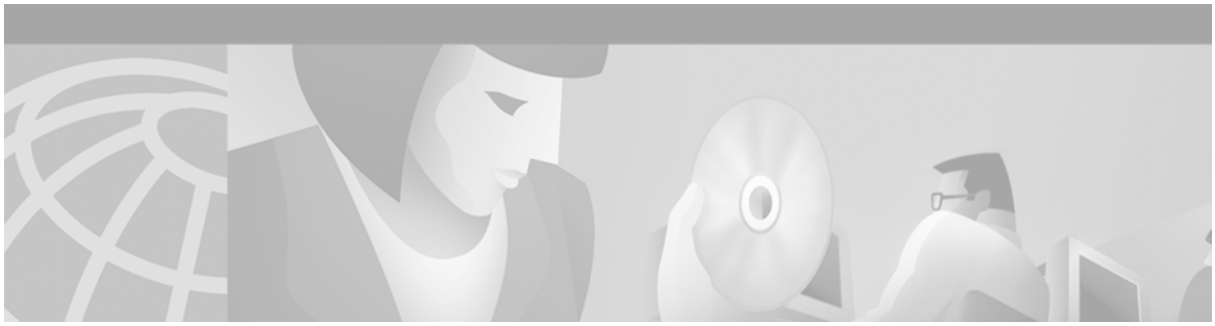
Corporate Headquarters
Cisco Systems, Inc.
170 West Tasman Drive
San Jose, CA 95134-1706
USA
http://www.cisco.com
Tel: 408 526-4000
800 553-NETS (6387)
Fax: 408 526-4100

Customer Order Number: DOC-7813952=
Text Part Number: 78-13952-01



CONTENTS

CHAPTER 2 **Cisco ACNS Software Commands** 2-1



Preface

This preface describes who should read the *Cisco Cache Software Command Reference*, how it is organized, and its document conventions. It contains the following sections:

Audience

This command reference is for experienced network administrators familiar with TCP/IP networking concepts and router configuration.

Document Organization

This command reference includes the following chapters:

Chapter	Title	Description
Chapter 1	Command-Line Interface Command Summary	Describes how to use the command-line interface and presents the commands and command syntax in tables.
Chapter 2	Cisco ACNS Software Commands	Lists ACNS software commands in alphabetical order and provides detailed descriptions of their use.

Document Conventions

This command reference uses basic conventions to represent text and table information.

Convention	Description
boldface font	Commands, keywords, and button names are in **boldface**.
italic font	Variables for which you supply values are in *italics*. Directory names and filenames are also in italics.
`screen` font	Terminal sessions and information the system displays are printed in `screen` font.
`boldface screen` font	Information you must enter is in **`boldface screen`** font.
`italic screen` font	Variables you enter are printed in *`italic screen`* font.
plain font	Enter one of a range of options as listed in the syntax description.
^D or **Ctrl-D**	Hold the **Ctrl** key while you press the **D** key.
string	Defined as a nonquoted set of characters. For example, when setting a community string for SNMP to "public," do not use quotation marks around the string, or the string will include the quotation marks.
Vertical bars (|)	Vertical bars separate alternative, mutually exclusive, elements.
{ }	Elements in braces are required elements.
[ ]	Elements in square brackets are optional.
{**x** | **y** | **z**}	Required keywords are grouped in braces and separated by vertical bars.
[x | y | z]	Optional keywords are grouped in brackets and separated by vertical bars.
[{ }]	Braces within square brackets indicate a required choice within an optional element.

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

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

Additional Documentation

For additional information on Cisco Content Delivery Networking products, refer to the following documentation.

- *Cisco ACNS Software Documentation Roadmap*
- *Release Notes for Cisco Application and Content Networking Software, Release 4.1*

- *Regulatory Compliance and Safety Information for the Cisco Content Networking Product Series*
- *Cisco Content Delivery Networking Products Getting Started Guide*
- *Cisco Content Engine 500 Series Hardware Installation Guide*
- *Cisco Content Distribution Manager 4630 Hardware Installation Guide*
- *Cisco Content Router 4430 Hardware Installation Guide*
- *Cisco Content Networking Hardware Installation Guide for the Seven-Rack Unit Chassis*
- *Cisco Application and Content Networking Software Caching Configuration Guide*
- *Cisco Application and Content Networking Software E-CDN Administrator's Guide*
- *Cisco ACNS System Maintenance and Troubleshooting Guide* (Available online only)
- *Cisco Storage Array 6 Installation and Configuration Guide*
- *Cisco Storage Array 12 Installation and Configuration Guide*
- *Release Notes for the Cisco Storage Array*

The following sections provide sources for obtaining documentation from Cisco Systems.

Obtaining Documentation

The following sections explain how to obtain documentation from Cisco Systems.

World Wide Web

You can access the most current Cisco documentation on the World Wide Web at the following URL:

http://www.cisco.com

Translated documentation is available at the following URL:

http://www.cisco.com/public/countries_languages.shtml

Documentation CD-ROM

Cisco documentation and additional literature are available in a Cisco Documentation CD-ROM package, which is shipped with your product. The Documentation CD-ROM is updated monthly and may be more current than printed documentation. The CD-ROM package is available as a single unit or through an annual subscription.

Ordering Documentation

Cisco documentation is available in the following ways:

- Registered Cisco Direct Customers can order Cisco product documentation from the Networking Products MarketPlace:

  http://www.cisco.com/cgi-bin/order/order_root.pl

- Registered Cisco.com users can order the Documentation CD-ROM through the online Subscription Store:

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

- Nonregistered Cisco.com users can order documentation through a local account representative by calling Cisco corporate headquarters (California, USA) at 408 526-7208 or, elsewhere in North America, by calling 800 553-NETS (6387).

Documentation Feedback

If you are reading Cisco product documentation on Cisco.com, you can submit technical comments electronically. Click **Leave Feedback** at the bottom of the Cisco Documentation home page. After you complete the form, print it out and fax it to Cisco at 408 527-0730.

You can e-mail your comments to bug-doc@cisco.com.

To submit your comments by mail, use the response card behind the front cover of your document, or write to the following address:

Cisco Systems
Attn: Document Resource Connection
170 West Tasman Drive
San Jose, CA 95134-9883

We appreciate your comments.

Obtaining Technical Assistance

Cisco provides Cisco.com as a starting point for all technical assistance. Customers and partners can obtain documentation, troubleshooting tips, and sample configurations from online tools by using the Cisco Technical Assistance Center (TAC) Web Site. Cisco.com registered users have complete access to the technical support resources on the Cisco TAC Web Site.

Cisco.com

Cisco.com is the foundation of a suite of interactive, networked services that provides immediate, open access to Cisco information, networking solutions, services, programs, and resources at any time, from anywhere in the world.

Cisco.com is a highly integrated Internet application and a powerful, easy-to-use tool that provides a broad range of features and services to help you to:

- Streamline business processes and improve productivity.
- Resolve technical issues with online support.
- Download and test software packages.
- Order Cisco learning materials and merchandise.
- Register for online skill assessment, training, and certification programs.

You can self-register on Cisco.com to obtain customized information and service. To access Cisco.com, go to the following URL:

http://www.cisco.com

Technical Assistance Center

The Cisco TAC is available to all customers who need technical assistance with a Cisco product, technology, or solution. Two types of support are available through the Cisco TAC: the Cisco TAC Web Site and the Cisco TAC Escalation Center.

Inquiries to Cisco TAC are categorized according to the urgency of the issue:

- Priority level 4 (P4)—You need information or assistance concerning Cisco product capabilities, product installation, or basic product configuration.
- Priority level 3 (P3)—Your network performance is degraded. Network functionality is noticeably impaired, but most business operations continue.
- Priority level 2 (P2)—Your production network is severely degraded, affecting significant aspects of business operations. No workaround is available.
- Priority level 1 (P1)—Your production network is down, and a critical impact to business operations will occur if service is not restored quickly. No workaround is available.

Which Cisco TAC resource you choose is based on the priority of the problem and the conditions of service contracts, when applicable.

Cisco TAC Web Site

The Cisco TAC Web Site allows you to resolve P3 and P4 issues yourself, saving both cost and time. The site provides around-the-clock access to online tools, knowledge bases, and software. To access the Cisco TAC Web Site, go to the following URL:

http://www.cisco.com/tac

All customers, partners, and resellers who have a valid Cisco services contract have complete access to the technical support resources on the Cisco TAC Web Site. The Cisco TAC Web Site requires a Cisco.com login ID and password. If you have a valid service contract but do not have a login ID or password, go to the following URL to register:

http://www.cisco.com/register/

If you cannot resolve your technical issues by using the Cisco TAC Web Site, and you are a Cisco.com registered user, you can open a case online by using the TAC Case Open tool at the following URL:

http://www.cisco.com/tac/caseopen

If you have Internet access, it is recommended that you open P3 and P4 cases through the Cisco TAC Web Site.

Cisco TAC Escalation Center

The Cisco TAC Escalation Center addresses issues that are classified as priority level 1 or priority level 2; these classifications are assigned when severe network degradation significantly impacts business operations. When you contact the TAC Escalation Center with a P1 or P2 problem, a Cisco TAC engineer will automatically open a case.

To obtain a directory of toll-free Cisco TAC telephone numbers for your country, go to the following URL:

http://www.cisco.com/warp/public/687/Directory/DirTAC.shtml

Before calling, please check with your network operations center to determine the level of Cisco support services to which your company is entitled; for example, SMARTnet, SMARTnet Onsite, or Network Supported Accounts (NSA). In addition, please have available your service agreement number and your product serial number.

CHAPTER 1

Command-Line Interface Command Summary

This chapter provides a summary of the command-line interface (CLI) commands included in the *Cisco Application and Content Networking Software Command Reference*. The command summary tables are grouped alphabetically in five categories: user-level EXEC commands, privileged-level EXEC commands, global configuration commands, interface configuration commands, and **show** EXEC commands. The CLI can be accessed through the console port or Telnet.

Using Command-Line Processing

ACNS software commands are not case sensitive. You can abbreviate commands and parameters as long as they contain enough letters to be different from any other currently available commands or parameters. You can scroll through the last 20 commands stored in the history buffer and enter or edit the command at the prompt. (See Table 1-1.)

Table 1-1 Command-Line Processing Keystroke Combinations

Keystroke Combinations	Function
Ctrl-A	Jumps to the first character of the command line.
Ctrl-B or the left arrow key	Moves the cursor back one character.
Ctrl-C	Escapes and terminates prompts and tasks.
Ctrl-D	Deletes the character at the cursor.
Ctrl-E	Jumps to the end of the current command line.
Ctrl-F or the right arrow key[1]	Moves the cursor forward one character.
Ctrl-K	Deletes from the cursor to the end of the command line.
Ctrl-L	Repeats the current command line on a new line.
Ctrl-N or the down arrow key[1]	Enters the next command line in the history buffer.
Ctrl-P or the up arrow key[1]	Enters the previous command line in the history buffer.
Ctrl-T	Transposes the character at the cursor with the character to the left of the cursor.
Ctrl-U; Ctrl-X	Deletes from the cursor to the beginning of the command line.
Ctrl-W	Deletes the last word typed.

Table 1-1 Command-Line Processing Keystroke Combinations (continued)

Keystroke Combinations	Function
Esc-B	Moves the cursor back one word.
Esc-D	Deletes from the cursor to the end of the word.
Esc-F	Moves the cursor forward one word.
Delete key or Backspace key	Erases a mistake when entering a command; reenter the command after using this key.

1. The arrow keys function only on ANSI-compatible terminals such as VT100s.

Command Modes

The four modes are:

- EXEC
- Domain configuration
- Global configuration
- Interface configuration

EXEC Mode

The two EXEC access levels are privileged and user. The **enable** and **disable** commands switch between the two levels. The user-level EXEC command line is available to users if they enter a valid password. The user-level EXEC commands are a subset of the privileged-level EXEC commands. The user-level EXEC prompt is the host name followed by a right angle bracket (>). The prompt for the privileged-level EXEC command line is the pound sign (#). To execute an EXEC command, enter the command at the EXEC system prompt and press the **Return** key. In the following example, a user accesses the privileged-level EXEC command line from the user level.

```
Console> enable
Console#
```

Use the **Delete** or **Backspace** key sequences to edit commands when you type commands at the EXEC prompt.

As a shortcut, you can abbreviate commands to the fewest letters that make them unique. For example, the letters **sho** can be entered for the **show** command.

Certain EXEC commands display multiple screens with the following prompt at the bottom of the screen:

```
--More--
```

Press the **Spacebar** to continue the output, or press **Return** to display the next line. Press any other key to return to the prompt. Also, at the --More-- prompt, you can enter a **?** to display the help message.

To leave EXEC mode, use the **exit** command at the system prompt:

```
Console# exit
```

Domain Configuration Mode

Domain configuration mode allows you to configure the Content Engine as a content routing agent for specific domains. To enter domain configuration mode, use the **boomerang dns** global configuration command. You must be in domain configuration mode to enter domain configuration commands.

```
Console(config)# boomerang dns domain www.foobar.com
Console(config-domain)#
```

To exit domain configuration mode, use the **end** global configuration command:

```
Console(config-domain)# end
```

You can also exit domain configuration mode by entering the **exit** command or by pressing **Ctrl-Z**.

Note For a description of how to configure the Content Engine as a content routing agent, refer to the *Cisco Cache Software Configuration Guide, Release 2.5*. For more information about the boomerang content routing process, refer to the *Cisco Content Routing Software Configuration Guide and Command Reference*.

Global Configuration Mode

To enter the global configuration mode, use the **configure** EXEC command. You must be in global configuration mode to enter global configuration commands.

```
Console# configure
Console(config)#
```

To exit global configuration mode, use the **end** global configuration command:

```
Console(config)# end
```

You can also exit global configuration mode by entering the **exit** command or by pressing **Ctrl-Z**.

Interface Configuration Mode

To enter interface configuration mode, use the **interface** global configuration command. The following example demonstrates how to enter interface configuration mode:

```
Console# config
Console(config)# interface ?
FastEthernet     Select a fast ethernet interface to configure
GigabitEthernet  Select a gigabit ethernet interface to configure
Console(config)# interface fastethernet ?
<0-3>/  FastEthernet slot/port
Console(config)# interface fastethernet 0/1
Console(config-if)#
```

The interface configuration commands are:

autosense

bandwidth

cdp

exit

fullduplex

halfduplex

ip

no

shutdown

standby

These commands are described in the "Interface Configuration Command Summary" section on page 1-16.

To exit interface configuration mode, enter **exit** to return to global configuration mode:

```
Console(config-if)# exit
Console(config)#
```

Check Command Syntax

The user interface provides error isolation in the form of an error indicator, a caret symbol (^). The ^ symbol appears at the point in the command string where you have entered an incorrect command, keyword, or argument.

In the following example, suppose you want to set the clock. Use context-sensitive help to check the syntax for setting the clock.

An example of a mistake is:

```
Console# clock 1222
               ^
%Invalid input detected at '^' marker.
Console# clock ?
read-calendar    Read the calendar and update system clock
set              Set the time and date
update-calendar  Update the calendar with system clock
Console# clock
```

The help output shows that the **set** keyword is required. Check the syntax for entering the time:

```
Console# clock set ?
<0-23>:  Current Time (hh:mm:ss)
Console# clock set
```

Enter the current time in 24-hour format with hours, minutes, and seconds separated by colons:

```
Console# clock set 13:32:00
% Incomplete command.
```

The system indicates that you need to provide additional arguments to complete the command. Press the **Up Arrow** to automatically repeat the previous command entry. Then add a space and question mark (?) to display the additional arguments:

```
Console# clock set 13:32:00 ?
<1-31> Day of the month
January Month of the year
February
March
. . .
```

Enter the day and month as prompted and use the question mark for additional instructions.

```
Console# clock set 13:32:00 23 December ?
  <1993-2035>  Year
```

Now you can complete the command entry by entering the year:

```
Console# clock set 13:32:00 23 December 00
                                        ^
%Invalid input detected at '^' marker.
Console#
```

The caret symbol (^) and help response indicate an error with the 00 entry. To display the correct syntax, press **Ctrl-P** or the **Up Arrow**. You can also reenter the command string, and then enter a space character, a question mark, and press **Enter**:

```
Console# clock set 13:32:00 23 December ?
<1993-2035> Year
Console# clock set 13:32:00 23 December
```

Enter the year using the correct syntax and press **Return** to execute the command:

```
Console# clock set 13:32:00 23 December 2000
WARNING: Setting the clock may cause a temporary service interruption.
Do you want to proceed? [no] yes
Sat Dec 23 13:32:00 EST 2000
Console#
```

System Help

You can obtain help when you enter commands by using the following methods:

- For a brief description of the context-sensitive help system, enter **help**.
- To list all commands for a command mode, enter a question mark (**?**) at the system prompt.
- To obtain a list of commands that start with a particular character set, enter an abbreviated command immediately followed by a question mark (**?**).

  ```
  Console# cl ?
  clear clock
  ```

- To list the command keywords or arguments, enter a space and a question mark (**?**) after the command:

  ```
  Console# clock ?
    clear  Clear the current time from the battery-backed clock
    save   Save the current time into the battery-backed clock
    set    Set the local time and date
  ```

Save Configuration Changes

To avoid losing new configurations, save them to NVRAM using the **copy** or **write** commands, as shown in the following example:

```
Console# copy running-config startup-config
```

or

```
Console# write
```

See the command description for the **copy running-config startup-config** command for more information on "running" and "saved" configuration modes.

EXEC Command Summary

The EXEC commands are entered in the EXEC mode. Table 1-2 lists the user-level EXEC commands. Table 1-3 lists the privileged-level EXEC commands.

Table 1-2 ACNS Software User-Level EXEC Commands

User EXEC Command	Syntax	Description
cd	**cd** *directoryname*	Changes the current directory.
cpfile	**cpfile** *oldfilename newfilename*	Copies sysfs files.
delfile	**del** *filename*	Deletes a file.
deltree	**deltree** *directory*	Deletes directory and all subdirectories.
dir	**dir** [*directory*]	Displays files in long list format.
dnslookup	**dnslookup** {*hostname* | *domainname*}	Resolves host name (DNS).
enable	**enable**	Accesses privileged EXEC commands.
exit	**exit**	Exits from terminal session.
help	**help**	Provides assistance for command line-interface.
lls	**lls** [*directory*]	Displays directory files in long list format.
ls	**ls** [*directory*]	Displays files in directory.
mkdir	**mkdir** *directory*	Makes directory.
mkfile	**mkfile** *filename*	Makes file (for testing).
ping	**ping** {*hostname* | *ip-address*}	Sends echo packets.
pwd	**pwd**	Displays path name of the present working directory.
rename	**rename** *sourcefile destinationfile*	Renames a file (path name).
rmdir	**rmdir** *directory*	Removes directory.
type	**type** *filename*	Displays a file.
whoami	**whoami**	Displays current user's login name.

Table 1-3 ACNS Software Privileged-Level EXEC Commands

Privileged EXEC Command	Syntax	Description
boomerang log dump	**boomerang log dump**	Writes boomerang memory data to a disk file.
boomerang send-packet	**boomerang send-packet** {**tcp** | **udp**} *dest-port source-port* {*dest-ip-address* | *dest-hostname*} {*source-ip-address* | *source-hostname*}	Sends test packets to determine whether or not a destination accepts boomerang-altered source IP addresses.
cache	**cache** {**clear** [**force**] | **reset** | **synchronize**}	Specifies cache commands.

Table 1-3 ACNS Software Privileged-Level EXEC Commands (continued)

Privileged EXEC Command	Syntax	Description
cd	**cd** *directoryname*	Changes directory.
cfs	**cfs** {**clear** *partition* [**force**] | **format** *partition* | **mount** *partition* | **reset** *partition* | **sync** *partition* | **unmount** *partition*}	Partitions cache file system.
clear	**clear** {**bypass** {**counters** | **list**} | **cache** [**dns** [**domain** *domainname* | **hostname** *hostname*] | **http** [**url** *url*] | **http-authentication** | **real-proxy** | **wmt**] | **cdp** {**counters** | **table**} | **logging** | **statistics** {**all** | **authentication** | **boomerang** | **dns-cache** | **ftp** | **history** | **http** {**all** | **cluster** | **errors** | **ims** | **object** | **outgoing** | **proxy outgoing** | **requests** | **response** | **savings**} | **http-authcache** | **https** | **icp** {**all** | **client** | **server**} | **ip** | **ldap** | **mediacache real** | **ntlm** | **pre-load** | **radius** | **rule** {**action** {action-type {**all** | **pattern** pattern-type} | **all**}} | **running** | **tacacs** | **tcp** | **transaction-logs** | **url-filter** {**N2H2** | **websense**} | **wmt**} | **transaction- log**}	Resets counters and listed functions to default settings.
clock	**clock** {**read-calendar** | **set** *time day month year* | **update-calendar**}	Manages the system clock.
configure	**configure**	Enters configuration mode from privileged EXEC mode.
copy	**copy** {**compactflash install** *filename* | **disk ftp** {*hostname* | *ip-address*} *remotefiledir remotefilename localfilename* | **disk startup-config** *filename* | **ftp** {**disk** {*hostname* | *ip-address*} *remotefiledir remotefilename localfilename* | **install** {*hostname* | *ip-address*} *remotefiledir remotefilename*} | **running-config** {**disk** *filename* | **startup-config** | **tftp** {*hostname* | *ip-address*} *remotefilename*} | **startup-config** {**disk** *filename* | **running-config** | **tftp** {*hostname* | *ip-address*} *remotefilename*} | **system-status disk** *filename* | **tech-support** {**disk** *filename* | **tftp** {*hostname* | *ip-address*} *remotefilename*} | **tftp disk** {*hostname* | *ip-address*} *remotefilename localfilename* | **tftp startup-config** {*hostname* | *ip-address*} *remotefilename* | **tftp running-config** {*hostname* | *ip-address*} *remotefilename*}	Copies configuration or image files to and from Flash memory, disk, or remote hosts.
cpfile	**cpfile** *oldfilename newfilename*	Copies a file.
debug	**debug** {**all** option | **authentication** {**http-request** | **user**}}	Configures debugging options.
delfile	**del** *filename*	Removes a file.
deltree	**deltree** *directory*	Removes a directory and its subdirectories.
dir	**dir** [*directory*]	Displays files in long list format.

Table 1-3 ACNS Software Privileged-Level EXEC Commands (continued)

Privileged EXEC Command	Syntax	Description
disable	**disable**	Turns off privileged EXEC commands.
disk	**disk** {**add** *diskname* {**cfs** | **ecdnfs** | **mediafs** | **sysfs**} {**remaining** | *partitionsize*} [{**cfs** | **ecdnfs** | **mediafs** | **sysfs**}{**remaining** | *partitionsize*}] | **cancel-config** | **config sysfs** {**remaining** | *partitionsize*} [{**cfs** | **ecdnfs** | **mediafs**} {**remaining** | *partitionsize*}] | **raid-array** {**add-array** | **repair** *diskname*}}	Configures disk space among functions.
dnslookup	**dnslookup** {*host* | *domain-name*}	Resolves a host name (DNS).
ecdn	**ecdn force-downgrade** [**disable**]	Forces E-CDN downgrade.
enable	**enable**	Accesses privileged EXEC commands.
exit	**exit**	Exits from the EXEC and configuration command levels to user level.
help	**help**	Provides assistance for command-line interface.
install	**install** *paxfilename*	Installs a new version of the Cache software.
lls	**lls** [*directory*]	Displays files in long list format.
ls	**ls** [*directory*]	Lists files in directory.
mediafs	**mediafs** {**format** *partition_name* | **mount** *partition_name* | **sync** *partition_num* | **unmount** *partition_num*}	Performs maintenance on the media file system.
mkdir	**mkdir** *directory*	Makes a directory.
mkfile	**mkfile** *filename*	Makes a file (for testing).
no debug	**no debug**	Disables debugging.
ntpdate	**ntpdate** {*hostname* | *ip-address*}	Sets the NTP server name.
ping	**ping** {*hostname* | *ip-address*}	Sends echo packets.
pre-load force	**pre-load force**	Forces the preload operation.
pwd	**pwd**	View the current working directory.
reload	**reload**	Halts and performs a cold restart.
rename	**rename** *sourcefile destinationfile*	Renames a file (path name).
rmdir	**rmdir** *directory*	Removes a directory.
restore	**restore factory-default**	Restores the Content Engine to its manufactured default status.
sysfs	**sysfs** {**check** *disk name* | **format** *disk name* | **mount** {*disk name* {**local1** | **local2**}} | **repair** *disk name* | **sync** | **unmount** {**local1** | **local2**}}	Maintains system file system.
terminal	**terminal** {**length** *lines* | **monitor** [**disable**]}	Sets terminal commands.
transaction-log force	**transaction-log force** {**archive** | **export**}	Forces archive of working log file to make a transaction log file.

Table 1-3 ACNS Software Privileged-Level EXEC Commands (continued)

Privileged EXEC Command	Syntax	Description
type	**type** *filename*	Displays a file.
type-tail	**type-tail** *filename* [*line* | **follow**]	Displays the last several lines of a file.
undebug	**undebug**	Disables debugging functions (see also **debug**).
url-filter	**url-filter local-list-reload**	Reloads new good site or bad site lists when this feature is enabled.
whoami	**whoami**	Displays current user's name.
write	**write** [**erase** | **memory** | **terminal**]	Writes running configuration to memory or terminal.
wmt	**wmt** {**multicast-station** {**start** *name* | **stop** *name*} | **test-command**}	Starts, stops, and tests WMT multicast stations.

Domain Configuration Command Summary

The domain configuration Content Engine commands are entered in the domain configuration mode. Table 1-4 lists the domain configuration command.

Table 1-4 ACNS Software Domain Configuration Command

Domain Configuration Command	Syntax	Description
boomerang	**boomerang** {**dns** {**domain** *domain-name* [**alias** *alias-name* | **content-server** *ip-address* [**file** *filename*] | **dns-ttl** *seconds* | **hops** *hops* | **key** {**0** *keyword* | **7** *keyword* | *keyword*} | **origin-server** *ip-address*] | **enable**} | **log-races enable**}	Configures boomerang DNS, enables content routing, and enables logging the result of races on boomerang distributed reverse proxy.

Global Configuration Command Summary

The global configuration Content Engine commands are entered in the global configuration mode. Table 1-5 lists the global configuration commands.

Table 1-5 ACNS Software Global Configuration Commands

Global Configuration Command	Syntax	Description
asset tag	**asset tag** *name*	Configures CISCO-ENTITY-ASSETT-MIB.
authentication	**authentication** {**configuration** {**local** | **tacacs**} **enable** [**primary** | **secondary**] | **login** {**local** | **tacacs**} **enable** [**primary** | **secondary**]}	Configures authentication parameters.
bypass	**bypass** {**auth-traffic enable** | **load** {**enable** | **in-interval** *seconds* | **out-interval** *seconds* | **time-interval** *minutes*} | **static** {*clientipaddress* {*serveripaddress* | **any-server**} | **any-client** *serveripaddress*} | **timer** *minutes*}	Configures bypass functions.
cdp	**cdp** {**enable** | **holdtime** *seconds* | **timer** *seconds*}	Configures CDP packets and timing.
clock	**clock** {**summertime** *timezone* {**date** *startday startmonth startyear starthour endday endmonth endyear offset* | **recurring** {**1-4** *startweekday startmonth starthour endweekday endmonth endhour offset* | **first** *startweekday startmonth starthour endweekday endmonh endhour offset* | **last** *startweekday startmonth starthour endweekday endmonh endhour offset*}} | **timezone** {*timezone hoursoffset minutesoffset*}}	Sets summer daylight saving time of day and time zone.
dns-cache	**dns-cache size** *maxsize*	Configures DNS cache.
ecdn	**ecdn** {**cdm ip** *ip-address* [**port** *port-number*] | **enable**}	Configures Enterprise Content Delivery Network (E-CDN).
end	**end**	Exits configuration and privileged EXEC modes.
error-handling	**error-handling** {**reset-connection** | **send-cache-error** | **transparent**}	Customizes how Content Engine handles errors.
exception	**exception** {**coredump** | **debug**}	Enables exception debug mode.
exec-timeout	**exec-timeout** *timeout*	Configures the length of time that an inactive Telnet session remains open.
exit	**exit**	Exits configuration and privileged EXEC modes.
external-ip	**external-ip** *ip-address*	Configures up to a maximum of eight external IP addresses.

Table 1-5 ACNS Software Global Configuration Commands (continued)

Global Configuration Command	Syntax	Description
ftp	**ftp** {**age-multiplier directory-listing** *dl_time* **file** *fo_time* | **max-ttl** {**days directory-listing** *dlmax_days* **file** *fmax_days* | **hours directory-listing** *dlmax_hours* **file** *fmax_hours* | **minutes directory-listing** *dlmax_min* **file** *fmax_min* | **seconds directory-listing** *dlmax_sec* **file** *fmax_sec*} | **min-ttl** *min_minutes* | **object max-size** *size* | **proxy** {**active-mode enable** | **anonymous-pswd** *passwd* | **incoming** *port* | **outgoing host** {*hostname* | *ip-address*} *port* | **reval-each-request** {**all** | **directory-listing** | **none**}}	Configures FTP caching services.
gui-server	**gui-server** {**enable** | **port** *port*}	Configures GUI server.
help	**help**	Provides assistance for command-line interface.
hostname	**hostname** *name*	Configures the system's network name.
http	**http** {**age-multiplier text** *num* **binary** *num* | **anonymizer enable** | **append** {**proxy-auth-header** {*hostname* | *ip-address*} | **via-header** | **www-auth-header** {*hostname* | *ip-address*} | **x-forwarded-for-header**} | **authenticate-strip-ntlm** | **authentication** {**cache** {**max-entries** *entries* | **timeout** *minutes*} | **header** {**401** | **407**}} | **cache-authenticated** {**all** | **basic** | **ntlm**} | **cache-cookies** | **cache-on-abort** {**enable** | **max-threshold** *maxthresh* | **min-threshold** *minthresh* | **percent** *percenthresh*} | **client-no-cache-request** {**ignore** | **revalidate**} | **cluster** {**heal-port** *number* | **http-port** *number* | **max-delay** *seconds* | **misses** *number*} | **l4-switch enable** | **max-ttl** {**days text** *textdays* **binary** *bindays* | **hours text** *texthours* **binary** *binhours* | **minutes text** *textminutes* **binary** *binminutes* | **seconds text** *textseconds* **binary** *binseconds*} | **min-ttl** *minutes* | **object** {**max-size** *maxsize* | **url-validation enable**} | **persistent-connections** {**all** | **client-only** | **server-only** | **timeout** *seconds*} | **proxy** {**incoming** *ports* | **outgoing** {**connection-timeout** *microsecs* | **host** {*hostname* | *ip-address*} *port* [**primary**] **monitor** *seconds* | **origin-server** | **preserve-407**}} | **reval-each-request** {**all** | **none** | **text**} | **serve-ims** {**text** *percentage* **binary** *percentage*} | **strict-request-content-length-checking enable**}	Configures HTTP-related parameters.
https	**https** {**destination-port allow** {*port* | **all**} | **deny** {*port* | **all**} | **proxy** {**incoming** *port* | **outgoing host** {*hostname* | *ip-address*} *port*}}	Configures HTTPS-related parameters.

Table 1-5 ACNS Software Global Configuration Commands (continued)

Global Configuration Command	Syntax	Description
icp	**icp** {**client** {{**add-remote-server** {*hostname* | *ip-address*} {**parent** | **sibling**} **icp-port** *icpport* **http-port** *httpport* [**restrict** *domainnames*]} | **enable** | **exclude** *domainnames* | **max-fail** *retries* | **max-wait** *timeout* | **modify-remote-server** {*hostname* | *ip-address*} {**http-port** *port* | **icp-port** *port* | **parent** | **restrict** *domainnames* | **sibling**}} | **server** {**enable** | **http-port** *port* | **port** *icpport* | **remote-client** {*hostname* | *ip-address*} {**fetch** | **no-fetch**}}}	Configures Internet Cache Protocol parameters.
inetd	**inetd enable** *service concurrent_tasks*	Configures, enables, and disables TCP/IP TFP, RCP, and TFTP services.
interface	**interface** {**FastEthernet** | **GigabitEthernet**} *slot/port* {**autosense** | **bandwidth** *linespeed* | **cdp enable** | **fullduplex** | **halfduplex** | **ip address** *ip-address netmask* [**secondary**] | **shutdown** | **standby** *grpnumber* {**errors** *maxerrors* | **ip** *ip-address netmask* | **priority** *priority*}} For more detail, see the "Interface Configuration Command Summary" section on page 1-16.	Configures a Fast Ethernet or Gigabit Ethernet interface.
ip	**ip** {**default-gateway** *ip-address* | **domain-name** *name* | **dscp** {**client** {**cache-hit** {**match-server** | **set-dscp** *dscp-packets* | **set-tos** *tos-packets*} | **cache-miss** {**match-server** | **set-dscp** *dscp-packets* | **set-tos** *tos-packets*}} | **server** {**match-server** | **set-dscp** *dscp-packets* | **set-tos** *tos-packets*}} | **name-server** {*ip-addresses* | *serial-lookup*} | **route** *dest_addrs netmask gateway*}	Configures Internet Protocol.
ldap server	**ldap server** {**administrative-dn** *name* | **administrative-passwd** *passwd* | **allow-mode** | **base** *baseword* | **enable** | **filter** *filterword* | **host** {*hostname* | *hostipaddress*} [**primary** | **secondary**] | **port** *portnumber* | **timeout** *seconds* | **userid-attribute** *useidword* | **version** *number*}	Configures LDAP server parameters.
logging	**logging** {**console** {**enable** | **priority** *loglevel*} | **cw2K** | **disk** {**enable** | **filename** *filename* | **priority** *loglevel* | **recycle** *size*} | **facility** *facility* | **host** {*ip-address* | **priority** *loglevel*}}	Configures system logging (syslog).
mediafs-division	**mediafs-division** {**wmt-cache-space** *percent_space* **real-cache-space** *percent_space*}	Configures the media file system space allocation to WMT and RealProxy cache.
multicast-client	**multicast-client** {**accept-license-agreement** | **enable** | **evaluate** | **license-key** *key*}	Configures multicast client license options.
no	**no** *command*	Negates a command or sets its defaults.

Table 1-5 ACNS Software Global Configuration Commands (continued)

Global Configuration Command	Syntax	Description
ntlm server	**ntlm server** {**domain** *name* | **enable** | **host** {*hostname* | *ip-address* [**primary** | **secondary**}}	Configures NTLM server parameters.
ntp	**ntp** {**server** {*hostname* | *ip-address*} | **enable** {**cdm** | **manual**}	Configures Network Time Protocol (NTP).
pre-load	**pre-load** {**concurrent-requests** *number* | **depth-level-default** *level_number* | **enable** | **fetch** {**directory** *dir_names* | **domain** *domain_names* | **suffix** *suffix_names*} | **no-fetch** {**directory** *dir_names* | **domain** *domain_names* | **suffix** *suffix_names*} | **schedule** {**every-day** [**start-time** *time* [**end-time** *time*]] | **every-week** {*days of week* [**start-time** *time* [**end-time** *time*]]}} | **traverse-other-domains** | **url-list-file** *path*}	Configures the Content Engine to fetch and preload content.
proxy-auto-config	**proxy-auto-config download** {*ftp-hostname* | *ftp-ip-address*} *remotedir pacfile*	Automatically downloads proxy configuration in browser.
proxy-protocols	**proxy-protocols** {**outgoing-proxy exclude** {**enable** | **list** *word*} | **transparent** {**default-server** | **original-proxy**}}	Configures proxy protocols-related parameters.
radius-server	**radius-server** {**enable** | **host** {*hostname* | *hostipaddr*} [**auth-port** *port*] | **key** *keyword* | **redirect** {**enable** | **message** *reply url*} | **retransmit** *retries* | **timeout** *seconds*}	Configures RADIUS authentication.
real-subscriber	**real-subscriber** {**accept-license-agreement** | **enable** | **evaluate** | **license-key** *key* | **publisher** {*host-name* | *ip-address*} *admin-port-number user-name user-password*}	Configures RealSubscriber parameters.
rtsp proxy	**rtsp proxy** {**incoming** *port* | **l4-switch enable** | **media-real** {**accept-license-agreement** | **enable** | **evaluate** | **ip-address** *ip-address* | **license-key** *keyword*}}	Enables or disables the Real-Time Streaming Protocol (RTSP) proxy, manages license, and configures the RTSP proxy IP address and the redirector port number.
rule	**show rule** {**action** {action-type {**all** | **pattern** pattern-type} | **all**} For a more complete explanation of specific rules, see the "rule" section on page 2-147.	Sets the rules by which the Content Engine filters HTTP web traffic.
snmp-server community	**snmp-server community** *string* [**group** | **rw**]	Enables SNMP; sets community string and optionally names group and enables read-write access.
snmp-server contact	**snmp-server contact** *line*	Text for MIB object sysContact.
snmp-server enable	**snmp-server enable traps** [**config** | **content-engine** {**disk-fail** | **disk-read** | **disk-write** | **overload-bypass** | **transaction-log**} | **entity** | **snmp** [**authentication** | **cold-start**]]	Enables SNMP traps.

Table 1-5 ACNS Software Global Configuration Commands (continued)

Global Configuration Command	Syntax	Description
snmp-server group	**snmp-server group** *name* {**v1** [**notify** *name* | **read** *name* | **write** *name*] | **v2c** [**notify** *name* | **read** *name* | **write** *name*] | **v3** {**auth** [**notify** *name* | **read** *name* | **write** *name*] | **noauth** [**notify** *name* | **read** *name* | **write** *name*] | **priv** *name* [**notify** *name* | **read** *name* | **write** *name*]}}	Defines a user security model group.
snmp-server host	**snmp-server host** {*hostname* | *ip-address*} *communitystring username* [**v2c** [**retry** *number* | **timeout** *seconds*] | **v3** {**auth** [**retry** *number* | **timeout** *seconds*] | **noauth** [**retry** *number* | **timeout** *seconds*] | **priv** [**retry** *number* | **timeout** *seconds*]}]	Specifies hosts to receive SNMP traps.
snmp-server location	**snmp-server location** *line*	Specifies path for MIB object sysLocation.
snmp-server notify inform	**snmp-server notify inform**	Configures the SNMP inform request.
snmp-server user	**snmp-server user** *name group* [**auth** {**md5** *password* [**priv** *password*] | **sha** *password* [**priv** *password*]} | **remote** *octetstring* [**auth** {**md5** *password* [**priv** *password*] | **sha** *password* [**priv** *password*]}]	Defines a user who can access the SNMP engine.
snmp-server view	**snmp-server view** *viewname familyname* {**excluded** | **included**}	Defines a Version 2 SNMP (SNMPv2) MIB view.
ssh-key-generate	**ssh-key-generate** [**key-length** *length*]	Generates an Secure Shell (SSH) host key.
sshd	**sshd** {**enable** | **password-guesses** *number* | **timeout** *seconds*}	Configures SSH service parameters.
standby	**standby** *group_number* {**errors** *max_errors* | **ip** *ip-address netmask* | **priority** *priority_level*}	Configures an interface to be a backup for another interface.
tacacs	**tacacs** {**key** *keyword* | **retransmit** *retries* | **server** {*hostname* | *ip-address*} [**primary**] | **timeout** *seconds*}	Configures TACAS+ authentication parameters.
telnet enable	**telnet enable**	Enables Telnet services.
tcp	**tcp** {**client-mss** *maxsegsize* | **client-receive-buffer** *kbytes* | **client-rw-timeout** *seconds* | **client-satellite** | **client-send-buffer** *kbytes* | **cwnd-base** *segments* | **ecdn enable** | **increase-xmit-timer-value** *value* | **init-ss-threshold** *value* | **keepalive-probe-cnt** *count* | **keepalive-probe-interval** *seconds* | **keepalive-timeout** *seconds* | **server-mss** *maxsegsize* | **server-receive-buffer** *kbytes* | **server-rw-timeout** *seconds* | **server-satellite** | **server-send-buffer** *kbytes* | **type-of-service enable**}	Configures TCP parameters.

Table 1-5 ACNS Software Global Configuration Commands (continued)

Global Configuration Command	Syntax	Description
tftp-server	**tftp-server dir** *directory*	Sets the Trivial File Transfer Protocol (TFTP) server directory.
transaction-logs	**transaction-logs** {**archive** {**interval** {**every-day** {**at** *time* | **every** *hour*} | **every-hour** {**at** *minute* | **every** *minute*} | *second*}} | **max-file-size** *filesize*} | **ecdn enable** | **enable** | **export** {**compress** | **enable** | **ftp-server** {*hostname* | *servipaddrs*} *login passw directory* | **interval** {**every-hour** {**at** *minute* | **every** *minute*} | **every-day** {**at** *hour:minute* | **every** *hour*} | **every-week on** *weekday* **[at** *hour:minute*] | *minute*}} | **file-marker** | **format** {**apache** | **extended-squid** | **squid**} | **sanitize**	Configures transaction logging parameters.
trusted-host	**trusted-host** {*hostname* | *ip-address* | **domain-lookup**}	Enables trusted hosts.
url-filter	**url-filter bad-sites-deny** {**enable** | *filename*} | **custom-message** *dirname* | **good-sites-allow** {**enable** | *filename*} | **N2H2** {**allowmode enable** | **enable** | **server** {*hostname* | *ip-address*} [**port** *portnum* [**timeout** *seconds*]]} | **smartfilter enable** | **websense** {**allowmode enable** | **enable** | **server** {*hostname* | *ip-address*} [**port** *portnum* [**timeout** *seconds*]]}	Configures URL filtering.
username	**username** *name* {**password** {**0** *word* | **1** *word* | *word*} | **privilege** {**0-0** | **15-15** | **200-300**}}	Establishes username authentication.
wccp custom-web-cache	**wccp custom-web-cache** {**router-list-num** *num* **port** *port* [[**hash-destination-ip**] [**hash-destination-port**] [**hash-source-ip**] [**hash-source-port**] [**l2-redirect**] [**password** *key*] [**weight** *percentage*]]}	Configures custom web caching service.
wccp flow-redirect	**wccp flow-redirect enable**	Redirects moved flows.
wccp home-router	**wccp home-router** *ip-address*	Specifies WCCP Version 1 home router IP address.
wccp media-cache	**wccp media-cache** {**router-list-num** *num* [[**l2-redirect**] [**password** *key*] [**weight** *percentage*]]}	Configures WCCP Version 2 media caching service.
wccp port-list	**wccp port-list** *listnum portnum*	Associates ports with specific WCCP Version 2 dynamic services.
wccp reverse-proxy	**wccp reverse-proxy** {**router-list-num** *num* [[**l2-redirect**] [**password** *key*] [**weight** *percentage*]]}	Configures WCCP Version 2 reverse proxy web caching service.
wccp router-list	**wccp router-list** *number ip-address*	Creates a router list for use in WCCP services.

Table 1-5 ACNS Software Global Configuration Commands (continued)

Global Configuration Command	Syntax	Description
wccp service-number	**wccp service-number** *servnumber* {**router-list-num** *num* **port** *port* **application** {**cache** | **streaming**} [[**hash-destination-ip**] [**hash-destination-port**] [**hash-source-ip**] [**hash-source-port**] [**l2-redirect**] [**password** *key*] [**weight** *percentage*]]}	Configures WCCP Version 2 service number.
wccp shutdown	**wccp shutdown max-wait** *seconds*	Sets the maximum time interval over which the Content Engine will perform a clean shutdown.
wccp slow-start	**wccp slow-start enable**	Accepts traffic load in slow-start mode.
wccp version	**wccp version** {**1** | **2**}	Specifies WCCP version number.
wccp web-cache	**wccp web-cache** {**router-list-num** *num* [[**l2-redirect**] [**password** *key*] [**weight** *percentage*]]}	Configures standard web caching service.
wccp wmt	**wccp wmt** {**router-list-num** *num* [[**l2-redirect**] [**password** *key*] [**weight** *percentage*]]}	Configures the web cache service to run with WCCP Windows Media Technologies (WMT).
wmt	**wmt** {**accept-license-agreement** | **broadcast** {**alias-name** *name* **source** *url*} | **cache** {**enable** | **max-obj-size** *size*} | **disallowed-client-protocols** [**HTTP** | **TCP** | **UDP**] | **enable** | **evaluate** | **incoming** *number* | **l4-switch** {**enable**} | **license-key** *key* | **max-bandwidth** *size* | **max-bitrate** *bit_rate* | **max-current-sessions** *number* | **multicast** {**schedule-start** *name minute hour day month* | **station-configuration** *name dest_addr dest_port media_source* [**play-forever**]}}	Configures Windows Media Technologies (WMT).

Interface Configuration Command Summary

The interface configuration commands are entered in the interface configuration mode.

The following example demonstrates how to enter the interface configuration mode for a Fast Ethernet port:

```
Console# config
Console(config)# interface ?
FastEthernet     Select a fast ethernet interface to configure
GigabitEthernet  Select a gigabit ethernet interface to configure
Console(config)# interface fastethernet ?
<0-3>/  FastEthernet slot/port
Console(config)# interface fastethernet 0/1
Console(config-if)#
Console(config-if)# ?
Configure Interface commands:
autosense    Interface autosense
bandwidth    Interface bandwidth
```

```
exit         Exit from this submode
full-duplex  Interface fullduplex
  half-duplex  Interface halfduplex
  ip           Interface Internet Protocol Config commands
  no           Negate a command or set its defaults
  shutdown     Shutdown the specific interface
```

To exit the interface configuration mode, enter **exit** to return to the global configuration mode.

```
Console(config-if)# exit
Console(config)#
```

Table 1-6 lists the interface configuration commands.

Table 1-6 ACNS Software Interface Configuration Commands

Interface Command	Syntax	Description
autosense	**autosense**	Sets current interface to autosense.
bandwidth	**bandwidth** *mbits*	Sets specified interface line speed (10, 100 Mbps).
cdp	**cdp** {**enable** | **full-duplex** | **half-duplex** | **ip address** *ip-address ip-subnet*}	Sets Cisco Discovery Protocol interface configuration commands.
exit	**exit**	Exits from interface mode.
fullduplex	**fullduplex**	Sets current interface to full-duplex mode.
halfduplex	**halfduplex**	Sets current interface to half-duplex mode.
ip	**ip** {**address** *ip-address ip-subnet*}	Configures specified interface Internet Protocol parameters.
no	**no** {**autosense** | **bandwidth** | **cdp** | **fullduplex** | **halfduplex** | **ip** | **shutdown** | **standby**}	Negates a command or sets its defaults.
shutdown	**shutdown**	Shuts down the specified interface.
standby	**standby** *group_number* {**errors** *max_errors* | **ip** *ip-address netmask* | **priority** *priority_level*}	Configures an interface to be a backup for another interface.

show Command Summary

The Content Engine show commands are entered in the EXEC mode. Table 1-7 lists the **show** commands.

Table 1-7 ACNS Software show Commands

EXEC show Command	Syntax	Description
show arp	**show arp**	Displays Address Resolution Protocol (ARP) entries.
show authentication	**show authentication {http-request | user}**	Displays authentication configuration.
show boomerang	**show boomerang**	Displays boomerang content routing information.

Table 1-7 ACNS Software show Commands (continued)

EXEC show Command	Syntax	Description
show bypass	**show bypass** [**list**] [**statistics** {**auth-traffic** | **load**}] [**summary**]	Displays Content Engine bypass configuration.
show cdp	**show cdp** [**entry** *neighbor* {*protocol* | *version*} | **holdtime** | **interface** {**fastEthernet** *slot/port* | **gigabitEthernet** *slot/port*} | **neighbors** {**detail** | **fastEthernet** {*slot/port* | *detail*} | **gigabitEthernet** {*slot/port* | *detail*}} | **run** | **timer** | **traffic**]	Displays Cisco Discovery Protocol configuration.
show cfs	**show cfs** {**statistics** | **volumes**}	Displays cache file system status.
show clock	**show clock** [**detail**]	Displays the system clock.
show debugging	**show debugging**	Displays the state of each debugging option.
show disks	**show disks** [**configured** | **current** | **details** | **raid-info**]	Displays disk configurations.
show dns-cache	**show dns-cache**	Displays DNS cache information.
show ecdn	**show ecdn**	Displays Enterprise CDN information.
show ecdnfs volumes	**show ecdnfs volumes**	Displays Enterprise CDN file system (ecdnfs) information.
show error-handling	**error-handling** {**reset-connection** | **send-cache-error** | **transparent**}	Sets error-handling options.
show flash	**show flash**	Displays Flash memory information.
show ftp	**show ftp**	Displays File Transfer Protocol (FTP) caching-related configuration.
show gui-server	**show gui-server**	Displays the graphical user interface (GUI) server configuration.
show hardware	**show hardware**	Displays system hardware information.
show hosts	**show hosts**	Displays IP domain name, name servers, and host table.
show http	**show http** {**age-mult** | **all** | **anonymizer** | **append** | **authenticate-strip-ntlm** | **authentication** | **cache-authenticated** | **cache-cookie** | **cache-on-abort** | **client-no-cache-request** | **cluster** | **object** | **persistent-connections** | **proxy** | **reval-each-request** | **strict-request-content-length- checking** | **ttl**}	Displays HTTP-related caching configuration.
show http-authcache	**show http-authcache**	Displays authentication cache.
show https	**show https {all** | **destination-port** | **proxy**}	Displays HTTPS-related parameters.

Table 1-7 ACNS Software show Commands (continued)

EXEC show Command	Syntax	Description
show icp	**show icp** {**client** | **root** | **server**}	Displays Internet Cache Protocol (ICP) information.
show interface	**show interface** {**FastEthernet** *slot/port* | **GigabitEthernet** *slot/port* | **scsi** *number*}	Displays hardware interface information.
show inetd	**show inetd**	Displays the status of TCP/IP services.
show ip routes	**show ip routes**	Displays IP routing table.
show ldap	**show ldap**	Displays LDAP parameters.
show logging	**show logging**	Displays system logging configuration.
show mediafs	**show mediafs volumes**	Displays media file system (mediafs) information.
show memory	**show memory**	Displays memory blocks and statistics.
show multicast-client	**show multicast-client** [**license-agreement**]	Displays multicast client configuration and license parameters.
show ntlm	**show ntlm**	Displays NTLM parameters.
show ntp	**show ntp status**	Displays the NTP configuration status.
show pre-load	**show pre-load**	Displays preload configuration.
show processes	**show processes** [**cpu** | **memory**]	Displays process status.
show proxy-auto-config	**show proxy-auto-config**	Displays the state of the browser automatic configuration feature.
show proxy-protocols	**show** {**all** | **outgoing-proxy** | **transparent**}	Displays proxy protocols parameters.
show radius-server	**show radius-server**	Displays RADIUS server information.
show real-subscriber	**show real-subscriber** [**license-agreement**]	Displays RealSubscriber configuration and license parameters.
show rtsp	**show rtsp** {**all** | **license-agreement** | **proxy**}	Displays the RTSP configurations.
show rule	**show rule** {**action** {action-type {**all** | **pattern** pattern-type} | **all**} For a more complete explanation of specific rules, refer to the "show rule" section on page 2-210	Displays the Rules Template configuration information.
show running-config	**show running-config**	Displays the current operating configuration.
show services	**show services** {**ports** [*portnum*] | **summary**}	Displays services-related information.
show snmp	**show snmp** {**engineID** | **group** | **stats**}	Displays SNMP parameters.
show ssh	**show ssh**	Displays Secure Shell (SSH) status and configuration.
show standby	**show standby**	Displays standby interface-related information.

Table 1-7 ACNS Software show Commands (continued)

EXEC show Command	Syntax	Description
show startup-config	**show startup-config**	Displays the startup configuration.
show statistics	**show statistics** {**authentication** | **boomerang** [**domain** *domainname*] | **bypass** [**auth-traffic** | **load** | **summary**] | **cfs** | **dns-cache** | **ftp** | **http** {**cluster** | **ims** | **object** | **performance** | **proxy outgoing** | **requests** | **savings** | **usage**} | **http-authcache** | **https** | **icmp** | **icp** {**client** | **server**} | **ip** | **ldap** | **mediacache real** {**requests** | **savings**} | **netstat** | **ntlm** | **pre-load** | **radius** | **rule** {**action** {action-type {**all** | **pattern** pattern-type} | **services** | **all**}} | **snmp** | **streamstat** | **tacacs** | **tcp** | **transaction-logs** | **udp** | **url-filter** {**N2H2** | **websense**} | **wmt** {**all** | **bytes** | **errors** | **multicast** | **requests** | **savings** | **usage** [**detail** | **summary**]}	Displays statistical system information.
show sysfs	**show sysfs volumes**	Displays system file system (sysfs) information.
show tacacs	**show tacacs**	Displays TACACS+ configuration.
show tcp	**show tcp**	TCP configuration.
show tech-support	**show tech-support** [**page**]	Displays system information for Cisco technical support.
show tftp-server	**show tftp-server**	Displays Trivial File Transfer Protocol (TFTP) server configuration.
show transaction-logging	**show transaction-logging**	Displays transaction logging information.
show trusted-host	**show trusted-host**	Displays the name of the trusted host.
show url-filter	**show url-filter**	Displays URL filter configurations.
show user	**show user username** *name*	Displays user information.
show users	**show users** {**administrative** | **request-authenticated**}	Displays specified users.
show version	**show version**	Displays system version.
show wccp	**show wccp** {**content-engines** | **flows** {**custom-web-cache** | **media-cache** | **reverse-proxy** | **web-cache** | **wmt**} [**summary**] | **gre** | **modules** | **port- list** | **routers** | **services** [**detail**] | **slowstart** {**custom-web-cache** | **media-cache** | **reverse-proxy** | **web- cache** | **wmt**} | **status**}	Displays WCCP information.
show wmt	**show wmt** [**license-agreement** | **proxy**]	Displays WMT configuration.

CHAPTER 2

Cisco ACNS Software Commands

This chapter contains an alphabetical listing of all commands of Cisco ACNS software.

asset tag

To set the tag name for the CISCO-ENTITY-ASSETT-MIB, use the **asset** command in global configuration mode.

asset tag *name*

no asset tag *name*

Syntax Description

name	Tag name for the CISCO-ENTITY-ASSETT-MIB.

Defaults

No default behaviors or values

Command Modes

Interface configuration

Usage Guidelines

Use this command to set the tag name.

Examples

```
Console(config)# asset tag entitymib
```

authentication

To configure user authentication options, use the **authentication** command in global configuration mode. Use the **no** form of the command to selectively disable options.

authentication {**configuration** {**local** | **tacacs**} **enable** [**primary** | **secondary**] | **login** {**local** | **tacacs**} **enable** [**primary** | **secondary**]}

no authentication {**configuration** {**local** | **tacacs**} **enable** [**primary** | **secondary**] | **login** {**local** | **tacacs**} **enable** [**primary** | **secondary**]}

Syntax Description

configuration	Sets configuration authentication (authorization).
local	Selects local database for authentication.
tacacs	Selects TACACS+ database for authentication.
enable	Enables database for configuration authentication.
primary	(Optional) Sets selected authentication database as the primary.
secondary	(Optional) Sets selected authentication database as the secondary.
login	Sets login authentication.
enable	Enables database for login authentication.

Defaults

Local authentication methods are enabled by default.

Command Modes

Global configuration

Usage Guidelines

Authentication, also referred to as "login," is the act of verifying usernames and passwords. Authorization refers to the setting of privileges to authenticated users in to a network.

The **authentication** command configures the authentication and authorization methods that govern login and configuration access to the Content Engine. ACNS 4.1 software supports local and Terminal Access Controller Access Control System Plus (TACACS+) authentication and authorization methods.

Login and configuration privileges can be obtained from both the local database or the TACACS+ remote database. If both databases are enabled, then both databases are queried; if the user data cannot be found in the first database queried, then the second database is tried.

The **authentication login** command specifies the method that determines whether the user has any level of access permission to the Content Engine. The **authentication configuration** command specifies the method that authorizes the user with privileged access (configuration access) to the Content Engine.

The **authentication login local** and the **authentication configuration local** commands use a local database for authentication and authorization.

The **authentication login tacacs** and **authentication configuration tacacs** commands use a remote TACACS+ server to determine the level of user access. The Content Engine **tacacs** global configuration command and a TACACS+ server must be configured to use the TACACS+ authentication and authorization method.

When the **primary** keyword is entered for TACACS+ login or configuration authentication, the TACACS+ database is queried first, and the local database is queried second. If the TACACS+ database is not designated as primary, and both the local and the TACACS+ database are enabled, the local database is queried first. If both the local and the TACACS+ databases are disabled (**no authentication**), the Content Engine verifies that both are disabled and if so, sets the Content Engine to the default state.

By default, the local method is enabled and TACACS+ is disabled for both login and configuration. Whenever TACACS+ is disabled, local is automatically enabled. Both TACACS+ and local methods can be enabled at the same time. The **primary** option specifies the first method to attempt; the **secondary** option specifies the method to use if the primary method fails. If both methods of an **authentication login** or **authentication configuration** command are configured as primary, or both as secondary, local is attempted first, then TACACS+.

Examples

The following example enables local and TACACS+ authentication and authorization, setting TACACS+ as the first method used and local as the secondary method to use if TACACS+ fails.

```
Console(config)# authentication login tacacs enable primary
Console(config)# authentication login local enable secondary
Console(config)# authentication configuration local enable secondary
Console(config)# authentication configuration tacacs enable primary
```

This is an example of the **show authentication** command.

```
Console# show authentication
Login Authentication:         Console/Telnet Session
----------------------------- ----------------------
local                         enabled
tacacs                        enabled (primary)

Configuration Authentication: Console/Telnet Session
----------------------------- ----------------------
local                         enabled
tacacs                        enabled
```

This is an example of the **show statistics authentication** command.

```
Console# show statistics authentication

Authentication Statistics
-------------------------------------
Number of access requests: 37
Number of access deny responses: 14
Number of access allow responses: 23
```

Related Commands

show authentication

show statistics authentication

tacacs

autosense

To enable autosense on an interface, use the **autosense** interface configuration command. To disable this function, use the **no** form of this command.

autosense

no autosense

Syntax Description

This command has no arguments or keywords.

Defaults

Autosense is enabled by default.

Command Modes

Interface configuration

Usage Guidelines

Cisco router Ethernet interfaces do not negotiate duplex settings. If the Content Engine is connected to a router directly with a crossover cable, the Content Engine interface must be manually set to match the router interface settings. Disable **autosense** before configuring an Ethernet interface. When **autosense** is on, manual configurations are overridden. You must reboot the Content Engine to start autosensing.

Examples

```
ContentEngine(config-if)# autosense

ContentEngine(config-if)# no autosense
```

bandwidth

To configure an interface bandwidth, use the **bandwidth** interface configuration command. To restore default values, use the **no** form of this command.

bandwidth *mbits*

no bandwidth

Syntax Description

mbits	Bandwidth size in megabits per second (Mbps) (10 or 100).

Defaults

1000 Mbps on Gigabit Ethernet interfaces.

Command Modes

Interface configuration

Usage Guidelines

Use this command to set the bandwidth on Fast Ethernet interfaces. Gigabit Ethernet interfaces run at 1000 Mbps only.

Examples

```
ContentEngine(config-if)# bandwidth 10

ContentEngine(config-if)# no bandwidth
```

boomerang

To enable boomerang content routing on the Content Engine and enter domain configuration mode, use the **boomerang** domain configuration command.

boomerang {**dns** {**domain** *domain-name* [**alias** *alias-name* | **content-server** *ip-address* [**file** *filename*] | **dns-ttl** *seconds* | **hops** *hops* | **key** {**0** *keyword* | **7** *keyword* | *keyword*} | **origin-server** *ip-address*] | **enable**} | **log-races enable**}

no boomerang {**dns** {**domain** *domain-name* [**alias** *alias-name* | **content-server** *ip-address* [**file** *filename*] | **dns-ttl** *seconds* | **hops** *hops* | **key** {**0** *keyword* | **7** *keyword* | *keyword*} | **origin-server** *ip-address*] | **enable**} | **log-races enable**}

Syntax Description

dns	Establishes a domain name to be served by boomerang and configures a DNS boomerang distributed reverse proxy.
domain	Establishes support for a client domain. Enters domain configuration mode.
domain-name	Domain name string (maximum string length is 99 characters).
alias	(Optional) Creates an alias domain name.
alias-name	Alias name.
content-server	(Optional) Specifies a server IP address of the local cache or mirror.
ip-address	IP address of local cache or mirror.
file	(Optional) File to check if content server is alive.
filename	Filename to probe (for example, /index.html).
dns-ttl	Sets the DNS TTL (Time To Live).
seconds	Time To Live value in seconds (0–4294967294).
hops	Sets the number of hops to live.
hops	Number of hops to live (0–255).
key	Shared secret string.
0	Specifies clear text key.
7	Specifies type 7 encrypted key.
keyword	RC4 Shared Secret (clear text).
origin-server	Sets IP address of origin server.
ip-address	IP address of origin server.
enable	Enables the boomerang software.
log-races enable	Enables logging the result of races.

Defaults

dns-ttl: 20 seconds

key: **0** (clear text)

log-races enable: disabled

Command Modes

Domain configuration

Usage Guidelines

Use the **boomerang dns enable** command to enable content routing software on a Content Engine that you want to configure as a content routing agent. Use the **boomerang dns domain** command to configure the Content Engine as a content routing agent for a specified domain and to enter the domain configuration mode to establish operating parameters for the specified domain name.

Boomerang agents support multiple domains, where each agent domain may be associated with a different boomerang server. Other than memory limits, there are no limits to the number of domains supported on the agent. For more information on the boomerang agent, see the *Cisco Content Routing Software Configuration and Command Reference, Release 1.1*.

Caution A Content Engine cannot be used for transparent caching if it has been configured as a content routing agent. Therefore, if you want to use a Content Engine for transparent caching, make sure that none of the **boomerang** commands are enabled on the Content Engine.

Use the **boomerang dns domain** *domain-name* **alias** *alias-name* command to set alternate boomerang domain names that share the same operating parameters. If you are using the Content Engine as a content routing agent, use this command on both the Content Router and the Content Engine to establish an alternative name for a domain.

Note Corresponding alias domain names must also be configured on the boomerang server. Each client domain can be associated with a different boomerang server.

If the Content Engine is not used to serve web pages, use the **content-server** *ip-address* **file** *filename* option to specify the address of the cache to be used. If it wins the race, the content server is the local web cache or mirror cache that serves content for the requesting web client that initiated the DNS race. The boomerang client probes the content server periodically to ensure that it is running and able to serve web pages. The probe consists of an HTTP GET request for the configured filename. A response of 200 OK indicates that the content server is running. If a filename is not given, attempts to connect are made only through port 80.

If you are using the Content Engine as a content routing agent, use the **boomerang dns domain** *domain-name* **dns-ttl** command to specify the DNS TTL value contained in the DNS response generated by the agent. In general, a lower DNS TTL value ensures more recent content, whereas a higher DNS TTL value reduces the Content Router load. The higher the DNS TTL value, the less the load on the Content Router. A lower value means an increased Content Router load, but also means that the addresses of Content Engines that won DNS races are used for a shorter length of time in the annealing process. For example, if the DNS TTL is set at 60 seconds, a DNS server returns to the Content Router to look up a domain name no more than once a minute. In other words, the name server uses the winning Content Engine address for 60 seconds before consulting the Content Router again. Use **no dns-ttl** to reset the delay to its default value.

Note A **dns-ttl** command entered on a Content Engine overrides a **dns-ttl** command entered on the Content Router.

The number of hops to live value of the DNS response is generated by the client. The value specified by the **hops** option overrides the value specified by the boomerang server.

The **key** shared secret string specifies the secret that is matched against the secret contained in the packets sent by the server. The shared secret configured on the client domain needs to be the same as the secret configured on the server.

Use the **origin-server** *ip-address* subcommand if the Content Engine is used as a cache rather than mirror site. If the web cache does not have the requested content, or there is a cache miss, it must get the content from the origin server and cache it for future requests. Because the Content Engine web cache does not have the IP address of the origin server, this sub-command must be set to provide the IP address to get content from the origin server.

The **boomerang log-races enable** command enables logging of the DNS IP address resolution timing results between the boomerang server and the agent. To disable the command, enter the **no boomerang log-races** command.

To delete a domain, enter the **no boomerang** command. It is not necessary to enter arguments and variables before deleting the current domain name.

Examples

In the following example, assume that a domain is named www.foobar.com. It is given the domain name www.foobar.com on the Content Router.

```
Console(config)# boomerang dns enable
Console(config)# boomerang dns domain www.foobar.com
```

In the following example, assume that a domain is named www.foobar.com. It is given the alias www.foobar.net on the Content Router.

```
Console(config-domain)# alias www.foobar.net
```

When configuring www.foobar.com on the agent, enter the alias on the Content Engine as follows.

```
Console(config-domain)# alias www.foobar.net
```

Related Commands

show boomerang

boomerang dump-log

To write boomerang memory data to a local disk file, use the **boomerang dump-log** EXEC command.

boomerang dump-log

no boomerang dump-log

Syntax Description

This command has no arguments or keywords.

Defaults

No default behaviors or values

Command Modes

EXEC

Usage Guidelines

Enable the boomerang logging function with the **boomerang log-races enable** command, and then dump the log to file using the **boomerang dump-log** EXEC command. The command writes data in memory to a disk file, for example:

```
/local/local1/logs/boomerang/boomlog.txt
```

Examples

```
Console(config)# boomerang dump-log
writing Boomerang events to /local1/logs/boomerang/boomlog.txt file ..
 ....finished
```

Related Commands

boomerang send-packet

boomerang send-packet

To send test packets to determine whether or not a destination accepts boomerang-altered source IP addresses, use the **boomerang send-packet** EXEC command.

boomerang send-packet {**tcp** | **udp**} *dest-port source-port* {*dest-ip-address* | *dest-hostname*} {*source-ip-address* | *source-hostname*}

no boomerang send-packet {**tcp** | **udp**} *dest-port source-port* {*dest-ip-address* | *dest-hostname*} {*source-ip-address* | *source-hostname*}

Syntax Description

tcp	Sends a TCP packet.
udp	Sends a UDP packet.
dest-port	Destination port number (1–65535).
source-port	Source port number (1–65535).
dest-ip-address	IP address of the destination site.
dest-hostname	Name of the destination host.
source-ip-address	IP address of the source.
source-hostname	Name of the source host.

Defaults

No default behavior or values

Command Modes

EXEC

Usage Guidelines

Some networks may have filters that prevent the transmission of packets with source addresses outside the address space of the network. If you are using the Content Engine as a content routing agent, such filters could inhibit the content routing process. To determine whether such filters exist, use a sniffer and the **boomerang send-packet** command to send a packet with a source address outside the subnet on which the Content Engine resides. The sniffer should be set up to monitor traffic on the network of the destination site to which the packet is sent. If the sniffer detects this packet, you know that the destination can accept boomerang-altered source IP addresses.

Examples

```
Console# boomerang send-packet tcp 53 53 10.1.1.1 10.1.1.2
```

Related Commands

boomerang dump-log

bypass

To enable transparent error handling and dynamic authentication bypass, and to configure static bypass lists, use the **bypass** global configuration command. To disable the bypass feature, use the **no** form of the command.

bypass auth-traffic enable

bypass load {**enable** | **in-interval** *seconds* | **out-interval** *seconds* | **time-interval** *minutes*}

bypass static *clientipaddress* {*serveripaddress* | **any-server**}

bypass static any-client *serveripaddress*

bypass timer *minutes*

no bypass {**auth-traffic enable** | **load** {**enable** | **in-interval** *seconds* | **out-interval** *seconds* | **time-interval** *minutes*} | **static** {*clientipaddress* {*serveripaddress* | **any-server**} | **any-client** *serveripaddress*} | **timer** *minutes*}

Syntax Description

auth-traffic	Sets authenticated traffic bypass configuration.
enable	Enables authenticated traffic bypass.
load	Sets bypass load configuration.
enable	Enables bypass load.
in-interval	Sets time interval between buckets coming back.
seconds	Time in seconds (2–600).
out-interval	Sets time interval between bypassing buckets.
seconds	Time in seconds (4–600).
time-interval	Sets time that a bucket is bypassed.
minutes	Time in minutes (1–1440).
static	Adds a static entry to the bypass list.
clientipaddress	Requests from this IP address bypass the Content Engine.
serveripaddress	Requests from a specified client to this specific server bypass the Content Engine.
any-server	Requests from a specified client to any server bypass the Content Engine.
any-client	Bypasses HTTP traffic from any client destined to a particular server.
serveripaddress	IP address of the web server to be bypassed.
timer	Sets authentication bypass timer in minutes. The bypass entry is removed from the dynamic list when the timer expires.
minutes	Time in minutes (1–1440).

Defaults

bypass timer: 20 minutes

in-interval: 60 seconds

out-interval: 4 seconds

time-interval: 10 minutes

Command Modes

Global configuration

Usage Guidelines

Bypass features are available only with WCCP Version 2. The Content Engine can only set up a bypass for WCCP-redirected traffic, not proxy-style requests.

Authentication Traffic Bypass

Some web sites, because of IP authentication, do not allow the Content Engine to connect directly on behalf of the client. To preserve transparency and to avoid a disruption of service, the Content Engine can use authentication traffic bypass to automatically generate a dynamic access list for these client/server pairs. Authentication bypass triggers are also propagated upstream and downstream in the case of hierarchical caching. When a client/server pair goes into authentication bypass, it is bypassed for an amount of time set by the **bypass timer** command (20 minutes by default).

Dynamic Traffic Bypass

The following two scenarios describe typical dynamic traffic bypass situations:

Scenario 1—Dynamic Bypass Upon Receiving a Web Server Error

A user issues an HTTP request from a web browser. The request is transparently intercepted and redirected to the Content Engine. The Content Engine accepts the incoming TCP connection from the web browser, determines that the request is for an object not in storage (cache-miss), and issues a request for the object from the origin web server, but receives some kind of error (for instance, a protocol or authentication error) from the web server.

The Content Engine has already accepted the TCP connection from the web browser and the three-way TCP handshake has taken place. The Content Engine detects that the transaction with the web server is failed, but does not know the cause (the origin web server is performing authentication based on user source IP address, incompatibility between the TCP stacks, and so forth).

If **error-handling transparent** (the default) is configured and if the Content Engine receives an error from the origin server, the Content Engine sends a 200 OK response back to the browser with instructions to refresh the URL as follows.

```
HTTP/1.0 200 OK
Cache-Control; no-cache
Connection: Close
```

This refresh instruction causes the client to send the request again. On the connection retry, the Content Engine does not accept the connection. It passes the request back to the WCCP-enabled router or switch unintercepted. The router then sends the flow toward the origin web server directly from the web browser, thereby bypassing the Content Engine.

Scenario 2—Dynamic Bypass Upon Receiving an Unsupported Protocol

When the Content Engine receives non-HTTP requests over TCP port 80, the Content Engine issues a “retry” response, closes the connection, and does not accept subsequent connections in the same manner as in scenario 1.

Note Non-HTTP includes nonconforming HTTP as well as different protocols such as Secure Shell (SSH), Simple Mail Transfer Protocol (SMTP), or Network News Transport Protocol (NNTP). An example of nonconforming HTTP is the failure of a web server to issue two carriage return and line feeds at the end of the HTTP header section.

These two scenarios implement the WCCP return-path functionality in WCCP, which is a mechanism whereby a Content Engine can return traffic to the WCCP-enabled router or switch, telling the router or switch to forward the packets as if the Content Engine was not present.

It is typical for about 3 percent of all HTTP traffic flows to have some kind of failure condition. These failed flows are automatically retried using authentication bypass or dynamic client bypass, demonstrating that the failure conditions were preexisting and not due to the deployment of transparent caching.

Overload Bypass

If a Content Engine becomes overwhelmed with traffic, it can use the bypass load feature to reroute the overload traffic.

When the Content Engine is overloaded and **bypass load** is enabled, the Content Engine bypasses a bucket. If the load remains too high, another bucket is bypassed, and so on until the Content Engine can handle the load. The time interval between one bucket being bypassed and the next, is set by the **out-interval** option. The default is 4 seconds.

When the first bucket bypass occurs, a time interval must elapse before the Content Engine begins to again service the bypassed buckets. The duration of this interval is set by the **time-interval** option. The default is 10 minutes.

When the Content Engine begins to service the bypassed traffic again, it begins with a single bypassed bucket. If the load is serviceable, it picks up another bypassed bucket, and so on. The time interval between picking up one bucket and the next is set by the **in-interval** option. The default is 60 seconds.

Bypass Static

The **bypass static** command permits traffic from specified sources to bypass the Content Engine. The type of traffic sources are as follows:

- Specific web client to a specific web server
- Specific web client to any web server
- Any web client to a specific web server

Wildcards in either the source or the destination field are not supported.

To clear all static configuration lists, use the **no** form of the command.

Examples

This example forces HTTP traffic from a specified client to a specified server to bypass the Content Engine.

```
ContentEngine(config)# bypass static 10.1.17.1 172.16.7.52
```

This example forces all HTTP traffic destined to a specified server to bypass the Content Engine.

```
ContentEngine(config)# bypass static any-client 172.16.7.52
```

This example forces all HTTP traffic from a specified client to any web server to bypass the Content Engine.

```
ContentEngine(config)# bypass static 10.1.17.1 any-server
```

This example forces all authenticated HTTP traffic to bypass the Content Engine for 24 hours.

```
ContentEngine(config)# bypass auth-traffic enable
ContentEngine(config)# bypass timer 1440
```

A static list of source and destination addresses helps to isolate instances of problem-causing clients and servers.

- To display static configuration list items, use the **show bypass list** command.

```
ContentEngine# show bypass list
Client              Server          Entry type
------              ------          ----------
10.1.17.1:0         172.16.7.52:0   static-config
any-client:0        172.16.7.52:0   static-config
10.1.17.2:0         any-server:0    static-config
```

- The total number of entries in the bypass list is reported by the **show bypass summary** command.

```
Total number of HTTP connections bypassed = 0
        Connections bypassed due to system overload               = 0
        Connections bypassed due to authentication issues         = 0
        Connections bypassed due to facilitate error transparency = 0
        Connections bypassed due to static configuration          = 0

Total number of entries in the bypass list = 3
        Number of Authentication bypass entries = 0
        Number of Error bypass entries          = 0
        Number of Static Configuration entries  = 3
```

Related Commands

rule

show bypass

show statistics bypass

clear bypass

cache

To synchronize the cache file system (cfs) contents from memory to disk, use the **cache synchronize** EXEC command.

cache {**clear** [**force**] | **reset** | **synchronize**}

To clear the disk of all cached content, use the **cache clear** EXEC command.

Syntax Description

clear	Clears the cache.
force	(Optional) Forces deletion of all cached objects.
reset	Resets the cache (unmounts, formats, and mounts cache file system volumes).
synchronize	Synchronizes the cache.

Defaults

No default behavior or values

Command Modes

EXEC

Usage Guidelines

The **cache clear** command removes all cached contents from the currently mounted cfs volumes. Objects being read or written are removed when they cease being "busy." The equivalent to this command is the **clear cache** or **cfs clear** command.

Caution The **cache clear** command is irreversible, and all cfs cached content will be erased.

The **cache clear force** deletes all cfs objects, whether busy or not, and may generate broken GIF or HTML messages for objects that were being read from the disk when the command was executed. If an object is being written to the Content Engine disk when a **cache clear force** command is executed, the application stops caching that object but still delivers the object from the web server to the client.

The **cache synchronize** command synchronizes the cache file system and the media file system contents from memory to disk. Although synchronization is performed at regular intervals while the Content Engine is operating, this command can be used to ensure that all data is written to disk before you reset or turn off the Content Engine. Synchronization can also be done using the **cfs sync** and **mediafs sync** command.

Examples

```
ContentEngine# cache clear force
```

Related Commands

clear cache

cfs

mediafs

cdp

To configure Cisco Discovery Protocol (CDP) options, use the **cdp** command in global configuration mode.

cdp {**enable** | **holdtime** *seconds* | **timer** *seconds*}

no cdp {**enable** | **holdtime** *seconds* | **timer** *seconds*}

Syntax Description

enable	Enables CDP.
holdtime	Specifies in seconds the period of time a receiver keeps CDP packets. The default is 180 seconds.
seconds	Time in seconds (10–255).
timer	Specifies the rate at which CDP packets are sent. The default is 60 seconds.
seconds	Time in seconds (5–254).

Defaults

holdtime: 180 seconds

timer: 60 seconds

Command Modes

Global configuration

Usage Guidelines

When enabled using the **cdp enable** command, Cisco Discovery Protocol (CDP) obtains protocol addresses of neighboring devices and discovers the platform of those devices. It also shows information about the interfaces used by your router. CDP is media- and protocol-independent, and runs on Cisco-manufactured equipment.

Use of SNMP with the CDP Management Information Base (MIB) 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. Cisco Discovery Protocol uses the CISCO-CDP-MIB.

Each device configured for CDP sends periodic messages, known as advertisements, to a multicast address. The **cdp time** *seconds* command specifies the rate at which CDP packets are sent. Each device advertises at least one address at which it can receive SNMP messages. The advertisements also contain Time To Live or hold time information. To set the hold time, use the **cdp holdtime** *seconds* command to specify the period of time in seconds that a receiver is to keep CDP packets. Each device also listens to the periodic CDP messages sent by others to learn about neighboring devices.

Examples

In the following example, three command lines are entered in sequence. CDP is first enabled, the hold time is set to 10 seconds for keeping CDP Packets, and then the rate at which CDP packets are sent (15 seconds) is set.

```
ContentEngine(config)# cdp enable
ContentEngine(config)# cdp holdtime 10
ContentEngine(config)# cdp timer 15
```

Related Commands

clear cdp counters

clear cdp table

show cdp

cfs

To manipulate the cache object file system of the Content Engine, use the **cfs** EXEC command.

cfs clear *partition* [**force**]

cfs format *partition*

cfs mount *partition*

cfs reset *partition*

cfs sync *partition*

cfs unmount *partition*

no cfs {**clear** *partition* [**force**] | **format** *partition* | **mount** *partition* | **reset** *partition* | **sync** *partition* | **unmount** *partition*}

Syntax Description

clear	Deletes nonbusy objects from the specified cfs volume.
force	(Optional) Forcibly deletes all objects from the specified cfs volume.
format	Erases and formats or creates a file system for caching.
mount	Mounts a cache file system.
reset	Resets (unmounts-formats-mounts) a cache file system.
sync	Synchronizes a cache file system.
unmount	Unmounts a cache file system.
partition	Partition name (for example, disk00/00, disk00/01, disk01/00).

Defaults

No default behavior or values

Command Modes

EXEC

Usage Guidelines

Cache objects retrieved from the web are saved and manipulated with the cache file system (cfs) on a cfs partition of the hard disk. This does not affect the sysfs, swfs, or mediafs partitions. The **cfs** commands are used to manage the cache object file system.

The **cfs clear** command deletes nonbusy objects from the specified cfs volume. A nonbusy object is an object that is not being accessed (read or written). The **cfs clear** command (without force) deletes all possible objects without generating a broken GIF or HTML message to the client.

The **cfs clear force** command deletes all objects, busy or nonbusy, and may generate broken GIF or HTML messages for objects that were being read from the disk when the command was executed. If an object is being written to the Content Engine disk when a **cfs clear force** command is executed, the application stops caching that object but still delivers the object from the web server to the client.

The **cfs reset** command unmounts, formats, and mounts a specified volume. Unmounting a volume can result in broken GIF or HTML messages for objects that are being read from the disk (cache hits) when the command is executed. When a cfs volume is reset, all cfs data on that volume is lost.

Note The **cfs reset** command can be invoked on unmounted volumes.

The **cfs format** command creates the cache file system internal "dbs" for the cfs partition of the disk if the volume is unmounted. It formats the cfs partition to prepare it for a cfs mount. The **cfs mount** command creates and maps data structures in memory to the cfs partition.

Caution All cached content is erased with the format command.

The **cfs unmount** command frees the in-memory data structures that map to the physical (disk) cfs partition.

The **cfs sync** command synchronizes the cache file system contents from memory to disk. Although synchronization is performed at regular intervals while the Content Engine is running, this command can be used to ensure that all data is written to disk before you reset or turn off the Content Engine. Synchronization can also be done with the **cache synchronize** command.

Examples

```
ContentEngine# cfs sync disk05
```

Related Commands

show cfs

cache clear

clear cache

clear

To clear the HTTP object cache, the hardware interface, statistics, archive working transaction logs, and other settings, use the **clear** EXEC command.

clear bypass {**counters** | **list**}

clear cache [**dns** [**domain** *domainname* | **hostname** *hostname*] | **http** [**url** *url*] | **http-authentication** | **real-proxy** | **wmt**]

clear cdp {**counters** | **table**}

clear logging

clear statistics {**all** | **authentication** | **boomerang** | **dns-cache** | **ftp** | **history** | **http** {**all** | **cluster** | **errors** | **ims** | **object** | **outgoing** | **proxy outgoing** | **requests** | **response** | **savings**} | **http-authcache** | **https** | **icp** {**all** | **client** | **server**} | **ip** | **ldap** | **mediacache real** | **ntlm** | **pre-load** | **radius** | **rule** {**action** {action-type {**all** | **pattern** pattern-type} | **all**}} | **running** | **tacacs** | **tcp** | **transaction-logs** | **url-filter** {**N2H2** | **websense**} | **wmt**}

clear transaction-log

Syntax Description

bypass	Clears bypass commands.
counters	Clears all bypass counters.
list	Clears all bypass lists.
cache	Clears the HTTP object from the cfs cache.
dns	(Optional) Clears cached DNS entries in the HTTP proxy.
domain	(Optional) Specifies the DNS cache domain name.
domainname	DNS cache domain name.
hostname	(Optional) Specifies the DNS cache host name.
hostname	DNS cache host name.
http	(Optional) Clears the HTTP objects cache.
url	(Optional) Specifies the HTTP URL.
url	HTTP URL.
http-authentication	(Optional) Clears the authentication cache.
real-proxy	(Optional) Clears RealProxy cache content.
wmt	(Optional) Clears the WMT cache.
cdp	Resets CDP statistical data.
counters	Clears CDP counters.
table	Clears CDP tables.
logging	Clears syslog messages saved in disk file.
statistics	Clears statistics as specified.
all	Clears all statistics.
authentication	Clears authentication statistics.
boomerang	Clears boomerang statistics.

dns-cache	Clears DNS cache statistics.
ftp	Clears FTP caching statistics.
history	Clears the statistics history.
http	Clears HTTP statistics.
all	Clears all HTTP statistics.
cluster	Clears healing mode statistics.
errors	Clears HTTP errors statistics.
ims	Clears HTTP if-modified-since (IMS) statistics.
object	Clears HTTP object statistics.
outgoing	Clears HTTP outgoing proxy statistics.
proxy outgoing	Clears outgoing proxy monitor statistics.
requests	Clears HTTP requests statistics.
response	Clears HTTP response statistics.
savings	Clears HTTP savings statistics.
http-authcache	Clears authentication cache statistics.
https	Clears HTTPS statistics.
icp	Selects ICP statistics.
all	Clears all ICP statistics.
client	Clears ICP client statistics.
server	Clears ICP server statistics.
ip	Clears IP statistics.
ldap	Clears LDAP statistics.
mediacache	Clears mediacache statistics.
real	Clears RealProxy media cache statistics.
ntlm	Clears NTLM statistics.
pre-load	Clears preload statistics.
radius	Clears RADIUS statistics.
rule	Clears rule statistics.
action	Clears statistics of all the rules with same action.

action-type	Specifies one of the following actions: **block** **dscp** **freshness-factor** **no-auth** **no-cache** **no-proxy** **redirect** **refresh** **reset** **rewrite** **selective-cache** **use-proxy** **use-proxy-failover** **use-server** See the "rule" section on page 2-147 for explanations of actions and patterns.
all	Clears statistics of all the patterns for this action.
pattern	Clears statistics of rules with the specified pattern.
pattern-type	Specifies one of the following patterns: **domain** **dst-ip** **dst-port** **mime-type**[1] **header-field** **src-ip** **url-regex** **header-field** **url-regsub** **client** **server** See the "rule" section on page 2-147 for explanations of patterns and actions.
all	Clears statistics of all the rules.
running	Clears the running statistics.
tacacs	Clears TACACS+ statistics.
tcp	Clears TCP statistics.
transaction-logs	Clears transaction log export statistics.
url-filter	Clears URL filter statistics.
N2H2	Clears N2H2 URL filter statistics.
websense	Clears Websense URL filter statistics.
wmt	Clears all Windows Media Technologies (WMT) statistics.
transaction-log	Archives working transaction log files.

1. mime-type is an option for the freshness-factor, no-cache, and selective-cache actions only.

Defaults No default behavior or values

Command Modes

EXEC

Usage Guidelines

The **clear cache** command removes all cached contents from the currently mounted cfs volumes. Objects being read or written are removed when they cease being "busy." The equivalent to this command is the **cache clear** or **cfs clear** command.

Caution This command is irreversible, and all cached content will be erased.

The **clear cache force** command deletes all objects, whether busy or not, and may generate broken GIF or HTML messages for objects that were being read from the disk when the command was executed. If an object is being written to the Content Engine disk when a **clear cache force** command is executed, the application stops caching that object but still delivers the object from the web server to the client.

The **clear logging** command removes all current entries from the syslog.txt file, but does not make an archive of the file. It does put a syslog in syslog.txt to indicate that the syslog has been cleared as shown in the following example:

```
Feb 14 12:17:18 ContentEngine# exec_clear_logging:Syslog cleared
```

The **clear statistics** command clears all statistical counters from the parameters given. Use this command to monitor fresh statistical data for some or all features without losing cached objects or configurations.

The **clear transaction-log** command causes the transaction log to be archived immediately to the Content Engine hard disk. This command has the same effect as the **transaction-log force archive** command.

Examples

To purge all the entries in the bypass list, use the **clear bypass list** option.

```
ContentEngine# clear bypass list
```

To force the working transaction log file to be archived, use the **clear transaction-log** option.

```
ContentEngine# clear transaction-log
```

In the following example, the **clear statistics http cluster** command resets the healing mode statistics.

```
Console(config)# clear statistics http cluster
```

Related Commands

cache clear

cfs clear

show statistics

show interface

show wccp

clock

To set, clear, or save the battery-backed clock functions, use the **clock** EXEC command.

clock {**read-calendar** | **set** *time day month year* | **update-calendar**}

no clock {**read-calendar** | **set** *time day month year* | **update-calendar**}

Syntax Description

read-calendar	Reads the calendar and update system clock.
set	Sets the time and date.
time	Current time in hh:mm:ss format (hh: 00–23; mm: 00–59; ss: 00–59).
day	Day of the month (1–31).
month	Month of the year (April, August, December, February, January, July, June, March, May, November, October, September).
year	Year (1993–2035).
update-calendar	Updates the calendar with the system clock.

Defaults

No default behavior or values

Command Modes

EXEC

Usage Guidelines

If you have an outside source on your network that provides time services (such as a Network Time Protocol [NTP] server), you do not need to set the system clock manually. When setting the clock, enter the local time. The Content Engine calculates Coordinated Universal Time (UTC) based on the time zone set by the **clock timezone** global configuration command.

Two clocks exist in the system: the software clock and the hardware clock. The software uses the software clock. The hardware clock is used only at bootup to initialize the software clock.

The **set** keyword sets the software clock.

When the E-CDN application is enabled, all commands that can change the local time are disabled: **clock read-calendar**, **clock set**, **ntpdate**, and **ntp**.

Examples

```
ContentEngine# clock set 13:32:00 01 February 2000
```

Related Commands

clock timezone

show clock detail

clock

To set the summer daylight savings time and time zone for display purposes, use the **clock** global configuration command. To disable this function, use the **no** form of this command.

clock {**summertime** *timezone* {**date** *startday startmonth startyear starthour endday endmonth endyear offset* | **recurring** {**1-4** *startweekday startmonth starthour endweekday endmonth endhour offset* | **first** *startweekday startmonth starthour endweekday endmonh endhour offset* | **last** *startweekday startmonth starthour endweekday endmonh endhour offset*}} | **timezone** {*timezone hoursoffset minutesoffset*}}

no clock {**summertime** *timezone* {**date** *startday startmonth startyear starthour endday endmonth endyear offset* | **recurring** {**1-4** *startweekday startmonth starthour endweekday endmonh endhour offset* | **first** *startweekday startmonth starthour endweekday endmonh endhour offset* | **last** *startweekday startmonth starthour endweekday endmonh endhour offset*}} | **timezone** {*timezone hoursoffset minutesoffset*}}

Syntax Description

summertime	Configures summer or daylight saving time.
timezone	Name of summer time zone.
date	Configures absolute summer time.
startday	Date (1–31) to start.
startmonth	Month (January through December) to start.
startyear	Year (1993–2032) to start.
starthour	Hour (0–23) to start in (hh:mm) format.
endday	Date (1–31) to end.
endmonth	Month (January through December) to end.
endyear	Year (1993–2032) to end.
endhour	Hour (0–23) to end in (hh:mm) format.
offset	Minutes offset (see Table 2-1) from UTC (0–59).
recurring	Configures recurring summer time.
1-4	Configures starting week number 1–4.
first	Configures to recur beginning the first week of the month.
last	Configures to recur beginning the last week of the month.
startweekday	Weekday (Monday–Friday) to start.
startmonth	Month (January–December) to start.
starthour	Hour (0–13) to start in (hh:mm) format.
endweekday	Weekday (Monday–Friday) to end.
endmonth	Month (January–December) to end.
endhour	Hour (0–13) to end in hour:minute (hh:mm) format.
offset	Minutes offset (see Table 2-1) from UTC (0–59).
timezone	Configures standard time zone.
timezone	Name of time zone.
hoursoffset	Hours offset (see Table 2-1) from Coordinated Universal Time (–23, +23).
minutesoffset	Minutes offset (see Table 2-1) from UTC (0–59).

Defaults No default behavior or values

Command Modes Global configuration

Usage Guidelines To set and display the local and UTC current time of day without an NTP server, use the **clock timezone** command with the **clock set** command. The **clock timezone** parameter specifies the difference between UTC and local time, which is set with the **clock set** EXEC command. The UTC and local time are displayed with the **show clock detail** EXEC command.

Use the **clock** *timezone offset* command to specify a time zone, where *timezone* is the desired time zone entry from Table 2-1 and *0 0* is the offset (ahead or behind) Coordinated Universal Time (UTC) in hours and minutes. UTC was formerly known as Greenwich mean time (GMT).

```
ce(config)# clock timezone timezone 0 0
```

Note The time zone entry is case-sensitive and must be specified in the exact notation listed in the following time zone table. When using a time zone entry from the following time zone table, the system is automatically adjusted for daylight saving time.

Table 2-1 Offset from UTC

Time Zone	Offset from UTC
Africa/Algiers	+1
Africa/Cairo	+2
Africa/Casablanca	0
Africa/Harare	+2
Africa/Johannesburg	+2
Africa/Nairobi	+3
America/Buenos_Aires	–3
America/Caracas	–4
America/Mexico_City	–6
America/Lima	–5
America/Santiago	–4
Atlantic/Azores	–1
Atlantic/Cape_Verde	–1
Asia/Almaty	+6
Asia/Baghdad	+3
Asia/Baku	+4
Asia/Bangkok	+7
Asia/Colombo	+6
Asia/Dacca	+6
Asia/Hong_Kong	+8

Table 2-1 Offset from UTC (continued)

Time Zone	Offset from UTC
Asia/Irkutsk	+8
Asia/Jerusalem	+2
Asia/Kabul	+4.30
Asia/Karachi	+5
Asia/Katmandu	+5.45
Asia/Krasnoyarsk	+7
Asia/Magadan	+11
Asia/Muscat	+4
Asia/New Delhi	+5.30
Asia/Rangoon	+6.30
Aisa/Riyadh	+3
Asia/Seoul	+9
Asia/Singapore	+8
Asia/Taipei	+8
Asia/Tehran	+3.30
Asia/Vladivostok	+10
Asia/Yekaterinburg	+5
Asia/Yakutsk	+9
Australia/Adelaide	+9.30
Australia/Brisbane	+10
Australia/Darwin	+9.30
Australia/Hobart	+10
Australia/Perth	+8
Australia/Sydney	+10
Canada/Atlantic	–4
Canada/Newfoundland	–3.30
Canada/Saskatchewan	–6
Europe/Athens	+2
Europe/Berlin	+1
Europe/Bucharest	+2
Europe/Helsinki	+2
Europe/London	0
Europe/Moscow	+3
Europe/Paris	+1
Europe/Prague	+1
Europe/Warsaw	+1

Table 2-1 Offset from UTC (continued)

Time Zone	Offset from UTC
Japan	+9
Pacific/Auckland	+12
Pacific/Fiji	+12
Pacific/Guam	+10
Pacific/Kwajalein	–12
Pacific/Samoa	–11
US/Alaska	–9
US/Central	–6
US/Eastern	–5
US/East-Indiana	–5
US/Hawaii	–10
US/Mountain	–7
US/Pacific	–8

Examples

The following example specifies the local time zone as Pacific Standard Time and offsets 8 hours behind UTC.

```
ContentEngine(config)# clock timezone PST -8

ContentEngine(config)# no clock timezone

ContentEngine(config)# clock summertime PDT date 10 October 2001 23:59 29 April 2002 23:59
60
```

Related Commands

clock

show clock detail

configure

To enter global configuration mode, use the **configure** EXEC command. You must be in global configuration mode to enter global configuration commands.

configure

To exit global configuration mode, use the **end**, **Ctrl-Z**, or **exit** commands.

Syntax Description

This command has no arguments or keywords.

Defaults

No default behavior or values

Command Modes

EXEC

Usage Guidelines

Use this command to enter global configuration mode.

Examples

```
ContentEngine# configure
Enter configuration commands, one per line. End with CNTL/Z.
ContentEngine(config)#
```

Related Commands

show running-config

show startup-config

end

exit

Ctrl-Z

copy

To copy configuration or image data from a source to a destination, use the **copy** EXEC command.

copy compactflash install *filename*

copy disk ftp {*hostname* | *ip-address*} *remotefiledir remotefilename localfilename*

copy disk startup-config *filename*

copy ftp disk {*hostname* | *ip-address*} *remotefiledir remotefilename localfilename*

copy ftp install {*hostname* | *ip-address*} *remotefiledir remotefilename*

copy running-config disk *filename*

copy running-config startup-config

copy running-config tftp {*hostname* | *ip-address*} *remotefilename*

copy startup-config disk *filename*

copy startup-config running-config

copy startup-config tftp {*hostname* | *ip-address*} *remotefilename*

copy system-status disk *filename*

copy tech-support disk *filename*

copy tech-support tftp {*hostname* | *ip-address*} *remotefilename*

copy tftp disk {*hostname* | *ip-address*} *remotefilename localfilename*

copy tftp startup-config {*hostname* | *ip-address*} *remotefilename*

copy tftp running-config {*hostname* | *ip-address*} *remotefilename*

no copy {**compactflash install** *filename* | **disk ftp** {*hostname* | *ip-address*} *remotefiledir remotefilename localfilename* | **disk startup-config** *filename* | **ftp** {**disk** {*hostname* | *ip-address*} *remotefiledir remotefilename localfilename* | **install** {*hostname* | *ip-address*} *remotefiledir remotefilename*} | **running-config** {**disk** *filename* | **startup-config** | **tftp** {*hostname* | *ip-address*} *remotefilename*} | **startup-config** {**disk** *filename* | **running-config** | **tftp** {*hostname* | *ip-address*} *remotefilename*} | **system-status disk** *filename* | **tech-support** {**disk** *filename* | **tftp** {*hostname* | *ip-address*} *remotefilename*} | **tftp disk** {*hostname* | *ip-address*} *remotefilename localfilename* | **tftp startup-config** {*hostname* | *ip-address*} *remotefilename* | **tftp running-config** {*hostname* | *ip-address*} *remotefilename*}

Syntax Description

compactflash	Copies file from CompactFlash card.
install	Installs a software release file.
filename	Image filename.
disk ftp	Copies a local disk file to an FTP server.
hostname	Host name of FTP server.

ip-address	IP address of FTP server.
remotefiledir	Directory on the FTP server to which the local file is copied.
remotefilename	Name of local file when copied to the FTP server.
localfilename	Name of the local file to be copied.
disk startup-config	Copies configuration file from disk to startup configuration (NVRAM).
filename	Name of existing configuration file.
ftp disk	Copies file from an FTP server to a local disk.
hostname	Host name of FTP server.
ip-address	IP address of FTP server.
remotefiledir	Directory on the FTP server where the file to be copied is located.
remotefilename	Name of the file to be copied to the local disk.
localfilename	Name of the copied file as it appears on the local disk.
ftp install	Copies the file from an FTP server and installs the file to the local device.
hostname	Name of the FTP server.
ip-address	IP address of the FTP server.
remotefiledir	Remote file directory.
remotefilename	Remote filename.
running-config disk	Copies current system configuration to disk.
filename	Name of file to be created on disk.
running-config startup-config	Copies running configuration to startup configuration (NVRAM).
running-config tftp	Copies running configuration to a file on a TFTP server.
hostname	Host name of TFTP server.
ip-address	IP address of TFTP server.
remotefilename	Remote filename of configuration file to be created on TFTP server. Use the complete path name.
startup-config disk	Copies startup configuration to a disk file.
filename	Name of startup configuration file to be copied to the local disk.
startup-config running-config	Copies startup configuration to running configuration.
startup-config tftp	Copies startup configuration to a file on a TFTP server.
hostname	Host name of TFTP server.
ip-address	IP address of TFTP server.
remotefilename	Remote filename of startup configuration file to be created on TFTP server. Use the complete path name.
system-status disk	Copies system status to disk.
filename	Name of file to be created on disk.
tech-support disk	Copies system information for technical support to disk.
filename	Name of file to be created on disk.
tech-support tftp	Copies system information to a TFTP server.
hostname	Host name of TFTP server.
ip-address	IP address of TFTP server.

remotefilename	Remote filename of system information file to be created on TFTP server. Use the complete path name.
tftp disk	Copies image from a TFTP server to disk.
hostname	Host name of TFTP server.
ip-address	IP address of TFTP server.
remotefilename	Name of the remote image file to be copied from the TFTP server. Use the complete path name.
localfilename	Name of the image file to be created on the local disk.
tftp startup-config	Copies image from a TFTP server to startup configuration.
hostname	Host name of TFTP server.
ip-address	IP address of TFTP server.
remotefilename	Name of the remote image file to be coped from the TFTP server. Use the complete path name.
tftp running-config	Copies image from a TFTP server to running configuration.
hostname	Host name of TFTP server.
ip-address	IP address of TFTP server.
remotefilename	Name of the remote image file to be coped from the TFTP server. Use the complete path name.

Defaults

No default behavior or values

Command Modes

EXEC

Usage Guidelines

The **copy disk ftp** command copies files from a sysfs partition to an FTP server. The **copy disk startup-config** command copies a startup configuration file to Flash memory.

The **copy ftp disk** command copies a file from an FTP server to a sysfs partition.

Use the **copy ftp install** command to install an image file. Part of the image goes to disk and part goes to Flash memory.

Use the **copy running-config** command copies the running system configuration to a sysfs partition, Flash memory, or to a TFTP server. The **copy running-config startup-config** command is equivalent to the **write memory** command.

The **copy startup-config** command copies the startup configuration file to a TFTP server or to a sysfs partition.

The **copy system-status** command creates a file on a sysfs partition containing hardware and software status information.

The **copy tech-support tftp** command can copy technical support information to a TFTP server or to a a sysfs partition.

The **copy tftp disk** command copies a file from a TFTP server to disk.

Examples

The following example copies an image file from an FTP server and installs the file to the local device.

```
ce-590# copy ftp install 1.1.1.1 //users2/ACNS400BR/boot ce590-ACNS-400.bin
Enter username for remote ftp server:biff
Enter password for remote ftp server:
Initiating FTP download...
printing one # per 1MB downloaded
Sending:USER biff
1.1.1.1 FTP server (Version) Mon Feb 28 10:30:36 EST
2000) ready.
Password required for biff.
Sending:PASS *****
User biff logged in.
Sending:TYPE I
Type set to I.
Sending:PASV
Entering Passive Mode (128,107,193,244,55,156)
Sending:CWD //users2/ACNS400BR/boot
CWD command successful.
Sending PASV
Entering Passive Mode (128,107,193,244,55,156)
Sending:RETR ce590-ACNS-400.bin
Opening BINARY mode data connection for ruby.bin (87376881 bytes).
################################################################################
writing flash component:
.............................................................
The new software will run after you reload.
ce-590#
```

Related Commands

install

reload

show running-config

show startup-config

write

cpfile

To make a copy of a file, use the **cpfile** EXEC command.

cpfile *sourcefile destinationfile*

Syntax Description

sourcefile	Name of the file to copy.
destinationfile	Name of the copy to be created.

Defaults

No default behavior or values

Command Modes

EXEC

Usage Guidelines

Use this command to create a copy of a file. Only sysfs files can be copied.

Examples

```
ContentEngine# cpfile ce500-194616.bin cd500-194618.bin
```

Related Commands

copy
dir
lls
ls
mkfile
rmdir
rmname

debug

Note We recommend that the **debug** command be used only at the direction of Cisco Systems technical support personnel. Cache performance is impacted when you run the **debug** command.

To monitor and record cache software functions, use the **debug** EXEC command. Use the **no** form of the command to disable **debug**.

debug {**all** option | **authentication** {**http-request** | **user**}}

no debug authentication user

Command Modes

EXEC

Syntax Description

all	Enables all debugging.
http-request	Debugs HTTP request authentication.
user	Debugs user login against system authentication.
authmod	Debugs authentication module.
boomerang	Debugs boomerang.
all	Debugs all boomerang functions.
channel	Debugs boomerang channel.
cli	Debugs boomerang command-line interface (CLI).
events	Debugs boomerang events.
memory	Debugs boomerang memory allocation.
buf	Debugs buffer manager.
all	Debugs all buffer manager functions.
dmbuf	Debugs buffer manager dmbuf.
dmsg	Debugs buffer manager dmsg.
cdp	Debugs CDP.
adjacency	Debugs CDP neighbor.
events	Debugs CDP events.
ip	Debugs CDP IP.
packets	Debugs packet-related CDP.
cli	Debugs CLI.
all	Debugs all CLI.
bin	Debugs CLI binary program.
parser	Debugs CLI parser.
dataserver	Debugs data server.
all	Debuts all data server functions.
clientlib	Debugs data server clientlib module.
server	Debugs data server module.

exit	Exits from this submode.
ftp	Debugs FTP.
all	Debugs all FTP functions.
cache	Debugs FTP cache.
client	Debugs FTP client.
server	Debugs FTP server.
http	Debugs HTTP commands.
all	Debugs all HTTP functions.
cache	Debugs HTTP cache.
header	Debugs HTTP header.
hit	Debugs HTTP hit.
miss	Debugs HTTP miss.
parser	Debugs HTTP parser.
proxy	Debugs HTTP proxy.
server	Debugs HTTP server.
http-authcache	Debugs the authentication cache.
all	Debugs all the authentication cache functions.
application	Debugs application module.
cli	Debugs CLI module.
daemon	Debugs daemon client module.
https	Debugs HTTPS.
all	Debugs all HTTPS functions.
cli	Debugs HTTPS CLI.
header	Debugs HTTPS header.
parser	Debugs HTTPS parser.
proxy	Debugs HTTPS proxy.
icp	Debugs ICP.
all	Debugs all ICP functions.
client	Debugs ICP client.
ex	Debugs ICP ex.
heal	Debugs ICP healing.
main	Debugs ICP main.
parse	Debugs ICP parse.
print	Debugs ICP print.
server	Debugs ICP server.
utils	Debugs ICP utilities.
logging	Debugs logging.
all	Debugs all logging functions.
ntp	Debugs NTP.
pre-load	Debugs preload.
all	(Optional) Debugs all preload functions.

rtsp	Debugs RTSP.
all	Debugs all RTSP functions.
dnscache	Debugs RTSP proxy internal DNS cache.
manager	Debugs RTSP manager.
protocol	Debugs RTSP proxy protocol.
proxy	Debugs RTSP proxy.
real-all	Debugs all RealProxy plug-ins.
real-allowance	Debugs RealProxy allowance plug-in.
real-cache	Debugs RealProxy cache plug-in debug.
real-stats	Debugs RealProxy statistics plug-in.
wccp-general	Debugs RTSP proxy general WCCP.
wccp-liveness	Checks if RTSP proxy WCCP is alive.
rule	Debugs Rules Template.
all	Debugs all rule functions.
ip	Debugs rule IP.
main	Debugs rule main.
port	Debugs rule port.
regex	Debugs rule regex.
regsub	Debugs rule regsub.
snmp	Debugs SNMP.
all	Debugs all SNMP functions.
cli	Debugs SNMP CLI.
main	Debugs SNMP main.
mib	Debugs SNMP MIB.
traps	Debugs SNMP traps.
standby	Debugs standby.
all	(Optional) Debugs all standby functions.
stats	Debugs statistics.
all	Debugs all statistics functions.
collection	Debugs statistics collection.
computation	Debugs statistics computation.
history	Debugs statistics history.
translog	Debugs transaction logging.
archive	Debugs transaction log archive.
export	Debugs transaction log FTP export.
url-filter	Debugs URL filter.
local-list	Debugs URL filter local bad or good list.
n2h2	Debugs URL filter N2H2.
websense	Debugs URL filter Websense.
wccp	Debugs WCCP information.
all	Debugs all WCCP functions.

detail	Debugs WCCP detail.
error	Debugs WCCP error.
events	Debugs WCCP events.
keepalive	Debugs WCCP keepalive to applications.
packets	Debugs WCCP packet-related information.
slowstart	Debugs WCCP slow start.
wi	Debugs web interface.
wmt	Debugs WMT component.
error	Debugs WMT level 1 functionality.
client-ip	Debugs request from a specific client.
ip-address	Debugs IP address of requesting client.
server-ip	Debugs request to a specific server.
ip-address	Debugs IP address of specific server.
trace	Debugs WMT level 2 functionality.
client-ip	Debugs request from a specific client.
ip-address	Debugs IP address of requesting client.
server-ip	Debugs request to a specific server.
ip-address	Debugs IP address of specific server.
authentication	Authentication debug commands.
http-request	Debugs HTTP request authentication.
user	Debugs user's login to the system authentication.

Usage Guidelines

We recommend that the **debug** command be used only at the direction of Cisco Systems technical support personnel. Cache performance is impacted when you run **debug**. Use the **show debugging** command to display enabled **debug** options.

Related Commands

show debugging

undebug

delfile

To delete a file, use the **delfile** EXEC command.

delfile *filename*

Syntax Description

filename	Name of the file to delete.

Defaults

No default behavior or values

Command Modes

EXEC

Usage Guidelines

Use this command to remove a file from a sysfs partition.

Examples

```
ContentEngine# delfile /local1/tempfile
```

Related Commands

cpfile
deltree
mkdir
mkfile
rmdir

deltree

To remove a directory with its subdirectories and files, use the **deltree** EXEC command.

deltree *directory*

Syntax Description		
	directory	Name of the directory tree to delete.

Defaults

No default behavior or values

Command Modes

EXEC

Usage Guidelines

Use this command to remove a directory and all files within the directory from the Content Engine sysfs file system. Do not remove files or directories required for proper Content Engine functioning.

Examples

```
ContentEngine# deltree /local1/testdir
```

Related Commands

delfile

mkdir

mkfile

rmdir

dir

To view a long list of files in a directory, use the **dir** EXEC command.

dir [*directory*]

Syntax Description

directory	(Optional) Name of the directory to list.

Defaults

No default behavior or values

Command Modes

EXEC

Usage Guidelines

Use this command to view a detailed list of files contained within the working directory, including names, sizes, and time created. The equivalent command is **lls**.

Examples

```
ContentEngine# dir
size            time of last change                name
--------------  -------------------------          -----------
       3931934  Tue Sep 19 10:41:32 2000           errlog-cache-20000918-164015
           431  Mon Sep 18 16:57:40 2000           ii.cfg
           431  Mon Sep 18 17:27:46 2000           ii4.cfg
           431  Mon Sep 18 16:54:50 2000           iii.cfg
          1453  Tue Sep 19 10:34:03 2000           syslog.txt
          1024  Tue Sep 19 10:41:31 2000  <DIR>    testdir
```

Related Commands

ls

lls

disable

To turn off privileged EXEC commands, use the **disable** EXEC command.

disable

Syntax Description

This command has no arguments or keywords.

Defaults

No default behavior or values

Command Modes

EXEC

Usage Guidelines

The **disable** command places you in the user-level EXEC shell. To turn privileged EXEC mode back on, use the **enable** command.

Examples

```
ContentEngine# disable
```

Related Commands

enable

disk

To configure the disks for devices that are using ACNS software, use the **disk** EXEC command.

disk add *diskname* {**cfs** | **ecdnfs** | **mediafs** | **sysfs**} {**remaining** | *partitionsize*} [{**cfs** | **ecdnfs** | **mediafs** | **sysfs**}{**remaining** | *partitionsize*}]

disk cancel-config

disk config sysfs {**remaining** | *partitionsize*} [{**cfs** | **ecdnfs** | **mediafs**}{**remaining** | *partitionsize*}]

disk raid-array add-array

disk raid-array repair *diskname*

Syntax Description

add	Adds a single disk.
diskname	Name of the disk to be added.
cfs	Allocates the disk space of the added disk to cache file system functions.
ecdnfs	Allocates the disk space of the added disk to Enterprise CDN file system functions.
mediafs	Allocates the disk space of the added disk to media file system functions.
sysfs	Allocates the disk space of the added disk to system file system functions.
remaining	Specifies that the remaining space be allocated to the file function.
partitionsize	Size of the allocation. (Size can be designated in megabytes, gigabytes, or as a percentage of the system total storage.)
cancel-config	Cancels the disk configuration.
config	Configures disk space among functions.
sysfs	Allocates disk space for system file system functions.
remaining	Specifies that the remaining space be allocated to the file function.
partitionsize	Size of the allocation. (Size can be designated in megabytes, gigabytes, or as a percentage of the system total storage.)
cfs	(Optional) Allocates disk space for cache file system functions.
ecdnfs	(Optional) Allocates disk space for E-CDN file system functions.
mediafs	(Optional) Allocates disk space for media file system functions.
remaining	(Optional) Specifies that the remaining space be allocated to the function.
partitionsize	Size of the allocation. (Size can be designated in megabytes, gigabytes, or as a percentage of the system total storage.)
raid-array	Manages Storage Array disk configuration for the CDM-4650.
add-array	Creates a logical disk for the Storage Array that is recognized by the CDM-4650 RAID controller.
repair	Rebuilds a RAID disk array after a single disk in the array fails.
diskname	Name of the disk to be repaired.

Defaults

No default behavior or values

Command Modes EXEC

Usage Guidelines Use the **disk config** command to configure disk allocations.

Note If you are using a CE-507, the maximum allocation for the cfs with the E-CDN application enabled is 6 GB because of memory restrictions in the CE-507.

For example, adjust the disk storage allocations as follows:

```
ContentEngine# disk config sysfs 2GB cfs 6GB mediafs 2GB ecdnfs remaining
```

Use the **disk cancel-config** command to cancel the configuration.

Use the **disk add** command to add a single disk with specified partitions.

Use the **disk raid-array add-array** command to create a logical disk for the Storage Array that is recognized by the CDM-4650 RAID controller.

Use the **disk raid-array repair** command to rebuild a RAID disk array after a single disk in the array fails.

Examples In the following example of the **disk config sysfs** command, 10 percent of the total storage is allocated to the sysfs and 30 percent to every other file system.

```
ContentEngine# disk config sysfs 10% mediafs 30% ecdnfs 30% cfs 30%
Disk configured successfully.
New configuration will take effect after reload.
Please remove this device from the ECDN CDM (if any) before reboot this device, as this
device's configuration will be stale due to disk repartition.
```

Related Commands **show disks**

show cfs

show ecdnfs

show mediafs

show statisitics

dns-cache

To configure the DNS cache, use the **dns-cache** global configuration command. To disable the DNS cache, use the **no** form of this command.

dns-cache size *maxnumber*

no dns-cache size

Syntax Description

size	Sets the DNS cache size.
maxnumber	Maximum number of cache records (4096–65536).

Defaults

No default behavior or values

Command Modes

Global configuration

Usage Guidelines

Cache size refers to the maximum number of DNS cache entries. Domain name resolution requires that at least one DNS name server be configured with the **ip name-server** command**.** The DNS cache goes online when the **ip name-server** command is configured, and goes offline when the last IP name server configuration is deleted with the **no ip name-server** *ip-address* command.

Examples

In the following example, the DNS cache size is set to 20,000 records.

```
ContentEngine(config)# dns-cache size 20000
```

In the following example, the DNS cache is disabled with the **no** form of the **dns-cache** command.

```
ContentEngine(config)# no dns-cache size
```

Related Commands

ip name-server

clear dns-cache

dnslookup

show statistics dns-cache

dnslookup

To resolve a host or domain name to an IP address, use the **dnslookup** EXEC command.

dnslookup {*hostname* | *domainname*}

Syntax Description

hostname	Name of host on network.
domainname	Domain name.

Defaults

No default behavior or values

Command Modes

EXEC

Examples

In the following three examples, the **dnslookup** command is used to resolve the host name **myhost** to IP address 172.31.69.11, **cisco.com** to IP address 192.168.219.25, and the host name IP address.

```
ContentEngine# dnslookup myhost
official hostname: myhost.cisco.com
          address: 172.31.69.11

ContentEngine#dnslookup cisco.com
official hostname: cisco.com
         address: 192.168.219.25

ContentEngine#dnslookup  10.0.11.0
official hostname: 10.0.11.0
          address: 10.0.11.0
```

ecdn

To force a downgrade of the Enterprise CDN (E-CDN) software, use the **ecdn** EXEC command.

ecdn force-downgrade [**disable**]

Syntax Description

force-downgrade	Forces downgrade without content preservation.
disable	(Optional) Disables the forced downgrade.

Defaults

No default behavior or values

Command Modes

EXEC

Examples

In the following two examples, the first demonstrates a forced downgrade of the Enterprise CDN software without preserving content, and the second demonstrates the optional **disable** downgrade command.

```
Console# ecdn force-downgrade

Console# ecdn force-downgrade disable
```

ecdn

To associate the Content Engine or Content Router with the IP address and (optionally) the port number of the Content Distribution Manager and enable the Enterprise CDN (E-CDN) application, use the **ecdn** command in global configuration mode. To clear these parameters, use the **no** form of this command.

ecdn {**cdm ip** *ip_address* [**port** *port_num*] | **enable**}

no ecdn {**cdm ip** *ip_address* [**port** *port_num*] | **enable**}

Syntax Description

cdm ip	Associates the Content Engine or Content Router with the IP address of the Content Distribution Manager.
ip_address	IP address of the Content Distribution Manager that the Content Engine or Content Router is associated with.
port	(Optional) Sets the port number associated with the Content Distribution Manager.
port_num	Port number of the Content Distribution Manager that the Content Engine or Content Router is associated with (1–65535). The default is 80. If no port number is specified, the default is used.
enable	Enables the E-CDN application.

Defaults

If you do not specify a port number, the default is port 80. The E-CDN application is not enabled by default.

Command Modes

Global configuration

Examples

This example associates the Content Engine with the Content Distribution Manager IP address 172.17.76.76 and port number 443.

```
ContentEngine(config)# ecdn cdm ip 172.17.76.76 port 443
```

This example cancels the association of the Content Engine with a Content Distribution Manager.

```
ContentEngine(config)# no ecdn cdm ip
```

This example enables the E-CDN application.

```
ContentEngine(config)# ecdn enable
Admin's passwd will be changed to what you have set on the CDM.
Please make sure that from now on until ecdn is turned off,
you make all "admin" passwd changes from the cdm GUI console.
```

This example disables the E-CDN application.

```
ContentEngine(config)# no ecdn enable
```

Related Commands

show ecdn

enable

To access privileged EXEC commands, use the **enable** EXEC command.

enable

Syntax Description

This command has no arguments or keywords.

Defaults

No default behavior or values

Command Modes

EXEC

Usage Guidelines

To access privileged EXEC mode from user EXEC mode, use the **enable** command. The **disable** command takes you from privileged EXEC mode to user EXEC mode.

Examples

```
ContentEngine> enable
ContentEngine#
```

Related Commands

disable

exit

end

To exit global configuration mode, use the **end** global configuration command.

end

Syntax Description

This command has no arguments or keywords.

Defaults

No default behavior or values

Command Modes

Global configuration

Usage Guidelines

Use the **end** command to exit global configuration mode after completing any changes to the running configuration. To save new configurations to NVRAM, use the **write** command.

The **Ctrl-Z** command also exits global configuration mode.

Examples

```
ContentEngine(config)# end
ContentEngine#
```

Related Commands

exit

Ctrl-Z

error-handling

Use the **error-handling** command to set error-handling options.

error-handling {**reset-connection** | **send-cache-error** | **transparent**}

no error-handling

Syntax Description

reset-connection	Resets the TCP connection without specifying any error.
send-cache-error	Sends cache error.
transparent	Makes the cache transparent to the client.

Defaults

The default is the **error-handling transparent** option.

Command Modes

Global configuration

Usage Guidelines

The **error-handling transparent** option is set by default, so that the Content Engine will not send errors to the client but will bypass the client connections to the server. Setting the **error-handling send-cache-error** command will send a Content Engine-generated error page to the client. Using the **reset-connection** option aborts the client connection.

If error handling is set to **transparent**, the Content Engine adds the client/server pair to the WCCP bypass list. The Content Engine will send a retry message to the client. The retried connection from the client is then bypassed by the Content Engine.

A transparent error bypass is triggered only if the following conditions exist:

- The Content Engine is configured to preserve transparency as opposed to preserving confinement and control.
- The transaction is transparently intercepted.
- The WCCP module (that is, WCCP Version 2 or later module) on the Content Engine is capable of performing a bypass.

For a client request, bypass occurs under the following conditions:

- If the request is malformed and fails to parse
- If the client is denied access
- If the client fails proxy authentication

For a server response, bypass occurs under the following conditions:

- If the response is not obtained explicitly through an outgoing proxy
- If the request is malformed and fails to parse
- If the request has a 501, 502, 503, 504, or 505 status code, which may indicate that an error exists on the server

The **error-handling transparent** command creates a bypass list entry for the client/server pair, and the **error-handling send-cache-error** command sends a Content Engine generated error page to the client.

With the transparent option enabled, end users can receive browser-generated messages rather than a Content Engine-generated HTML page for errors that the Content Engine encounters while processing a client request or response. Thus, the Content Engine remains transparent (invisible) to the end user.

Transparent error reporting is implemented as follows:

- Content Engine running WCCP Version 2

  To make the source of the error messages transparent to the user, the client/server pair is added to the bypass list and an HTTP redirect message is sent to the client, requesting the client to redirect the request to the same URL as before. The client, on receiving the redirect message, sends back the request once again. This time, the request is bypassed by the Content Engine because the client/server pair is on the bypass list. The request now goes to the server directly. Because the connection was not accepted by the Content Engine, any timeout error, failure to connect to the server, or mangled response from the server is handled by the browser. Currently all entries on the bypass list are kept for a configurable period of time (the default is 20 minutes).

  With the **reset connection** option, a reset is sent back to the client and the connection is closed if it encounters an error from the server. When a browser receives a connection reset, it displays a "Connection Reset By Peer" alert box.

- Content Engine running WCCP Version 1

  For all error conditions, the Content Engine sends back a reset and closes the connection. It does not send back any error pages. All errors seen by the clients are in the familiar browser error format.

- Content Engine acting as an incoming proxy server

  The Content Engine sends back HTML error pages. When clients are using the Content Engine as an incoming proxy server, they receive the HTML error pages generated by the Content Engine.

Examples

```
ContentEngine(config)# error-handling transparent
```

exception debug

Note We recommend that the **exception debug** and **exception coredump** commands be used only at the direction of Cisco Systems technical support personnel. Cache performance is impacted when you run the **exception debug** or **exception coredump** command.

To enable error handling or debug mode, use the **exception debug** global configuration command. To revert to the default value, use the **no** form of this command.

exception {**coredump** | **debug**}

no exception {**coredump** | **debug**}

Syntax Description

coredump	Causes proxy processes to do a core dump if the system crashes.
debug	Causes proxy processes to hang if the system crashes, until they are explicitly killed.

Defaults

The default is disabled.

Command Modes

Global configuration

Usage Guidelines

We recommend that the **exception debug** and **exception coredump** commands be used only at the direction of Cisco Systems technical support personnel. Cache performance is impacted when you run the **exception debug** or **exception coredump** command.

Examples

```
ContentEngine(config)# exception ?
debug  if enabled, proxy processes will hang there until someone kills it
ContentEngine(config)# exception disable
ContentEngine(config)# no exception disable
```

Related Commands

debug

exec-timeout

To configure the length of time that an inactive Telnet session remains open, use the **exec-timeout** global configuration command. To revert to the default value, use the **no** form of this command.

exec-timeout *timeout*

no exec-timeout

Syntax Description

timeout	Timeout in minutes (0–44,640).

Defaults

The default is 15 minutes.

Command Modes

Global configuration

Usage Guidelines

A Telnet session with the Content Engine can remain open and inactive for the interval of time specified by the **exec-timeout** command. When the exec-timeout interval elapses, the Content Engine automatically closes the Telnet session.

Examples

```
ContentEngine(config)# exec-timeout 100

ContentEngine(config)# no exec-timeout
```

exit

To access the EXEC command shell from the global, interface, and debug configuration command shells, use the **exit** command.

exit

Syntax Description

This command has no arguments or keywords.

Defaults

No default behavior or values

Command Modes

EXEC, global, and interface configuration

Usage Guidelines

Use the **exit** command in any configuration mode to return to EXEC mode. This is equivalent to the **Ctrl-Z** or the **end** command.

The **exit** command issued in the user level EXEC shell terminates the console or Telnet session.

Examples

```
ContentEngine(config)# exit
ContentEngine# exit
ContentEngine>
```

Related Commands

end

external-ip

To configure up to eight external Network Address Translation (NAT) IP addresses, use the **external-ip** command in global configuration mode.

external-ip *ip-address*

no external-ip *ip-address*

Syntax Description

ip-address	A maximum of eight external (NAT) IP addresses can be configured.

Defaults

No default behavior or values

Command Modes

Global configuration

Usage Guidelines

Use this command to configure up to eight Network Address Translation IP addresses to allow the router to translate up to eight internal addresses to registered unique addresses and translate external registered addresses to addresses that are unique to the private network.

Examples

```
ContentEngine(config)# external-ip 192.168.43.1 192.168.43.2 192.168.43.3 192.168.43.4
```

ftp

To configure FTP caching services on the Content Engine, use the **ftp** global configuration command. Use the **no** form of this command to selectively disable options.

ftp age-multiplier directory-listing *dl_time* **file** *fo_time*

ftp max-ttl days directory-listing *dlmax_days* **file** *fmax_days*

ftp max-ttl hours directory-listing *dlmax_hours* **file** *fmax_hours*

ftp max-ttl minutes directory-listing *dlmax_min* **file** *fmax_min*

ftp max-ttl seconds directory-listing *dlmax_sec* **file** *fmax_sec*

ftp min-ttl *min_minutes*

ftp object max-size *size*

ftp proxy active-mode enable

ftp proxy anonymous-pswd *passwd*

ftp proxy incoming *port*

ftp proxy outgoing host {*hostname* | *ip-address*} *port*

ftp reval-each-request {**all** | **directory-listing** | **none**}

no ftp {**age-multiplier directory-listing** *dl_time* **file** *fo_time* | **max-ttl** {**days directory-listing** *dlmax_days* **file** *fmax_days* | **hours directory-listing** *dlmax_hours* **file** *fmax_hours* | **minutes directory-listing** *dlmax_min* **file** *fmax_min* | **seconds directory-listing** *dlmax_sec* **file** *fmax_sec*} | **min-ttl** *min_minutes* | **object max-size** *size* | **proxy** {**active-mode enable** | **anonymous-pswd** *passwd* | **incoming** *port* | **outgoing host** {*hostname* | *ip-address*} *port* | **reval-each-request** {**all** | **directory-listing** | **none**}}

Syntax Description

age-multiplier	FTP caching heuristic modifiers.
directory-listing	Specifies heuristic modifier of directory listing objects.
dl_time	Expiration time of directory listing objects as a percentage of their age (0–100). The default is 30.
file	Specifies heuristic modifier of file objects.
fo_time	Expiration time of file objects as a percentage of their age (0–100). The default is 60.
max-ttl	Sets maximum Time To Live for objects in the cache.
days	Sets maximum Time To Live units in days.
directory-listing	Sets maximum Time To Live for directory listing objects in days.
dlmax_days	Maximum Time To Live in days for directory listing objects (1–1825). The default is 7 days.
file	Sets maximum Time To Live for file objects in days.
fmax_days	Maximum Time To Live in days (1–1825). The default is 3 days.

hours	Sets maximum Time To Live units in hours.
directory-listing	Sets maximum Time To Live for directory listing objects in hours.
dlmax_hours	Maximum Time To Live for directory listing objects in hours (1–43800). The default is 72 hours.
file	Sets maximum Time To Live for file objects in hours.
fmax_hours	Maximum Time To Live for file objects in hours (1–43800). The default is 168 hours.
minutes	Sets maximum Time To Live units in minutes.
directory-listing	Sets maximum Time To Live for directory listing objects in minutes.
dlmax_ min	Maximum Time To Live for directory listing objects in minutes (1–2628000). The default is 4320 minutes.
file	Sets maximum Time To Live for file objects in minutes.
fmax_min	Maximum Time To Live for file objects in minutes (1–2628000). The default is 10080 minutes.
seconds	Sets maximum Time To Live units in seconds.
directory-listing	Sets maximum Time To Live for directory listing objects in seconds.
dlmax_ sec	Maximum Time To Live for directory listing objects in seconds (1–157680000). The default is 259200 seconds.
file	Sets maximum Time To Live for file objects in seconds.
fmax_sec	Maximum Time To Live for file objects in seconds (1–157680000). The default is 604800 seconds.
min-ttl	Sets minimum Time To Live for FTP objects in cache.
min_minutes	Minimum Time To Live in minutes for FTP objects in cache (0–86400).
object	Sets configuration of FTP objects.
max-size	Sets maximum size of a cacheable object.
size	Maximum size of a cacheable object in kilobytes (KB) (1–1048576).
proxy	Sets proxy configuration parameters.
ftp proxy active-mode	Configures FTP active mode to fetch files.
enable	Enables FTP active mode.
anonymous-pswd	Sets anonymous password string (for example, wwwuser@cisco.com).
passwd	Anonymous password. The default is anonymous@hostname.
incoming	Sets the incoming port for proxy-mode requests.
port	Up to eight ports to listen for requests (1–65535).
outgoing	Sets parameters to direct outgoing FTP requests to another proxy server.
host	Sets outgoing FTP proxy host parameters.
hostname	Host name of the outgoing FTP proxy.
ip-address	IP address of the outgoing FTP proxy.
port	Port of the outgoing FTP proxy (1–65535).
reval-each-request	Sets scope of revalidation for every request.
all	Revalidates all objects on every request.
directory-listing	Revalidates directory listing objects on every request.
none	Does not revalidate for each request.

Defaults

dl_time: 30 percent

fo_time: 60 percent

dlmax_days: 7 days

fmax_days: 3 days

dlmax_hours: 72 hours

fmax_hours: 168 hours

dlmax_min: 4320 minutes

fmax_min: 10080 minutes

dlmax_sec: 259200 seconds

fmax_sec: 604800 seconds

min_minutes: 86400 minutes

directory-listing *age_percent*: 50 percent

Maximum size of cacheable object: unlimited

Command Modes

Global configuration

Usage Guidelines

The Content Engine accepts FTP requests when URLs specify the FTP protocol (for example, GET ftp://ftp.cs.wisc.edu/pub/cao/READM). For these requests, the client uses HTTP as the transport protocol with the Content Engine, whereas the Content Engine uses FTP with the FTP server.

The FTP proxy supports passive and active mode for fetching files and directories. Passive mode is the default. The Content Engine automatically changes to active mode if passive mode is not supported by the FTP server. If **active-mode enable** is configured, FTP first attempts to fetch the file in active mode. If active mode fails, it attempts to fetch it again in passive mode.

The Content Engine caches both the FTP file objects and directory listings in the cfs. The Content Engine transforms the regular directory listings from the FTP server into HTML, with links that the client users can point to and click to download files.

When the Content Engine receives an FTP request from the web client, it first looks in its cache. If the object is not in its cache, it fetches the object from an upstream FTP proxy server (if one is configured), or directly from the origin FTP server.

The FTP proxy supports anonymous as well as authenticated FTP requests. Only base64 encoding is supported for authentication. The FTP proxy accepts all FTP URL schemes defined in RFC 1738. In the case of a URL in the form ftp://user@site/dir/file, the proxy sends back an authentication failure reply and the browser supplies a popup window for the user to enter login information.

The FTP proxy supports commonly used MIME types, attaches the corresponding header to the client, chooses the appropriate transfer type (binary or ASCII), and enables the browser to open the FTP file with the configured application. For unknown file types, the proxy uses binary transfer as the default and instructs the browser to save the download file instead of opening it. The FTP proxy returns a formatted directory listing to the client if the FTP server replies with a known format directory listing. The formatted directory listing has full information about the file or directory and provides the ability for users to choose the download transfer type.

The Content Engine caches FTP traffic only when the client uses the Content Engine as a proxy server for FTP requests. All FTP traffic that was sent directly from the web client to an FTP server, if transparently intercepted by the Content Engine, is treated as non-HTTP traffic.

The FTP proxy supports up to eight incoming ports. It can share the ports with transparent-mode services and also with the other proxy-mode protocols supported by the Content Engine, such as HTTP and HTTPS. In proxy mode, the Content Engine accepts and services the FTP requests only on the ports configured for FTP proxy. All the FTP requests on other proxy mode ports are rejected in accordance with the error-handling settings on the Content Engine.

The Content Engine can apply the Rules Template to FTP requests based on server name, domain name, server IP address and port, client IP address, and URL.

The Content Engine logs FTP transactions in the transaction log, in accordance with the Squid syntax. When URL tracking is enabled, the Content Engine logs FTP transaction information to the syslog. The syslog entries are prefixed with <ftp>.

Examples

This example configures an incoming FTP proxy on ports 8080, 8081, and 9090. Up to eight incoming proxy ports can be configured on the same command line.

```
ContentEngine(config)# ftp proxy incoming 8080 8081 9090
```

This example removes one FTP proxy port from the list entered in the previous example. Ports 8080 and 9090 remain FTP proxy ports.

```
ContentEngine(config)# no ftp proxy incoming 8081
```

This example disables all the FTP proxy ports.

```
ContentEngine(config)# no ftp proxy incoming
```

This example configures an upstream FTP proxy with the IP address 172.16.76.76 on port 8888.

```
ContentEngine(config)# ftp proxy outgoing host 172.16.76.76 8888
```

This example specifies an anonymous password string for the Content Engine to use when contacting FTP servers. The default password string is anonymous@hostname.

```
ContentEngine(config)# ftp proxy anonymous-pswd newstring@hostname
```

This example configures the maximum size in kilobytes of an FTP object that the Content Engine will cache. By default, the maximum size of a cacheable object is not limited.

```
ContentEngine(config)# ftp object max-size 15000
```

This example forces the Content Engine to revalidate all objects for every FTP request.

```
ContentEngine(config)# ftp reval-each-request all
```

This example configures a maximum Time To Live of 3 days in cache for directory listing objects and file objects.

```
ContentEngine(config)# ftp max-ttl days directory-listing 3 file 3
```

Related Commands

show ftp

fullduplex

To configure an interface for full-duplex operation, use the **fullduplex** interface configuration command. To disable this function, use the **no** form of this command.

fullduplex

no fullduplex

Syntax Description

This command has no arguments or keywords.

Defaults

No default behavior or values

Command Modes

Interface configuration

Usage Guidelines

Use this command to configure an interface for full-duplex operation. Full duplex allows data to travel in both directions at the same time through an interface or a cable. A half-duplex setting ensures that data only travels in one direction at any given time. Although full duplex is faster, the interfaces sometimes cannot operate effectively in this mode. If you encounter excessive collisions or network errors, configure the interface for half duplex rather than full duplex.

Examples

```
ContentEngine(config-if)# fullduplex
ContentEngine(config-if)# no fullduplex
```

Related Commands

halfduplex

gui-server

To disable or specify the number of the Content Engine management graphical user interface (GUI) server port, use the **gui-server** global configuration command.

gui-server {**enable** | **port** *port*}

no gui-server {**enable** | **port** *port*}

Syntax Description

enable	Enables the graphical user interface.
port	Configures the graphical user interface server port.
port	Port number (1–65535). The default is 8001.

Defaults

The default port is 8001.

Command Modes

Global configuration

Examples

The following example enables the Content Engine management GUI on port 8002.

```
ContentEngine(config)# gui-server enable
ContentEngine(config)# gui-server port 8002
```

Related Commands

show gui-server

halfduplex

To configure an interface for half-duplex operation, use the **halfduplex** interface configuration command. To disable this function, use the **no** form of this command.

halfduplex

no halfduplex

Syntax Description

This command has no arguments or keywords.

Defaults

No default behavior or values

Command Modes

Interface configuration

Usage Guidelines

Use this command to configure an interface for half-duplex operation. Full duplex allows data to travel in both directions at the same time through an interface or a cable. A half-duplex setting ensures that data only travels in one direction at any given time. Although full duplex is faster, the interfaces sometimes cannot operate effectively in this mode. If you encounter excessive collisions or network errors, configure the interface for half duplex rather than full duplex.

Examples

```
ContentEngine(config-if)# halfduplex
ContentEngine(config-if)# no halfduplex
```

Related Commands

fullduplex

help

To obtain online help for the command-line interface, use the **help** EXEC or global configuration command.

help

Syntax Description

This command has no arguments or keywords.

Defaults

No default behavior or values

Command Modes

EXEC and global configuration

Usage Guidelines

You can get help at any point in a command by entering a question mark (**?**). If nothing matches, the help list will be empty, and you must back up until entering a **?** shows the available options.

Two styles of help are provided:

- Full help is available when you are ready to enter a command argument (for example, **show ?**) and describes each possible argument.
- Partial help is provided when you enter an abbreviated command and you want to know what arguments match the input (for example, **show stat?**).

Examples

```
ContentEngine# help
```

hostname

To configure the Content Engine's network name, use the **hostname** global configuration command. To reset the host name to the default setting, use the **no** form of this command.

hostname *name*

no hostname *name*

Syntax Description

name	New host name for the Content Engine; the name is case sensitive. The name may be from 1 to 22 alphanumeric characters.

Defaults

The default host name is the Content Engine model number (for example CE590 or CE7320).

Command Modes

Global configuration

Usage Guidelines

Use this command to configure the host name for the Content Engine. The host name is used for the command prompts and default configuration filenames.

Examples

The following example changes the host name to sandbox.

```
ContentEngine(config)# hostname sandbox
sandbox(config)#
```

The following example removes the host name.

```
ContentEngine(config)# no hostname
(config)#
```

http

To configure HTTP-related parameters, use the **http** global configuration command. To disable HTTP related-parameters, use the **no** form of this command.

http age-multiplier text *num* **binary** *num*

http anonymizer enable

http append {**proxy-auth-header** {*hostname* | *ip-address*} | **via-header** | **www-auth-header** {*hostname* | *ip-address*} | **x-forwarded-for-header**}

http authenticate-strip-ntlm

http authentication {**cache** {**max-entries** *entries* | **timeout** *minutes*} | **header** {**401** | **407**}}

http cache-authenticated {**all** | **basic** | **ntlm**}

http cache-cookies

http cache-on-abort {**enable** | **max-threshold** *maxthresh* | **min-threshold** *minthresh* | **percent** *percenthresh*}

http client-no-cache-request {**ignore** | **revalidate**}

http cluster {**heal-port** *number* | **http-port** *number* | **max-delay** *seconds* | **misses** *number*}

http l4-switch enable

http max-ttl {**days text** *textdays* **binary** *bindays* | **hours text** *texthours* **binary** *binhours* | **minutes text** *textminutes* **binary** *binminutes* | **seconds text** *textseconds* **binary** *binseconds*}

http min-ttl *minutes*

http object {**max-size** *maxsize* | **url-validation enable**}

http persistent-connections {**all** | **client-only** | **server-only** | **timeout** *seconds*}

http proxy {**incoming** *ports* | **outgoing** {**connection-timeout** *microsecs* | **host** {*hostname* | *ip-address*} *port* [**primary**] **monitor** *seconds* | **origin-server** | **preserve-407**}}

http reval-each-request {**all** | **none** | **text**}

http serve-ims {**text** *percentage* **binary** *percentage*}

http strict-request-content-length-checking enable

no http {**age-multiplier text** *num* **binary** *num* | **anonymizer enable** | **append** {**proxy-auth-header** {*hostname* | *ip-address*} | **via-header** | **www-auth-header** {*hostname* | *ip-address*} | **x-forwarded-for-header**} | **authenticate-strip-ntlm** | **authentication** {**cache** {**max-entries** *entries* | **timeout** *minutes*} | **header** {**401** | **407**}} | **cache-authenticated** {**all** | **basic** | **ntlm**} | **cache-cookies** | **cache-on-abort** {**enable** | **max-threshold** *maxthresh* | **min-threshold** *minthresh* | **percent** *percenthresh*} | **client-no-cache-request** {**ignore** | **revalidate**} | **cluster** {**heal-port** *number* | **http-port** *number* | **max-delay** *seconds* | **misses** *number*} | **l4-switch enable** | **max-ttl** {**days text** *textdays* **binary** *bindays* | **hours text** *texthours* **binary** *binhours* | **minutes text** *textminutes* **binary** *binminutes* | **seconds text** *textseconds* **binary** *binseconds*} |

min-ttl *minutes* | **object** {**max-size** *maxsize* | **url-validation enable**} | **persistent-connections** {**all** | **client-only** | **server-only** | **timeout** *seconds*} | **proxy** {**incoming** *ports* | **outgoing** {**connection-timeout** *microsecs* | **host** {*hostname* | *ip-address*} *port* [**primary**] **monitor** *seconds* | **origin-server** | **preserve-407**}} | **reval-each-request** {**all** | **none** | **text**} | **serve-ims** {**text** percentage **binary** percentage} | **strict-request-content-length-checking enable**}

Syntax Description

age-multiplier	HTTP/1.0 caching heuristic modifiers.
text	Heuristic modifier for text object.
num	Expiration time of text objects as a percentage of their age (0–100).
binary	Heuristic modifier for binary object.
num	Expiration time of binary objects as a percentage of their age (0–100).
anonymizer enable	Sets HTTP anonymizer.
append	Configures HTTP headers to be appended by the Content Engine.
proxy-auth-header	Configures host to receive Proxy Authorization header.
hostname	Name of host receiving Proxy Authorization header.
ip-address	IP address of host receiving Proxy Authorization header.
via-header	Includes “Via” header in responses and replies.
www-auth-header	Configures host to receive WWW Authorization header.
x-forwarded-for-header	Notifies web server of client’s IP address through “X-Forwarded-For” header.
authenticate-strip-ntlm	Strips NT LAN Manager (NTLM) authentication headers.
authentication	Configures parameters related to HTTP authentication.
cache	Configures authentication cache parameters.
max-entries	Sets the maximum number of entries in the authentication cache.
entries	Maximum number of entries in the authentication cache (500–32000).
timeout	Sets the timeout value of records in the authentication cache.
minutes	Time in minutes (30–1440) between the user’s last Internet access and the removal of that user’s entry from the authorization cache, forcing reauthentication. The default is 480 minutes; the minimum is 30 minutes; and the maximum is 1440 minutes (24 hours).
header	Determines which HTTP header to use for authentication (user ID and password) when the style of the HTTP request indicates that no proxy server is present. Headers can be either HTTP 401 (Unauthorized) or HTTP 407 (Proxy Authentication Required). The default is HTTP 401.
401	Uses HTTP 401 to query users for credentials.
407	Uses HTTP 407 to query users for credentials.
cache-authenticated	Caches and revalidates authenticated web objects.
all	Authenticates web object cache using any scheme.
basic	Authenticates web object cache using basic scheme authorization.
ntlm	Authenticates web object cache using NTLM scheme authorization.
cache-cookies	Caches web objects with associated cookies.
cache-on-abort	Sets cache-on-abort configuration options.
enable	Enables cache-on-abort feature.

max-threshold	Sets maximum threshold.
maxthresh	Value in kilobytes of maximum threshold (1–99999). Default is 256.
min-threshold	Sets minimum threshold.
minthresh	Value in kilobytes of minimum threshold (1–99999). Default is 32.
percent	Sets percent threshold.
percenthresh	Percentage value (1–99). Default is 80 percent.
client-no-cache-request	Configures management of no-cache requests.
ignore	Ignores the no-cache header in client request.
revalidate	Revalidates object with the origin server before serving a no-cache client request.
cluster	Configures cache cluster options.
heal-port	Listening port number of healing server for healing requests.
number	Healing server listener port number (1–65535). Default is 14333.
http-port	Healing server HTTP request forwarding port number.
number	HTTP request forwarding port number (1–65535). Default is 80.
max-delay	Maximum wait for response.
seconds	Maximum delay in seconds (0–10).
misses	Duration of healing mode (misses).
number	Total number of misses (0–999) before healing mode is disabled.
l4-switch enable	Configures parameters for Layer 4 switch redirection.
max-ttl	Sets maximum Time To Live for objects in the cache.
days	Sets maximum Time To Live for units in days.
hours	Sets maximum Time To Live for units in hours.
minutes	Sets maximum Time To Live for units in minutes.
seconds	Sets maximum Time To Live for units in seconds.
text	Sets maximum Time To Live for text objects.
binary	Sets maximum Time To Live for binary objects.
textdays	Maximum Time To Live (1–1825). The default is 3 days.
bindays	Maximum Time To Live (1–1825). The default is 7 days.
texthours	Maximum Time To Live (1–43800). The default is 72 hours.
binhours	Maximum Time To Live (1–43800). The default is 168 hours.
textminutes	Maximum Time To Live (1–2628000). The default is 4320 minutes.
binminutes	Maximum Time To Live (1–2628000). The default is 10080 minutes.
textseconds	Maximum Time To Live (1–157680000). The default is 259200 seconds.
binseconds	Maximum Time To Live (1–157680000). The default is 604800 seconds.
min-ttl	Sets minimum Time To Live for objects in the cache.
minutes	Minimum Time To Live in minutes (0–86400).
object	Sets URL validation and maximum size of HTTP objects.
max-size	Sets the maximum size of a cacheable object.
maxsize	Maximum size of a cacheable object in kilobytes (1–204799).
url-validation enable	Enables each HTTP validation request.
persistent-connections	Sets persistent connections configuration options.

all	Makes client and server connections persistent.
client-only	Makes only a client connection persistent.
server-only	Makes only a server connection persistent.
timeout	Sets persistent connections timeout value.
seconds	Persistent connections timeout in seconds (1–86400).
proxy	Configures parameters for proxy mode.
incoming	Configures for incoming proxy-mode requests.
ports	Ports on which to listen for incoming HTTP, FTP, and HTTPS proxy requests (1–65535). Up to eight ports can be specified. The default is no incoming proxy.
outgoing	Configures direct outgoing requests to another proxy server.
connection-timeout	Defines a timeout period, in microseconds, for probing outgoing proxy servers.
microsecs	Timeout period in microseconds (200–5000000).
host	Uses outgoing HTTP proxy.
hostname	Host name of outgoing proxy.
ip-address	IP address of outgoing proxy.
port	Port number of outgoing proxy (1–65535).
primary	(Optional) Makes the configured proxy the primary proxy server.
monitor	Defines the interval for monitoring the outgoing proxy servers.
seconds	Monitoring interval in seconds (10–300).
origin-server	Uses origin server if all outgoing proxies fail.
preserve-407	Preserves HTTP authentication header 407 by sending header 407 to the client when asking the client for Internet proxy authentication credentials.
reval-each-request	Configures revalidation for every request.
all	Revalidates all objects on every request.
none	Does not revalidate objects for each request.
text	Revalidates text objects on every request.
serve-ims	Configures If-Modified-Since (IMS) requests.
text	Modifies IMS request handling of text objects.
percentage	Age percentage to serve a text object without revalidation.
binary	Modifies IMS request handling of binary objects.
percentage	Age percentage to serve a binary object without revalidation.
strict-request-content-length-checking	Enables request content length checking options.
enable	Enables strict request content length checks.

Defaults

age-multiplier: 30 percent for text objects and 60 percent for binary objects

timeout *minutes*: 480

header: HTTP 401

maxthresh: 256 KB

minthresh: 32 KB

ports: no incoming proxy

percenthresh: 80 percent

heal-port *number*: 14333

textdays: 3 days

bindays: 7 days

texthours: 72 hours

binhours: 168 hours

textminutes: 4320 minutes

binminutes: 10080 minutes

textseconds: 259200 seconds

binseconds: 604800 seconds

misses *number*: 0 misses

object max-size: no maximum size

outgoing connection-timeout: 300 milliseconds

http strict-request-content-length-checking: disabled

The Content Engine strips the hop-hop 407 sent by the Internet proxy by default.

http cache-on-abort: disabled

Command Modes

Global configuration

Usage Guidelines

Use these commands to configure specific parameters for caching HTTP objects.

The **http anonymizer** command zeros out client IP addresses in the log files.

Note Text objects refer to HTML pages. Binary objects refer to all other web objects (for example, GIFs or JPEGs).

If a cached object's HTTP header does not specify an expiration time, the **age-multiplier** and **max-ttl** options provide a means for the Content Engine to age cached objects. The Content Engine's algorithm to calculate an object's cache expiration date is as follows:

Expiration date = (Today's date – Object's last modified date) * Freshness factor

The freshness factor is computed from the text and binary percentage parameters of the **age-multiplier** command. Valid age-multiplier values are 0 to 100 percent of the object's age. Default values are 30 percent for text and 60 percent for binary objects. After the expiration date, the object is considered stale and subsequent requests result in a fresh retrieval by the Content Engine.

When the Content Engine authenticates a user through a server, a record of that authentication is stored locally in the Content Engine RAM (authentication cache). As long as the authentication entry is kept, subsequent attempts to access restricted Internet content by that user do not require LDAP server lookups.

The **max-entries** option sets the maximum number of authentication cache entries retained.

The **timeout** command specifies how long an inactive entry can remain in the authentication cache before it is purged. Once a record has been purged, any subsequent access attempt to restricted Internet content requires a server lookup for reauthentication.

The **max-ttl** option sets the upper limit on estimated expiration dates. An explicit expiration date in the HTTP header (set by the web server) takes precedence over the **max-ttl** value.

HTTP Request Considerations

The ACNS 4.1 software Cache application supports Microsoft NT LAN Manager (NTLM), Lightweight Directory Access Protocol (LDAP), and RADIUS server HTTP request authentication. The **http authentication** command authenticates a user's domain, username, and password with a preconfigured primary domain controller (PDC) before allowing requests from the user to be served by the Content Engine.

When the Content Engine authenticates a user through an NTLM, RADIUS, or LDAP server, a record of that authentication is stored locally in the Content Engine RAM (authentication cache). As long as the authentication entry is kept, subsequent attempts to access restricted Internet content by that user do not require server lookups.

The **http authentication cache timeout** command specifies how long an inactive entry can remain in the authentication cache before it is purged. Once a record has been purged, any subsequent access attempt to restricted Internet content requires reauthentication.

LDAP authentication can be used with Websense URL filtering, but not with RADIUS authentication. Both LDAP and RADIUS rely on different servers, which may require different user IDs and passwords, making LDAP and RADIUS authentication schemes mutually exclusive. Should both RADIUS and LDAP be configured on the Content Engine at the same time, LDAP authentication is executed, not RADIUS authentication.

Excluding Domains from HTTP Authentication Servers

To exclude domains from HTTP authentication servers, use the **rule no-auth domain** command. LDAP, NTLM, or RADIUS authentication takes place only if the site requested does not match the specified pattern.

Proxy Mode Server Authentication

The events listed below occur when the Content Engine is configured for HTTP request authentication and one of the following two scenarios is true:

- The Content Engine receives a proxy-style request from a client.
- The Content Engine receives a transparent (WCCP-style) request from a client and the Content Engine **http authentication header** command parameter is set to 407 (because there is an upstream proxy).

1. The Content Engine examines the HTTP headers of the client request to find user information (contained in the Proxy-Authorization header).
2. If no user information is provided, the Content Engine returns a 407 (Proxy Authorization Required) message to the client.
3. The client resends the request, including the user information.
4. The Content Engine searches its authentication cache (based on user ID and password) to see whether the client has been previously authenticated.
5. If a match is found, the request is serviced normally.
6. If no match is found, the Content Engine sends a request to the authentication server to find an entry for this client.

7. If the server finds a match, the Content Engine allows the request to be serviced normally and stores the client user ID and password in the authentication cache.
8. If no match is found, the Content Engine again returns a 407 (Proxy Authorization Required) message to the client.

Transparent Mode Authentication

The events listed below occur when the Content Engine is configured for authentication and both of the following are true:

- The Content Engine receives a redirected request from a client.
- The **http authentication header** command parameter is set to 401 (because there is no upstream proxy).

1. The Content Engine searches its authentication cache to see whether the user's IP address has been previously authenticated.
2. If a match is found, the Content Engine allows the request to be serviced normally.
3. If no match is found in the first step, the Content Engine examines the HTTP headers to find user information (contained in the Authorization header).
4. If no user information is provided, the Content Engine returns a 401 (Unauthorized) message to the client.
5. The client resends the request, including the user information.
6. The Content Engine sends a request to the authentication server to find an entry for this user.
7. If the server finds a match, the Content Engine allows the request to be serviced normally and stores the client IP address in the authentication cache.
8. If no match is found, the Content Engine again returns a 401 (Unauthorized) message to the client.

In transparent mode, the Content Engine uses the client IP address as a key for the authentication database.

If you are using user authentication in transparent mode, we recommend that the AuthTimeout interval configured with the **http authentication cache timeout** command be short. IP addresses can be reallocated, or different users can access the Internet through an already authenticated device (PC, workstation, and the like). Shorter AuthTimeout values help reduce the possibility that individuals can gain access using previously authenticated devices. When the Content Engine operates in proxy mode, it can authenticate the user with the user ID and password.

Server Redundancy

Two authentication servers can be specified with the **server host command option** to provide redundancy and improved throughput. Content Engine load-balancing schemes distribute the requests to the servers. If the Content Engine cannot connect to either server, no authentication can take place, and users who have not been previously authenticated are denied access.

Security Options

The Content Engine uses simple (nonencrypted) authentication to communicate with the LDAP server. Future expansion may allow for more security options based on Secure Socket Layer (SSL), SASL, or certificate-based authentication.

Hierarchical Caching

In some cases, users are located at branch offices. A Content Engine (CE1) can reside with them in the branch office. Another Content Engine (CE2) can reside upstream, with an NTLM, RADIUS, or LDAP server available to both Content Engines for user authentication.

Note The **http append proxy-auth-header** global configuration command must be configured on the downstream Content Engines to ensure that proxy authorization information, required by upstream Content Engines, is not stripped from the HTTP request by the downstream Content Engines. Up to 8 upstream IP addresses can be configured on each downstream Content Engine.

If branch office user 1 accesses the Internet, and content is cached at CE1, then this content cannot be served to any other branch office user unless that user is authenticated. CE1 must authenticate the local users.

Assuming that both CE1 and CE2 are connected to the server and authenticate the users, when branch office user 2 firsts requests Internet content, CE1 responds to the request with an authentication failure response (either HTTP 407 if in proxy mode, or HTTP 401 if in transparent mode). User 2 enters the user ID and password, and the original request is repeated with the credentials included. CE1 contacts the HTTP request authentication server to authenticate user 2.

Assuming authentication success, and a cache miss, the request along with the credentials is forwarded to CE2. CE2 also contacts the authentication server to authenticate user 2. Assuming success, CE2 either serves the request out of its cache or forwards the request to the origin server.

User 2 authentication information is now stored in the authentication cache in both CE1 and CE2. Neither CE1 nor CE2 needs to contact the authentication server for user 2's subsequent requests (unless user 2's entry expires and is removed from the authentication cache).

This scenario assumes that CE1 and CE2 use the same method for authenticating users. Specifically, both Content Engines must expect the user credentials (user ID and password) to be encoded in the same way.

Hierarchical Caching in Transparent Mode

When the Content Engine operates in transparent mode, the user IP address is used as a key to the authentication cache. When user 2 sends a request transparently to CE1, after authentication, CE1 inserts its own IP address as the source for the request. Therefore, CE2 cannot use the source IP address as a key for the authentication cache.

When CE1 inserts its own IP address as the source, it must also insert an X-Forwarded-For header in the request (**http append x-forwarded-for-header** command). CE2 must first look for an X-Forwarded-For header. If one exists, that IP address must be used to search the authentication cache. Assuming the user is authenticated at CE2, then CE2 must not change the X-Forwarded-For header, just in case there is a transparent CE3 upstream.

In this scenario, if CE1 does not create an X-Forwarded-For header (for example, if it is not a Cisco Content Engine and does not support this header), then authentication on CE2 will not work.

Hierarchical Caching, Content Engine in Transparent Mode with an Upstream Proxy

In a topology with two Content Engines, assume that CE1 is operating in transparent mode and CE2 is operating in proxy mode, with the browsers of all users pointing to CE2 as a proxy.

Because the browsers are set up to send requests to a proxy, an HTTP 407 message is sent from CE1 back to each user to prompt for credentials. By using the 407 message, the problem of authenticating based on source IP address is avoided. The username and password can be used instead.

This mode provides better security than using the HTTP 401 message. The Content Engine examines the style of the address to determine whether there is an upstream proxy. If there is, the Content Engine uses an HTTP 407 message to prompt the user for credentials even when operating in transparent mode.

Authentication Cache Size Adjustments

If the authentication cache is not large enough to accommodate all authenticated users at the same time, the Content Engine purges older entries that have not yet timed out.

Transaction Logging

Once a user has been authenticated through LDAP, NTLM, or a RADIUS server, all transaction logs generated by the Content Engine for that user contain user information. If the Content Engine is acting in proxy mode, the user ID is included in the transaction logs. If the Content Engine is acting in transparent mode, the user IP address is included instead.

If the **transaction-logs sanitize** command is invoked, the user information is suppressed.

In this example, the host for the LDAP server daemon is configured:

```
Console(config)# ldap server host www.someDomain.com port 390
```

To delete an LDAP server, use the **no ldap server** command.

```
Console(config)# no ldap server host 1.1.1.1
```

In this example, the host for the RADIUS server is configured:

```
Console(config)# radius-server 172.16.90.121
```

In this example, the length of time that entries are valid in the authentication cache is set:

```
Console(config)# http authentication cache timeout 1000
```

The following example specifies that the Content Engine should use header 407 when asking the end user for authentication credentials (user ID and password).

```
Console(config)# http authentication header 407
```

The **cache-cookies** option enables the Content Engine to cache binary content served with HTTP Set-cookie headers and no explicit expiration information.

The **reval-each-request** option enables the Content Engine to revalidate all objects requested from the cache, text objects only, or none at all.

Use the **object max-size** option to specify the maximum size in kilobytes of a cacheable object. The default is no maximum size for a cacheable object. The **no** form of the command resets the default value.

The **http proxy** options enable the Content Engine to operate in environments where WCCP is not enabled, or where client browsers have previously been configured to use a legacy proxy server. The Content Engine accepts proxy-style requests when the incoming proxy ports are configured with the **http proxy incoming** *ports* option. Up to eight incoming proxy ports can be specified on a single command line or on multiple command lines.

To configure the Content Engine to direct all HTTP miss traffic to a parent cache (without using ICP or WCCP), use the **http proxy outgoing host** *port* option, where **host** is the system name or IP address of the outgoing proxy server, and *port* is the port number designated by the outgoing (upstream) server to accept proxy requests.

The **cache-on-abort** option provides user-defined thresholds to determine whether or not the Content Engine will complete the download of an object when the client has aborted the request. When the download of an object aborts before it is completed, the object is not stored on the Content Engine

or counted in the hit-rate statistics. Client abort processing occurs when a client of the Content Engine aborts the download of a cacheable object before the download is complete. Typically, a client aborts a download by clicking the Stop icon on the browser, or by closing the browser during a download.

If the **cache-on-abort** option is enabled and all cache-on-abort thresholds are disabled, then the Content Engine always aborts downloading an object to the cache. If the Content Engine determines that there is another client currently requesting the same object, downloading is not aborted. The Content Engine only applies those thresholds that have been enabled.

To specify the port number over which requests from the healing Content Engine are sent to other Content Engines in the cluster, use the **http cluster http-port** option.

Note The default port number is 80. If you choose to configure a port other than the default, you must make sure that the port configured matches the port specified in the **http proxy incoming** command on healing servers in the farm. Otherwise, the healing client is not able to retrieve objects from the healing servers.

To return to the default port number, use the **no http cluster http-port** command.

The **client-no-cache-request** allows a choice between ignoring the no-cache client request or revalidating the object with the origin server before serving the no-cache client request. These choices are mutually exclusive, and the last selection takes effect.

The **l4-switch enable** option permits the Content Engine to transparently receive Layer 4 redirected traffic from Layer 4-enabled switches such as the Cisco CSS11000 series switches. Refer to the switch documentation for specific configuration information.

Configuring Healing Mode

When a Content Engine is added to an existing WCCP Version 2 cache group (cluster), it can receive requests for content that was formerly served by another cache in the cluster. This event is termed a "near-miss," because if the request had been sent to the former Content Engine, it would have been a cache hit. A near-miss lowers the overall cache hit rate of the Content Engine cluster.

Healing mode allows the newly added Content Engine to query and obtain cache objects from all other caches in the cluster on a cache miss event. If the object is not found in the cluster, the Content Engine processes the request through the outgoing proxy or origin server. The Content Engine in healing mode is called a healing client. The caches in the cluster that respond to healing client requests are called healing servers.

Note Healing mode is only invoked on a healing client when the request is transparently redirected to the Content Engine. Healing mode is not invoked when the request is sent to the Content Engine in proxy mode.

The **http cluster** command modifies the healing mode parameters. The **http cluster http-port** command specifies the port number over which requests from the healing Content Engine are sent to other Content Engines in the cluster.

Note The default port number is 80. If you choose to configure a port other than 80, you must make sure that the port that is configured matches the port specified in the **http proxy incoming** command on healing servers in the farm. Otherwise, the healing client is not able to retrieve objects from the healing servers.

The **http cluster heal-port** command specifies the port number over which the healing client sends healing queries and the healing server sends healing responses. The default port number is 14333. If a port other than the default is configured, make sure that all Content Engines in the cluster use the same port.

The **http cluster misses** command specifies the maximum number of misses that the healing Content Engine can receive from the cluster from the last healing mode hit response until the healing process is disabled. The default is 0 misses. The **http cluster max-delay** command specifies the maximum time interval in seconds for which a healing Content Engine waits for a healing response from the cluster before considering the healing request a miss.

To enable the healing client, you should, at the least, configure the **max-delay** and **misses** options. The default port number for **http-port** is 80. If you use the default port, you do not have to configure **http-port**. The default port number for **heal-port** is 14333.

To disable the healing client, you should, at the least, configure either **misses** or **max-delay** to 0, or you can use the **no** form of the command as follows:

http cluster misses 0

no http cluster misses

http max-delay 0

no http cluster max-delay

Note Healing mode existed in Cache software, 2.x releases.

HTTP Proxy Failover

The **http proxy outgoing** option can configure up to eight backup proxy servers for the HTTP proxy failover feature. One proxy server functions as the primary proxy server and all requests are redirected to it. If the primary proxy server fails to respond to the HTTP CONNECT, the server is noted as failed and the requests are redirected to the next outgoing proxy server until one of the proxies service the request. The **no http proxy outgoing connection-timeout** option causes the timeout to be set to the default value of 300 milliseconds.

To explicitly designate the primary proxy, use the **primary** keyword. If several proxies are configured with the **primary** keyword, the last one configured overrides the others. Failover to a proxy server occurs in the order the proxy servers were configured. If all the configured proxy servers fail, the Content Engine can optionally redirect requests to the origin server if the user enters the **http proxy outgoing origin-server** option. If the user has configured the **origin-server** option, the Content Engine directs HTTP requests to the original server specified in the HTTP header. If the option is not enabled, the client receives an error response. Response errors and read errors are returned to the client, because it is not possible to detect whether these errors are generated at the origin server or at the proxy.

The state of the proxy servers is maintained by active monitoring, which occurs in the background. The state of the proxy servers can be seen in the CLI and syslog NOTICE messages. This interval is configured with the **http proxy outgoing monitor** option. This outgoing monitor interval is the interval of time over which the proxy servers are polled. If one of the proxy servers is unavailable, the polling mechanism waits for the connect timeout (300 milliseconds) before polling the next server.

Requests with a destination specified in the **proxy-protocols outgoing-proxy exclude** command bypass the Content Engine proxy as well as the failover proxies.

By default, the Content Engine strips the hop-hop 407 (Proxy Authentication Required) error code sent by Internet proxy. If the **http proxy outgoing preserve-407** command is invoked, the Content Engine sends the 407 error code to the client, and the Internet proxy authenticates the client.

Note If the client is connected to the Content Engine in transparent mode and the user does preserve 407 error codes by invoking the **http proxy outgoing preserve-407** command, client browsers will not recognize the 407 error codes.

When an HTTP request intended for another proxy server is intercepted by the Content Engine in transparent mode, the Content Engine forwards the request to the intended proxy server if the **proxy-protocols transparent original-proxy** command was entered.

The proxy failover feature currently supports only HTTP, not HTTPS or FTP.

The **persistent-connections** option enables persistent connections on the Content Engine. To configure the number of seconds the Content Engine should wait for a connection response before it times out, use the **timeout** option.

The **http object url-validation enable** option has a dependency with the **ip name-server** CLI command. When the **ip name-server** option is not configured (for example, during transparent proxy), **http object url-validation enable** is dynamically turned off. When the **ip name-server** option is configured, **http object url-validation enable** is turned on automatically if and only if it was enabled.

Caution URL validation is on by default. Cisco Systems strongly recommends that you keep URL validation enabled, because disabling URL validation might make the Content Engine vulnerable to corruption from the HTTP objects in the cache.

Use the **proxy-protocols outgoing-proxy exclude** global configuration command to specify a domain for which the Content Engine should not use an upstream proxy. In the following example, the domain cisco.com is outgoing proxy-excluded.

```
ContentEngine(config)# proxy-protocols outgoing-proxy exclude cisco.com
```

The Content Engine will not use the upstream proxy for any domain that ends with the listed domain name. For example, if you specify cisco.com, the configured outgoing proxy server will be bypassed each time the Content Engine tries to retrieve a web page from videos.cisco.com, or personals.cisco.com.

For IP addresses, enter the full IP address or use the asterisk “*” as a wildcard for IP address fields as follows:

172.16.1.*

172.16.*.*

172.*.*.*

The syntax 172.16.*.* indicates that all requests to the domain host of 172.16.xxx.xxx will be excluded. Wildcard syntax does not support “0” or “?”.

The following forms of wildcard specification are not supported:

172.*.10.2

172.31.1*.8

Examples

This **http authentication** example sets the length of time that entries are valid in the authentication cache.

```
Console(config)# http authentication cache timeout 1000
```

The following **http authentication** example specifies that the Content Engine should use header 407 when asking the end user for authentication credentials (user ID and password).

```
Console(config)# http authentication header 407
```

In this **http proxy outgoing** example, the host 10.1.1.1 on port 8088 is designated the primary proxy server, and host 10.1.1.2 is a backup proxy server.

```
ContentEngine(config)# http proxy outgoing host 10.1.1.1 8088 primary
ContentEngine(config)# http proxy outgoing host 10.1.1.2 220
```

In this example, the Content Engine is configured to redirect requests directly to the origin server if all of the proxy servers fail.

```
ContentEngine(config)# http proxy outgoing origin-server
```

In this example, the Content Engine is configured to monitor the proxy servers every 120 seconds.

```
ContentEngine(config)# http proxy outgoing monitor 120
```

To disable any of the preceding commands, use the **no** version of the command.

Proxy Failover show Commands

```
ContentEngine# show http proxy
Incoming Proxy-Mode:
  Servicing Proxy mode HTTP connections on ports:   8080

Outgoing Proxy-Mode:
  Primary proxy server: 172.16.63.150   port 1 Failed
  Backup proxy servers: 172.16.236.151  port 8005
                        172.16.236.152  port 123
                        172.16.236.153  port 65535 Failed
                        172.16.236.154  port 10
Monitor Interval for Outgoing Proxy Servers is 60 seconds
Use of Origin Server upon Proxy Failures is disabled.
```

Statistics

```
ContentEngine# show statistics http requests
Statistics - Requests
                                          Total             % of Requests
                              ------------------------------------------------
     Total Received Requests:                49103                        -
              Forced Reloads:                  109                      0.2
               Client Errors:                   23                      0.0
               Server Errors:                  348                      0.7
                 URL Blocked:                    0                      0.0
      Sent to Outgoing Proxy:                    0                      0.0
Failures from Outgoing Proxy:                    0                      0.0
Excluded from Outgoing Proxy:                    0                      0.0
             ICP Client Hits:                    0                      0.0
             ICP Server Hits:                    0                      0.0
           HTTP 0.9 Requests:                    2                      0.0
           HTTP 1.0 Requests:                49101                    100.0
           HTTP 1.1 Requests:                    0                      0.0
       HTTP Unknown Requests:                    0                      0.0
           Non HTTP Requests:                    0                      0.0
         Non HTTP Responses:                    46                      0.1
      Chunked HTTP Responses:                    0                      0.0
        Http Miss Due To DNS:                    0                      0.0
    Http Deletes Due To DNS:                     0                      0.0
  Objects cached for min ttl:                 2674                      5.

ContentEngine# show statistics http proxy outgoing
```

```
HTTP Outgoing Proxy Statistics
IP               PORT    ATTEMPTS   FAILURES
--------------------------------------------------
172.16.23.150    8000    0          0
172.16.23.151    8080    0          0
172.16.23.152    9000    0          0
172.16.23.153    9001    0          0
172.16.23.154    9005    0          0


 Requests when all proxies were failed: 0

ContentEngine(config)# http append via-header
ContentEngine(config)# http append x-forwarded-for-header
ContentEngine(config)# http age-multiplier text 30 bin 60
ContentEngine(config)# no http age-multiplier text 30 bin 60
ContentEngine(config)# http reval-each-request text
ContentEngine(config)# no http reval-each-request text
```

In this example, with the default configuration (all **cache-on-abort** thresholds disabled), client abort processing is configured to always abort downloading an object to the cache:

```
ContentEngine(config)# http cache-on-abort enable
```

In this example, the Content Engine is configured to always continue downloading an object to the cache (this is the default configuration):

```
ContentEngine(config)# no http cache-on-abort
```

In this example, the Content Engine is configured to use the default minimum threshold when the **cache-on-abort** option has been enabled, and the threshold is set to 16 kilobytes:

```
ContentEngine(config)# http cache-on-abort min 16
```

In this example, the Content Engine is configured to not consider the minimum threshold:

```
ContentEngine(config)# no http cache-on-abort min
```

The **cache-on-abort max-threshold** and **percent** thresholds are configured like the minimum threshold shown in the examples.

This example enables the healing mode feature by setting the HTTP port 8080 for forwarding HTTP requests to a specific port (3144) on a healing server, setting the maximum delay to wait for a response from the cluster in seconds before considering the healing request a miss, and setting the maximum number of misses that the healing Content Engine can receive from the cluster before healing mode is disabled at healing client.

```
Console(config)# http cluster http-port 8080

Console(config)# http cluster heal-port 3144

Console(config)# http cluster max-delay 5

Console(config)# http cluster misses 5
```

In this example, the **show statistics http cluster** command displays the statistics of the healing client and the healing server. The **clear statistics http cluster** command resets the healing mode statistics:

```
Console(config)# show statistics http cluster
Yimin-507#show stat http cluster
Healing mode max attempts              = 0
Healing mode max latency               = 10
Healing mode current cumulative misses = 0
```

```
Healing mode client statistics
------------------------------
Client Requests  Sent     = 0
Client Responses Received = 0
Client Responses Hit      = 0
Client Responses Miss     = 0
Client Responses Error    = 0
Client Responses Timeout  = 0

Healing mode server statistics
------------------------------
Server Requests Received  = 0
Server Responses Sent     = 0
Server Responses Hit      = 0
Server Responses Miss     = 0
Server Responses Error    = 0

Yimin-507#
Console(config)# clear statistics http cluster
```

The **show http cluster** command displays **max-delay**, **misses**, **http-port**, and **heal-port** values. In the first example, the values are set to 0 and the healing client is disabled.

```
Console(config)# show http cluster
Healing client is disabled

Timeout for responses = 10 seconds
Max number of misses allowed before stop healing mode = 0
Port number for healing request/response = 14333
Http-port to forward http request to healing server = 80
```

In this example the healing client is enabled.

```
Console(config)# show http cluster
Healing client is enabled

Timeout for responses = 10 seconds
Max number of misses allowed before stop healing mode = 999
Port number for healing request/response = 14333
Http-port to forward http request to healing server = 80
```

Related Commands

proxy-protocols

rule no-proxy

rule use-proxy

rule use-proxy-failover

show http

show http proxy

show statistics http requests

show statistics http proxy outgoing

https

To configure the Content Engine for HTTPS proxy services, use the **https** global configuration command.

https destination-port allow {*ports* | **all**}

https destination-port deny {*ports* | **all**}

https proxy incoming *ports*

https proxy outgoing host {*hostname* | *ip-address*} *port*

no https {**destination-port allow** {*port* | **all**} | **deny** {*port* | **all**} | **proxy** {**incoming** *port* | **outgoing host** {*hostname* | *ip-address*} *port*}}

Syntax Description

destination-port	Destination port restrictions.
allow	Allows HTTPS traffic to specified ports.
ports	Up to eight port numbers (1–65535).
all	Specifies all ports.
deny	Denies HTTPS traffic to specified ports.
ports	Up to eight port numbers (1–65535).
all	Specifies all ports.
proxy	Sets configuration parameters for proxy mode.
incoming	Sets configuration for incoming proxy-mode requests.
ports	Up to eight port numbers (1–65535) to listen for HTTPS requests.
outgoing	Sets configuration to direct outgoing requests to another proxy server.
host	Uses outgoing HTTPS proxy.
hostname	Host name of outgoing proxy.
ip-address	IP address of outgoing proxy.
port	Port of outgoing proxy (1–65535).

Defaults

No default behavior or values

Command Modes

Global configuration

Usage Guidelines

The following table shows CLI commands associated with HTTPS proxy features.

HTTPS Proxy Features	Related CLI Commands (Abbreviated Syntax)
Supports up to eight incoming proxy ports.	**https proxy incoming** *port_1–65535, port, ...*
Shares proxy port with transparent services by configuring a WCCP service and an HTTPS incoming proxy on the same port.	**https proxy incoming** *ports_1–65535* **wccp custom-web-cache** ...
Configures outgoing HTTPS proxy server using the global **exclude** option for the HTTPS proxy.	**proxy-protocols outgoing-proxy exclude** *domain_name* **https proxy outgoing host** {*hostname* | *ip_address*} *port_1–65535*
Original versus default outgoing HTTPS proxy decision process.	**proxy-protocols transparent** {**default-server** | **-proxy**}
Handles in transparent mode an HTTPS request bound for another proxy host	**proxy-protocols transparent** {**default-server** | **-proxy**}

The order in which the CLI commands are entered is not important.

Cisco ACNS software supports HTTPS in the following two scenarios:

- The Content Engine receives an HTTPS request sent by a web client configured to use the Content Engine as an HTTPS proxy server.
- The Content Engine in transparent mode intercepts a request sent by a web client to another HTTPS proxy server.

In both cases the Content Engine creates a connection to the origin server (directly or through another proxy server) and allows the web client and origin server to set up an SSL tunnel through the Content Engine.

HTTPS traffic is encrypted and cannot be interpreted by the Content Engine or any other device between the web client and the origin server. HTTPS objects are not cached.

The Content Engine as an HTTPS proxy server supports up to eight ports. It can share the ports with transparent-mode services and with HTTP. In proxy mode, the Content Engine accepts and services the HTTPS requests on the ports specified with the **https proxy incoming** command. All HTTPS requests on other proxy-mode ports are rejected in accordance with the error-handling settings on the Content Engine. In transparent mode, all HTTPS proxy-style requests intended for another HTTPS proxy server are accepted. The Content Engine acts on these transparently received requests in accordance with the **proxy-protocols transparent** command.

When the Content Engine is configured to use an HTTPS outgoing proxy with the **https proxy outgoing host** command, all incoming HTTPS requests are directed to this outgoing proxy. The **proxy-protocols outgoing-proxy exclude** command specifies a global proxy exclude domain effective for all proxy server protocols including HTTPS. The Content Engine applies the following logic when an outgoing proxy server is configured:

- If the destination server is specified by the global **exclude** option, then go directly to the destination server.
- If the destination server is not specified by the global **exclude** option and the request is HTTP, go directly to the destination server.
- If the destination server is not specified by the global **exclude** option, then go to the outgoing proxy server.

When a Content Engine intercepts a proxy request intended for another proxy server and there is no outgoing proxy configured for HTTPS, and the **proxy-protocols transparent default-server** command is invoked, the Content Engine addresses the request to the destination server directly and not to the client's intended proxy server.

Statistics Reporting

Only connection statistics are reported. Because requests and responses are sent through the secure tunnel, the Content Engine is not able to identify the number of requests sent, or the number of bytes per request. Thus, the request and transaction per second (TPS) statistics are not available for HTTPS.

Transaction Logging

The Content Engine logs HTTPS transactions in the transaction log in accordance with Squid syntax. One log entry is made for each HTTPS connection, though many transactions are performed per connection. The Content Engine is not aware of objects conveyed through the SSL tunnel, only the HTTPS server name.

Syslog and URL Tracking

When URL tracking is enabled, the Content Engine logs HTTPS transaction information to the syslog file. The syslog entries have the prefix <https>. For HTTPS there are no "misses" or "hits." Because the Content Engine ignores objects transferred through an SSL tunnel, there is only one URL tracking entry per HTTPS connection (similar to the transaction log).

Examples

In this example, the Content Engine is configured as an HTTPS proxy server, and accepts HTTPS requests on port 8081. Only a single port is supported in the HTTPS protocol.

```
ContentEngine(config)# https proxy incoming 8081
```

In this example, the Content Engine is configured to forward HTTPS requests to an outgoing proxy server (10.1.1.1) on port 8880.

```
ContentEngine(config)# https proxy outgoing host 10.1.1.1 8880
```

In this example, a domain name is excluded from being forwarded to an outgoing proxy server.

```
ContentEngine(config)# proxy-protocols transparent default-server
ContentEngine(config)# proxy-protocols outgoing-proxy exclude cruzio.com
```

Related Commands

proxy-protocols

http proxy

show proxy-protocols

show http proxy

icp

To configure the Internet Cache Protocol (ICP) client and server, use the **icp** global configuration command. To disable the ICP client and server, use the **no** form of this command.

icp client add-remote-server {*hostname* | *ip-address*} {**parent** | **sibling**} **icp-port** *icpport* **http-port** *httpport* [**restrict** *domainnames*]

icp client enable

icp client exclude *domainnames*

icp client max-fail *retries*

icp client max-wait *timeout*

icp client modify-remote-server {*hostname* | *ip-address*} {**http-port** *port* | **icp-port** *port* | **parent** | **restrict** *domainnames* | **sibling**}

icp server enable

icp server http-port *port*

icp server port *icpport*

icp server remote-client {*hostname* | *ip-address*} {**fetch** | **no-fetch**}

no icp {**client** {{**add-remote-server** {*hostname* | *ip-address*} {**parent** | **sibling**} **icp-port** *icpport* **http-port** *httpport* [**restrict** *domainnames*]} | **enable** | **exclude** *domainnames* | **max-fail** *retries* | **max-wait** *timeout* | **modify-remote-server** {*hostname* | *ip-address*} {**http-port** *port* | **icp-port** *port* | **parent** | **restrict** *domainnames* | **sibling**}} | **server** {**enable** | **http-port** *port* | **port** *icpport* | **remote-client** {*hostname* | *ip-address*} {**fetch** | **no-fetch**}}}

Syntax Description

client	Sets ICP client functionality.
add-remote-server	Adds an ICP client remote server.
hostname	Host name of remote server.
ip-address	IP address of remote server.
parent	ICP server that acts like a parent.
sibling	ICP server that acts like a sibling.
icp-port	Sets ICP port to receive remote requests.
icpport	ICP port number (0–65535).
http-port	Sets HTTP port to receive HTTP requests.
httpport	HTTP request port number (0–65535).
restrict	(Optional) Sets a restricted list of domains.
domainnames	Space-delimited restricted domain list.
enable	Enables the ICP client.
exclude	Excludes ICP client local domains.
domainnames	Space-delimited local domain list.

max-fail	Sets maximum number of retries allowed.
retries	Number of retries (0–100).
max-wait	Sets maximum wait for ICP responses before timeout occurs.
timeout	Timeout period for ICP responses in seconds (0–30).
modify-remote-server	Modifies the ICP client remote server parameters.
hostname	Host name of remote server.
ip-address	IP address of remote server.
http-port	Sets HTTP port.
port	HTTP request port number (0–65535).
icp-port	Sets ICP port.
port	ICP request port number (0–65535).
parent	ICP remote server that acts like a parent.
restrict	Sets restricted list of domains.
domainnames	Space delimited local domain list.
sibling	ICP remote server that acts like a sibling.
server	Sets ICP server functionality.
enable	Enables the ICP server.
http-port	HTTP proxy port to listen for ICP-generated requests.
port	HTTP server port number (0–65535) for ICP.
port	ICP server listener port that listens for ICP requests.
icpport	ICP request port number (0–65535).
remote-client	Sets ICP server remote client.
hostname	Host name of remote client.
ip-address	IP address of remote client.
fetch	Sets ICP remote client to fetch cache miss.
no-fetch	Sets ICP remote client to not fetch cache miss.

Defaults

http-port: 3128

Command Modes

Global configuration

Usage Guidelines

Use these commands to establish and configure the ICP server and client functionality of the Content Engine. Configurations made without enabling ICP functionality are stored within the configuration until removed. To enable the ICP server or client functionality, use the **icp** {**server** | **client**} **enable** command. Be sure to enable the ICP on any other Content Engines or ICP servers or clients within the ICP environment to ensure proper service. You can monitor the statistical data of the ICP service using the **show statistics icp** EXEC command.

Examples

The following example restricts ICP parent and sibling to specific domain sets.

```
ContentEngine(config)# icp client add-remote-server 172.16.0.0 parent icp-port 3130
http-port 3128 domain_x.com domain_y.com domain_z.com

ContentEngine(config)# icp client add-remote-server 172.16.0.0 sibling icp-port 3130
http-port 3128 domain_a.com domain_b.com domain_c.com

ContentEngine(config)# icp client enable
Icp Client started
```

Related Commands

show icp client

show icp server

show statistics icp

inetd

To configure, enable, and disable TCP/IP FTP, RCP, and TFTP services, use the **inetd** global configuration command. To disable these same TCP/IP services, use the **no** form of this command.

inetd enable *service concurrent_tasks*

no inetd enable *service concurrent_tasks*

Syntax Description

enable	Enables TCP/IP services.
service	Name of service to be enabled: FTP, RCP, and TFTP.
concurrent_tasks	Maximum number of concurrent sessions supported for the specified service (1–20).

Defaults

No default behavior or values

Command Modes

Global configuration

Usage Guidelines

Use the FTP, RCP, and TFTP commands to enable and disable TCP/IP services on the Content Engine. To disable the service, enter the **no** form of the **inetd enable** command. The maximum number of sessions for any service is 20. Use the **show inetd** command to view whether current **inetd** sessions are enabled or disabled.

To configure TFTP files for write or copy, follow these steps.

Note These directories must be created if they do not already exist.

Step 1 Create or specify the directories that you want to allow clients to access through the TFTP service on the Content Engine.

```
ContentEngine(config)# tftp-server dir /local1/tftpboot
ContentEngine(config)# tftp-server dir /local1/tftpboot/uploads
```

Note Multiple directories can be specified, depending on the write or copy file destination. The first file entered in your list is the default directory.

Step 2 Configure the Content Engine trusted hosts (clients) to allow access to your directories.

```
ContentEngine(config)# trusted-host ip-address
```

or

```
ContentEngine(config)# trusted-host hostname
```

The **trusted-host domain-lookup** is enabled by default. In the enabled condition, the Content Engine attempts to resolve addresses to host names and host names to addresses, requiring addresses or host names entered to be in DNS. To avoid IP address resolution, disable domain lookup by entering **no trusted-host domain-lookup** and entering your IP address. (This entry is an address of an "unknown" host name.) Then re-enable the domain lookup by entering **trusted-host domain-lookup** and access the IP address entered.

An alternative is to disable the **trusted-host domain-lookup** command, allowing anyone to access the files. However, you still need to add the directories you want to make accessible. Although the **no trusted-host** {*ip-address* | *hostname*} appears to be an unusable command, it is actually a hidden command.

Examples

This example enables an FTP service session.

```
Console(config)# inetd enable ftp
```

This example disables FTP services.

```
Console(config)# no inetd enable ftp
```

Related Commands

show inetd

install

To install Content Engine software, use the **install** EXEC command.

install *imagefilename*

Note The **install** command does not accept .pax files. Files should be of the type .bin (for example, cache-sw.bin). Also, if the release being installed does not require a new system image, then it may not be necessary to write to Flash memory. If the newer version has changes that require a new system image to be installed, then the **install** command may result in a write to Flash memory.

Syntax Description

imagefilename	Name of the .bin file you want to install.

Defaults

No default behavior or values

Command Modes

EXEC

Usage Guidelines

The **install** command loads the system image into Flash memory and copies components of the optional software to the swfs partition.

Note If you are installing a system image that contains optional software, make sure that a software file system (swfs) partition is mounted on disk00.

To install a system image, copy the image file to the sysfs directory local1 or local2. Before executing the **install** command, change the present working directory to the directory where the system image resides. When the **install** command is executed, the image file is expanded. The expanded files overwrite the existing files in the Content Engine. The newly installed version takes effect after the system image is reloaded.

Examples

```
ContentEngine# install ce7320-cache-311.bin
```

Related Commands

reload

interface

To configure a Fast Ethernet or Gigabit Ethernet interface, use the **interface** global configuration command. To disable selected options, restore default values, or enable a shut down interface, use the **no** form of this command.

interface {**FastEthernet** | **GigabitEthernet**} *slot/port* **autosense**

interface {**FastEthernet** | **GigabitEthernet**} *slot/port* **bandwidth** *linespeed*

interface {**FastEthernet** | **GigabitEthernet**} *slot/port* **cdp enable**

interface {**FastEthernet** | **GigabitEthernet**} *slot/port* {**fullduplex** | **halfduplex**}

interface {**FastEthernet** | **GigabitEthernet**} *slot/port* **ip address** *ip-address netmask* [**secondary**]

interface {**FastEthernet** | **GigabitEthernet**} *slot/port* **shutdown**

interface {**FastEthernet** | **GigabitEthernet**} *slot/port* **standby** *grpnumber* {**errors** *maxerrors* | **ip** *ip-address netmask* | **priority** *priority*}

no interface {**FastEthernet** | **GigabitEthernet**} *slot/port* {**autosense** | **bandwidth** *linespeed* | **cdp enable** | **fullduplex** | **halfduplex** | **ip address** *ip-address netmask* [**secondary**] | **shutdown** | **standby** *grpnumber* {**errors** *maxerrors* | **ip** *ip-address netmask* | **priority** *priority*}}

Syntax Description

FastEthernet	Selects a Fast Ethernet interface to configure.
GigabitEthernet	Selects a Gigabit Ethernet interface to configure.
slot/port	Slot and port number for the selected interface. Slot range is 0–3; port range is 0–3. The slot number and port number are separated with a forward slash character (/).
autosense	Sets interface to autosense.
bandwidth	Sets bandwidth of specified interface.
linespeed	Bandwidth of interface in megabits per second (Mbps) (10, 100, or 1000).
cdp enable	Enables Cisco Discovery Protocol interface.
fullduplex	Sets interface to full-duplex operation.
halfduplex	Sets interface to half-duplex operation.
ip address	Sets IP address and netmask.
ip-address	IP address of interface.
netmask	Netmask of interface.
secondary	(Optional) Makes this IP address a secondary address.
shutdown	Shuts down the specified interface.
standby	Sets standby interface configuration commands.
grpnumber	Standby group number (1–4).
errors	Sets the maximum number of errors allowed in a standby group.
maxerrors	Maximum number of errors allowed (0–42949667295).
ip	Sets the IP address of a standby group.
ip-address	IP address of a standby group.

netmask	Netmask of the standby group.
priority	Sets the priority of an interface for the standby group.
priority	Interface priority for the standby group (0–4294967295).

Defaults

No default behavior or values

Command Modes

Global configuration

Usage Guidelines

To display the interface identifiers (for example, interface FastEthernet 0/1), use the **show running-config** or **show startup-config** commands. The **autosense**, **bandwidth**, **fullduplex**, **halfduplex**, **ip**, and **shutdown** commands are listed separately in this command reference.

Examples

The following example configures an attribute of an interface with a single CLI command.

```
ContentEngine(config)# interface FastEthernet 0/1 half-duplex
```

An interface can be configured in a sequence of CLI commands as follows.

```
ContentEngine(config)# interface FastEthernet 0/1
ContentEngine(config-if): ?

Configure Interface commands:
  autosense    Interface autosense
  bandwidth    Interface bandwidth
  exit         Exit from this submode
  full-duplex  Interface fullduplex
  half-duplex  Interface halfduplex
  ip           Interface Internet Protocol Config commands
  no           Negate a command or set its defaults
  shutdown     Shutdown the specific interface
ContentEngine(config-if)# half-duplex
ContentEngine(config-if)# exit
ContentEngine(config)#
```

The following example enables a shut down interface.

```
ContentEngine(config)# no interface FastEthernet 0/1 shutdown
```

Related Commands

show interface

show running-config

show startup-config

ip

To configure the Content Engine IP interface, use the **ip** interface configuration command. To disable this function, use the **no** form of this command.

ip address [*ip-address ip-subnet*] [**secondary**]

no ip address [*ip-address ip-subnet*] [**secondary**]

Syntax Description

address	Sets the IP address of an interface.
ip-address	(Optional) IP address.
ip-subnet	(Optional) IP subnet mask.
secondary	(Optional) Makes this IP address a secondary address.

Defaults

No default behavior or values

Command Modes

Interface configuration

Usage Guidelines

Use this command to set or change the IP address and subnet mask of the Content Engine network interfaces. The Content Engine requires a reboot for the new IP address to take effect.

The **ip address** interface configuration command allows configuration of secondary IP addresses for a specified interface as follows.

```
Console(config)# ip address ip_address netmask [secondary]
```

Up to four secondary IP addresses can be specified for each interface. The same IP address cannot be assigned to more than one interface. The secondary IP address becomes active only after a primary IP address is configured. The following command configures the primary IP address.

```
Console(config)# ip address ip_address netmask
```

The secondary IP addresses are disabled when the interface is shut down, and are enabled when the interface is brought up. Use the **no** form of the command to disable a specific IP address.

```
Console(config)# no ip address ip_address netmask
```

Note No two interfaces can have IP addresses in the same subnet.

Examples

```
ContentEngine(config-if)# ip address 10.10.10.10 255.0.0.0

ContentEngine(config-if)# no ip address
```

ip

To change initial network device configuration settings, use the **ip** global configuration command. To delete or disable these settings, use the **no** form of this command. The **dscp** option allows you to set the global type of service (ToS) or differentiated services code point (DSCP) values in IP packets.

ip default-gateway *ip-address*

ip domain-name *name*

ip dscp {**client** {**cache-hit** {**match-server** | **set-dscp** *dscp-packets* | **set-tos** *tos-packets*} | **cache-miss** {**match-server** | **set-dscp** *dscp-packets* | **set-tos** *tos-packets*}} | **server** {**match-server** | **set-dscp** *dscp-packets* | **set-tos** *tos-packets*}}

ip name-server {*ip-addresses* | *serial-lookup*}

ip route *dest_addrs netmask gateway*

no ip {**default-gateway** *ip-address* | **domain-name** *name* | **dscp** {**client** {**cache-hit** {**match-server** | **set-dscp** *dscp-packets* | **set-tos** *tos-packets*} | **cache-miss** {**match-server** | **set-dscp** *dscp-packets* | **set-tos** *tos-packets*}} | **server** {**match-server** | **set-dscp** *dscp-packets* | **set-tos** *tos-packets*}} | **name-server** {*ip-addresses* | *serial-lookup*} | **route** *dest_addrs netmask gateway*}

Syntax Description

default-gateway	Specifies the default gateway (if not routing IP).
ip-address	IP address of default gateway.
domain-name	Specifies the domain name.
name	Domain name.
dscp	Configures IP differentiated services code point (DSCP) and type of service ToS fields.
client	Configures for responses to client.
cache-hit	Cache hit responses to client.
cache-miss	Cache miss responses to client.
server	Configures outgoing requests.
match-server	Uses the original ToS/DSCP value of server.
set-dscp	Configures differentiated services code point (DSCP) values.
dscp-packets:	0–63—Sets DSCP values:
af11	Sets packets with AF11 DSCP (001010).
af12	Sets packets with AF12 DSCP (001100).
af13	Sets packets with AF13 DSCP (001110).
af21	Sets packets with AF21 DSCP (010010).
af22	Sets packets with AF22 DSCP (010100).
af23	Sets packets with AF23 DSCP (010110).
af31	Sets packets with AF31 DSCP (011010).
af32	Sets packets with AF32 DSCP (011100).
af33	Sets packets with AF33 DSCP (011110).

af41	Sets packets with AF41 DSCP (100010).
af42	Sets packets with AF42 DSCP (100100).
af43	Sets packets with AF43 DSCP (100110).
cs1	Sets packets with CS1 (precedence 1) DSCP (001000).
cs2	Sets packets with CS2 (precedence 2) DSCP (010000).
cs3	Sets packets with CS3 (precedence 3) DSCP (011000).
cs4	Sets packets with CS4 (precedence 4) DSCP (100000).
cs5	Sets packets with CS5 (precedence 5) DSCP (101000).
cs6	Sets packets with CS6 (precedence 6) DSCP (110000).
cs7	Sets packets with CS7 (precedence 7) DSCP (111000).
default	Sets packets with default DSCP (000000).
ef	Sets packets with EF DSCP (101110).
set-tos	Configures Type of Service (ToS).
tos-packets:	0–127—Sets ToS value:
critical	Sets packets with critical precedence (80).
flash	Sets packets with flash precedence (48).
flash-override	Sets packets with flash override precedence (64).
immediate	Sets packets with immediate precedence (32).
internet	Sets packets with internetwork control precedence (96).
max-reliability	Sets packets with max reliable ToS (2).
max-throughput	Sets packets with max throughput ToS (4).
min-delay	Sets packets with min delay ToS (8).
min-monetary-cost	Sets packets with min monetary cost ToS (1).
network	Sets packets with network control precedence (112).
normal	Sets packets with normal ToS (0).
priority	Sets packets with priority precedence (16).
name-server	Specifies the address of name server.
ip-addresses	IP addresses of name servers (up to a maximum of 8).
serial-lookup	Queries each of the configured name servers iteratively if primary server responds in the negative.
route	Specifies net route.
dest_addrs	Destination route address.
netmask	Netmask.
gateway	Gateway address.

Defaults

No default behavior or values

Command Modes

Global configuration

Usage Guidelines

To define a default gateway, use the **ip default-gateway** global configuration command. To delete the IP default gateway, use the **no** form of this command.

The Content Engine uses the default gateway to route IP packets when there is no specific route found to the destination.

To define a default domain name, use the **ip domain-name** global configuration command. To remove the IP default domain name, use the **no** form of this command.

The Content Engine appends the configured domain name to any IP host name that does not contain a domain name. The appended name is resolved by the DNS server and then added to the host table. The Content Engine must have at least one domain name server specified for the host name resolution to work correctly. Use the **ip name-server** *hostname* command to specify domain name servers.

To specify the address of one or more name servers to use for name and address resolution, use the **ip name-server** global configuration command. To disable IP name servers, use the **no** form of this command.

For proper resolution of host name to IP address or IP address to host name, the Content Engine uses DNS servers. Use the **ip name-server** command to point the Content Engine to a specific DNS server. You can configure up to eight servers.

To configure static IP routing, use the **ip route** global configuration command. To disable an IP routing, use the **no** form of this command.

Use the **ip route** command to add a specific static route for a network host. Any IP packet designated for the specified host uses the configured route.

Examples

```
ContentEngine(config)# ip default-gateway 192.168.7.18

ContentEngine(config)# no ip default-gateway

ContentEngine(config)# ip route 172.16.227.128 ffffff80 172.16.227.250

ContentEngine(config)# no ip route 172.16.227.128 ffffff80 172.16.227.250

ContentEngine(config)# ip domain-name cisco.com

ContentEngine(config)# no ip domain-name

ContentEngine(config)# ip name-server 10.11.12.13

ContentEngine(config)# no ip name-server 10.11.12.14
```

Related Commands

show ip routes

ldap server

To configure the Content Engine to perform user authentication with a Lightweight Directory Access Protocol (LDAP) server, use the **ldap** global configuration command. To disable **ldap** options, use the **no** form of this command.

ldap server {**administrative-dn** *name* | **administrative-passwd** *passwd* | **allow-mode** | **base** *baseword* | **enable** | **filter** *filterword* | **host** {*hostname* | *hostipaddress*} [**primary** | **secondary**] | **port** *portnumber* | **timeout** *seconds* | **retransmit** *retries* | **userid-attribute** *useidword* | **version** *number*}

no ldap server {**administrative-dn** *name* | **administrative-passwd** *passwd* | **allow-mode** | **base** *baseword* | **enable** | **filter** *filterword* | **host** {*hostname* | *hostipaddress*} [**primary** | **secondary**] | **port** *portnumber* | **timeout** *seconds* | **retransmit** *retries* | **userid-attribute** *useidword* | **version** *number*}

Syntax Description

server	Configures LDAP server parameters.
administrative-dn	Sets the administrative distinguished name.
name	Administrative distinguished name.
administrative-passwd	Sets the administrative password.
passwd	Administrative password.
allow-mode	Allows access to users when the LDAP server is unavailable.
base	Sets the base distinguished name of the starting point for the search in the LDAP database.
baseword	Base value. There is no default.
enable	Enables HTTP request authentication with the LDAP server.
filter	Sets the LDAP filter for the authentication group.
filterword	Text for the LDAP filter. There is no default.
host	Sets host parameters.
hostname	Host name of the LDAP server. Two servers can be named.
hostipaddress	IP address of the LDAP server.
primary	(Optional) Specifies the host as the primary host.
secondary	(Optional) Specifies the host as the secondary host.
port	Sets the TCP port for the LDAP authentication server.
portnumber	LDAP server port number (1–65535). The default is 389.
timeout	Sets the time to wait for an LDAP server to reply.
seconds	Waiting time in seconds (1–20). The default is 5 seconds; minimum is 1 second; maximum is 20 seconds.
retransmit	Specifies the number of transmission attempts to an active server.
retries	Number of transmission attempts for a transaction (1–3). The default is 2.
userid-attribute	Sets the user ID attribute on the LDAP server.
useidword	Value for the user ID attribute. The default is “uid.”
version	Sets the LDAP version number.
number	LDAP version number (2–3). The default is 2.

Defaults

version *number*: 2

allow-mode: enabled

port *portnumber*: 389

timeout *seconds*: 5

useidword: uid

version *number*: 2

Usage Guidelines

System administrators can now use the Content Engine to restrict user Internet access using a Lightweight Directory Access Protocol (LDAP) server for authentication purposes, which provides most of the services of the X.500 protocol with less complexity and overhead.

Use the **ldap** global configuration command to enable LDAP authentication. Use the **no** form of the command to disable LDAP functions. An LDAP-enabled Content Engine authenticates users with an LDAP server. With an HTTP query, the Content Engine obtains a set of credentials from the user (user ID and password) and compares them against those in an LDAP server.

ACNS 4.1 software supports LDAP version 2 and version 3 and supports all LDAP features except for Secure Authentication and Security Layer (SASL).

Note The HTTP authentication featuring RADIUS and LDAP existed in Cache software 2.x releases and were configured through the **radius-server** and **ldap** commands, respectively. For ACNS 4.1 software, the **radius-server authtimeout** option and the **ldap authcache max-entries** and **ldap authcache auth-timeout** options have been removed and are now configurable through the **http authentication cache max-entries** and **timeout** commands, respectively. The **ldap client auth-header** option has been removed and is now configurable through the **http authentication header** command. The **multi-user-prompt** has been removed and replaced by the **http avoid-multiple-user-prompts** option. In addition, the **radius-server** command options **exclude** has been removed. The **rule no-auth domain** command replaces **radius-server exclude**; however, there is no replacement available for the **multi-user-prompt** option. The **ldap server** command has the following added options: **enable** and **version**.

An LDAP-enabled Content Engine authenticates user login and HTTP requests with an LDAP server. With an HTTP query, the Content Engine obtains a set of credentials from the user (user ID and password) and compares them against those in an LDAP server.

All LDAP Version 3 features are supported except for Secure Authentication and Security Layer (SASL).

Proxy Mode LDAP Authentication

The events listed below occur when the Content Engine is configured for LDAP authentication and one of the following two scenarios is true:

- The Content Engine receives a proxy-style request from a client.
- The Content Engine receives a transparent (WCCP-style) request from a client and the Content Engine **http authentication header** command parameter is set to 407 (because there is an upstream proxy).

1. The Content Engine examines the HTTP headers of the client request to find user information (contained in the Proxy-Authorization header).
2. If no user information is provided, the Content Engine returns a 407 (Proxy Authorization Required) message to the client.

3. The client resends the request, including the user information.
4. The Content Engine searches its authentication cache (based on user ID and password) to see if the client has been previously authenticated.
5. If a match is found, the request is serviced normally.
6. If no match is found, the Content Engine sends a request to the LDAP server to find an entry for this client.
7. If the server finds a match, the Content Engine allows the request to be serviced normally and stores the client user ID and password in the authentication cache.
8. If no match is found, the Content Engine again returns a 407 (Proxy Authorization Required) message to the client.

Transparent Mode LDAP Authentication

The events listed below occur when the Content Engine is configured for LDAP authentication and both of the following are true:

- The Content Engine receives a redirected request from a client.
- The **http authentication header** command parameter is set to 401 (because there is no upstream proxy).

1. The Content Engine searches its authentication cache to see if the user's IP address has been previously authenticated.
2. If a match is found, the Content Engine allows the request to be serviced normally.
3. If no match is found in the first step, the Content Engine examines the HTTP headers to find user information (contained in the Authorization header).
4. If no user information is provided, the Content Engine returns a 401 (Unauthorized) message to the client.
5. The client resends the request, including the user information.
6. The Content Engine sends a request to the LDAP server to find an entry for this user.
7. If the server finds a match, the Content Engine allows the request to be serviced normally and stores the client IP address in the authentication cache.
8. If no match is found, the Content Engine again returns a 401 (Unauthorized) message to the client.

In transparent mode, the Content Engine uses the client IP address as a key for the authentication database.

If you are using LDAP user authentication in transparent mode, we recommend that the AuthTimeout interval configured with the **http authentication cache timeout** command be short. IP addresses can be reallocated, or different users can access the Internet through an already authenticated device (PC, workstation, and the like). Shorter AuthTimeout values help reduce the possibility that individuals can gain access using previously authenticated devices. When the Content Engine operates in proxy mode, it can authenticate the user with the user ID and password.

Security Options

The Content Engine uses simple (nonencrypted) authentication to communicate with the LDAP server. Future expansion may allow for more security options based on Secure Socket Layer (SSL), SASL, or certificate-based authentication.

Excluding Domains

To exclude domains from LDAP authentication, use the **rule no-auth domain** command. Authentication challenges from LDAP, RADIUS, TACACS+, or SSH take place only if the request does not match the specified **no-auth** pattern.

LDAP and RADIUS Considerations

LDAP authentication can be used with Websense and N2H2 URL filtering, but not with RADIUS authentication. Both LDAP and RADIUS rely on different servers, which may require different user IDs and passwords, making RADIUS and LDAP authentication schemes mutually exclusive. Should both RADIUS and LDAP be configured on the Content Engine at the same time, LDAP authentication is executed, not RADIUS authentication.

Hierarchical Caching

In some cases, users are located at branch offices. A Content Engine (CE1) can reside with them in the branch office. Another Content Engine (CE2) can reside upstream, with an LDAP server available to both Content Engines for user authentication.

Note The **http append proxy-auth-header** global configuration command must be configured on the downstream Content Engines to ensure that proxy-authorization information, required by upstream Content Engines, is not stripped from the HTTP request by the downstream Content Engines. Up to 16 upstream IP addresses can be configured on each downstream Content Engine.

If branch office user 1 accesses the Internet, and content is cached at CE1, then this content cannot be served to any other branch office user unless that user is authenticated. CE1 must authenticate the local users.

Assuming that both CE1 and CE2 are connected to the LDAP server and authenticate the users, when branch office user 2 firsts requests Internet content, CE1 responds to the request with an authentication failure response (either HTTP 407 if in proxy mode, or HTTP 401 if in transparent mode). User 2 enters the user ID and password, and the original request is repeated with the credentials included. CE1 contacts the LDAP server to authenticate user 2.

Assuming authentication success, and a cache miss, the request along with the credentials is forwarded to CE2. CE2 also contacts the LDAP server to authenticate user 2. Assuming success, CE2 either serves the request out of its cache or forwards the request to the origin server.

User 2 authentication information is now stored in the authentication cache in both CE1 and CE2. Neither CE1 nor CE2 needs to contact the LDAP server for user 2's subsequent requests (unless user 2's entry expires and is removed from the authentication cache).

This scenario assumes that CE1 and CE2 use the same method for authenticating users. Specifically, both Content Engines must expect the user credentials (user ID and password) to be encoded in the same way.

Hierarchical Caching in Transparent Mode

When the Content Engine operates in transparent mode, the user IP address is used as a key to the authentication cache. When user 2 sends a request transparently to CE1, after authentication, CE1 inserts its own IP address as the source for the request. Therefore, CE2 cannot use the source IP address as a key for the authentication cache.

When CE1 inserts its own IP address as the source, it must also insert an X-Forwarded-For header in the request (**http append x-forwarded-for-header** command). CE2 must first look for an X-Forwarded-For header. If one exists, that IP address must be used to search the authentication cache. Assuming the user is authenticated at CE2, then CE2 must not change the X-Forwarded-For header, just in case there is a transparent CE3 upstream.

In this scenario, if CE1 does not create an X-Forwarded-For header (for example, if it is not a Cisco Content Engine and does not support this header), then authentication on CE2 will not work.

Hierarchical Caching, Content Engine in Transparent Mode with an Upstream Proxy

In a topology with two Content Engines, assume that CE1 is operating in transparent mode and CE2 is operating in proxy mode, with the browsers of all users pointing to CE2 as a proxy.

Because the browsers are set up to send requests to a proxy, an HTTP 407 message is sent from CE1 back to each user to prompt for credentials. By using the 407 message, the problem of authenticating based on source IP address is avoided. The username and password can be used instead.

This mode provides better security than using the HTTP 401 message. The Content Engine examines the style of the address to determine whether there is an upstream proxy. If there is, the Content Engine uses an HTTP 407 message to prompt the user for credentials even when operating in transparent mode.

Server Redundancy

Two LDAP servers can be specified with the **ldap server host command** to provide redundancy and improved throughput. Content Engine load-balancing schemes distribute the requests to the servers. If the Content Engine cannot connect to either server, no authentication can take place, and users who have not been previously authenticated are denied access.

Authentication Cache Size Adjustments

If the authentication cache is not large enough to accommodate all authenticated users at the same time, the Content Engine purges older entries that have not yet timed out.

Transaction Logging

After a user has been authenticated through LDAP, all transaction logs generated by the Content Engine for that user contain user information. If the Content Engine is acting in proxy mode, the user ID is included in the transaction logs. If the Content Engine is acting in transparent mode, the user IP address is included instead.

If the **transaction-logs sanitize** command is invoked, the user information is suppressed.

Examples

This example specifies an LDAP server with IP address 10.1.1.1 on port 88, and excludes the domain name, mydomain.net, from LDAP authentication with the **rule** global configuration command.

```
Console(config)# ldap server enable
Console(config)# ldap server host 10.1.1.1 port 88
Console# show ldap
LDAP parameters:
        State:           Enabled
        Base DN:         <none>
        Filter:          <none>
        Timeout:         5 seconds
        UID Attribute:   uid
        Primary:        10.1.1.1
        Secondary:      <none>
        LDAP port:      88
        Administrative DN:      <none>
        Administrative Password:        <none>
```

```
        LDAP version: 2
Console(config)# rule enable
Console(config)# rule no-auth domain mydomain.net

Console# show rule all
Rules Template Configuration
----------------------------
Rule Processing Enabled
rule no-auth domain mydomain.net
```

To delete an LDAP server, use the **no ldap server** command.

```
Console(config)# no ldap server host 10.1.1.1 port 88
```

Related Commands

rule

show ldap

show rule

debug authentication http-request

lls

To view a long list of directory names, use the **lls** EXEC command.

lls [*directory*]

Syntax Description

directory	(Optional) Name of the directory for which you want a long list of files.

Defaults

No default behavior or values

Command Modes

EXEC

Usage Guidelines

This command provides detailed information about files and subdirectories stored in the present working directory (including size, date, time of creation, sysfs name, and long name of the file). This information can also be viewed with the **dir** command.

Examples

```
ContentEngine# lls
     size          time of last change            name
--------------  -------------------------         -----------
             0  Tue Mar 18 01:52:41 1980          dir
       1959099  Tue Mar 18 01:52:41 1980          errlog-cache-19800317-171249
            62  Mon Mar 17 17:12:33 1980          errlog-dataserver-19800317-17
          1233
        439190  Tue Mar 18 01:52:34 1980          errlog-webserver-19800317-171
           243
         26758  Tue Mar 18 01:47:25 1980          syslog.txt
             0  Tue Mar 18 01:52:21 1980          testee
             0  Tue Mar 18 01:52:10 1980          tester
```

Related Commands

dir

ls

logging

To configure system logging, use the **logging** global configuration command. To disable logging functions, use the **no** form of this command.

logging {**console** {**enable** | **priority** *loglevel*} | **cw2K** | **disk** {**enable** | **filename** *filename* | **priority** *loglevel* | **recycle** *size*} | **facility** *facility* | **host** {*ip-address* | **priority** *loglevel*}}

no logging {**console** {**enable** | **priority** *loglevel*} | **cw2K** | **disk** {**enable** | **filename** *filename* | **priority** *loglevel* | **recycle** *size*}| **facility** *facility* | **host** {*ip-address* | **priority** *loglevel*}}

Syntax Description

console	Sets log to console.
enable	Enables log to a console.
priority	Sets which priority level messages to log.
loglevel	Use one of the following keywords:
• **alert**	Immediate action needed. Priority 1.
• **critical**	Immediate action needed. Priority 2.
• **debug**	Debugging messages. Priority 7.
• **emergency**	System is unusable. Priority 0.
• **error**	Error conditions. Priority 3.
• **information**	Informational messages. Priority 6.
• **notice**	Normal but significant conditions. Priority 5.
• **warning**	Warning conditions. Priority 4.
cw2k	Logs messages in CiscoWorks2000 format. This command is only effective when used in conjunction with the **logging host** command.
disk	Sets log to disk file.
enable	Enables log to disk file.
filename	Sets log filename.
filename	Name of the log file.
recycle	Overwrites syslog.txt when it surpasses the recycle size.
size	Size of syslog file in bytes (1000000–50000000).
facility	Sets facility parameter for syslog messages.
facility	Use one of the following keywords:
• **auth**	Authorization system.
• **daemon**	System daemons.
• **kernel**	Kernel.
• **local0**	Local use.
• **local1**	Local use.
• **local2**	Local use.
• **local3**	Local use.
• **local4**	Local use.
• **local5**	Local use.
• **local6**	Local use.

• **local7**	Local use.
• **mail**	Mail system.
• **news**	USENET news.
• **syslog**	Syslog itself.
• **user**	User process.
• **uucp**	UUCP system.
host	Sets log to a host.
ip-address	Host IP address.

Defaults

Logging: on

Priority of message for console: warning

Priority of message for file: debug

Log file: /local1/var/log/syslog.txt

Log file recycle size: 10,000,000 bytes

Command Modes

Global configuration

Usage Guidelines

Use this command to set specific parameters of the system log file. System logging is always enabled internally. The system log file is located on the sysfs partition as /local1/syslog.txt. To configure the Content Engine to send varying levels of event messages to an external syslog host, use the **logging host** option. Logging can be configured to send various levels of messages to the console using the **logging console priority** option.

The RealProxy generates error messages and writes them to the RealProxy log file. These error messages are captured by the Cache software and passed to the system log file. There is a one-to-one mapping between the RealProxy error codes and the syslog priority levels, as shown in Table 2-2.

Table 2-2 Mapping of RealProxy Error Level to syslog Priority Level

RealProxy Error Code	RealProxy Condition	RealProxy Usage	syslog Priority Level
0	Panic	Error potentially causing a system failure. RealSystem takes actions necessary to correct the problem.	Priority 0—LOG_EMERG, Emergency. System is unusable.
1	Severe	Error requiring immediate user intervention to prevent a problem.	Priority 1—LOG_ALERT, Alert. Immediate action needed.
2	Critical	Error that may require user intervention to correct.	Priority 2—LOG_CRI, Critical. Critical conditions.
3	General	Error that does not cause a significant problem with normal system operation.	Priority 3—LOG_ERR, Error. Error conditions.

Table 2-2 Mapping of RealProxy Error Level to syslog Priority Level (continued)

RealProxy Error Code	RealProxy Condition	RealProxy Usage	syslog Priority Level
4	Warning	Warning about a condition that does not cause system problems but may require attention.	Priority 4—LOG_WARNING Warning. Warning conditions.
5	Notice	Notice about a condition that does not cause system problems but should be noted.	5—LOG_NOTICE Notice. Normal but significant conditions.
6	Informational	Informational message only.	6—LOG_INFO Information. Informational messages.
7	Debug	Information of use only when debugging a program.	7—LOG_DEBUG Debug. Debugging messages.

The **no logging disk recycle size** command sets the file size to the default value. Whenever the current log file size surpasses the recycle size, the log file is rotated. The log file cycles through at most five rotations, and they are saved as [*log file name*].[*1-5*] under the same directory as the original log. The rotated log file is the one configured using the **logging disk filename** command.

Examples

```
ContentEngine(config)# logging console priority warnings

ContentEngine(config)# no logging console warnings
```

Related Commands

clear logging

show logging

ls

To view a list of files or subdirectory names within a directory, use the **ls** EXEC command.

ls [*directory*]

Syntax Description

directory	(Optional) Name of the directory for which you want a list of files.

Defaults

No default behavior or values

Command Modes

EXEC

Usage Guidelines

To list the filenames and subdirectories within a particular directory, use the **ls** *directory* command; to list the filenames and subdirectories of the current working directory, use the **ls** command. To view the present working directory, use the **pwd** command.

Examples

```
ContentEngine# ls
/local1

ContentEngine# ls /local1
core_dir
crash
errorlog
logs
lost+found
service_logs
smartfilter
syslog.txt
```

Related Commands

dir

lls

pwd

mediafs-division

To divide the media file system (mediafs) space percentage between the WMT cache and the RealProxy cache, use the **mediafs-division** global configuration command.

mediafs | mediafs-division {**wmt-cache-space** *percent_space* **real-cache-space** *percent_space*}

no mediafs | mediafs-division {**wmt-cache-space** *percent_space* **real-cache-space** *percent_space*}

Syntax Description

mediafs-division	Divides the media file system space between the WMT cache and the RealProxy cache.
wmt-cache-space	Defines the percentage of media file system space allocated to the WMT cache.
percent_space	Percentage of the cache allocated to WMT (0–100).
real-cache-space	Defines the percentage of media file system space allocated to the RealProxy cache.
percent_space	Percentage of the cache allocated to RealProxy (0–100).

Defaults

No default behavior or values

Command Modes

Global configuration

Usage Guidelines

Use this command to allocate the total media file system cache space between WMT and RealProxy on a percentage basis. The total combined media file system cache space divided between WMT and RealProxy equals 100 percent.

Examples

```
ContentEngine# mediafs-division wmt-cache-space 34 real-cache-space 66
```

Related Commands

mediafs

mkdir

To create a directory, use the **mkdir** EXEC command.

mkdir *directory*

Syntax Description		
	directory	Name of the directory to create.

Defaults

No default behavior or values

Command Modes

EXEC

Usage Guidelines

Use this command to create a new directory or subdirectory in the Content Engine file system.

Examples

```
ContentEngine# mkdir /oldpaxfiles
```

Related Commands

dir
lls
ls
pwd
rmdir

mkfile

To create a new file, use the **mkfile** EXEC command.

mkfile *filename*

Syntax Description

filename	Name of the file you want to create.

Defaults

No default behavior or values

Command Modes

EXEC

Usage Guidelines

Use this command to create a new file in any directory of the Content Engine.

Examples

```
ContentEngine# mkfile traceinfo
```

Related Commands

lls

ls

mkdir

multicast-client

To configure multicast client options, use the **multicast-client** command in global configuration mode.

multicast-client {**accept-license-agreement** | **enable** | **evaluate** | **license-key** *key*}

no multicast-client {**accept-license-agreement** | **enable** | **evaluate** | **license-key** *key*}

Syntax Description

accept-license-agreement	Accepts multicast client license agreement.
enable	Enables multicast client.
evaluate	Starts or continues the 30-day evaluation period of multicast client.
license-key	Requires license key for multicast client.
key	Multicast client license key parameters.

Defaults

The default evaluation period for the multicast client license is 30 days.

Command Modes

Global configuration

Examples

```
ContentEngine# multicast-client accept-license-agreement
ContentEngine# multicast-client enable
ContentEngine# multicast-client evaluate
ContentEngine# multicast-client license-key 123456789
```

Related Commands

show multicast-license agreement

no

To negate an interface configuration command or set its defaults, use the **no** interface configuration command.

no {**autosense** | **bandwidth** | **cdp** | **fullduplex** | **halfduplex** | **ip** | **shutdown** | **standby**}

Syntax Description

autosense	Autosense capability on an interface.
bandwidth	Interface speed.
cdp	Configures Cisco Discovery Protocol interface.
fullduplex	Full-duplex interface.
halfduplex	Half-duplex interface.
ip	Interface Internet Protocol (IP) configuration commands.
shutdown	Shuts down the specific interface.
standby	Standby interface configuration commands.

Defaults

No default behavior or values

Command Modes

Interface configuration

Usage Guidelines

Use this command to negate an interface configuration mode command or set its defaults. See the individual commands for syntax options and descriptions.

Examples

```
ContentEngine(config-if)# no autosense
```

no

To undo a global configuration command or set its defaults, use the **no** form of a global configuration command to undo the original command.

no *command*

Syntax Description

asset	Configures CISCO-ENTITY-ASSETT-MIB.
authentication	Configures authentication.
boomerang	Configures boomerang agent.
bypass	Configures bypass.
cdp	Configures Cisco Discovery Protocol (CDP).
clock	Configures time-of-day clock.
dns-cache	Configures DNS cache.
ecdn	Configures Enterprise Content Delivery Network (E-CDN).
error-handling	Customizes how Content Engine should handle errors.
exception	Configures exceptions.
exec-timeout	Configures exec timeout.
external-ip	Configures up to eight external (NAT) IP addresses.
ftp	Configures FTP caching-related parameters.
gui-server	Configures GUI server.
help	Configures assistance for command-line interface.
hostname	Configures the system's network name.
http	Configures HTTP-related parameters.
https	Configures HTTPS-related parameters.
icp	Configures Internet Cache Protocol (ICP) parameters.
interface	Configures a Fast Ethernet or Gigabit Ethernet interface.
ip	Internet Protocol (IP) configuration commands.
ldap	Configures LDAP parameters.
logging	Configures system logging (syslog).
mediafs-division	Configures the media file system space allocation for the Windows Media Technologies (WMT) cache and the RealProxy cache.
multicast-client	Configures multicast client parameters.
ntlm	Configures NTLM parameters.
ntp	Configures Network Time Protocol (NTP).
pre-load	Configures content preloading.
primary-interface	Configures a primary interface.
proxy-auto-config	Configures browser proxy auto-configuration feature.
proxy-protocols	Configures proxy protocols-related parameters.
radius-server	Configures RADIUS server authentication.
real-subscriber	Configures RealSubscriber parameters.
rtsp	Configures Real-Time Streaming Protocol-related parameters.

rule	Configures Rules Template.
snmp-server	Configures SNMP.
ssh-key-generate	Generates Secure Shell (SSH) host key.
sshd	Configures Secure Shell service.
tacacs	Configures TACACS+ authentication.
tcp	Configures global TCP parameters.
transaction-logs	Configures transaction logging.
url-filter	Configures URL filtering.
username	Establishes username authentication.
wccp	Configures Web Cache Communication Protocol.
wmt	Configures Windows Media Technologies (WMT) parameters.

Defaults

No default behavior or values

Command Modes

Global configuration

Usage Guidelines

Use the **no** command to disable functions or negate a command. If you need to negate a specific command, such as the default gateway IP address, you must include the specific string in your command, such as **no ip default-gateway** *ip-address*.

Examples

```
ContentEngine(config)# wccp version 2

ContentEngine(config)# no wccp version 2
```

ntlm server

To configure Microsoft Windows NT LAN Manager (NTLM) parameters, use the **ntlm** command in global configuration mode.

ntlm server {**domain** *name* | **enable** | **host** {*hostname* | *ip-address* [**primary** | **secondary**]}}

no ntlm server {**domain** *name* | **enable** | **host** {*hostname* | *ip-address* [**primary** | **secondary**]}}

Syntax Description

server	Configures NTLM server-related parameters.
domain	Specifies NTLM domain name.
name	Name of NTLM domain.
enable	Enables NTLM authentication.
host	Configures NTLM NT controller name or IP address.
hostname	Host name.
ip-address	Host IP address.
primary	(Optional) Sets selected host name or address as the primary.
secondary	(Optional) Sets selected host name or address as the secondary.

Defaults

No default behavior or values

Command Modes

Global configuration

Usage Guidelines

Use the **ntlm server** command to enable NTLM authentication and configure the NTLM server domain name, NT primary domain controller (PDC) name or IP address, and optionally set the host name or address as primary or secondary.

The NTLM protocol can be used to authenticate and block user access to the Internet. When a user logs in to a Windows NT or a Windows 2000 domain, the information is stored by the browser and later used as NTLM credentials to access the Internet. The browser sends the NTLM credentials with the domain name to the ACNS cache, which in turns sends a request to the Windows NT domain controller to check the validity of the user in the domain. If the user is not a valid user in the domain, then the request to access the Internet is denied. If authentication succeeds, the source IP address is entered in the authentication cache. Future requests from this IP address are not challenged until the authentication cache entry expires, or is cleared.

Before invoking an NTLM authentication request, make sure that the following conditions exist.

- The NTLM primary domain controller has an entry in the Domain Name System (DNS) that matches its NetBIOS-named computer account.
- The primary domain controller is both forward and reverse DNS-resolvable.
- The domain name configured on the Content Engine matches the domain of which the primary domain controller is a part.

In the following example, server1 must be in the cisco.com domain and must have an entry in DNS that matches its NetBIOS-named computer account.

```
ip domain-name cisco.com
ntlm server host server1
```

For clients within the domain using the Internet Explorer browser in proxy mode, authentication is "popless"; that is, the user is not prompted with a dialog box to enter a username and password. In transparent mode, authentication is transparent only if the Internet options security settings are customized and set to **User Authentication > Logon > Automatic logon with current username and password**.

For clients outside the domain using the Netscape browser, a dialog box appears and the first authentication request asks the client to enter a username and password. Once the client is successfully authenticated, the entry is placed in the cache, and no reauthentication requests are made to the client until the entry lease expires.

Examples

This example configures a Content Engine for NTLM request authentication and blocking.

```
ContentEngine(config)# ntlm server enable
ContentEngine(config)# ntlm server domain cisco_abc
ContentEngine(config)# ntlm server host 172.16.10.10 primary
ContentEngine(config)# ntlm server host 172.16.10.12 secondary
```

Related Commands

show ntlm

ntp

To configure the Network Time Protocol (NTP) and to allow the system clock to be synchronized by a time server, use the **ntp** global configuration command. To disable this function, use the **no** form of this command.

ntp {**server** {*hostname* | *ip-address*} | **enable** {**cdm** | **manual**}

no {**server** {*hostname* | *ip-address*} | **enable** {**cdm** | **manual**}

Syntax Description

server	Configures the NTP Server host name or IP address.
hostname	Host name of the time server providing the clock synchronization (maximum of 4).
ip-address	IP address of the time server providing the clock synchronization (maximum of 4).
enable	Enables NTP services.
cdm	Configures NTP using the settings from the Content Distribution Manager.
manual	Configures the NTP Server IP address manually.

Defaults

The default NTP version number is 3.

Command Modes

Global configuration

Usage Guidelines

Use this command to synchronize the Content Engine clock with the specified server. The server does not synchronize to this machine. The **ntp enable** command configures the NTP settings. The three possible configurations are NTP is disabled, NTP is set to manual and uses IP addresses from the NTP server, or NTP settings match Content Distribution Manager settings. The third configuration allows the Content Distribution Manager to select the timekeeping method, which can be either NTP or to set the time on reboot. Currently, only the option to set the time upon reboot is supported in the E-CDN application, and the Content Distribution Manager does not yet announce NTP settings.

Examples

```
ContentEngine(config)# ntp server 172.16.22.44
ContentEngine(config)# no ntp server 172.16.22.44
```

Related Commands

clock

show clock

show ntp status

ntpdate

To set the software clock (time and date) using a Network Time Protocol (NTP) server, use the **ntpdate** EXEC command.

ntpdate {*hostname* | *ip-address*}

Syntax Description

hostname	NTP host name.
ip-address	NTP server IP address.

Defaults

No default behavior or values

Command Modes

EXEC

Usage Guidelines

Use NTP to find the current time of day and set the Content Engine current time to match. The time must be saved to the hardware clock using the **clock save** command if it is to be restored after a reload.

Examples

```
ContentEngine# ntpdate 10.11.23.40
```

Related Commands

clock save

clock set

show clocks

ping

To send echo packets for diagnosing basic network connectivity on networks, use the **ping** EXEC command.

ping {*hostname* | *ip-address*}

Syntax Description

hostname	Host name of system to ping.
ip-address	IP address of system to ping.

Defaults

No default behavior or values

Command Modes

EXEC

Usage Guidelines

To use this command with the *hostname* argument, be sure DNS functionality is configured on your Content Engine. To force the timeout of a nonresponsive host, or to eliminate a loop cycle, press **Ctrl-C**.

Examples

```
ContentEngine# ping mycacheengine
```

pre-load

To configure the Content Engine to fetch and preload content, use the **pre-load** global configuration command.

pre-load concurrent-requests *number*

pre-load depth-level-default *level_number*

pre-load enable

pre-load fetch {**directory** *dir_names* | **domain** *domain_names* | **suffix** *suffix_names*}

pre-load no-fetch {**directory** *dir_names* | **domain** *domain_names* | **suffix** *suffix_names*}

pre-load schedule every-day [**start-time** *time* [**end-time** *time*]]

pre-load schedule every-hour [**start-time** *time* [**end-time** *time*]]

pre-load schedule every-week *days of week* [**start-time** *time* [**end-time** *time*]]

pre-load traverse-other-domains

pre-load url-list-file *path*

no pre-load {**concurrent-requests** *number* | **depth-level-default** *level-number* | **enable** | **fetch** {**directory** *dir_names* | **domain** *domain_names* | **suffix** *suffix_names*} | **no-fetch** {**directory** *dir_names* | **domain** *domain_names* | **suffix** *suffix_names*} | **schedule** {**every-day** [**start-time** *time* [**end-time** *time*]] | **every-hour** [**start-time** *time* [**end-time** *time*]] | **every-week** {*days of week* [**start-time** *time* [**end-time** *time*]]}} | **traverse-other-domains** | **url-list-file** *path*}

Syntax Description

concurrent-requests	Configures the maximum number of concurrent requests.
number	Number of concurrent requests (1–100). The default is 25.
depth-level-default	Configures default depth level.
level_number	Depth level of URL download (1–50). The default is 1.
enable	Enables the preload feature.
fetch	Configures the filter for the objects to be fetched.
directory	Configures the directories to be fetched.
dir_names	List of directory names separated by spaces.
domain	Configures the domains to be fetched.
domain_names	List of domain names separated by spaces.
suffix	Configures the suffixes to be fetched
suffix_names	List of suffixes separated by spaces.
no-fetch	Configures the filter for the objects that should not be fetched.
directory	Configures the directories that should not be fetched.
dir_names	List of directory names separated by spaces.
domain	Configures the domains that should not be fetched.
domain_names	List of domain names separated by spaces.

suffix	Configures the suffixes that should not be fetched.
suffix_names	List of suffixes separated by spaces.
schedule	Configures the schedule time for preload.
every-day	Preloads in intervals of 1 day.
start-time	(Optional) Sets preload start time. The default is 00:00.
time	(Optional) Time of day to start the preload (00:00–23:59 in hh:mm format).
end-time	(Optional) Sets preload end time. The default is until the downloading of all the objects.
time	(Optional) Time of day to end the preload (00:00–23:59 in hh:mm format).
every-hour	Preloads in intervals of 1 hour or less.
start-time	(Optional) Sets preload start time. The default is 0.
time	(Optional) Minute of the hour to start the preload (0–59).
end-time	(Optional) Sets preload end time. The default is until the downloading of all the objects.
time	(Optional) Minute of the hour to end the preload (0–59).
every-week	Preloads in intervals of 1 week or less.
days of week	Adds one or more weekdays.
• **Fri**	Every Friday.
• **Mon**	Every Monday.
• **Sat**	Every Saturday.
• **Sun**	Every Sunday.
• **Thu**	Every Thursday.
• **Tue**	Every Tuesday.
• **Wed**	Every Wednesday.
start-time	(Optional) Sets preload start time. The default is 00:00.
time	(Optional) Time of day to start the preload (00:00–23:59 in hh:mm format).
end-time	(Optional) Sets preload end time. Default is till the downloading of all the objects.
time	(Optional) Time of day to end the preload (00:00–23:59 in hh:mm format).
traverse-other-domains	Allows other domains in an HTML page to be traversed.
url-list-file	Sets URL list file path.
path	Path of the file containing the URL list.

Defaults

concurrent-requests *number*: 25

depth-level-default *level_number*: 1

every-day: default

start-time *time*: 00:00

end-time *time*: until downloading of all objects has occurred

traverse-other-domains: other domains in an HTML page are not traversed by default

Command Modes

Global configuration

Usage Guidelines

Cisco ACNS software can read a file of URLs and download the specified content to the Content Engine. This preloading can be scheduled with the **pre-load schedule** global configuration command, or triggered immediately with the **pre-load force** EXEC command.

A list file (URL list) of URLs to be preloaded is maintained by the administrator. The URL list must be created on a remote system and transferred to a sysfs volume on the Content Engine. The path of the URL list is specified by the **pre-load url-list-file** option. Each URL in the list has an optional depth parameter. The depth parameter specifies how many levels down the preloading is performed. For example: http://www.espn.com 3 means download http://www.espn.com and all content three levels deep. If the depth level is not specified, then the CLI-configured depth level is used. The URLs are delimited with a carriage return as follows:

```
<cr>
. . .
http://www.cnn.com  3 <cr>
ftp://ftp.lehigh.edu/  2 <cr>
http://www.yahoo.com <cr>
. . .
<cr>
```

Whenever an administrator wants to preload authenticated content to the Content Engine, the URL list file entry must be as follows:

```
http://username:password@www.authenticatedsite.com/  <depth level>
```

Note For the Content Engine to cache authenticated content, HTTP cache authentication feature must be set using **http cache-authenticated** command.

Use the **pre-load schedule** command to specify the time intervals at which the preload event executes, or use the **pre-load force** EXEC command to launch a preload event at any time.

If the content to be preloaded is already available in the Content Engine, then the Content Engine revalidates the freshness of the stored copy.

A preload request process (wget) is spawned for every URL in the list. These processes operate concurrently. Use the **pre-load concurrent-requests** option to configure the maximum number of preload processes to run at the same time. If the number of URLs in the URL list file is less than the number of specified concurrent requests, then the lesser number is active.

All configured HTTP parameters and rules apply to the preloaded objects.

The **no pre-load enable** command stops the preload operation and is identical to the preload schedule end-time behavior, meaning that no new wget() sessions are started, and existing sessions are not terminated abruptly.

Examples

This example enables the preload feature.

```
ContentEngine(config)# pre-load enable
```

This example specifies the pathname of the preload URL list file.

```
ContentEngine(config)# pre-load url-list-file /local1/myurllist
```

This example specifies the depth level for URL retrieval at 4.

```
ContentEngine(config)# pre-load depth-default 4
```

This example specifies the filter for the objects that need to be excluded.

```
ContentEngine(config)# pre-load no-fetch suffix .mil .su .ca
```

This example specifies the filter for the domain to be fetched.

```
ContentEngine(config)# pre-load fetch domain cisco.com
```

This example allows other domains in a HTML page to be traversed (by default, other domains in an HTML page are not traversed).

```
ContentEngine(config)# pre-load traverse-other-domain
```

This example specifies the maximum number of concurrent connections.

```
ContentEngine(config)# pre-load concurrent-requests 5
```

This example specifies a daily interval for scheduling the preload event.

```
ContentEngine(config)# pre-load schedule every-day start-time 01:00 end-time 02:00
```

This example specifies an hourly interval for scheduling the preload event.

```
ContentEngine(config)# pre-load schedule every-hour start-time 8 end-time 20
```

The **pre-load schedule every-week** option permits configuring a preload event on more than one day of the week.

This example specifies a twice-weekly interval for scheduling the preload event.

```
ContentEngine#(config)# pre-load schedule every-week Sun Wed start-time 01:00 end-time
06:00
```

The default start time for the preloading operation is 00:00 (that is, the start of the day). If the end time is not specified, the preload operation is completed after all the objects have been downloaded.

The following are examples of preload-related **show** commands:

```
ContentEngine# show statistics pre-load
Statistics of last Preloading operation
---------------------------------------

Preloading was initiated by cron.
Preloading started at Sat Feb 10 21:00:01 2001
Preloading ended   at Sun Feb 11 00:45:25 2001

Number of invalid entries in URL list file  =          0
Total number of preloaded objects           =      44178
Total number of preloaded bytes             =  895723727


ContentEngine# show pre-load
Preloading is enabled
Number of concurrent sessions: 10
Depth level: 3
URL List File: /local1/preload/preload.txt
Preload will not traverse other domains.

Fetch Domains:
Fetch Suffix:
Fetch Directory:
No-fetch Domain:
No-Fetch Suffix:
No-Fetch Directory:
```

```
Scheduling on:
Sunday
        Start Time: 00:00
        End Time  : Till completion
```

Related Commands

pre-load force

show pre-load

show statistics pre-load

pre-load force

To force a preload operation, use the **pre-load force** command.

pre-load force

Syntax Description		
	force	Forces a preload operation.

Defaults

No default behavior or values

Command Modes

EXEC

Usage Guidelines

Use the **pre-load force** command to immediately begin a previously scheduled preload operation. Use the **no pre-load enable** global configuration command to stop a preload process in progress.

Examples

This example initiates a previously configured and scheduled preload operation and then terminates it after an interval of time.

```
ContentEngine# pre-load force
. . .
ContentEngine(config)# no pre-load enable
```

Related Commands

pre-load

show pre-load

show statistics pre-load

primary-interface

To configure the primary interface for the Enterprise CDN (E-CDN) application, use the **primary-interface** command in global configuration mode. Use the **no** form of the command to revert to the default primary interface.

primary-interface {**FastEthernet** | **GigabitEthernet**} *slot/port* [**dhcp**]

no primary-interface {**FastEthernet** | **GigabitEthernet**} *slot/port* [**dhcp**]

Syntax Description

FastEthernet	Selects a Fast Ethernet interface as the E-CDN primary interface.
GigabitEthernet	Selects a Gigabit Ethernet interface as the E-CDN primary interface.
slot	Slot number of the specified interface.
port	Port number of the specified interface.
dhcp	(Optional) Enables the Dynamic Host Configuration Protocol (DHCP) on the specified interface.

Defaults

The default primary interface is the first operational interface on which a link beat is detected. Interfaces with lower-number IDs are polled first (for example, FastEthernet 0/0 is checked before 1/0). For hardware with Gigabit Ethernet interfaces, the Gigabit Ethernet interfaces are polled before the Fast Ethernet interfaces.

Command Modes

Global configuration

Usage Guidelines

The **primary-interface** global configuration command permits the administrator to specify the primary interface for the E-CDN application.

Note If the E-CDN application is enabled without specifying the primary interface, the E-CDN application chooses a default interface as primary.

The primary interface can be changed without disabling the E-CDN application. To change the primary interface, reenter the command string and specify a different interface. To enable DHCP services with the specified interface, include the **dhcp** option.

Examples

```
Console(config)# primary-interface FastEthernet 0/0 dhcp
Console(config)# primary-interface FastEthernet 0/1
```

proxy-auto-config

To download the proxy automatic configuration file from an FTP server, use the **proxy-auto-config download** command in EXEC mode.

proxy-auto-config download {*ftp-hostname* | *ftp-ip-address*} *remotedir pacfile*

Syntax Description

download	Downloads and installs a configuration file from the FTP server.
ftp-hostname	Host name of FTP server.
ftp-ip-address	IP address of FTP server.
remotedir	Directory on the FTP server where the .pac file is located.
pacfile	Filename of the remote proxy autoconfiguration file.

Defaults

No default behavior or values

Command Modes

EXEC

Usage Guidelines

A browser obtains proxy IP address and port configuration information from the proxy automatic configuration file (.pac file) when the browser's automatic configuration URL field is configured with the Content Engine IP address, incoming port number, file directory, and .pac filename.

The **proxy-auto-config download** EXEC command downloads an automatic configuration file from an FTP server to the present working directory of the Content Engine.

Examples

This example demonstrates how to download an autoconfiguration file from an FTP server to the Content Engine.

```
Console# proxy-auto-config download 172.16.10.10 remotedirname theproxyfile.pac
```

This example shows the URL that you enter in the browser's automatic proxy configuration URL field.

http://CCNScache-ipaddress:portnumber/theproxyfile.pac

Related Commands

show proxy-auto-config

proxy-auto-config (global configuration mode)

proxy-auto-config

To enable the browser automatic configuration feature, use the **proxy-auto-config enable** command in global configuration mode. To disable the proxy autoconfiguration feature, use the **no** form of this command.

proxy-auto-config enable

no proxy-auto-config enable

Syntax Description

enable	Enables the automatic browser configuration feature.

Defaults

Proxy autoconfiguration is disabled by default.

Command Modes

Global configuration

Usage Guidelines

A browser obtains proxy IP address and port configuration information from the proxy automatic configuration (.pac) file when the browser's autoconfiguration URL field is configured with the Content Engine IP address, incoming port number, file directory, and .pac filename.

To enable the proxy automatic configuration file feature, enter the **proxy-auto-config enable** global configuration command. Each time you download a new autoconfiguration file to the Content Engine, enter a **no proxy-auto-config enable** and then a **proxy-auto-config enable** command.

The autoconfiguration feature is supported by Microsoft Internet Explorer and Netscape Communicator browsers. The browser must be manually configured for automatic proxy configuration.

Examples

The following example demonstrates how browser autoconfiguration is enabled on the Content Engine.

```
Console(config)# proxy-auto-config enable
```

The following example shows the URL that you enter in the browser automatic proxy configuration URL field.

http://*Content_Engine_ip_address:portnumber/theproxyfile.pac*

Note Use the port number specified by the **http proxy incoming** ***portnumber*** command for configuring proxy incoming ports. For instance, if port **8080** is specified with the **http proxy incoming 8080** command, then use 8080 as your port number in the example shown.

Related Commands

proxy-auto-config (EXEC mode)

show proxy-auto-config

proxy-protocols

Use the **proxy-protocols** global configuration command to specify a domain name, host name, or IP address to be excluded from proxy forwarding. To selectively turn off outgoing-proxy exclude lists or to force transparently received proxy-style requests to be fulfilled by the Content Engine, use the **no** form of this command.

proxy-protocols outgoing-proxy exclude {**enable** | **list** *word*}

proxy-protocols transparent {**default-server** | **original-proxy**}

no proxy-protocols {**outgoing-proxy exclude** {**enable** | **list** *word*} | **transparent** {**default-server** | **original-proxy**}}

Syntax Description

outgoing-proxy exclude	Sets global outgoing proxy exclude criteria.
enable	Enables global outgoing proxy exceptions.
list	Sets the global outgoing proxy exclude list.
word	Domain names, host names, or IP addresses to be excluded from proxy forwarding (supports 64 exclude list entries).
transparent	Sets transparent mode behavior for proxy requests.
default-server	Uses the Content Engine to go to the origin server or the outgoing proxy, if configured.
original-proxy	Uses the intended proxy server from the original request.

Defaults

No default behavior or values

Command Modes

Global configuration

Usage Guidelines

The **proxy-protocols outgoing-proxy exclude** option allows the administrator to specify a single domain name, host name, or IP address to be globally excluded from proxy forwarding. Domains are entered as an ASCII string, separated by spaces. The wildcard character * (asterisk) can be used for IP addresses (for instance, 174.12.*.*). Only one exclusion can be entered per command line. Enter successive command lines to specify multiple exclusions.

When you enter the **proxy-protocols transparent default-server** global configuration command, the Content Engine forwards intercepted HTTP, HTTPS, and FTP proxy-style requests to the corresponding outgoing proxy server, if one is configured. If no outgoing proxy server is configured for the protocol, the request is serviced by the Content Engine and the origin server.

The **proxy-protocols transparent original-proxy** option specifies that requests sent by a web client to another proxy server, but intercepted by the Content Engine in transparent mode, be directed back to the intended proxy server.

Examples

The following example configures the Content Engine to forward intercepted HTTPS proxy-style requests to an outgoing proxy server. The domain name cruzio.com is excluded from proxy forwarding. The **show proxy-protocols** command verifies the configuration.

```
ContentEngine(config)# https proxy outgoing host 172.16.10.10 266
ContentEngine(config)# proxy-protocols transparent default-server
ContentEngine(config)# proxy-protocols outgoing-proxy exclude cruzio.com

ContentEngine# show proxy-protocols all
Transparent mode forwarding policies: default-server
Outgoing exclude domain name: cruzio.com
```

The following example configures the Content Engine to forward intercepted HTTP proxy-style requests to the intended proxy server.

```
ContentEngine(config)# proxy-protocols transparent original-proxy
```

Related Commands

http proxy outgoing

https proxy outgoing

show proxy-protocols

pwd

To view the present working directory, use the **pwd** EXEC command.

pwd

Syntax Description

This command has no arguments or keywords.

Defaults

No default behavior or values

Command Modes

EXEC

Usage Guidelines

Use this command to display the present working directory of the Content Engine.

Examples

```
ContentEngine# pwd
```

Related Commands

cd

dir

lls

ls

radius-server

To configure RADIUS parameters, use the **radius-server** command in global configuration mode. To disable RADIUS authentication parameters, use the **no** form of this command.

radius-server {**enable** | **host** {*hostname* | *hostipaddr*} [**auth-port** *port*] | **key** *keyword* | **redirect** {**enable** | **message** *reply url*} | **retransmit** *retries* | **timeout** *seconds*}

no radius-server {**enable** | **host** {*hostname* | *hostipaddr*} [**auth-port** *port*] | **key** *keyword* | **retransmit** *retries* | **timeout** *seconds*}

Syntax Description

enable	Enables HTTP RADIUS authentication.
host	Specifies a RADIUS server.
hostname	Host name of RADIUS server.
hostipaddr	IP address of RADIUS server.
auth-port	Sets UDP port for RADIUS authentication server.
port	UDP port number (1–65535). The default is 1645.
key	Encryption key shared with the RADIUS servers.
keyword	Text of shared key (15 characters maximum).
redirect	Redirects response if authentication request fails.
enable	Enables redirect feature.
message	Replies with authentication failure message.
reply	Reply message text string (24 characters maximum).
url	URL destination of authentication failure instructions.
retransmit	Specifies the number of transmission attempts to an active server.
retries	Number of transmission attempts for a transaction (1–3). The default is 2.
timeout	Time to wait for a RADIUS server to reply.
seconds	Wait time in seconds (1–20). The default is 5 seconds.

Defaults

retransmit *retries*: 2

timeout *seconds*: 5

auth-port *port*: UDP port 1645

Command Modes

Global configuration

Usage Guidelines

RADIUS authentication clients reside on the Content Engine running ACNS software. When enabled, these clients send authentication requests to a central RADIUS server, which contains user authentication and network service access information.

To configure RADIUS parameters, use the **radius-server** command in global configuration mode. To disable RADIUS authentication parameters, use the **no** form of this command.

RADIUS Authentication Redirection

The **redirect** option of the **radius-server** command redirects an authentication response to a different authentication server if an authentication request using the RADIUS server fails.

Note The **rule** command is relevant to RADIUS only if **redirect** has been configured.

Excluding Domains

To exclude domains from RADIUS authentication, use the **rule no-auth domain** command. RADIUS authentication takes place only if the site requested does not match the specified pattern.

Examples

The following example enables the RADIUS client, specifies a RADIUS server, specifies the RADIUS key, accepts retransmit defaults, and excludes the domain name, mydomain.net, from RADIUS authentication. The configuration is verified with the **show radius-server** and **show rule all** commands.

```
Console(config)# radius-server enable
Console(config)# radius-server host 172.16.90.121
Console(config)# radius-server key myradiuskey
Console(config)# rule enable
Console(config)# rule no-auth domain mydomain.net

Console(config)# show radius-server
Radius Configuration:
---------------------
Radius Authentication is on
    Timeout       = 5
    Retransmit    = 3
    Key           = ****
    Servers
    -------
    IP 172.16.90.121 Port =   1645    State: ENABLED

Console# show rule all
Rules Template Configuration
----------------------------
Rule Processing Enabled
rule no-auth domain mydomain.net
```

The following example disables RADIUS authentication on the Content Engine.

```
Console(config)# no radius-server enable
```

Related Commands

rule

show radius

debug authentication http-request

no debug authentication http-request

real-subscriber

To configure RealSubscriber parameters, use the **real-subscriber** command in global configuration mode.

real-subscriber {**accept-license-agreement** | **enable** | **evaluate** | **license-key** *key* | **publisher** {*host-name* | *ip-address*} *admin-port-number user-name user-password*}

no real-subscriber {**accept-license-agreement** | **enable** | **evaluate** | **license-key** *key* | **publisher** {*host-name* | *ip-address*} *admin-port-number user-name user-password*}

Syntax Description

accept-license-agreement	Accepts RealSubscriber license agreement.
enable	Enables RealSubscriber.
evaluate	Starts or continues the 30-day evaluation period of RealSubscriber.
license-key	Requires licence key for RealSubscriber.
key	RealSubscriber license key.
publisher	Configures RealPublisher parameters.
host-name	Host name of RealPublisher.
ip-address	IP address of RealPublisher.
admin-port-number	Administration port number of RealPublisher (1–65535).
user-name	Username on RealPublisher.
user-password	User password on RealPublisher.

Defaults

Thirty-day evaluation period for the RealSubscriber license

Command Modes

Global configuration

Usage Guidelines

When **real-subscriber enable** is invoked while E-CDN is enabled, as shown in the following example, the following message appears:

```
Console(config)# real-subscriber enable

RealSubscriber server can only use up to 100Kbps of bandwidth initially. Please go to ECDN
bandwidth page in the CDM GUI to adjust it accordingly.
```

Examples

```
Console(config)# real-subscriber enable

RealSubscriber server can only use up to 100Kbps of bandwidth initially. Please go to ECDN
bandwidth page in the CDM GUI to adjust it accordingly.

Console(config)# real-subscriber accept-license-agreement
Console(config)# real-subscriber evaluate
Console(config)# real-subscriber license-key 1234567
Console(config)# real-subscriber publisher 209.165.200.224 12 Leonidas 654321
```

Related Commands

show real-subscriber

clear real-proxy-cache

clear statistics mediacache real

mediafs-division

rtsp proxy media-real

show statistics mediacache-real

reload

To halt and perform a cold restart on your Content Engine, use the **reload** EXEC command.

reload

Syntax Description

This command has no arguments or keywords.

Defaults

No default behavior or values

Command Modes

EXEC

Usage Guidelines

To reboot the Content Engine, use the **reload** command. If no configurations are saved to Flash memory, you are prompted to enter configuration parameters upon restart. Any open connections are dropped after you issue this command, and the file system is reformatted upon restart. To save any file system contents to disk from memory before a restart, use the **cache synchronize** command.

Examples

```
ContentEngine# reload
```

Related Commands

cache synchronize

write

write erase

rmdir

To delete a directory, use the **rmdir** EXEC command.

rmdir *directory*

Syntax Description

directory	Name of the directory you want to delete.

Defaults

No default behavior or values

Command Modes

EXEC

Usage Guidelines

Use this command to remove any directory from the Content Engine file system. The **rmdir** command only removes empty directories.

Examples

```
ContentEngine# rmdir /local1/oldfiles
```

Related Commands

lls

ls

mkdir

rename

To rename a file on your Content Engine, use the **rename** EXEC command.

rename *oldfilename newfilename*

Syntax Description

oldfilename	Original filename.
newfilename	New filename.

Defaults

No default behavior or values

Command Modes

EXEC

Usage Guidelines

Use this command to rename any sysfs file without making a copy of the file.

Examples

```
ContentEngine# rename errlog.txt old_errlog.txt
```

Related Commands

cpfile

restore

To restore the Content Engine to its manufactured default status, removing user data from disk and in Flash memory, use the **restore** EXEC command.

restore factory-default

Syntax Description

factory-default	Resets Content Engine configuration and data to their manufactured default status.

Defaults

No default behavior or values

Command Modes

EXEC

Usage Guidelines

Use this command to restore data on disk and in Flash memory to the factory default, while preserving particular time stamp evaluation data. This command erases user-specified configuration information stored in the Flash image and removes data on disk and user-defined disk partitions. User-defined disk partitions that are removed include the sysfs, cfs, mediafs, and ecdnfs partitions. The configuration being removed includes the starting configuration of the device.

Examples

The following example illustrates the results of invoking the **restore** command. Because configuration parameters and data are lost, prompts are given before initiating the restore operation to ensure that you want to proceed.

```
ContentEngine# restore factory-default
Are you sure you want to go ahead?[yes/no]
Are you really sure you want to go ahead?[yes/no]

( Process of restoring the device to factory-default starts …)
( The device reboots … )

ACNS boot: detected no saved system configuration
Do you want to enter basic configuration now?
Hit RETURN to enter basic configuration
```

Note The user can enter basic configuration parameters (such as IP address, host name, and name server) at this point, or later through entries in the command-line interface.

In the following example, entering the **show disk** command after the **restore** command is invoked verifies that the **restore** command has removed data from the partitioned file systems: sysfs, cfs, mediafs, and ecdnfs.

```
ContentEngine# show disk

SYSFS         0.0GB         0.0%
CFS           0.0GB         0.0%
MEDIAFS       0.0GB         0.0%
```

```
ECDNFS        0.0GB         0.0%
FREE         29.9GB       100.0%
```

Since Flash memory configurations were removed after the **restore** command was invoked, the **show startup-config** command does not return any Flash memory data. The **show running-config** command returns the default running configurations.

The **show real-subscriber** or **show wmt** commands continue to display the same license evaluation periods as before the **restore factory-default** command was invoked, because the evaluation period is not affected by this **restore** command. For example, if there were 21 days remaining in the evaluation period before the **restore factory-default** command was used, there would continue to be 21 days remaining in the evaluation period.

Related Commands

show disk

show startup-config

show running-config

show real-subscriber

show wmt

rtsp proxy

To enable or disable the Real-Time Streaming Protocol (RTSP) proxy and to configure the RTSP proxy IP address and the redirector port number, use the **rtsp proxy** global configuration command.

rtsp proxy {**incoming** *port* | **l4-switch enable** | **media-real** {**accept-license-agreement** | **enable** | **evaluate** | **ip-address** *ip-address* | **license-key** *keyword*}}

no rtsp proxy {**incoming** *port* | **l4-switch enable** | **media-real** {**accept-license-agreement** | **enable** | **evaluate** | **ip-address** *ip-address* | **license-key** *keyword*}}

Syntax Description

incoming	Sets the listener port for incoming RTSP proxy-mode requests.
port	Port number for incoming requests (1–65535).
l4-switch	Configures Layer 4 switch interoperability for the RTSP media cache.
enable	Enables Layer 4 switch interoperability.
media-real	Configures the RealProxy cache.
accept-license-agreement	Accepts license. The license agreement can be viewed with the **show rtsp proxy license-agreement** command.
enable	Enables RealProxy media cache.
evaluate	Starts or continues the 30-day evaluation period of the RealProxy cache.
ip-address	Sets IP address of RealProxy server.
ip-address	IP address of RealProxy server.
license-key	Sets required license key for the RealProxy.
keyword	Keyword string.

Defaults

The default is RTSP proxy disabled.

Command Modes

Global configuration

Usage Guidelines

Use the **rtsp proxy** global configuration command to configure the Content Engine to accept redirected RTSP traffic or to configure the Content Engine as a media proxy to receive RTSP proxy-style requests from RealPlayer clients. The **wccp media-cache** global configuration command registers the Content Engine with WCCP Version 2-enabled routers that can transparently redirect RTSP traffic to the Content Engine. Streaming media objects are cached in the media file system (mediafs) disk partitions. The RealProxy software is copied to the software file system (swfs) partition as part of the installation procedure. Use the **disk** EXEC command to create swfs and mediafs partitions. Use the **mediafs** EXEC command to maintain the mediafs partitions. The **rtsp proxy** global configuration command configures the Content Engine RTSP proxy redirector, which redirects RTSP requests from RealPlayer clients to the RealProxy and subsequently to the media cache. RTSP requests from other than RealPlayer client are directed to the specified origin server.

The RTSP proxy redirector listens to port 554 traffic, and if the player is a RealPlayer, it redirects the RTSP request to use the RealProxy for RealMedia traffic. For traffic that it does not support (for instance, QuickTime), the Content Engine (WCCP) bypasses the traffic.

The RTSP proxy registers with the WCCP process to receive transparently redirected RTSP requests.

RealProxy software is configured with the RealSystem administrator GUI, which is accessed from the RealProxy page of the Content Engine management GUI.

Procedure to Enable Transparent RTSP Proxy Service

To enable transparent RTSP proxy service with the RealProxy on the Cisco Content Engine, follow these steps.

Requirements

- Content Engine running Cache software, Release 3.1 or more recent version
- RealProxy software installed with mediafs partitions mounted
- RealMedia license key
- IP addresses of the RealProxy and routers

Complete the following steps to enable transparent redirection of RTSP traffic to the RealProxy:

Step 1 On the WCCP Version 2 routers, configure the outbound interfaces to the Internet.

In the following example, the outbound interface is the Ethernet 0 device.

```
router(config)# ip wccp 80
router(config)# interface Ethernet 0
router(interface)# ip wccp 80 redirect out
```

Step 2 Set the WCCP Version 2 parameters on the Content Engine.

In the following example, the WCCP Version 2-enabled routers have the IP addresses 172.16.25.25 and 172.16.25.24.

```
ContentEngine(config)# wccp version 2
ContentEngine(config)# wccp router-list 1 172.16.25.25 172.16.25.24
ContentEngine(config)# wccp media-cache router-list-num 1
```

Step 3 Set the IP address for the RealProxy. Make sure that the IP address of the RealProxy is visible to the RealPlayers that use it.

```
ContentEngine(config)# rtsp proxy media-real ip-address 172.16.16.16
```

Step 4 Enter the RealProxy license number.

```
ContentEngine(config)# rtsp proxy media-real license-key mylicense
```

Step 5 Enable the RealProxy.

```
ContentEngine(config)# rtsp proxy media-real enable
```

Step 6 Save the new configuration.

```
ContentEngine# copy running-config startup-config
```

Step 7 Configure the RealProxy parameters with the RealSystem administrator GUI.

A RealProxy page has been added to the management GUI. To access the RealSystem administrator, click the **Admin** button on the RealProxy page. The **Admin** button is active when the RealProxy software is installed and enabled.

Step 8 Use the following commands to display RealProxy statistics.

```
ContentEngine# show statistics mediacache real requests
ContentEngine# show statistics mediacache real savings
```

Note The **mediacache real** statistics relate only to objects transported over RTSP that were requested by a RealPlayer client. Objects transported over HTTP are counted in the HTTP statistics. Streaming objects requested by other clients or transported over other protocols, bypass the Content Engine.

Procedure to Enable Conventional RTSP Proxy Services

To enable the Content Engine to service RealPlayer clients with the RealProxy on the Content Engine, follow these steps:

Step 1 Set the IP address for the RealProxy. Make sure that the IP address of the RealProxy is visible to the RealPlayers that use it.

```
ContentEngine(config)# rtsp proxy media-real ip-address 172.16.16.16
```

Step 2 Enter the RealProxy license number shipped with the Content Engine.

```
ContentEngine(config)# rtsp proxy media-real license-key mylicense
```

Step 3 Enable the RealProxy.

```
ContentEngine(config)# rtsp proxy media-real enable
```

Step 4 Configure the Content Engine to listen for RTSP traffic on a specified port. The default RTSP port is 554.

```
ContentEngine# rtsp proxy incoming 554
```

Step 5 Configure RealPlayer clients to use RealProxy on the Content Engine.

a. Open RealPlayer.

b. Choose **View > Preferences**.

c. Click the **Proxy** tab.

d. Check the **Use RTSP proxy** check box.

e. Enter the IP address of the Content Engine in the Use RTSP proxy address field.

f. Specify the port number that you entered with the Cache software **rtsp proxy** global configuration CLI command.

g. Click **OK**.

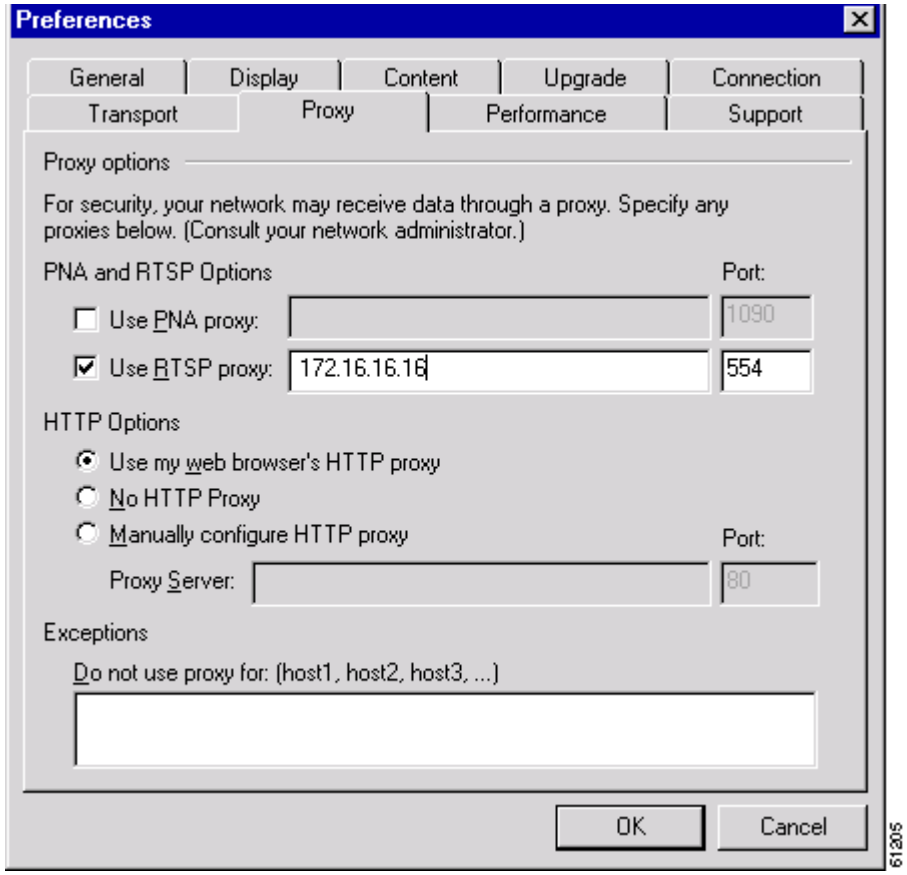
The RealPlayer configuration is shown in Figure 2-1.

RealPlayer is now able to use the Content Engine RealProxy to fetch streaming objects.

For more information on setting up RealPlayer, refer to the RealProxy readme “Setting Up RealServer” and “Setting Up RealPlayer” sections at the following URL:

http://service.real.com/help/library/guides/proxy/readme.htm#5

Figure 2-1 RealPlayer Configured to Use Content Engine as Traditional Proxy for RTSP Traffic



Step 6 Save the Content Engine configuration to Flash memory.

```
ContentEngine# copy running-config startup-config
```

Step 7 Use the following commands to display RealProxy statistics.

```
ContentEngine# show statistics mediacache real requests
ContentEngine# show statistics mediacache real savings
```

Note The **mediacache real** statistics relate only to objects transported over RTSP that were requested by a RealPlayer client. Objects transported over HTTP are counted in the HTTP statistics. Streaming objects requested by other clients or transported over other protocols, bypass the Content Engine.

Examples

To start proxy RTSP traffic where the user agent is RealPlayer (proxy-directed as well as transparently redirected), enter this command.

```
ContentEngine(config)#: rtsp proxy media-real enable
```

To stop proxy RTSP traffic from RealPlayer (proxy-directed as well as transparently redirected), enter this command.

```
ContentEngine(config)# no rtsp proxy media-real
```

To set the port number for proxy-directed RTSP traffic, thus setting up the Content Engine to do traditional RTSP proxy, enter this command.

```
ContentEngine(config)# rtsp proxy incoming portnum
```

To clear the port number for proxy-directed (all) RTSP traffic, thus disabling the Content Engine from doing further traditional proxy of RTSP traffic, enter this command.

```
ContentEngine(config)# no rtsp proxy incoming
```

To enable proxy of proxy-directed RTSP traffic, enter the following CLI commands.

```
ContentEngine(config)# rtsp proxy media-real enable
ContentEngine(config)# rtsp proxy incoming port
ContentEngine(config)# rtsp proxy media-real enable
ContentEngine(config)# wccp router-list 1 172.16.25.25 172.16.25.24
ContentEngine(config)# wccp media-cache router-list-num 1
ContentEngine(config)#
```

In this example, on the router side, Ethernet0 is the outbound interface to the Internet.

```
router(config)# ip wccp 80
router(config)# interface Ethernet 0
router(interface)# ip wccp 80 redirect out
```

Related Commands

show rtsp

rule

To set the rules by which the Content Engine filters web traffic, use the **rule** global configuration command.

rule block {**domain** *LINE* | **dst-ip** *d_ipaddress d_subnet* | **dst-port** *port* | **header-field** {**referer** *LINE* | **request-line** *LINE* | **user-agent** *LINE*} | **src-ip** *s_ipaddress s_subnet* | **url-regex** *LINE*}

rule dscp client cache-hit {**match-server** {**domain** *LINE* | **dst-ip** *d_ipaddress d_subnet* | **dst-port** *port* | **mime-type** *LINE* | **src-ip** *s_ipaddress s_subnet* | **url-regex** *LINE*} | **set-dscp** *dscpvalue* | **set-tos** *tosvalue*}

rule dscp client cache-miss {**match-server** {**domain** *LINE* | **dst-ip** *d_ipaddress d_subnet* | **dst-port** *port* | **mime-type** *LINE* | **src-ip** *s_ipaddress s_subnet* | **url-regex** *LINE*} | **set-dscp** *dscpvalue* | **set-tos** *tosvalue*}}

rule dscp server {**match-client** {**domain** *LINE* | **dst-ip** *d_ipaddress d_subnet* | **dst-port** *port* | **src-ip** *s_ipaddress s_subnet* | **url-regex** *LINE*} | **set-dscp** *dscpvalue* | **set-tos** *tosvalue*}

rule enable

rule freshness-factor *exp_time* {**domain** *LINE* | **dst-ip** *d_ipaddress d_subnet* | **dst-port** *port* | **mime-type** *LINE* | **src-ip** *s_ipaddress s_subnet* | **url-regex** *LINE*}

rule no-auth {**domain** *LINE* | **dst-ip** *d_ipaddress d_subnet* | **dst-port** *port* | **src-ip** *s_ipaddress s_subnet* | **url-regex** *LINE*}

rule no-cache {**domain** *LINE* | **dst-ip** *d_ipaddress d_subnet* | **dst-port** *port* | **mime-type** *LINE* | **src-ip** *s_ipaddress s_subnet* | **url-regex** *LINE*}

rule no-proxy {**domain** *LINE* | **dst-ip** *d_ipaddress d_subnet* | **dst-port** *port* | **src-ip** *s_ipaddress s_subnet* | **url-regex** *LINE*}

rule redirect {**header-field** {**referer** *LINE* | **request-line** *LINE* | **user-agent** *LINE* } | **url-regsub** *regexpr substitute*}

rule refresh {**domain** *LINE* | **dst-ip** *d_ipaddress d_subnet* | **dst-port** *port* | **mime-type** *LINE* | **src-ip** *s_ipaddress s_subnet* | **url-regex** *LINE*}

rule reset {**domain** *LINE* | **dst-ip** *d_ipaddress d_subnet* | **dst-port** *port* | **header-field** {**referer** *LINE* | **request-line** *LINE* | **user-agent** *LINE*} | **src-ip** *s_ipaddress s_subnet* | **url-regex** *LINE*}

rule rewrite {**header-field** {**referer** *LINE* | **request-line** *LINE* | **user-agent** *LINE*} | **url-regsub** *regexpr substitute*}

rule selective-cache {**domain** *LINE* | **dst-ip** *d_ipaddress d_subnet* | **dst-port** *port* | **mime-type** *LINE* | **src-ip** *s_ipaddress s_subnet* | **url-regex** *LINE*}

rule use-proxy {*hostname* | *ip-address*} *port* {**domain** *LINE* | **dst-ip** *d_ipaddress d_subnet* | **dst-port** *port* | **src-ip** *s_ipaddress s_subnet* | **url-regex** *LINE*}

rule use-proxy-failover {*hostname* | *ip-address*} *port* {**domain** *LINE* | **dst-ip** *d_ipaddress d_subnet* | **dst-port** *port* | **src-ip** *s_ipaddress s_subnet* | **url-regex** *LINE*}

rule use-server {*hostname* | *ip-address*} *port* {**domain** *LINE* | **dst-ip** *d_ipaddress d_subnet* | **dst-port** *port* | **src-ip** *s_ipaddress s_subnet* | **url-regex** *LINE*}

no rule block {**domain** *LINE* | **dst-ip** *d_ipaddress d_subnet* | **dst-port** *port* | **header-field** {**referer** *LINE* | **request-line** *LINE* | **user-agent** *LINE*} | **src-ip** *s_ipaddress s_subnet* | **url-regex** *LINE*} | **dscp client** {**cache-hit** {**match-server** {**domain** *LINE* | **dst-ip** *d_ipaddress d_subnet* | **dst-port** *port* | **mime-type** *LINE* | **src-ip** *s_ipaddress s_subnet* | **url-regex** *LINE*} | **set-dscp** *dscpvalue* | **set-tos** *tosvalue*} | **cache-miss** {**match-server** {**domain** *LINE* | **dst-ip** *d_ipaddress d_subnet* | **dst-port** *port* | **mime-type** *LINE* | **src-ip** *s_ipaddress s_subnet* | **url-regex** *LINE*} | **set-dscp** *dscpvalue* | **set-tos** *tosvalue*}} | **dscp server** {**match-client** {**domain** *LINE* | **dst-ip** *d_ipaddress d_subnet* | **dst-port** *port* | **src-ip** *s_ipaddress s_subnet* | **url-regex** *LINE*} | **set-dscp** *dscpvalue* | **set-tos** *tosvalue*} | **enable** | **freshness-factor** *exp_time* {**domain** *LINE* | **dst-ip** *d_ipaddress d_subnet* | **dst-port** *port* | **mime-type** *LINE* | **src-ip** *s_ipaddress s_subnet* | **url-regex** *LINE*} | **no-auth** {**domain** *LINE* | **dst-ip** *d_ipaddress d_subnet* | **dst-port** *port* | **src-ip** *s_ipaddress s_subnet* | **url-regex** *LINE*} | **no-cache** {**domain** *LINE* | **dst-ip** *d_ipaddress d_subnet* | **dst-port** *port* | **mime-type** *LINE* | **src-ip** *s_ipaddress s_subnet* | **url-regex** *LINE*} | **no-proxy** {**domain** *LINE* | **dst-ip** *d_ipaddress d_subnet* | **dst-port** *port* | **src-ip** *s_ipaddress s_subnet* | **url-regex** *LINE*} | **redirect** {**header-field** {**referer** *LINE* | **request-line** *LINE* | **user-agent** *LINE* } | **url-regsub** *regexpr substitute*} | **refresh** {**domain** *LINE* | **dst-ip** *d_ipaddress d_subnet* | **dst-port** *port* | **mime-type** *LINE* | **src-ip** *s_ipaddress s_subnet* | **url-regex** *LINE*} | **reset** {**domain** *LINE* | **dst-ip** *d_ipaddress d_subnet* | **dst-port** *port* | **header-field** {**referer** *LINE* | **request-line** *LINE* | **user-agent** *LINE*} | **src-ip** *s_ipaddress s_subnet* | **url-regex** *LINE*} | **rewrite** {**header-field** {**referer** *LINE* | **request-line** *LINE* | **user-agent** *LINE*} | **url-regsub** *regexpr substitute*} | **selective-cache** {**domain** *LINE* | **dst-ip** *d_ipaddress d_subnet* | **dst-port** *port* | **mime-type** *LINE* | **src-ip** *s_ipaddress s_subnet* | **url-regex** *LINE*} | **use-proxy** {*hostname* | *ip-address*} *port* {**domain** *LINE* | **dst-ip** *d_ipaddress d_subnet* | **dst-port** *port* | **src-ip** *s_ipaddress s_subnet* | **url-regex** *LINE*} | **rule use-proxy-failover** {*hostname* | *ip-address*} *port* {**domain** *LINE* | **dst-ip** *d_ipaddress d_subnet* | **dst-port** *port* | **src-ip** *s_ipaddress s_subnet* | **url-regex** *LINE*} | **use-server** {*hostname* | *ip-address*} *port* {**domain** *LINE* | **dst-ip** *d_ipaddress d_subnet* | **dst-port** *port* | **src-ip** *s_ipaddress s_subnet* | **url-regex** *LINE*}

Syntax Description

block	Action—Blocks the request.
domain	Pattern type—Regular expression to match the domain name.
LINE	Regular expression to be matched with domain name.
dst-ip	Pattern type—Destination IP address of the request.
d_ipaddress	Destination IP address of the request.
d_subnet	Destination IP subnet mask.
dst-port	Pattern type—Destination port number.
port	Destination port number (1–65535).
header-field	Pattern type—Request header field pattern.
referer	Referer request header.
LINE	Regular expression to be matched with referer header.
request-line	Request method line.
LINE	Regular expression to be matched with the request method line.
user-agent	User agent request header.
LINE	Regular expression to be matched with the user agent header.
src-ip	Pattern type—Source IP address of the request.
s_ipaddress	Source IP address of the request.

s_subnet	Source IP subnet mask.
url-regex	Pattern type—Regular expression to match a substring of the URL.
LINE	Regular expression to be matched with URL string.
dscp client	Action—Configures IP Type of Service/differentiated services code point (ToS/DSCP) field responses to client.
cache-hit	Configures cache hit responses to client.
match-server	Uses original ToS/DSCP value of server.
mime-type	Pattern type—MIME type to be matched with the Content-Type HTTP header.
LINE	Regular expression to be matched with the MIME type.
set-dscp	Configures differentiated services code point (DSCP) values.
dscp-packets:	0–63—Sets DSCP values:
af11	Sets packets with AF11 DSCP (001010).
af12	Sets packets with AF12 DSCP (001100).
af13	Sets packets with AF13 DSCP (001110).
af21	Sets packets with AF21 DSCP (010010).
af22	Sets packets with AF22 DSCP (010100).
af23	Sets packets with AF23 DSCP (010110).
af31	Sets packets with AF31 DSCP (011010).
af32	Sets packets with AF32 DSCP (011100).
af33	Sets packets with AF33 DSCP (011110).
af41	Sets packets with AF41 DSCP (100010).
af42	Sets packets with AF42 DSCP (100100).
af43	Sets packets with AF43 DSCP (100110).
cs1	Sets packets with CS1 (precedence 1) DSCP (001000).
cs2	Sets packets with CS2 (precedence 2) DSCP (010000).
cs3	Sets packets with CS3 (precedence 3) DSCP (011000).
cs4	Sets packets with CS4 (precedence 4) DSCP (100000).
cs5	Sets packets with CS5 (precedence 5) DSCP (101000).
cs6	Sets packets with CS6 (precedence 6) DSCP (110000).
cs7	Sets packets with CS7 (precedence 7) DSCP (111000).
default	Sets packets with default DSCP (000000).
ef	Sets packets with EF DSCP (101110).
set-tos	Configures type of service (ToS) values.
tos-packets:	0–127—Sets ToS value:
critical	Sets packets with critical precedence (80).
flash	Sets packets with flash precedence (48).

flash-override	Sets packets with flash override precedence (64).
immediate	Sets packets with immediate precedence (32).
internet	Sets packets with internetwork control precedence (96).
max-reliability	Sets packets with max reliable ToS (2).
max-through-put	Sets packets with max throughput ToS (4).
min-delay	Sets packets with min delay ToS (8).
min-monetary-cost	Sets packets with min monetary cost ToS (1).
network	Sets packets with network control precedence (112).
normal	Sets packets with normal ToS (0).
priority	Sets packets with priority precedence (16).
cache-miss	Configures cache miss responses to client.
dscp server	Action—Configures IP Type of Service/differentiated services code point (ToS/DSCP) server for outgoing requests.
match-client	Uses original ToS/DSCP value of client.
enable	Enables rules processing.
freshness-factor	Action—Caches heuristic modifiers.
exp_time	Expiration time of object as a percentage of age (0–100).
no-auth	Action—Does not authenticate.
no-cache	Action—Does not cache the object.
no-proxy	Action—Does not use any upstream proxy.
redirect	Action—Redirects request to rewritten URL.
url-regsub	Pattern type—Sets regular expression to match URL and replacement pattern.
regexpr	Regular expression to match.
substitute	Pattern to substitute for *regexpr*.
refresh	Action—Revalidates the object with the web server.
reset	Action—Issues a TCP RST.
rewrite	Action—Rewrites URL and fetch.
selective-cache	Action—Caches this object if permitted by HTTP.
use-proxy	Action—Makes use of a specific upstream proxy.
hostname	Host name of the specific proxy.
ip-address	IP address of the specific proxy.
port	Port number of the specific proxy (1–65535).
use-proxy-failover	Action—Causes outgoing proxy to fail over to outgoing HTTP proxy servers.
hostname	Host name of the specific proxy.
ip-address	IP address of the specific proxy.
port	Port number of the specific proxy (1–65535).
use-server	Action—Makes use of a specific server.

hostname	Host name of the specific server.
ip-address	IP address of the specific server.
port	Port number of the specific server (1–65535).

Defaults

The default is rule processing disabled.

Command Modes

Global configuration

Usage Guidelines

Configuring the Rules Template

The Rules Template feature allows for requests to be *matched* using an arbitrary number of parameters with an arbitrary number of *policies* applied against those matches. Requests can be matched against regular expressions symbolizing domain names, source IP addresses and network masks, destination IP addresses and network masks, destination port numbers, MIME-types, or regular expressions symbolizing a URL.

Policies that can be applied include:

- Blocking the request
- Using a specific object freshness calculation factor
- Not caching an object
- Bypassing an upstream proxy for the request
- Redirecting the request to a different URL
- Revalidating the object with the origin server
- Rewriting the URL
- Selectively caching the object
- Using a specific upstream proxy
- Using a specific server for the request

The **rule** command options **freshness-factor**, **redirect**, **rewrite**, and **use-server** were introduced in Cache software, Release 3.1.

The Rules Template feature is applicable only for HTTP, FTP, and HTTPS traffic and is not applicable for streaming protocols (RTSP, PNA, and WMT) implemented in ACNS 4.1 software.

Note To enter a question mark (?) character in a rule regular expression configuration from the command-line interface, use the escape character (\) followed by a question mark (?) character. This prevents the command-line interface from displaying context-sensitive help.

Actions and Patterns

A rule is an action and a pattern. If an HTTP request matches the pattern, the corresponding action is performed on the request. To enable rule processing, use the **rule enable** command.

An action is something the Content Engine performs when processing an HTTP request, for instance, blocking the request, using an alternative proxy, and so forth.

A pattern defines the limits of an HTTP request; for instance, a pattern may specify that the source IP address fall in the subnet range 172.16.*.*.

Rules can be dynamically added, displayed, or deleted from the Content Engine. The rules are preserved across reboots because they are written into persistent storage such as NVRAM. Only the system resources limit the number of rules the Content Engine can support. Because rules consume resources, the more rules there are defined, the more Content Engine performance may be affected.

Actions

The Rules Template feature supports the following types of action:

- **Block**—Blocks this request.
- **DSCP**—Configures the IP ToS/DSCP code point field.
  - **client cache-hit**—Configures the IP ToS/DSCP code point field for **cache-hit** responses to the client.
  - **client cache-miss**—Configures the IP ToS/DSCP code point field for **cache-miss** responses to the client.

  Setting the Type of Service (ToS) or differentiated services code point (DSCP) is called packet marking, allowing you to partition network data into multiple priority levels or types of service. With this release, you can now set the ToS or DSCP values in IP packets based on a URL match, a file type, a domain, a destination IP address, a source IP address, or a destination port.

  You can set specific ToS or DSCP values for the following:
  - Requests from the Content Engine to the server
  - Responses to the client on cache hit
  - Responses to the client on cache miss

  The ToS or DSCP may be set based on any of the policies matching the **src-ip-address**, **dst-ip-address**, **dst-port-number**, **domain regex**, **url-regex**, or **mime-type regex** options. In addition, you can now configure global ToS or DSCP settings with the **ip dscp** command.

  Note The Rules Template configuration takes precedence over the **ip dscp** command, and the **url-filter** command takes precedence over the **rule** command to the extent that even the rule **no-block** command is executed only if the **url-filter** command has not blocked the request.

- **DSCP server**—Configures the IP ToS/DSCP code point field for requests to the origin server.
- **Freshness-factor**—Determines the Time To Live if the request URL matches a specified regular expression. The **refresh** configuration takes priority over **freshness-factor** configurations.
- **No-auth**—Does not authenticate.

  Note that the **no-auth** rules result in the display of multiple authentication windows in the following scenario:
  - When the main page (for example, index.htm) is excluded from proxy authentication by using **no-auth** rules
  - When the user entry is not already included in the Content Engine authentication cache
  - When the index.htm page contains objects belonging to different domains

To avoid multiple authentication windows, configure the hidden **http avoid-multiple-auth-prompts** command in global configuration mode. Once it is configured, check the configuration with the **show http avoid-multiple-auth-prompts** command as shown the following example.

```
ContentEngine# show http avoid-multiple-auth-prompts
Avoiding multiple authentication prompts due to no-auth rules is enabled
```

Note The command in the example is hidden because it is applicable only to this specific scenario.

- **No-cache**—Does not cache this object. If both **no-cache** and **selective-cache** actions are matched, **no-cache** takes precedence.
- **No-proxy**—For a cache miss, does not use the configured upstream proxy but rather contacts the server directly.
- **Redirect**—Redirects the original request to a specified URL. Redirect is relevant to the RADIUS server only if the RADIUS server has been configured for **redirect**.
- **Refresh**—For a cache hit, forces an object freshness check with the server.
- **Reset**—Issues a TCP RST. This reset request in useful when resetting Code Red or Nimda virus requests.
- **Rewrite**—Rewrites the original request as a specified URL. The Content Engine searches for the rewritten URL in cache, and then on cache miss, fetches the rewritten URL and returns the object transparently to the client. It is preferable to use a **redirect** rule rather than **rewrite** because of possible performance impacts.

  The URL rewrite could change the domain name of the URL, which necessitates a DNS lookup to find the destination (dst) IP address of the new rewritten server to which the request must be sent. The original dst IP address derived from the WCCP redirect packet cannot be used.
- **Selective-cache**—Caches this object only if it is a match and is allowed to be cached by HTTP. If one or more rules specify this action, an object is cached if and only if it matches at least one of the **selective-cache** rules and passes every other caching restriction such as the object-size check and the no-cache-on-authenticated-object check. If the object does not match any of the **selective-cache** rules, the object is *not* cached.
- **Use-proxy**—For a cache miss, uses a specific upstream proxy. Specify the upstream proxy IP address (or domain name) and port number. If both **no-proxy** and **use-proxy** are matched, **no-proxy** takes precedence.
- **Use-proxy-failover**—Supports failover capability. The **use-proxy-failover** rule is similar to the **use-proxy** rule, except that if the connection attempt on the configured outgoing proxy fails, the requests fail over to the outgoing proxies configured with the HTTP proxy outgoing configuration. The rule requests use the HTTP proxy outgoing **origin-server** option, if it is configured. The **use-proxy-failover** rule takes precedence over the **use-proxy** rule. If both **no-proxy** and **use-proxy-failover** are matched, **no-proxy** takes precedence.

  The HTTP failover does not apply if the destination is on the exclude list. When in transparent mode, the setting for the original proxy takes precedence.
- **Use-server**—Sends server-style HTTP requests from the Content Engine to the specified IP address and port on a cache miss.

  Among **use-server**, **no-proxy**, and **use-proxy** rules, the **use-server** rule is the first one to be checked. If it results in a rule miss, **no-proxy** and **use-proxy** rules are executed in succession (**use-proxy** is not checked if a **no-proxy** rule matches).

If a rule is configured with a fully qualified domain name (FQDN) and a request is received with the partial domain name in transparent mode, the rule fails to be executed, as the FQDN is not in the request URL. In transparent mode, if a request is destined for a particular domain (for which a domain rule is configured) and does not contain the Host header, the rule pattern match fails.

Patterns

The Rules Template feature supports the following types of patterns.

- **Domain**—Matches the domain name in the URL or the Host header against a regular expression. For example, ".*ibm.*" matches any domain name that contains the "ibm" substring. "\.foo\.com$" matches any domain name that ends with the ".foo.com" substring.

  Note In regular expression syntax, the dollar sign "$" metacharacter directs that a match is made only when the pattern is found at the end of a line.

- **Dst-ip**—Matches the request's destination IP address and netmask. Specify an IP address and a netmask. In proxy mode, the Content Engine does a DNS lookup to resolve the destination IP address of the HTTP request, making the response time longer, and possibly negating the benefit of setting a **dst-ip** rule. When an outgoing proxy is configured, cache miss requests are forwarded by the Content Engine to the outgoing proxy without examination of the destination server IP address, making the **dst-ip** rule unenforceable on the first Content Engine.
- **Dst-port**—Matches the request's destination port number. Specify a port number.
- **Mime-type**—Matches the MIME type of the response. Specify a MIME type string, for example, "image/gif," as defined in RFC 2046. The administrator can specify a substring, for example, "java" and have it apply to all MIME types with the "java" substring, such as "application/x-javascript."
- **Src-ip**—Matches the request's source IP address and netmask. Specify an IP address and a netmask.
- **URL-regex**—Matches the URL against a regular expression. The match is case insensitive. Specify a regular expression whose syntax can be found at:

  http://yenta.www.media.mit.edu/projects/Yenta/Releases/Documentation/regex-0.12/.

- **Header-field**—Requests header field pattern.

  Request header field patterns **referer**, **request-line**, and **user-agent** are supported for actions **block**, **reset**, **redirect**, and **rewrite**. The **referer** pattern matches against the Referer header in the request, **request-line** pattern matches against the first line of the request, and **user-agent** pattern matches against the User-Agent header in the request.

- **URL-regsub**—For the **rewrite** and **redirect** actions, matches the URL against a regular expression to form a new URL per pattern substitution specification. The match is case insensitive. The valid substitution index range is from 1 to 9.

Rules Template Processing Considerations

There is a predefined order of execution among the actions and patterns. A group of rules with the same action is always executed either before or after another group of rules with a different action. This order of execution is predefined and not affected by the order in which the rules are entered using CLI commands.

Among the rules of the same action, there is a predefined execution order among the rules pattern. This means that within a group of rules of the same action, one group of rules with the same pattern is always executed either before or after another group of rules with a different pattern. This order is predefined and not affected by the order in which the rules are entered using CLI commands.

Rule Action Execution Order

The order of rule action execution is as follows:

1. **No-Auth**—Before authentication using RADIUS/LDAP/NTLM
2. **Reset**—Before cache lookup
3. **Block**—Before cache lookup
4. **Redirect**—Before cache lookup
5. **Rewrite**—Before cache lookup
6. **Refresh**—On cache hit
7. **Freshness-factor**—On cache hit
8. **Use-server**—On cache miss
9. **No-proxy**—On cache miss
10. **Use-proxy-failover**—On cache miss
11. **Use-proxy**—On cache miss
12. **TOS/DSCP server**—On cache miss
13. **TOS/DSCP client**
14. **No-cache**—On cache miss
15. **Selective-cache**—On cache miss

Note The commands **rule no-proxy**, **rule use-proxy-failover**, and **rule use-proxy** take precedence over **https proxy outgoing**, **http proxy outgoing**, and **ftp proxy outgoing** commands.

During a request using the rules template CLI commands, rule actions 1 through 4 use the original URL request for pattern matches. After a URl rewrite (rule action 5), rule actions 6 through 15 use the transformed URL for rule executions.

The commands **rule reset**, **rule block**, **rule rewrite**, and **rule redirect** support the following additional patterns for rule templates request:

- request-line—matches first line.
- referer—matches referer header.
- user-agent—matches user-agent header.

Rule Pattern Execution Order

The order of rule pattern execution is as follows:

1. **Dst-port**—Destination port check.
2. **Src-ip**—Source IP address check.
3. **URL-regex**—URL regex check.
4. **Domain**—Domain rule check.
5. **Dst-ip**—Destination IP address check.
6. **MIME-type**—Mime-type regex check.

Note Because the MIME type exists only in the response, only the actions **freshness-factor**, **refresh**, **no-cache**, and **selective-cache** apply to a rule of MIME type.

A search for a rule match with the remaining pattern is not performed if a match has already been found. For instance, if a match for the **rule block** action is found with a **URL-regex** request, then the remaining patterns **Domain**, **Dst-ip**, or **MIME-type** are not searched.

Rules are ORed together. Multiple rules may all match a request, and then all actions are taken with precedence among conflicting actions. Each rule contains one pattern, and patterns cannot be ANDed together. In future releases, ANDed patterns may be supported.

It is possible to circumvent some rules. For example, to circumvent a rule with the **domain** pattern, enter the web server IP address instead of the domain name in the browser. A rule may have unintended effects. For instance, a rule with the **domain** pattern specified as "ibm" that is intended to match "www.ibm.com" can also match domain names like www.ribman.com.

An **src-ip** rule may not apply as intended to requests that are received from another proxy because the original client IP address is in an X-forwarded-for header.

If a rule pattern match occurs, then the rest of the patterns are not searched. If the server has already marked an object as non-cacheable, **no-cache** rules are not checked at all, because the server already recognizes that this object is not cached. Any **no-cache** rule checks are performed only for cacheable requests.

Order of Execution Among Rules of Same Action and Same Pattern

Among the rules of the same action and the same pattern, the order of execution is in the reverse order in which the rules are entered. For instance, if the **use-proxy** commands are entered in the following order:

use-proxy 1.2.3.4 abc.abc.com

use-proxy 2.3.4.5 *.abc.com

then a request to abc.abc.com is sent to proxy 2.3.4.5 because the **use-proxy 2.3.4.5 *.abc.com** command is entered last and evaluated first. However, if the same commands are entered in a reverse order as follows:

use-proxy 2.3.4.5 *.abc.com

use-proxy 1.2.3.4 abc.abc.com

then a request to abc.abc.com is sent to proxy 1.2.3.4, as the **use-proxy 1.2.3.4 abc.abc.com** command is entered last and evaluated first.

The following rule pattern also applies for all rule actions. Mime-type checks are applicable only for **no-cache**, **selective-cache**, and **refresh** rules. If a rule pattern match occurs, then the rest of the patterns are not searched. If the server has already marked an object as non-cacheable, **no-cache** rules are not checked at all, since the server already recognizes that this object is not cached. Any **no-cache** rule checks are performed only for cacheable requests.

Examples

Multiple patterns can be input on the same line. If any of them matches the incoming HTTP request, the corresponding action is taken.

```
ContentEngine(config)# rule block domain \.foo.com ?
 LINE    <cr>
ContentEngine(config)# rule block domain \.foo.com bar.com
ContentEngine(config)#
```

This example sets the ToS value to "minimize delay" for outbound requests to a specified destination IP address, in this case 1.1.1.1.

```
Console(config)# rule dscp server set-tos min-delay dst-ip 1.1.1.1 255.255.255.255
```

This example sets the ToS value to "minimize delay" for all outbound requests.

```
Console(config)# ip dscp server set-tos min-delay
```

Using the IP command, this example uses the ToS or DSCP value that was originally sent by the server (when the object was first fetched) for all future cache hit responses for the same object:

```
Console(config)# ip dscp client cache-hit match-server

ContentEngine(config)# rule no-cache url-regex \.*cgi-bin.* ?
 LINE     <cr>
ContentEngine(config)# rule no-cache url-regex \.*cgi-bin.*
ContentEngine(config)#

ContentEngine(config)# rule no-cache dst-ip 172.77.120.0 255.255.192.0
```

Most actions do not have any parameters, as in the preceding examples. One exception is **use-proxy**, as in the following example.

```
ContentEngine(config)# rule use-proxy ?
 Hostname or A.B.C.D. IP address of the specific proxy
ContentEngine(config)# rule use-proxy CE.foo.com ?
<1-65535>  Port number of the specific proxy
ContentEngine(config)# rule use-proxy CE.foo.com 8080 ?
  domain     Regular expression to match with the domain name
  dst-ip     Destination IP address of the request
  dst-port   Destination port number
  src-ip     Source IP address of the request
  url-regex  Regular expression to substring match with the URL
ContentEngine(config)# rule use-proxy CE.foo.com 8080 url-regex ?
 LINE  Regular expression to substring match with the URL
ContentEngine(config)# rule use-proxy CE.foo.com 8080 url-regex .*\.jpg$ ?
 LINE     <cr>
ContentEngine(config)# rule use-proxy CE.foo.com 8080 url-regex .*\.jpg$ .*\.gif$ .*\.pdf$
ContentEngine(config)#
```

Other branches of the **rule** command work similarly to the above examples.

To delete rules, use **no** in front of the rule creation command.

```
ContentEngine(config)#no rule block url-regex .*\.jpg$ .*\.gif$ .*\.pdf$
```

The following example redirects a request for old-domain-name, which has been changed to new-domain-name.

```
cache(config)# rule redirect url-regsub http://old-domain-name/ http://new-domain-name/
```

The following example redirects requests from an IETF site to one that is locally mirrored.

```
cache(config)# rule redirect url-regsub http://www.ietf.org/rfc/(.*) http://wwwin-eng.cisco.com/RFC/RFC/\1
```

For the preceding example, if the request URL is http://www.ietf.org/rfc/rfc1111.txt, the Content Engine rewrites the URL as http://wwwin-eng.cisco.com/RFC/RFC/rfc1111.txt and sends a 302 Temporary Redirect response with the rewritten URL in the Location header to the client. The browser automatically initiates a request to the rewritten URL.

The following example redirects all requests for linux.org to a local server in India that is closer to where the Content Engine is located.

```
cache(config)# rule redirect url-regsub http://linux.org/(.*) http://linux.org.in/\1
```

The following example rewrites requests from an IETF site to one that is locally mirrored.

```
cache(config)# rule rewrite url-regsub http://www.ietf.org/rfc/.*
http://wwwin-eng.cisco.com/RFC/$1
```

The **no-auth** option permits specific login and content requests to bypass authentication and authorization features such as LDAP, RADIUS, SSH, or TACACS+.

```
Console(config)# rule enable
Console(config)# rule no-auth src-ip 172.16.53.88 255.255.255.255
Console(config)# rule no-auth dst-ip 172.22.73.34 255.255.255.255
Console(config)# rule no-auth dst-port 9090
Console(config)# rule no-auth url-regex .*cgi-bin.*
Console(config)# rule no-auth domain cisco.com
```

In the following example, any requests from src-ip 172.16.53.88 are not authenticated.

```
Console(config)# rule no-auth src-ip 172.16.53.88 255.255.255.255
```

In the following example, any requests to dst-ip 172.22.73.34 are not authenticated.

```
Console(config)# rule no-auth dst-ip 172.22.73.34 255.255.255.255
```

In the following example, any requests with the destination port 9090 are not authenticated.

```
Console(config)# rule no-auth dst-port 9090
```

In the following example, any requests with “cisco.com” as the domain are not authenticated. (For example, requests for roti.cisco.com or badal.cisco.com are excluded from the Content Engine authentication.)

```
Console(config)# rule no-auth domain cisco.com
```

In the following example, any requests with “cgi-bin” in the URL are not authenticated.

```
Console(config)# rule no-auth url-regex .*cgi-bin.*
```

The **dscp** option allows you to set the Type of Service (ToS) or differentiated services code point (DSCP) values in IP packets based on a URL match, a file type, a domain, a destination IP address, a source IP address, or a destination port.

```
 Console(config)# rule dscp ?
   client  Settings for responses to client
   server  Settings for outgoing requests

Console(config)# rule dscp client ?
   cache-hit   Cache hit responses to client
   cache-miss  Cache miss responses to client

Console(config)# rule dscp client cache-hit ?
   match-server  Use server's original ToS/DSCP value
   set-dscp      Set DSCP
   set-tos       Set Type of Service

Console(config)# rule dscp client cache-hit set-dscp ?
   <0-63>   Set DSCP value
   af11     Set packets with AF11 dscp (001010)
   af12     Set packets with AF12 dscp (001100)
   af13     Set packets with AF13 dscp (001110)
   af21     Set packets with AF21 dscp (010010)
   af22     Set packets with AF22 dscp (010100)
   af23     Set packets with AF23 dscp (010110)
```

```
  af31     Set packets with AF31 dscp (011010)
  af32     Set packets with AF32 dscp (011100)
  af33     Set packets with AF33 dscp (011110)
  af41     Set packets with AF41 dscp (100010)
  af42     Set packets with AF42 dscp (100100)
  af43     Set packets with AF43 dscp (100110)
  cs1      Set packets with CS1(precedence 1) dscp (001000)
  cs2      Set packets with CS2(precedence 2) dscp (010000)
  cs3      Set packets with CS3(precedence 3) dscp (011000)
  cs4      Set packets with CS4(precedence 4) dscp (100000)
  cs5      Set packets with CS5(precedence 5) dscp (101000)
  cs6      Set packets with CS6(precedence 6) dscp (110000)
  cs7      Set packets with CS7(precedence 7) dscp (111000)
  default  Set packets with default dscp (000000)
  ef       Set packets with EF dscp (101110)

Console(config)# rule dscp client cache-hit set-tos ?
  <0-127>            Set ToS value
  critical           Set packets with critical precedence (80)
  flash              Set packets with flash precedence (48)
  flash-override     Set packets with flash override precedence (64)
  immediate          Set packets with immediate precedence (32)
  internet           Set packets with internetwork control precedence (96)
  max-reliability    Set packets with max reliable ToS (2)
  max-throughput     Set packets with max throughput ToS (4)
  min-delay          Set packets with min delay ToS (8)
  min-monetary-cost  Set packets with min monetary cost ToS (1)
  network            Set packets with network control precedence (112)
  normal             Set packets with normal ToS (0)
  priority           Set packets with priority precedence (16)

Console(config)# rule dscp client cache-hit set-dscp af11 ?
  domain     Regular expression to match with the domain name
  dst-ip     Destination IP address of the request
  dst-port   Destination port number
  mime-type  Regular expression to match with MIME type
  src-ip     Source IP address of the request
  url-regex  Regular expression to substring match with the URL

Console(config)# rule dscp client cache-miss ?
  match-server  Use server's original ToS/DSCP value
  set-dscp      Set DSCP
  set-tos       Set Type of Service

Console(config)# rule dscp server ?
  match-client  Use client's ToS/DSCP value
  set-dscp      Set DSCP
  set-tos       Set Type of Service

Console# show rule action ?
  block             Block the request
  dscp              IP ToS/DSCP (Differentiated Services)
  freshness-factor  Caching heuristic modifiers
  no-auth           Do not authenticate
  no-cache          Do not cache the object
  no-proxy          Do not use any upstream proxy
  redirect          Redirect request to rewritten URL
  refresh           Revalidate the object with the web server
  rewrite           Rewrite URL and fetch
  selective-cache   Cache this object
  use-proxy         Use a specific upstream proxy
  use-server        Use a specific server

Console# show rule action dscp ?
```

```
  client  Settings for responses to client
  server  Settings for outgoing requests

Console# show rule action dscp client ?
  cache-hit   Cache hit responses to client
  cache-miss  Cache miss responses to client

Console# show rule action dscp client cache-hit ?
  all      Display all the patterns for this action
  pattern  Display all the rules with specific type of pattern

Console# show rule action dscp client cache-hit pattern ?
  domain     Regular expression to match with the domain name
  dst-ip     Destination IP address of the request
  dst-port   Destination port number
  mime-type  Regular expression to match with MIME type
  src-ip     Source IP address of the request
  url-regex  Regular expression to substring match with the URL
```

The following examples illustrate DSCP information obtained using the **show** command.

```
Console# show rule action dscp client cache-hit pattern src-ip
Rules Template Configuration
----------------------------
Rule Processing Disabled
rule dscp client cache-hit set-tos min-monetary-cost src-ip 10.1.1.1 255.255.255.0

Console# show stat rule action dscp client cache-hit pattern src-ip

Rules Template Statistics
-------------------------
Rule hit count = 0   Rule:rule dscp client cache-hit set-tos min-monetary-cost src-ip
10.1.1.1 255.255.255.0
```

Related Commands

bypass static

clear statistics rule

http proxy outgoing

proxy-protocols outgoing exclude

show rule

show statistics rule

show arp

To display the Address Resolution Protocol (ARP) table, use the **show arp** EXEC command.

show arp

Syntax Description

This command has no arguments or keywords.

Defaults

No default behavior or values

Command Modes

EXEC

Examples

```
ContentEngine# show arp
Protocol  Address          Flags      Hardware Addr     Type  Interface
Internet  172.16.55.1      Adj        00:D0:D3:39:6F:BC ARPA  eth0
```

The **show arp** command displays the Internet-to-Ethernet address translation tables of the Address Resolution Protocol. Without flags, the current ARP entry for the host name is displayed.

Table 2-3 describes the fields shown in the display.

Table 2-3 show arp Field Descriptions

Field	Description
Protocol	Type of protocol.
Address	Ethernet address of host name.
Flags	Current ARP flag status.
Hardware Addr	Hardware Ethernet address given as six hexadecimal bytes separated by colons.
Type	Type of wide-area network.
Interface	Type of Ethernet interface.

show authentication

To display the authentication configuration, use the **show authentication** EXEC command.

show authentication {**http-request** | **user**}

Syntax Description

http-request	Displays authentication configuration for HTTP request.
user	Displays authentication configuration for user login to the system.

Defaults

No default behavior or values

Command Modes

EXEC

Examples

The following example output displays the HTTP authentication configuration for the LDAP, RADIUS, and NTLM servers.

```
ContentEngine# show authentication http-request
HTTP Request Authentication via:
--------------------------------
  LDAP server: disabled
RADIUS server: disabled
  NTLM server: disabled
```

This example output displays the authentication configuration for local and TACACS+ the user login.

```
ContentEngine# show authentication user
Login Authentication:         Console/Telnet Session
----------------------------- ----------------------
local                         enabled (primary)
tacacs                        disabled

Configuration Authentication: Console/Telnet Session
----------------------------- ----------------------
local                         enabled (primary)
tacacs                        disabled
```

Related Commands

authentication configuration

authentication login

show http authentication

show statistics authentication

clear statistics authentication

show http authentication

show boomerang

To display the domain configuration of boomerang content routing on the Content Engine, enter the **show boomerang** EXEC command.

show boomerang [**domain** *domainname*]

Syntax Description

domain	(Optional) Displays the configuration for the boomerang domain.
domainname	Name of boomerang domain.

Defaults

No default behavior or values

Command Modes

EXEC

Examples

```
ContentEngine# show boomerang
Boomerang Configuration:
------------------------
  Boomerang agent is disabled

  Logging the result of races is disabled
```

Related Commands

boomerang

boomerang send-packet

show bypass

To display bypass configuration information, use the **show bypass** EXEC command.

show bypass [**list**] [**statistics** {**auth-traffic** | **load**}] [**summary**]

Syntax Description

list	(Optional) Bypass list entries.
statistics	(Optional) IP bypass statistics.
auth-traffic	Authenticated traffic bypass statistics.
load	Load bypass statistics.
summary	(Optional) Summary of bypass information.

Defaults

No default behavior or values

Command Modes

EXEC

Examples

```
ContentEngine# show bypass

Total number of HTTP connections bypassed = 3
        Connections bypassed due to system overload           = 0
        Connections bypassed due to authentication issues     = 3
        Connections bypassed to facilitate error transparency = 0
        Connections bypassed due to static configuration      = 0

Total number of entries in the bypass list = 2
        Number of Authentication bypass entries = 0
        Number of Error bypass entries          = 0
        Number of Static Configuration entries  = 2

ContentEngine# show bypass list

         Client                Server       Entry type
         ------                ------       ----------
   172.16.11.11:0         any-server:0    static-config
     any-client:0       172.31.23.23:0    static-config

ContentEngine# show bypass statistics auth-traffic

Authentication bypass statistics
--------------------------------
HTTP connections bypassed due to authentication = 0
Number of authentication bypass entries         = 0

ContentEngine# show bypass statistics load

Load bypass statistics
----------------------
Load Bypass is enabled
System bypass mode - not available
Number of bypassed buckets not available
Number of bypassed connections not available
```

```
Number of transitions from Bypass mode to Normal mode   = 0
Number of transitions from Normal mode to Bypass mode   = 0

MODULE          Normal     Overload   Inundated  Cum Secs   Current State
------          ------     --------   ---------  --------   -------------
load monitor    1          0          0          0          NORMAL

ContentEngine# show bypass summary

Total number of HTTP connections bypassed = 0
        Connections bypassed due to system overload               = 0
        Connections bypassed due to authentication issues         = 0
        Connections bypassed due to facilitate error transparency = 0
        Connections bypassed due to static configuration          = 0

Total number of entries in the bypass list = 0
        Number of Authentication bypass entries = 0
        Number of Error bypass entries          = 0
        Number of Static Configuration entries  = 0
```

Related Commands

bypass

show bypass statistics

clear bypass

show cdp

To display Cisco Discovery Protocol (CDP) configuration information, use the **show cdp** EXEC command.

show cdp [**entry** *neighbor* {*protocol* | *version*} | **holdtime** | **interface** {**fastEthernet** *slot/port* | **gigabitEthernet** *slot/port*} | **neighbors** {**detail** | **fastEthernet** {*slot/port* | *detail*} | **gigabitEthernet** {*slot/port* | *detail*}} | **run** | **timer** | **traffic**]

Syntax Description

entry	(Optional) Displays information for specific neighbor entry.
neighbor	Name of CDP neighbor entry.
protocol	CDP Protocol.
version	CDP version.
holdtime	(Optional) Displays length of time that CDP information is held by neighbors.
interface	(Optional) Displays interface status and configuration.
fastEthernet	Displays Fast Ethernet configuration.
slot/port	Fast Ethernet slot (0–3) and port number.
gigabitEthernet	Displays Gigabit Ethernet configuration.
slot/port	Gigabit Ethernet slot (1–2) and port number.
neighbors	(Optional) Displays CDP neighbor entries.
detail	Displays detailed neighbor entry information.
fastEthernet	Displays Fast Ethernet information.
slot/port	Neighbor Fast Ethernet slot (0–3) and port number.
detail	Detailed neighbor Fast Ethernet network information.
gigabitEthernet	Displays neighbor Gigabit Ethernet information.
slot/port	Neighbor Gigabit Ethernet slot (1–2) and port number.
detail	Detailed Gigabit Ethernet neighbor network information.
run	(Optional) Displays the CDP process status.
timer	(Optional) Displays the time when CDP information is resent to neighbors.
traffic	(Optional) Displays CDP statistical information.

Defaults

No default behavior or values

Command Modes

EXEC

Examples

The following examples display CDP information regarding how frequently CDP packets are resent to neighbors, the length of time that CDP packets are held by neighbors, disabled status of CDP Version 2 multicast advertisements, CDP Ethernet interface ports, and general CDP traffic information.

```
ContentEngine# show cdp
Global CDP information:
        Sending CDP packets every 60 seconds
        Sending a holdtime value of 180 seconds
        Sending CDPv2 advertisements is not enabled
ContentEngine# show cdp holdtime
180 seconds
ContentEngine# show cdp interface fastEthernet 1/2
FastEthernet0/1 is down, line protocol is down
  Encapsulation ARPA
  Sending CDP packets every 60 seconds
  Holdtime is 180 seconds
ContentEngine# show cdp neighbors fastEthernet 1/2 detail
ContentEngine# show cdp traffic
CDP counters :
        Total packets Output: 20197, Input: 80840
        Hdr syntax: 0, Chksum error: 0, Encaps failed: 0
        No memory: 0, Invalid packet: 0, Fragmented: 0
        CDP version 1 advertisements Output: 20197, Input: 60662
        CDP version 2 advertisements Output: 0, Input: 20178
```

Related Commands

cdp enable

cdp full-duplex

cdp half-duplex

no cdp

clear cdp counters

clear cdp table

show cfs

To display information about your cache file system (cfs), use the **show cfs** EXEC command.

show cfs {**statistics** | **volumes**}

Syntax Description

statistics	Displays the cfs statistics.
volumes	Displays the cfs volumes.

Defaults

No default behavior or values

Command Modes

EXEC

Examples

The cfs caches HTTP and FTP objects. The following **show cfs statistics** command summarizes disk statistics for logical Disk 0 and Disk 1.

```
ContentEngine# show cfs statistics
CFS statistics
--------------
Disk 0
Total disk space               = 18119393280
Total disk space used          = 941621248
Total disk objects read        = 0
Total disk objects write       = 0
Total bytes of disk read       = 0
Total bytes of disk write      = 0
Disk read errors               = 0
Disk write errors              = 0
Disk 1
Total disk space               = 18119393280
Total disk space used          = 485490688
Total disk objects read        = 0
Total disk objects write       = 0
Total bytes of disk read       = 0
Total bytes of disk write      = 0
Disk read errors               = 0
Disk write errors              = 0
```

The **show cfs volumes** command output displays different disk names and does not indicate whether or not a cfs partition is mounted. The cfs size is displayed in kilobytes. For example:

```
Console# show cfs volumes
cfs 00:/dev/raw/raw1        17783224KB
cfs 01:/dev/raw/raw2        17783224KB
cfs 02:/dev/raw/raw3        17783224KB
cfs 03:/dev/raw/raw4        17783224KB
```

Related Commands

cfs

show disks

show disk-partitions

show statistics cfs

show clock

To display the system clock, use the **show clock** EXEC command.

show clock [**detail**]

Syntax Description

detail	(Optional) Displays detailed information; indicates the clock source (NTP) and the current summer-time setting (if any).

Defaults

No default behavior or values

Command Modes

EXEC

Examples

The following example shows date and time information, such as day of the week, month, time (hh:mm:ss), and year in Greenwich mean time (GMT).

```
ContentEngine# show clock
Wed Apr 28 20:52:48 1999 GMT
```

The following example shows optional detail date and time information, including local time relative to GMT. In addition to the information shown in the previous example, **show clock detail** provides the Universal Coordinated Time (UTC) offset, standard time zone, daylight saving time zone.

```
ContentEngine# show clock detail
Tue Jun 1 14:48:18 1999 GMT

Tue Jun 1 07:48:18 1999 LocalTime
Epoch: 928248498 seconds
UTC offset: -25200 seconds (-7 hr 0 min)
timezone: PST
summerzone: PDT
summer offset: 0 minutes
daylight: summer
```

Related Commands

clock clear

clock save

clock set

show debugging

To display the state of each debugging option, use the **show debugging** EXEC command.

show debugging

Syntax Description

This command has no arguments or keywords.

Defaults

No default behavior or values

Usage Guidelines

This command only displays the type of debugging enabled, not the specific subset of the command.

Command Modes

EXEC

Examples

In the following example, the **debug icp client** coupled with the **show debugging** commands displays that ICP debugging is enabled, but it does not indicate whether debugging is monitoring ICP client or server packet transfer.

```
ContentEngine# debug icp client
ContentEngine# show debugging
Debug icp (client) is on
```

Related Commands

debug

no debug

undebug

show disks

To view information about your disks, use the **show disks** EXEC command.

show disks [**configured** | **current** | **details** | **raid-info**]

Syntax Description

configured	(Optional) Displays new configurations after reboot.
current	(Optional) Displays currently effective configurations.
details	(Optional) Displays currently effective configurations with more details.
raid-info	(Optional) Displays physical disk information for the CDM-4650 RAID controller.

Defaults

No default behavior or values

Command Modes

EXEC

Usage Guidelines

The **show disks** command displays the names of the disks currently attached to the Content Engine.

The **show disks configured** command displays the percentage or amount of disk space allocated to each file system, instead of the names of the disks, after reboot.

```
ContentEngine# show disks configured
SYSFS                  10%
CFS                    30%
MEDIAFS                30%
ECDNFS                 30%
```

Examples

The following example displays the logical names of the disks.

```
ContentEngine# show disks
disk00 (scsi bus 0, unit 0, id 0)
disk01 (scsi bus 0, unit 1, id 0)
```

Related Commands

disk partition

disk prepare

show disk-partitions

show dns-cache

To display DNS cache information, use the **show dns-cache** EXEC command.

show dns-cache

Syntax Description

This command has no arguments or keywords.

Defaults

No default behavior or values

Command Modes

EXEC

Usage Guidelines

Use this command to check the DNS cache status and cache size. The cache status is displayed as configured and online when the domain name server is online. The DNS cache goes online when the **ip name-server** command is configured and goes offline when the last IP name server configuration is deleted with the **no ip name-server** *ip-address* command.

Examples

The following example lists the DNS cache status and size.

```
ContentEngine# show dns-cache
DNS cache status : CONFIGURED and ONLINE
Max cache size   : 16384
Hash table size  : 4093
```

show ecdn

To view information about the state of the Enterprise CDN (E-CDN) application, use the **show ecdn command** in EXEC mode.

show ecdn

Syntax Description

This command has no arguments or keywords.

Defaults

No default behavior or values

Command Modes

EXEC

Examples

This example displays information about the state of the E-CDN application when it is disabled.

```
ContentEngine# show ecdn
ECDN content engine is not running; server 'bootnet' is 11314, 'checkup' is 0.
If you'd like to turn on ECDN, use config command 'ecdn'.
If you have, then the watchdog will restart in 10 min.
```

This example displays information about the state of the E-CDN application when it is enabled.

```
ContentEngine# show ecdn
ECDN content engine is running, state is online, Running normally.
For more information, point your browser to CDM 172.31.76.76
```

Related Commands

ecdn cdm ip

ecdn enable

no ecdn cdm ip

no ecdn enable

ecdn force-downgrade

show ecdnfs volumes

To view information about your Enterprise CDN (E-CDN) file system (ecdnfs), use the **show ecdnfs volumes** command in EXEC mode.

show ecdnfs volumes

Syntax Description

This command has no arguments or keywords.

Defaults

No default behavior or values

Command Modes

EXEC

Usage Guidelines

The ecdnfs stores pre-positioned E-CDN media content on disk. The **show ecdnfs volumes** command displays the ecdnfs volume number, its directory, and its size in kilobytes.

Examples

This example output provides information about the ecdnfs.

```
ContentEngine# show ecdnfs volumes
ecdnfs 00: /sonoma/state      7340029KB
```

Related Commands

show disks

show error-handling

To display the error-handling configuration, use the **show error-handling** EXEC command.

show error-handling

Syntax Description

This command has no arguments or keywords.

Defaults

No default behavior or values

Command Modes

EXEC

Examples

```
ContentEngine(config)# error-handling
  reset-connection  Reset TCP connection without specifying any error
  send-cache-error  Send Cache Error
  transparent       Make the Cache look transparent to the client

ContentEngine(config)# error-handling send-cache-error

ContentEngine# show error-handling
error-handling is set to send-cache-error
```

Related Commands

error-handling

show flash

To display the Flash memory version and usage information, use the **show flash** EXEC command.

show flash

Syntax Description

This command has no arguments or keywords.

Defaults

No default behavior or values

Command Modes

EXEC

Examples

```
ContentEngine# show flash
ACNS software version (disk-based code): ACNS-4.1.0-b12

System image on flash:
Version: 4.1.0

System flash directory:
System image: 72 sectors
120 sectors total, 29 sectors free.

Bootloader on flash:built Mon Dec 10 07:38:18 PST 2001
```

show ftp

To display the caching configuration of the File Transfer Protocol (FTP), use the **show ftp** EXEC command.

show ftp

Syntax Description

This command has no arguments or keywords.

Defaults

No default behavior or values

Command Modes

EXEC

Examples

This example lists the caching configuration of FTP.

```
ContentEngine# show ftp

FTP heuristic age-multipliers: directory-listing 30% file 60%
Maximum Time To Live in days : directory-listing 3 file 7
Minimum Time To Live in minutes: 60
No objects are revalidated on every request.
Serve-IMS without revalidation if...
Directory listing object is less than 50% of max age
File object is less than 80% of max age
Incoming Proxy-Mode:
Servicing Proxy mode FTP connections on ports: 22 23 88 66 48 488 449 90
Outgoing Proxy-Mode:
Not using outgoing proxy mode.
Maximum size of a cacheable object is unlimited.
ContentEngine#
```

To show the FTP caching statistics, enter the following command.

```
ContentEngine# show statistics ftp
ims          If-Modified-Since statistics
object       Object statistics
requests     Request statistics
savings      Savings statistics
errors       error statistics
```

To clear the FTP caching statistics, enter the following command.

```
ContentEngine# clear statistics ftp
```

Related Commands

ftp

show statistics ftp

clear statistics ftp

show gui-server

To display the current port assignment and operational status of the Cache software management graphical user interface (GUI) server, use the **show gui-server** EXEC command.

show gui-server

Syntax Description

This command has no arguments or keywords.

Defaults

No default behavior or values

Command Modes

EXEC

Examples

Many parameters are configurable by making entries into the management tropical user interface. The **show gui-server** command in the following example displays whether or not the graphical user interface is enabled and its listener port.

```
ContentEngine# show gui-server
GUI Server is enabled
Listen on port 8001
```

Related Commands

gui-server

show hardware

To display system hardware status, use the **show hardware** EXEC command.

show hardware

Syntax Description

This command has no arguments or keywords.

Defaults

No default behavior or values

Command Modes

EXEC

Examples

The following example lists the system hardware status, including version number, startup date and time, run time since startup, microprocessor type and speed, amount of physical memory available, and a list of disk drives.

```
ContentEngine# show hardware
Application and Content Networking Software (ACNS)
Copyright (c) 1999-2001 by Cisco Systems, Inc.
Application and Content Networking Software Release 4.1.0 (build b14 Jan  3 2002
)
Version: ce507-4.1.0

Compiled 13:10:04 Jan  3 2002 by acme
Compile Time Options: PP

System was restarted on Fri Jan  4 20:00:19 2002.
The system has been up for 1 hour, 52 minutes, 32 seconds.

Core CPU is GenuineIntel Pentium III (Coppermine) (rev 8) running at 598MHz.
246 Mbytes of Physical memory.

List of disk drives:
disk00: Normal          (h00 c00 i00 100)    17499MB( 17GB)
        disk00/03: ECDNFS         4256MB(  4GB) mounted at /sonoma/state
        disk00/04: SYSFS          1418MB(  1GB) mounted at /local/local1
        disk00/05: CFS            4256MB(  4GB)
        disk00/06: MEDIAFS        4256MB(  4GB) mounted at /media/media1
        System use:               3308MB(  3GB)
        FREE:                        0MB(  0GB)
disk01: Not present
```

Related Commands

show version

show hosts

To view the hosts on your Content Engine, use the **show hosts** EXEC command.

show hosts

Syntax Description

This command has no arguments or keywords.

Defaults

No default behavior or values

Command Modes

EXEC

Examples

The following **show hosts** example lists the name servers and their corresponding IP addresses. It also lists the host names, their corresponding IP addresses, and their corresponding aliases (if applicable) in a host table summary.

```
ContentEngine# show hosts
Domain name = cisco.com

Name Servers:
-----------
10.2.2.3
172.27.2.111

Host Table:
hostname         inet address       aliases
--------         ------------       -------
localhost        172.27.1.5
ContentEngine    172.28.117.254
```

Related Commands

show trusted-host

trusted host

show http

To display the HTTP caching parameters, use the **show http** EXEC command.

show http {**age-mult** | **all** | **anonymizer** | **append** | **authenticate-strip-ntlm** | **authentication** | **cache-authenticated** | **cache-cookie** | **cache-on-abort** | **client-no-cache-request** | **cluster** | **object** | **persistent-connections** | **proxy** | **reval-each-request** | **strict-request-content-length-checking** | **ttl**}

Syntax Description

age-mult	HTTP/1.0 caching heuristic modifiers.
all	Displays all HTTP-related caching configurations.
anonymizer	HTTP anonymizer configuration.
append	Configuration of HTTP headers appended by the Content Engine.
authenticate-strip-ntlm	NT LAN Manager (NTLM) authentication header configuration.
authentication	HTTP authentication parameters.
cache-authenticated	Configuration for caching of authenticated web objects.
cache-cookie	Configuration for caching of web objects with associated cookies.
cache-on-abort	Configuration of cache-on-abort.
client-no-cache-request	Configuration for managing of no-cache requests.
cluster	Configuration of cache cluster.
object	Configuration of HTTP object.
persistent-connections	Configuration of persistent connections.
proxy	Proxy-mode configuration.
reval-each-request	Revalidation configuration for every request.
strict-request-content-length-checking	Strict request content length checking configuration.
ttl	Time To Live for objects in cache.

Defaults

No default behavior or values

Command Modes

EXEC

Examples

The following **show http all** command example lists the configuration and status of HTTP.

```
ContentEngine# show http all
Basic authenticated objects are not cached.
NTLM authenticated objects are not cached.
HTTP heuristic age-multipliers: text 30% binary 60%
Serve-IMS without revalidation if...
  Text object is less than 50% of max age
  Binary object is less than 80% of max age
Objects with associated cookies are not cached
Client no-cache requests are retrieved from the origin server
Cache on abort feature is disabled
```

```
Objects will always continue to be cached on a client abort
  Maximum threshold is disabled
  Minimum threshold is disabled
  Percent threshold is disabled
Maximum time to live in days: text 3 binary 7
Minimum time to live for all objects in minutes: 5
Objects are not revalidated on each request
Incoming Proxy-Mode:
  Not servicing incoming proxy mode connections.
Outgoing Proxy-Mode:
  Not using outgoing proxy mode.

 Monitor Interval for Outgoing Proxy Servers is 60 seconds

 Timeout period for probing Outgoing Proxy Servers is 300000 microseconds

 Use of Origin Server upon Proxy Failures is disabled.
Persistent connection is enabled and set to all
Persistent connection timeout is 600 seconds
WWW-Authenticate headers containing NTLM authentication are preserved
Append Via-header is disabled
Append x-forward header is disabled
No host configured to receive Proxy-Authorization header
No host configured to receive WWW-Authorization header
Maximum size of a cacheable object is unlimited
Requested Object URL validation is enabled
HTTP anonymizer is disabled
Healing client is disabled

Timeout for responses = 0 seconds
Max number of misses allowed before stop healing mode = 0
Port number for healing request/response = 14333
Http-port to forward http request to healing server = 80

HTTP Authentication:
        Authentication Header : Based on URL syntax
        Authentication Cache Timeout : 480 (minutes)
        Authentication Cache Maximum entries: 8000
Strict request content length checking disabled
```

The **show http cluster** command displays **max-delay**, **misses**, **http-port**, and **heal-port** values. In the first example, the values are set to 0 and the healing client is disabled.

```
Console(config)# show http cluster
Healing client is disabled

Timeout for responses = 10 seconds
Max number of misses allowed before stop healing mode = 0
Port number for healing request/response = 14333
Http-port to forward http request to healing server = 80
```

In this example the healing client is enabled.

```
Console(config)# show http cluster
Healing client is enabled

Timeout for responses = 10 seconds
Max number of misses allowed before stop healing mode = 999
Port number for healing request/response = 14333
Http-port to forward http request to healing server = 80
```

The following **show http proxy** command example shows the IP address and port numbers of the HTTP incoming and outgoing proxy modes.

```
ContentEngine# show http proxy
Incoming Proxy-Mode:
  Servicing Proxy mode HTTP connections on port: 8080
Outgoing Proxy-Mode:
  Directing request to proxy server at 10.1.1.1 port 7777
```

Related Commands

http

show statistics http

proxy-protocols

show http proxy

clear statistics http

show http-authcache

To display authentication cache configuration information, use the **http-authcache** EXEC command.

show http-authcache

Syntax Description

This command has no arguments or keywords.

Defaults

No default behavior or values

Command Modes

EXEC

Examples

```
ContentEngine# show http-authcache
The authcache is empty
```

Related Commands

clear statistics http-authcache

show statistics http-authcache

show https

To display HTTPS proxy status and port policies, use the **show https** EXEC command.

show https {**all** | **destination-port** | **proxy**}

Syntax Description

all	Displays all HTTPS configuration parameters.
destination-port	Displays destination port restrictions.
proxy	Displays proxy-mode configuration.

Defaults

No default behavior or values

Command Modes

EXEC

Examples

```
ContentEngine# show https proxy
Incoming HTTPS proxy:
  Servicing Proxy mode HTTPS connections on port 9090
Outgoing HTTPS proxy:
  Directing request to proxy server at 1.1.1.2 port 8888
```

Related Commands

proxy-protocols

show statistics https

show icp

To display the Internet Cache Protocol (ICP) client, root, or server information, use the **show icp** EXEC command.

show icp {**client** | **root** | **server**}

Syntax Description

client	Displays ICP client detailed information.
root	Displays ICP brief client/server information.
server	Displays ICP server detailed information.

Defaults

No default behavior or values

Command Modes

EXEC

Examples

The following examples of **show icp** commands list the cache parameters of the client, root, and client/server root, respectively.

```
ContentEngine# show icp client
ICP client is disabled
max wait for replies = 2 seconds
remove from wait list after 20 failures
local_domain ""
Number of remote servers = 0  "

ContentEngine# show icp root
ICP client is disabled
max wait for replies = 2 seconds
remove from wait list after 20 failures
local_domain ""
Number of remote servers = 0
ICP server is disabled
Listen on port 3130
Number of remote clients = 0

ContentEngine# show icp server
ICP server is ebabled
Listen on port 3130
HTTP proxy server for ICP on port 3128
```

Related Commands

icp client

icp server

show inetd

To display the status of TCP/IP services, use the **show inetd** global configuration command.

show inetd

Syntax Description

This command has no arguments or keywords.

Defaults

No default behavior or values

Command Modes

Global configuration

Examples

This **show inetd** command example displays the enabled or disabled status of TCP/IP services.

```
ContentEngine# show inetd
Inetd service configurations:
ftp             enable
rcp             disabled
tftp            disabled
```

Related Commands

inetd

show interface

To display hardware interface information, use the **show interface** EXEC command.

show interface {**FastEthernet** *slot/port* | **GigabitEthernet** *slot/port* | **scsi** *number*}

Syntax Description

FastEthernet	Selects Fast Ethernet interface.
slot/port	Slot and port number for selected interface. Slot range is 0–3; port range is 0–3. The slot number and port number are separated with a forward slash character (/).
GigabitEthernet	Selects Gigabit Ethernet interface.
slot/port	Slot and port number for selected interface. Slot range is 0–3; port range is 0–3. The slot number and port number are separated with a forward slash character (/).
scsi	Selects SCSI interface.
number	SCSI device number (0–20).

Defaults

No default behavior or values

Command Modes

EXEC

Examples

```
ContentEngine# show interface scsi 0
Max Transfer Size: 33554432
Sync: yes
Wide: yes
ContentEngine# # show interface FastEthernet 0/1
Type:Ethernet
Ethernet address:01:64:FE:D0:99
Maximum Transfer Unit Size:1500
Metric:1
Packets Received:0
Input Errors:0
Input Packets Dropped:0
Input Packets Overruns:0
Input Packets Frames:0
Packet Sent:0
Output Errors:0
Output Packets Dropped:0
Output Packets Overruns:0
Output Packets Carrier:0
Output Queue Length:100
Collisions:0
Interrupts:10
Base address:0x6000
```

Related Commands

interface

show running-config

show ip routes

To display the IP routing table, use the **show ip routes** EXEC command.

show ip routes

Syntax Description

routes	Displays routing table.

Defaults

No default behavior or values

Command Modes

EXEC

Examples

```
ContentEngine# show ip routes
Destination      Gateway          Netmask
---------------- ---------------- ----------------
172.16.55.190    0.0.0.0          255.255.255.255
172.16.55.0      0.0.0.0          255.255.255.0
0.0.0.0          172.16.55.1      0.0.0.0
```

Related Commands

ip route

show ldap

To display Lightweight Directory Access Protocol (LDAP) parameters, use the **show ldap** EXEC command.

show ldap

Syntax Description

This command has no arguments or keywords.

Defaults

No default behavior or values

Command Modes

EXEC

Examples

```
ContentEngine# show ldap
LDAP parameters:
        State:          Disabled
        Allow mode:     Enabled
        Base DN:        <none>
        Filter:         <none>
        Timeout:        5 seconds
        UID Attribute:  uid
        Primary :       <none>
        Secondary :     <none>
        LDAP port :      389
        Administrative DN:<none>
        Administrative Password:<none>
        LDAP version : 2
```

Table 2-4 describes the fields shown in the display.

Table 2-4 show ldap Field Descriptions

Field	Description
State	Displays the state of LDAP (enabled or disabled).
Base DN	Displays base domain name.
Filter	Displays LDAP filter for authentication group.
Timeout	Displays wait time in seconds for an LDAP server to reply.
UID Attribute	Displays the user ID attribute on the LDAP server.
Primary	Displays host as the primary host.
Secondary	Displays host as the secondary host.

Table 2-4 show ldap Field Descriptions (continued)

Field	Description
LDAP port	Displays the TCP port for the LDAP authentication server.
Administrative DN	Displays the administrative distinguished name.
Administrative Password	Displays the administrative password.
LDAP version	Displays the current version of LDAP.

Related Commands **ldap server**

show logging

To display the system message log configuration, use the **show logging** EXEC command.

show logging

Syntax Description

This command has no arguments or keywords.

Defaults

No default behavior or values

Command Modes

EXEC

Examples

```
ContentEngine# show logging
Syslog to host is disabled
Priority for host logging is set to:  warning

Syslog to console is disabled
Priority for console logging is set to:  warning

Syslog to disk is enabled
Priority for disk logging is set to:  notice
Filename for disk logging is set to:  /local1/syslog.txt

Syslog facility is set to *

Syslog disk file recycle size is set to 500000
```

Related Commands

logging

clear logging

show mediafs

To display the disk name, partition numbers, and size in kilobytes of all volumes with media file system (mediafs) information, use the **show mediafs** EXEC command.

show mediafs volumes

Syntax Description

volumes	Displays media file system volumes.

Defaults

No default behavior or values

Command Modes

EXEC

Usage Guidelines

The media file system caches RealProxy and WMT files. The **show mediafs** command displays the disk volume and its corresponding size.

Examples

```
ContentEngine# show mediafs volumes
disk01/03 :    mounted  size: 14226579
disk07/00 :    mounted  size: 35566448
```

Related Commands

mediafs

mediafs-division

show memory

To display memory blocks and statistics, use the **show memory** EXEC command.

show memory

Syntax Description

This command has no arguments or keywords.

Defaults

No default behavior or values

Command Modes

EXEC

Examples

```
ContentEngine# show memory
Total physical memory    :    1013008 KB
Total free memory        :     516352 KB
Total memory shared      :          0 KB
Total buffer memory      :        284 KB
Total cached memory      :       8044 KB
```

show multicast-client

To display a multicast client configuration and license, use the **show multicast-client** EXEC command.

show multicast-client [**license-agreement**]

Syntax Description		
	license-agreement	(Optional) Displays multicast client license agreement.

Defaults

No default behavior or values

Command Modes

EXEC

Examples

The following **show multicast-client** example displays the current status of the multicast client. If the **license-agreement** option is included in the command string, the full text of the multicast license agreement is displayed.

```
ContentEngine# show multicast-client
multicast client product model: Cisco CE507

multicast client enabled
multicast client running
multicast client end user license agreement accepted
multicast client license key not installed
multicast client evaluation enabled. Estimated 60 days left for evaluation.
```

Related Commands

multicast-client

show ntlm

To display Microsoft Windows NT LAN Manager (NTLM) parameters, use the **show ntlm** EXEC command.

show ntlm

Syntax Description

This command has no arguments or keywords.

Defaults

No default behavior or values

Command Modes

EXEC

Examples

```
ContentEngine# show ntlm
NTLM parameters:
        Primary :       <none>
        Secondary :     <none>
        State:          Disabled
        Domain name:    <none>
```

Related Commands

ntlm server

show ntp

To display the Network Time Protocol (NTP) parameters, use the **show ntp** EXEC command.

show ntp status

Syntax Description

status	Displays NTP status.

Defaults

No default behavior or values

Command Modes

EXEC

Examples

```
ContentEngine# show ntp status
NTP disabled
server list:

ContentEngine# ntp server 172.16.10.80 172.16.10.150

ContentEngine# show ntp status
NTP manually enabled
server list:172.16.10.80 172.16.10.150
     remote           refid      st t when poll reach   delay   offset  disp
==============================================================================
 ntp-sj2.cisco.c .GPS.            1 u   21   64    7     7.23    0.990 1938.23
*ntp-sj1.cisco.c .GPS.            1 u   10   64   17     5.54   -0.226  938.17

ContentEngine# ntp enable cdm
ContentEngine# show ntp status
ntp settings slaved to CDM
server list:172.16.10.80 172.16.10.150

ContentEngine# no ntp enable cdm
ContentEngine# show ntp status
ntp disabled
server list:172.16.10.80 172.16.10.150
```

Table 2-5 describes the fields shown in the display.

Table 2-5 show ntp Field Descriptions

Field	Description
NTP	Indicates whether NTP is enabled or disabled.
remote	Name (first 15 characters) of remote NTP server.
*	In the remote column, identifies the system peer to which the clock is synchronized.
+	In the remote column, identifies a valid or eligible peer for NTP synchronization.
space	In the remote column, indicates peer rejected. (Peer could not be reached or excessive delay in reaching the NTP server.)
x	In the remote column, indicates a false tick and is ignored by the NTP server.

Table 2-5 show ntp Field Descriptions (continued)

Field	Description
-	In the remote column, indicates a reading outside the clock tolerance limits and is ignored by the NTP server.
refid	Clock reference ID to which the remote NTP server is synchronized. In this example, .GPS. is a GPS satellite receiver.
st	Clock server stratum or layer. In this example, stratum 1 is the top layer.
t	Type of peer (**l**ocal, **u**nicast, **m**ulticast, or **b**roadcast).
when	Indicates when the last packet was received from the server.
poll	Time check or correlation polling interval.
reach	Eight-bit reachability register. If the server was reachable during the last polling interval, a 1 is recorded; otherwise, a 0 is recorded. Octal values 377 and above indicate that every polling attempt reached the server.
delay	Estimated delay (in milliseconds) between requestor and server.
offset	Clock offset relative to the server.
disp	Dispersion, or clock jitter.

Related Commands

ntp enable cdm

ntp enable manual

ntp server

clock set

clock timezone

show pre-load

To display information about the preload configuration, use the **show pre-load** EXEC command.

show pre-load

Syntax Description

This command has no arguments or keywords.

Defaults

Maximum number of concurrent requests: 25

Depth level of URL download: 1

Start Time: 00:00

Command Modes

EXEC

Examples

```
ContentEngine# show pre-load
Preloading is disabled
Number of concurrent sessions: 10
Depth level: 3
URL List File:
Preload will not traverse other domains.

Fetch Domains:
Fetch Suffix:
Fetch Directory:
No-fetch Domain:
No-Fetch Suffix:
No-Fetch Directory:
Scheduling on all days
        Start Time: 00:00
        End Time  : Till completion
```

Table 2-6 describes the fields shown in the display.

Table 2-6 show pre-load Field Descriptions

Field	Description
Preloading is disabled	Indicates whether preload is enabled or disabled.
Number of concurrent sessions	Maximum number of concurrent requests (1–100). The default is 25.
Depth level	Depth level of URL download (1–50). The default is 1.
URL List File	Path name or the FTP or HTTP location where the preload URL list file resides.
Fetch Domains	Domains to be fetched.
Fetch Suffix	Suffixes to be fetched.
Fetch Directory	Directories to be fetched.
No-fetch Domain	Domains to be excluded by object filter.

Table 2-6 show pre-load Field Descriptions (continued)

Field	Description
No-Fetch Suffix	Suffixes to be excluded by object filter.
No-Fetch Directory	Directories to be excluded by object filter.
Scheduling on all days	Preload in scheduling intervals of 1 day, 1 hour or less, or 1 week or less.
Start Time	Preload start time. The default is 00:00.
End Time	Preload end time. If no end time appears, preload defaults and continues until completion.

Related Commands

pre-load

pre-load force

show pre-load statistics

show processes

To display CPU or memory processes, use the **show processes** EXEC command.

show processes [**cpu** | **memory**]

Syntax Description

cpu	(Optional) Displays CPU utilization.
memory	(Optional) Displays memory allocation processes.

Defaults

No default behavior or values

Command Modes

EXEC

Examples

```
ContentEngine# show processes cpu
CPU Usage:
   cpu: 0.27% User,  0.70% System,  0.00% User(nice),  99.03% Idle
  cpu0: 0.27% User,  0.70% System,  0.00% User(nice),  99.03% Idle
------------------------------------------------------------------
 PID  STATE PRI User T  SYS T        COMMAND
----- ----- --- ------ ------ --------------------
    1     S   0    111    258 (init)
    2     S   0      0      0 (kswapd)
    3     S   0      0      0 (kflushd)
    4     S   0      0      0 (kupdate)
  128     S   0      0      0 (mingetty)
  129     S   0      0      0 (mingetty)
  130     S   0      0      0 (mingetty)
  131     S   0      0      0 (mingetty)
  132     S   0      0      0 (mingetty)
  133     S   0      0      0 (mingetty)
  134     S   0      1    126 (exec)
  135     S   0      9    132 (start)
  137     S   0      0      0 (inetd)
  138     S   0     10     13 (dataserver)
  144     S   0      0      0 (parser_server)
  151     S   0      6      1 (parser_server)
  247     S   0      0      1 (syslogd)
  250     S   0      0      1 (wccp)
  251     S   0      0      0 (overload)
  252     S   0      0      8 (cache)
  253     S   0      1      0 (webserver)
  254     S   0      0      0 (snmpced)
  260     S   0      0      0 (cache)
  261     S   0      0      0 (cache)
  347     S   0      0      0 (cache)
  351     S   0      0      0 (cache)
  392     S   0      0      0 (parser_server)
  395     S   0      0      0 (in.telnetd)
  396     S   0      3      8 (exec)
  397     S   0      0      0 (parser_server)
  408     R   0      0      0 (exec_show_proce)
```

```
ContentEngine# show processes memory
Total       Used       Free     Shared    Buffers     Cached
1037320192  508776448  528543744          0     290816    8364032

   PID State    TTY  %MEM    VM Size RSS (pages) Name
------ ----- ------ ----- ---------- ----------- ----
     1     S      0  0.0     380928          53 (init)
     2     S      0  0.0          0           0 (kswapd)
     3     S      0  0.0          0           0 (kflushd)
     4     S      0  0.0          0           0 (kupdate)
   128     S   1025  0.0    1118208         102 (mingetty)
   129     S   1026  0.0    1118208         102 (mingetty)
   130     S   1027  0.0    1118208         102 (mingetty)
   131     S   1028  0.0    1118208         102 (mingetty)
   132     S   1029  0.0    1118208         102 (mingetty)
   133     S   1030  0.0    1118208         102 (mingetty)
   134     S   1088  0.0    1552384         164 (exec)
   135     S      0  0.0    1413120         144 (start)
   137     S      0  0.0    1179648         123 (inetd)
   138     S      0  0.0    1585152         186 (dataserver)
   144     S      0  0.3    4554752         864 (parser_server)
   151     S      0  0.3    4554752         864 (parser_server)
   247     S      0  0.0    1490944         166 (syslogd)
   250     S      0  0.0    1847296         144 (wccp)
   251     S      0  0.0    1462272         142 (overload)
   252     S      0 40.3  422227968      102285 (cache)
   253     S      0  0.3    4239360         767 (webserver)
   254     S      0  0.0    1622016         208 (snmpced)
   260     S      0 40.3  422227968      102285 (cache)
   261     S      0 40.3  422227968      102285 (cache)
   347     S      0 40.3  422227968      102285 (cache)
   351     S      0 40.3  422227968      102285 (cache)
   392     S      0  0.3    4554752         864 (parser_server)
   395     S      0  0.0    1675264         176 (in.telnetd)
   396     S  34816  0.0    1548288         163 (exec)
   397     S      0  0.3    4554752         864 (parser_server)
   409     R  34816  0.0    1589248         144 (exec_show_proce)

ContentEngine# show processes
CPU Usage:
   cpu: 0.27% User,  0.69% System,  0.00% User(nice),  99.04% Idle
  cpu0: 0.27% User,  0.69% System,  0.00% User(nice),  99.04% Idle
------------------------------------------------------------------------
 PID  STATE PRI User T  SYS T        COMMAND
----- ----- --- ------ ------ --------------------
    1     S   0    111    258 (init)
    2     S   0      0      0 (kswapd)
    3     S   0      0      0 (kflushd)
    4     S   0      0      0 (kupdate)
  128     S   0      0      0 (mingetty)
  129     S   0      0      0 (mingetty)
  130     S   0      0      0 (mingetty)
  131     S   0      0      0 (mingetty)
  132     S   0      0      0 (mingetty)
  133     S   0      0      0 (mingetty)
  134     S   0      1    126 (exec)
  135     S   0      9    132 (start)
  137     S   0      0      0 (inetd)
  138     S   0     10     13 (dataserver)
  144     S   0      0      0 (parser_server)
  151     S   0      6      1 (parser_server)
  247     S   0      0      1 (syslogd)
  250     S   0      0      1 (wccp)
  251     S   0      0      0 (overload)
```

```
252     S    0       0       8 (cache)
253     S    0       1       0 (webserver)
254     S    0       0       0 (snmpced)
260     S    0       0       0 (cache)
261     S    0       0       0 (cache)
347     S    0       0       0 (cache)
351     S    0       0       0 (cache)
392     S    0       0       0 (parser_server)
395     S    0       0       0 (in.telnetd)
396     S    0       4       9 (exec)
397     S    0       0       0 (parser_server)
410     R    0       0       0 (exec_show_proce)
```

Table 2-7 describes the fields shown in the display.

Table 2-7 show processes Field Descriptions

Field	Description
Total	Total available memory in bytes.
Used	Memory currently used in bytes.
Free	Free memory available in bytes.
Shared	Shared memory currently used in bytes.
Buffers	Buffer memory currently used in bytes.
Cached	Cache memory currently used in bytes.
CPU Usage	CPU utilization as a percentage for User, System overhead, and Idle.
PID	Process identifier.
STATE	Current state of corresponding processes. R = running. S = sleeping in an interruptible wait. D = sleeping in an uninterruptible wait or swapping. Z = zombie. T = traced or stopped on a signal.
PRI	Priority of processes.
User T	User time utilization.
Sys T	System time utilization.
COMMAND	Process command.
TTY	TTY to which the process is attached. For example, TTY may indicate which processes belong to network Telnet sessions.
%MEM	Percentage of memory used by corresponding processes.
VM Size	Virtual memory size (in bytes) allocated to the corresponding process.
RSS (pages)	Resident set size, which indicates the number of pages the process has in real memory minus three (–3) for administrative purposes. These are the pages that count toward text, data, and stack space, but do not count demand-loaded or swapped-out pages.
Name	Filename of the executable in parentheses.

show proxy-auto-config

To display the state of the browser auto-configuration feature, use the **show proxy-auto-config** EXEC command.

show proxy-auto-config

Syntax Description

This command has no arguments or keywords.

Defaults

No default behavior or values

Command Modes

EXEC

Examples

```
ContentEngine# show proxy-auto-config
Proxy auto-config is disabled.
Proxy auto-config file is NOT installed.
```

Related Commands

proxy-auto-config

show proxy-protocols

To display current global outgoing proxy exclude status and criteria, use the **show proxy-protocols** EXEC command.

show {**all** | **outgoing-proxy** | **transparent**}

Syntax Description

all	Displays all proxy protocols parameters.
outgoing-proxy	Displays global outgoing proxy exceptions.
transparent	Displays transparent mode protocol policies.

Defaults

No default behavior or values

Command Modes

EXEC

Examples

```
ContentEngine# show proxy-protocols all
Transparent mode forwarding policies: default-server
Outgoing exclude domain name: cisco.com
```

Related Commands

proxy-protocols

show radius-server

To display RADIUS information, use the **show radius-server** EXEC command.

show radius-server

Syntax Description

This command has no arguments or keywords.

Defaults

No default behavior or values

Command Modes

EXEC

Examples

```
ContentEngine# show radius-server
Radius Configuration:
---------------------
Radius Authentication is off
    Timeout       = 5
    Retransmit    = 2
    Radius Redirect is off
       There is no URL to authentication failure instructions
    Servers
    -------
```

Related Commands

radius-server

show real-subscriber

To display RealSubscriber configuration and license information, use the **show real-subscriber** EXEC command.

show real-subscriber [**license-agreement**]

Syntax Description

license-agreement	(Optional) Displays RealSubscriber license agreement.

Defaults

No default behavior or values

Command Modes

EXEC

Examples

```
ContentEngine# show real-subscriber
Real Subscriber version: ce507-8.0.1.446

Real Subscriber enabled
Real Subscriber not running
Real Subscriber end user license agreement accepted
Real Subscriber evaluation enabled. Estimated 60 days left for evaluation.
Real Subscriber license key not installed
Real Publisher not configured
```

Related Commands

real-subscriber

show rtsp

To display the Real-Time Streaming Protocol (RTSP) configurations, use the **show rtsp** EXEC command.

show rtsp {**all** | **license-agreement** | **proxy**}

Syntax Description

all	Displays all RTSP configurations.
license-agreement	Displays RealProxy end user license agreement.
proxy	Displays proxy mode configuration.

Defaults

No default behavior or values

Command Modes

EXEC

Examples

The following examples display output of the **show rtsp all** and **show rtsp proxy** EXEC commands. If the **show rtsp license-agreement** command is invoked, the full text of the RTSP license agreement appears.

```
ContentEngine# show rtsp all
Media Types:
  Not servicing Real Media
Real Proxy License Key:
  Not Installed
Media Specific Info:
  Real Proxy IP address: 172.16.193.8
Incoming Proxy-Mode:
  Servicing Proxy mode RTSP connections on port: 554
RTSP Transparent Proxy (WCCP):
  Configured for port: 554
RTSP Transparent Proxy (L4 Switch):
  Not configured.

ContentEngine# show rtsp proxy
Incoming Proxy-Mode:
  Configured Proxy mode RTSP connections on port: 667
RTSP Transparent Proxy (WCCP):
  Configured for port: 554
  WCCP Media-Cache NOT Enabled
RTSP Transparent Proxy (L4 Switch):
  Not configured.
WMT/REAL cache space partition:
  wmt 70%, real 30%
```

Related Commands

wccp media-cache

rtsp proxy

show rule

To display the rules configuration information, use the **show rule** EXEC command. For a more complete explanation of specific rules, see the "rule" section on page 2-147.

show rule {**action** {action-type {**all** | **pattern** pattern-type} | **all**}

Syntax Description

action	Displays all the rules with the same action.
action-type	**block** {**pattern** pattern-type}
	dscp {**client** {**cache-hit** {**match-server** {**pattern** pattern-type} | **set-dscp** *dscpvalue* | **set-tos** *tosvalue*} | **cache-miss** {**match-server** {**pattern** pattern-type} | **set-dscp** *dscpvalue* | **set-tos** *tosvalue*}} | **server** {**match-client** {**pattern** pattern-type} | **set-dscp** *dscpvalue* | **set-tos** *tosvalue*}}
	dscpvalue—Sets DSCP values 0–63 as follows: af11—Sets packets with AF11 DSCP (001010). af12—Sets packets with AF12 DSCP (001100). af13—Sets packets with AF13 DSCP (001110). af21—Sets packets with AF21 DSCP (010010). af22—Sets packets with AF22 DSCP (010100). af23—Sets packets with AF23 DSCP (010110). af31—Sets packets with AF31 DSCP (011010). af32—Sets packets with AF32 DSCP (011100). af33—Sets packets with AF33 DSCP (011110). af41—Sets packets with AF41 DSCP (100010). af42—Sets packets with AF42 DSCP (100100). af43—Sets packets with AF43 DSCP (100110). cs1—Sets packets with CS1 (precedence 1) DSCP (001000). cs2—Sets packets with CS2 (precedence 2) DSCP (010000). cs3—Sets packets with CS3 (precedence 3) DSCP (011000). cs4—Sets packets with CS4 (precedence 4) DSCP (100000). cs5—Sets packets with CS5 (precedence 5) DSCP (101000). cs6—Sets packets with CS6 (precedence 6) DSCP (110000). cs7—Sets packets with CS7 (precedence 7) DSCP (111000). default—Sets packets with default DSCP (000000) ef—Sets packets with EF DSCP (101110).
	tosvalue—Sets ToS value 0–127 as follows: critical—Sets packets with critical precedence (80). flash—Sets packets with flash precedence (48). flash-override—Sets packets with flash override precedence (64). immediate—Sets packets with immediate precedence (32). internet—Sets packets with internetwork control precedence (96). max-reliability—Sets packets with max reliable ToS (2). max-throughput—Sets packets with max throughput ToS (4). min-delay—Sets packets with min delay ToS (8). min-monetary-cost—Sets packets with min monetary cost ToS (1). network—Sets packets with network control precedence (112). normal—Sets packets with normal ToS (0). priority—Sets packets with priority precedence (16).
	enable

	freshness-factor *exp_time* {**pattern** pattern-type}
	no-auth {**pattern** pattern-type}
	no-cache {**pattern** pattern-type}
	no-proxy {**pattern** pattern-type}
	redirect {**pattern** pattern-type}
	refresh {**pattern** pattern-type}
	reset {**pattern** pattern-type}
	rewrite {**pattern** pattern-type}
	selective-cache {**pattern** pattern-type}
	use-proxy {*hostname* | *ip-address*} *port* {**pattern** pattern-type}
	use-proxy-failover {*hostname* | *ip-address*} *port* {**pattern** pattern-type}
	use-server {*hostname* | *ip-address*} *port* {**pattern** pattern-type}
all	Displays all the patterns for this action.
pattern-type	**domain** *LINE*
	dst-ip *d_ipaddress d_subnet*
	dst-port *port*
	mime-type[1] *LINE*
	src-ip *s_ipaddress s_subnet*
	url-regex *LINE*
	header-field {**referer** *LINE* | **request-line** *LINE* | **user-agent** *LINE*}
	url-regsub *regexpr substitute*
all	Displays all the rules.

1. mime-type is an option for freshness-factor, no-cache, and selective-cache actions only.

Defaults

No default behavior or values

Command Modes

EXEC

Examples

```
ContentEngine# show rule all
Rules Template Configuration
----------------------------
Rule Processing Enabled
rule block domain bar.com
rule block domain \.foo.com
rule rewrite url-regsub http://www.ietf.org/rfc/.* http://wwwin-eng.cisco.com/RFC/$1
rule no-cache dst-ip 172.31.120.0 255.255.192.0
rule no-cache url-regex \.*cgi-bin*

ContentEngine# show rule action use-proxy-failover all
Rules Template Configuration
----------------------------
Rule Processing Enabled
rule use-proxy-failover 172.16.0.0 80 domain acme.com
```

```
ContentEngine# show statistics rule action use-proxy-failover all
Rules Template Statistics
-------------------------
Rule hit count = 100 Rule: rule use-proxy-failover 172.16.0.0 80 domain acme.com
```

Related Commands

rule

show statistics rule

clear statistics rule

show running-config

To display the current running configuration information on the terminal, use the **show running-config** EXEC command. This command replaces the **write terminal** command.

show running-config

Syntax Description

This command has no arguments or keywords.

Defaults

No default behavior or values

Command Modes

EXEC

Usage Guidelines

Use this command in conjunction with the **show startup-config** command to compare the information in running memory to the startup configuration used during bootup.

Examples

```
ContentEngine# show running-config
hostname csbu-lab-ce590
!
http proxy incoming 8080
http proxy outgoing preserve-407
!
!
!
ip domain-name cisco.com
!
exec-timeout 60
!
!
!
interface ethernet 0
ip address 172.16.55.190 255.255.255.0
exit
!
!
ip name-server 172.16.2.200
!
ip route 10.0.0.0 0.0.0.0 172.16.55.1
!
!
!
icp client add-remote-server 10.1.1.1 parent icp-port 2222 http-port 888 restrict
 wertw
!
!
!
!
user modify username admin password 1 c4CaLFF29epFd
!
!
```

Related Commands

configure

copy running-config

copy startup-config

show services

To display services-related information, use the **show services** EXEC command.

show services {**ports** [*portnum*] | **summary**}

Syntax Description

ports	Displays services by port number.
portnum	(Optional) Up to eight port numbers (1–65535).
summary	Displays services summary.

Defaults

No default behavior or values

Command Modes

EXEC

Examples

This example displays services information by port number.

```
ContentEngine# show services ports
Service information by port
 ---------------------------
   21    FTP (Reserved)

   22    SSH (Reserved)

   23    Telnet (Reserved)

   42    Host Name Server (Reserved)

   49    Login (Reserved)

   53    DNS (Reserved)

   69    TFTP (Reserved)

   80    Started on Sat Jan  5 01:38:28 2002
         ECDN web server

  161    SNMP (Reserved)

 8001    Started on Sat Jan  5 01:38:28 2002
         GUI

65510    Started on Sat Jan  5 01:38:28 2002
         ECDN real server PNA

65520    Started on Sat Jan  5 01:38:28 2002
         ECDN real server HTTP

65530    Started on Sat Jan  5 01:38:28 2002
         ECDN real server MTR
```

This example displays services summary information.

```
ContentEngine# show services summary

             Service        Ports
-------------------------------------------------
                 GUI         8001
                 DNS           53
                 FTP           21
                 SSH           22
     ECDN web server           80
                SNMP          161
                TFTP           69
               Login           49
              Telnet           23
ECDN real server MTR        65530
ECDN real server PNA        65510
ECDN real server HTTP        65520
    Host Name Server           42
```

show snmp

To check the status of SNMP communications, use the **show snmp** EXEC command.

show snmp {**engine ID** | **group** | **stats**}

Syntax Description

engineID	Displays local SNMP engine identifier.
group	Displays SNMP groups.
stats	Displays SNMP statistics.

Defaults

No default behavior or values

Command Modes

EXEC

Usage Guidelines

This command provides counter information for SNMP operations.

Examples

```
ContentEngine# show snmp stats
Contact: Mary Brown, system admin, mbrown@acme.com 555-1111
Location: Building 2, 1st floor, Lab 1
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
    2048 Maximum packet size
    0 No such name errors
    0 Bad values errors
    0 General errors
    24 Response PDUs
    13 Trap PDUs
```

Table 2-8 describes the fields shown in the display.

Table 2-8 show snmp Field Descriptions

Field	Description
SNMP packets input	Total number of SNMP packets input.
Bad SNMP version errors	Number of packets with an invalid SNMP version.

Table 2-8 show snmp Field Descriptions (continued)

Field	Description
Unknown community name	Number of SNMP packets with an unknown community name.
Illegal operation for community name supplied	Number of packets requesting an operation not allowed for that community.
Encoding errors	Number of SNMP packets that were improperly encoded.
Number of requested variables	Number of variables requested by SNMP managers.
Number of altered variables	Number of variables altered by SNMP managers.
Get-request PDUs	Number of GET requests received.
Get-next PDUs	Number of GET-NEXT requests received.
Set-request PDUs	Number of SET requests received.
SNMP packets output	Total number of SNMP packets sent by the router.
Too big errors	Number of SNMP packets that were larger than the maximum packet size.
Maximum packet size	Maximum size of SNMP packets.
No such name errors	Number of SNMP requests that specified a MIB object that does not exist.
Bad values errors	Number of SNMP SET requests that specified an invalid value for a MIB object.
General errors	Number of SNMP SET requests that failed because of some other error. (It was not a No such name error, Bad values error, or any of the other specific errors.)
Response PDUs	Number of responses sent in reply to requests.
Trap PDUs	Number of SNMP traps sent.

show ssh

To display Secure Shell (SSH) status and configuration information, use the **show ssh** EXEC command.

show ssh

Syntax Description

This command has no arguments or keywords.

Defaults

No default behavior or values

Command Modes

EXEC

Examples

```
ContentEngine# show ssh
SSH server supports SSH1 protocol, ssh service is not enabled.
Currently there are no active ssh sessions.
Number of successful SSH sessions since last reboot: 0
Number of failed SSH sessions since last reboot: 0
SSH key has not been generated or previous key has been removed.
SSH login grace time value is 300 seconds.
Allow 3 password guess(es).
```

Related Commands

ssh-key-generate

sshd

show standby

To display standby interface information, use the **show standby** EXEC command.

show standby

Syntax Description This command has no arguments or keywords.

Defaults No default behavior or values

Command Modes EXEC

Examples

```
ContentEngine(config)# show standby
```

Related Commands **standby**

show startup-config

To display the startup configuration, use the **show startup-config** EXEC command.

show startup-config

Syntax Description

This command has no keywords or arguments.

Defaults

No default behavior or values

Command Modes

EXEC

Usage Guidelines

Use this command to display the configuration used during an initial bootup, stored in nonvolatile random-access memory (NVRAM).

Examples

```
ContentEngine# show startup-config
hostname ContentEngine
!
!
!
!
ip domain-name cisco.com
!
!
!
interface FastEthernet 0/0
ip address 172.16.55.190 255.255.255.0
exit
interface FastEthernet 0/1
shutdown
exit
!
!
ip name-server 172.16.2.200
!
!
!
!
!
!
username admin password 1 .9ENIWf7GjMm2
username admin privilege 15
!
!
```

Related Commands

configure

copy running-config

show running-config

show statistics

To display Content Engine statistics, use the **show statistics** EXEC command.

show statistics {**authentication** | **boomerang** [**domain** *domainname*] | **bypass** [**auth-traffic** | **load** | **summary**] | **cfs** | **dns-cache** | **ftp** | **http** {**cluster** | **ims** | **object** | **performance** | **proxy outgoing** | **requests** | **savings** | **usage**} | **http-authcache** | **https** | **icmp** | **icp** {**client** | **server**} | **ip** | **ldap** | **mediacache real** {**requests** | **savings**} | **netstat** | **ntlm** | **pre-load** | **radius** | **rule** {**action** {action-type {**all** | **pattern** pattern-type} | **all**}} | **services** | **snmp** | **streamstat** | **tacacs** | **tcp** | **transaction-logs** | **udp** | **url-filter** {**N2H2** | **websense**} | **wmt** {**all** | **bytes** | **errors** | **multicast** | **requests** | **savings** | **usage** [**detail** | **summary**]}

Syntax Description

authentication	Displays user authentication statistics.
boomerang	(Optional) Displays boomerang statistics.
domain	Displays statistics of boomerang domain.
domainname	Name of boomerang domain.
bypass	Displays bypass statistics.
auth-traffic	(Optional) Displays authenticated traffic bypass statistics.
load	(Optional) Displays load bypass statistics.
summary	(Optional) Displays a summary of bypass statistics.
cfs	Displays cache file system statistics.
dns-cache	Displays DNS caching statistics.
ftp	Displays FTP caching statistics.
http	Displays HTTP caching statistics.
cluster	Displays HTTP healing mode statistics.
ims	Displays HTTP if-modified-since statistics.
object	Displays HTTP object statistics.
performance	Displays HTTP performance statistics.
proxy outgoing	Displays outgoing proxy monitor statistics.
requests	Displays HTTP requests statistics.
savings	Displays HTTP savings statistics.
usage	Displays HTTP usage statistics.
http-authcache	Displays HTTP authentication cache characteristics.
https	Displays HTTPS statistics.
icmp	Displays Internet Control Message Protocol statistics.
icp	Displays ICP caching statistics.
client	Displays ICP client caching statistics.
server	Displays ICP server caching statistics.
ip	Displays IP statistics.
ldap	Displays LDAP statistics.
mediacache	Displays media caching statistics.
real	Displays RealProxy statistics.
requests	Displays media requests statistics.

savings	Displays media savings statistics.
netstat	Displays Internet socket connections.
ntlm	Displays NTLM statistics.
pre-load	Displays content preloading statistics.
radius	Displays RADIUS statistics.
rule	Selects rule statistics.
action	Displays rule statistics of the specified action.
action-type	Specifies one of the following actions: **block** **dscp** **freshness-factor** **no-auth** **no-cache** **no-proxy** **redirect** **refresh** **reset** **rewrite** **selective-cache** **use-proxy** **use-proxy-failover** **use-server** See the "rule" section on page 2-147 for explanations of actions and patterns.
all	Displays statistics of all the patterns for this action.
pattern	Displays statistics of rules with specified pattern.
pattern-type	Specifies one of the following patterns: **domain** **dst-ip** **dst-port** **mime-type**[1] **src-ip** **url-regex** **header-field** **url-regub** **client** **server** See the "rule" section on page 2-147 for explanations of patterns and actions.
all	Displays statistics of all the rules.
services	Displays services related statistics.
snmp	Displays SNMP statistics.
streamstat	Displays Windows Media streaming connections.
tacacs	Displays TACACS+ statistics.
tcp	Displays TCP statistics.
transaction-logs	Displays transaction log export statistics.
udp	Displays UDP statistics.
url-filter	Displays URL filter statistics.

N2H2	(Optional) Displays N2H2 URL filter statistics.
websense	(Optional) Displays websense URL filter statistics.
wmt	Displays Windows Media Technologies (WMT) statistics.
all	Displays all WMT statistics.
bytes	Displays unicast byte statistics.
errors	Displays error statistics.
multicast	Displays multicast statistics.
requests	Displays unicast request statistics.
savings	Displays savings statistics.
usage	Displays current usage statistics.
detail	(Optional) Displays detailed current usage statistics.
summary	(Optional) Displays summary of usage statistics.

1. mime-type is an option for freshness-factor, no-cache, and selective-cache actions only.

Defaults

No default behavior or values

Command Modes

EXEC

Usage Guidelines

To clear statistics without affecting configurations, use the **clear statistics** command. This sets all counters to zero.

In the **show statistics mediacache** command output, a cache hit is recorded if the requested media content is in the cache even though it may only be partial content. A live split request is considered a hit. A request for noncacheable media content is recorded as a cache miss (such as a demand pass-through or a live pass-through request).

Examples

```
ContentEngine# show statistics authentication
    Authentication  Statistics
    -------------------------------------
    Number of access requests:        5
    Number of access deny responses:  0
    Number of access allow responses: 5

ContentEngine# show statistics boomerang
  DNS packets with unknown domain            0
  HTTP hostname requests dropped             0
  Packets with bogus source address          0
  Packets with bogus length                  0
  Boomerang agent is disabled

ContentEngine# show statistics bypass

Total number of HTTP connections bypassed = 0
        Connections bypassed due to system overload              = 0
        Connections bypassed due to authentication issues        = 0
        Connections bypassed due to facilitate error transparency = 0
        Connections bypassed due to static configuration         = 0
```

```
Total number of entries in the bypass list = 0
        Number of Authentication bypass entries = 0
        Number of Error bypass entries          = 0
        Number of Static Configuration entries  = 0

ContentEngine# show statistics cfs
CFS statistics
--------------
Disk 0
        Total disk space               = 4440719360
        Total disk space used          = 0
        Total disk objects read        = 0
        Total disk objects write       = 0
        Total bytes of disk read       = 0
        Total bytes of disk write      = 0
        Disk read errors               = 0
        Disk write errors              = 0
```

In this example, the **show statistics http cluster** command displays the statistics of the healing client and the healing server.

```
Console(config)# show statistics http cluster
Healing mode max attempts              = 0
Healing mode max latency               = 10
Healing mode current cumulative misses = 0

Healing mode client statistics
------------------------------
Client Requests  Sent     = 0
Client Responses Received = 0
Client Responses Hit      = 0
Client Responses Miss     = 0
Client Responses Error    = 0
Client Responses Timeout  = 0

Healing mode server statistics
------------------------------
Server Requests Received  = 0
Server Responses Sent     = 0
Server Responses Hit      = 0
Server Responses Miss     = 0
Server Responses Error    = 0

ContentEngine# show statistics dns-cache
Max cache size: 10000
----- DNS Cache Statistics -----
Total DNS lookups      :        2
Adds                   :        1
Deletes                :        0
Drains                 :        0
Total Record           :        1
Hits                   :        0
Misses                 :        2
TTL Expired            :        0
ReqLazy Validity Calls :        0
ReqLazy Valid          :        0
ReqLazy Invalid        :        0
Total Accesses         :        2
Outstanding Queries    :        0
Max Outstanding Queries:        1
Lazy Accesses          :        0
Lazy Mismatches        :        0
Lazy Matches           :        0
```

```
Lazy Cache Hits        :        0
Lazy Cache Misses      :        0
Bad Hostname Recs      :        0
Bad Hostname Cache Hits:        0
Bytes Sent             :       35
Bytes Received         :      187
I/O Errors             :        0
DNS response not matching lookup:        0
Server down time       :        0
----- Hostname Hash Statistics -----
total number of members:        1
total number of lkups  :        1
number of lkup hits    :        0
cumu lkups hit cmps    :        0
number of lkup misses  :        1
cumu lkups miss cmps   :        0
----- IP Address Hash Statistics -----
total number of members:        0
total number of lkups  :        0
number of lkup hits    :        0
cumu lkups hit cmps    :        0
number of lkup misses  :        0
cumu lkups miss cmps   :        0

ContentEngine# show statistics ftp
FTP Statistics
--------------

FTP requests Received = 0

FTP Hits
                                Requests Percentage
 Number of hits =                      0        0.0 %
          Bytes =                      0        0.0 %

FTP Misses
                                Requests Percentage
 Number of misses =                    0        0.0 %
            Bytes =                    0        0.0 %

 Requests sent to Outgoing Proxy    = 0
 Requests sent to origin ftp server = 0

ContentEngine# show statistics mediacache real requests
                  Media Cache Statistics - Requests
                                              Total              % of Requests
                            ------------------------------------------------
     Total Received Requests:                     0                         -
            Demand Cache Hit:                     0                       0.0
          Demand Cache Miss:                      0                       0.0
         Demand Pass-Through:                     0                       0.0
                  Live Split:                     0                       0.0
           Live Pass-Through:                     0                       0.

ContentEngine# show statistics mediacache real savings
             Media Cache Statistics - Savings
                        Requests                          Bytes
         ------------------------------------------------------------
  Total:                      0                              0
   Hits:                      0                              0
   Miss:                      0                              0
Savings:                   0.0 %                          0.0 %
```

```
ContentEngine# show statistics tacacs user
TACACS+ Statistics
----------------------------------------------------------
Authentication:
  Number of access requests:                  3
  Number of acess deny responses:             1
Number of acess allow responses:              2

Authorization:
  Number of authorization requests:           1
  Number of authorization failure responses:  0
  Number of authorization success responses:  1
ContentEngine# show statistics url-filter websense
Websense URL Filtering Statistics:
             Lookup requests transmitted = 0
               Lookup requests timed-out = 0
               Lookup responses received = 0
    Lookup responses received with error = 0

            Requests BLOCKed by Websense = 0
               Requests OKed by Websense = 0
ContentEngine# show statistics wmt all
Unicast Requests Statistics
===========================
Total unicast requests received: 6
-----------------------------------

                                Total            % of Total
                                                Unicast Requests
                              ------------------------------------------

    Total Requests served:          6              100.00%

                                Total        % of Total Requests
                              ------------------------------------------

By Type of Content
------------------
             Live content:          3               50.00%
        On-Demand Content:          3               50.00%

By Transport Protocol
---------------------
                     MMSU:          6              100.00%
                     MMST:          0                0.00%
                     HTTP:          0                0.00%

By Source of Content
--------------------
                    Local:          0                0.00%
               Remote MMS:          6              100.00%
              Remote HTTP:          0                0.00%
                Multicast:          0                0.00%

Unicast Bytes Statistics
========================
Total unicast outgoing bytes: 14256853
---------------------------------
                                          Total     % of Total Unicast
                                                     Outgoing Bytes
                              ------------------------------------------
```

```
By Type of Content
------------------
             Live content:             13701789          96.11%
        On-Demand Content:               555064           3.89%

By Transport Protocol
---------------------
                     MMSU:             14256853         100.00%
                     MMST:                    0           0.00%
                     HTTP:                    0           0.00%

Unicast Savings Statistics
==========================
Total bytes saved: 353256
--------------------------
                                        Total       % of Total Bytes
                                                          Saved
                                    ------------------------------------------
By Pre-positioned content:                   0           0.00%
        By Live-splitting:                   0           0.00%
             By Cache-hit:              353256         100.00%

                                        Total         % of Total
                                                   Incoming Live Bytes
                                    ------------------------------------------
Live Splitting
--------------
           Incoming bytes:             13704857         100.00%
           Outgoing bytes:             13701789         100.00%
              Bytes saved:                    0           0.00%

                                        Total       % of Bytes Cache
                                                          Total
                                    ------------------------------------------
Caching
-------
         Bytes cache-miss:              201808          36.36%
          Bytes cache-hit:              353256          63.64%
        Bytes cache-total:              555064         100.00%

     Bytes cache-bypassed:                   0

                                        Total         % of Req Cache
                                                          Total
                                    ------------------------------------------

Cacheable requests
------------------
           Req cache-miss:                   1          33.33%
            Req cache-hit:                   2          66.67%
    Req cache-partial-hit:                   0           0.00%
          Req cache-total:                   3         100.00%

       Req cache-bypassed:                   0

Objects not cached
------------------
           Cache bypassed:                   0
          Exceed max-size:                   0
```

```
Multicast statistics
====================

Total Multicast Outgoing Bytes: 0

Aggregate Multicast Out Bandwidth (Kbps)
------------------------------------------------
        Current:     0.000
            Max:     0.000

Number of Concurrent Active Multicast Sessions
------------------------------------------------
        Current:         0
            Max:         0

List of All Configured Multicast Stations
------------------------------------------------
Total Number of Configured Multicast Stations: 0


Usage Summary
=============
Concurrent Unicast Client Sessions
----------------------------------
        Current:         0
            Max:         1

Concurrent Active Multicast Sessions
------------------------------------
        Current:         0
            Max:         0

Concurrent Remote Server Sessions
---------------------------------
        Current:         0
            Max:         1

Concurrent Unicast Bandwidth (Kbps)
-----------------------------------
        Current:     0.000
            Max:   107.125

Concurrent Multicast Out Bandwidth (Kbps)
-----------------------------------------
        Current:     0.000
            Max:     0.000

Concurrent Bandwidth to Remote Servers (Kbps)
---------------------------------------------
        Current:     0.000
            Max:   107.125

Error Statistics
================
                   Total request errors:         0

  Errors generated by this box
                  Reach MAX connections:         0
                    Reach MAX bandwidth:         0
                     Reach MAX bit rate:         0
                        MMSU under wccp:         0
                      MMSU not allowed:          0
                      MMST not allowed:          0
                     MMSU/T not allowed:         0
```

```
                        HTTP not allowed:          0
 1st tcp pkt error, possible port scan:            0
                              Illegal url:          0
                                No socket:          0
                           Cannot connect:          0
                     Authentication fail:           0
                     Remote server error:           0
                             Client error:          0
                           Internal error:          0
                Local vod file not found:           0
          Local vod file header corrupted:          0
            Local vod file data corrupted:          0
                           Unknown error:           0

Errors generated by remote servers
                    Reach MAX connections:          0
                      Reach MAX bandwidth:          0
                       Reach MAX bit rate:          0
                              Illegal url:          0
                          Invalid request:          0
                                No socket:          0
                           Cannot connect:          0
                        Conection refused:          0
                              Access deny:          0
                      Invalid stream type:          0
                      Remote server error:          0
                           Remote timeout:          0
                       Remote proxy error:          0
                           File not found:          0
                    File header corrupted:          0
                      File data corrupted:          0
                      Remote unknown error:         0

Authentication Retries from Clients:                0
```

Related Commands **clear statistics**

show sysfs

To display system file system (sysfs) information, use the **show sysfs** EXEC command.

show sysfs volumes

Syntax Description		
	volumes	Displays system file system volumes.

Defaults

No default behavior or values

Command Modes

EXEC

Examples

The system file system (sysfs) stores log files, including transaction logs, syslogs, and internal debugging logs. It also stores system image files and operating system files. The following example displays the disk volume number and its size.

```
ContentEngine# show sysfs volumes
disk00/01 :    mounted  size: 21338112
disk01/00 :    mounted  size: 35566448
```

Related Commands

sysfs

disk config sysfs

show tacacs

To display Terminal Access Controller Access Control System (TACACS+) authentication protocol configuration information, use the **show tacacs** EXEC command.

show tacacs

Syntax Description

This command has no arguments or keywords.

Defaults

No default behavior or values

Command Modes

EXEC

Examples

```
ContentEngine# show tacacs
    Login Authentication for Console/Telnet Session: disabled
    Configuration Authentication for Console/Telnet Session: disabled

    TACACS+ Configuration:
    ----------------------
    Key        =
    Timeout    = 5
    Retransmit = 2

    Server                          Status
    ----------------------------    ------
```

Related Commands

tacacs

show tcp

To display Transmission Control Protocol (TCP) configuration information, use the **show tcp** EXEC command.

show tcp

Syntax Description

This command has no arguments or keywords.

Defaults

No default behavior or values

Command Modes

EXEC

Examples

```
ContentEngine# show tcp
==TCP Configuration==
TCP keepalive timeout 300 sec
TCP keepalive probe count 4
TCP keepalive probe interval 75 sec
TCP server R/W timeout 120 sec
TCP client R/W timeout 120 sec
TCP server send buffer 8 k
TCP server receive buffer 32 k
TCP client send buffer 32 k
TCP client receive buffer 8 k
TCP Listen Queue 200
TCP server max segment size 1432
TCP server satellite (RFC1323) disabled
TCP client max segment size 1432
TCP client satellite (RFC1323) disabled
```

Related Commands

tcp

show tech-support

To view information necessary for Cisco's Technical Assistance Center (TAC) to assist you, use the **show tech-support** EXEC command.

show tech-support [**page**]

Syntax Description

page	(Optional) Pages through output.

Defaults

No default behavior or values

Command Modes

EXEC

Usage Guidelines

Use this command to view system information necessary for TAC to assist you with your Content Engine. We recommend that you log the output to a disk file.

Examples

This example shows the types of information available about ACNS software. Because the **show tech-support** output is comprehensive and can be extensive, only excepts are shown in the following example.

```
ContentEngine# show tech-support

CPU Usage:
   cpu: 0.39% User,  0.42% System,  0.33% User(nice),  98.86% Idle
  cpu0: 0.39% User,  0.42% System,  0.33% User(nice),  98.86% Idle
------------------------------------------------------------------------
 PID  STATE PRI User T  SYS T        COMMAND
----- ----- --- ------ ------ --------------------
    1     S   0   4386   1706 (init)
    2     S   0      0      0 (keventd)
    3     S  19      0      0 (ksoftirqd_CPU0)
    4     S   0      0      0 (kswapd)
    5     S   0      0      0 (bdflush)
    6     S   0      0      0 (kupdated)
    7     S   0      0      0 (scsi_eh_0)
   45     S   0   4733   4114 (nodemgr)
   46     S   0      0      0 (syslogd)
   47     R   0     83     65 (dataserver)
  920     S   0      0      0 (login)
  921     S   0    123     68 (inetd)
 1207     S   0      0      0 (parser_server)
 1208     S   0      0      0 (eval_timer_mana)
 1211     S   0     46      1 (parser_server)
 1442     S   0      0      0 (wccp)
 1443     S   0      0      0 (overload)
 1444     S   0      0      0 (standby)
 1445     S   0     13     29 (cache)
 1446     S   0      0      0 (proxy_poll)
 1447     S   0      0      0 (snmpced)
 1448     S   0      0      0 (http_authmod)
 1458     S   0      0      0 (http_authmod)
```

```
 1465    S   0      0      0 (http_authmod)
 1466    S   0      0      0 (http_authmod)
 1467    S   0      0      0 (http_authmod)
 1537    S   0      0      0 (cache)
 1538    S   0      0      0 (unified_log)
 1539    S   0      0      0 (boom_agent)
 1540    S   0      0      1 (webserver)
 1541    S   0      2      2  (mcm)
 1542    S   0      0      0 (cache)
 1543    S   0      0      0 (cache)
 1544    S   0      0      0 (boom_agent)
 1545    S   0      0      0 (boom_agent)
 1546    S   0      0      0 (boom_agent)
 1548    S   0      0      0 (ecdnURL_transla)
 1550    S   0      0      0 (cache)
 1551    S   0      0      0 (cache)
 1556    S   0      0      0 (cache)
 1567    S   0      0      0  (mcm)
 1568    S   0      0      0  (mcm)
 1629    S   0  18982   4140 (crond)
 1936    S   0   1669    611 (bootnet)
 1937    S  10      0      0 (tracknet)
 1938    S  10  33545   5556 (checkup)
 1983    S   0      0      0 (srcpd)
 2023    S   0      1      0 (admin-shell)
 2024    S   0      0      0 (parser_server)
 2150    S   0      0      0 (rsvpd)
 2152    S   0      0      0 (rtspd)
 2153    S   0   1635   1067 (httpsd)
 2164    S   0      0      0 (librarian)
 2167    S   0   1667   2105 (libaux)
 2170    S   0      0      0 (mapper)
 2178    S   0     32     37 (cache)
 2179    S   0      0      0 (router)
 2180    S   0      0      0 (fill)
 2183    S   0      0      0 (remotereq)
 2185    S -20      0      0 (videosvr)
 2188    S   0      9      4 (contentsvr)
 2189    S   0      0      0 (routeraux)
 2190    S   0      0      1 (dfcontrolsvr)
 2226    S   0      0      0 (smbd)
 2228    S   0      0      0 (nmbd)
 2973    Z   0      0      0 (cache)
 8446    S   0      0      0 (httpsd)
 8447    S   0      0      0 (gcache)
18173    S   0      0      0 (in.telnetd)
18174    S   0      0      0 (login)
18175    S   0      2      2 (admin-shell)
18176    S   0      0      0 (parser_server)
19426    S   0      0      0 (httpsd)
19427    S   0      0      0 (httpsd)
19456    Z   0      0      0 (cache)
19503    Z   0     30      3 (crond)
19515    S   0      0      0 (more)
19516    S   0      6     18 (exec_show_tech-)
19553    R   0      0      0 (exec_show_proce)
```

```
------------------ process memory --------------------

     Total       Used       Free     Shared    Buffers     Cached
1050943488  564785152  486158336          0    5222400  475176960

   PID State    TTY  %MEM    VM Size RSS (pages) Name
------ ----- ------ ----- ---------- ----------- ----
     1     S      0  0.0    1146880         119 (init)
     2     S      0  0.0          0           0 (keventd)
     3     S      0  0.0          0           0 (ksoftirqd_CPU0)
     4     S      0  0.0          0           0 (kswapd)
     5     S      0  0.0          0           0 (bdflush)
     6     S      0  0.0          0           0 (kupdated)
     7     S      0  0.0          0           0 (scsi_eh_0)
    45     S      0  0.0    1208320         143 (nodemgr)
    46     S      0  0.0    1630208         194 (syslogd)
    47     R      0  0.0    1974272         238 (dataserver)
   920     S   1088  0.0    1728512         236 (login)
   921     S      0  0.0    1191936         130 (inetd)
  1207     S      0  0.3    4980736         847 (parser_server)
  1208     S      0  0.0    1933312         151 (eval_timer_mana)
  1211     S      0  0.3    4980736         847 (parser_server)
  1442     S      0  0.0    2232320         163 (wccp)
  1443     S      0  0.0    1548288         154 (overload)
  1444     S      0  0.0    1724416         161 (standby)
  1445     S      0  5.9   65646592       15266 (cache)
  1446     S      0  0.0    1957888         173 (proxy_poll)
  1447     S      0  0.1    2097152         290 (snmpced)
  1448     S      0  0.0    1757184         205 (http_authmod)
  1458     S      0  0.0    1757184         205 (http_authmod)
  1465     S      0  0.0    1757184         205 (http_authmod)
  1466     S      0  0.0    1757184         205 (http_authmod)
  1467     S      0  0.0    1757184         205 (http_authmod)
  1537     S      0  5.9   65646592       15266 (cache)
  1538     S      0  0.0    1789952         169 (unified_log)
  1539     S      0  0.0    1392640         165 (boom_agent)
  1540     S      0  0.4   10817536        1164 (webserver)
  1541     S      0  0.0    2150400         251 (mcm)
  1542     S      0  5.9   65646592       15266 (cache)
  1543     S      0  5.9   65646592       15266 (cache)
  1544     S      0  0.0    1392640         165 (boom_agent)
  1545     S      0  0.0    1392640         165 (boom_agent)
  1546     S      0  0.0    1392640         165 (boom_agent)
  1548     S      0  0.2    5713920         723 (ecdnURL_transla)
  1550     S      0  5.9   65646592       15266 (cache)
  1551     S      0  5.9   65646592       15266 (cache)
  1556     S      0  5.9   65646592       15266 (cache)
  1567     S      0  0.0    2150400         251 (mcm)
  1568     S      0  0.0    2150400         251 (mcm)
  1629     S      0  0.0    1187840         137 (crond)
  1936     S      0  0.6    7532544        1605 (bootnet)
  1937     S      0  0.2    3215360         545 (tracknet)
  1938     S      0  0.2    3637248         654 (checkup)
  1983     S      0  0.3    4374528         838 (srcpd)
  2023     S   1088  0.0    2146304         182 (admin-shell)
  2024     S      0  0.3    4980736         847 (parser_server)
  2150     S      0  0.0    1679360         188 (rsvpd)
  2152     S      0  0.3    6217728         881 (rtspd)
  2153     S      0  0.1    2527232         329 (httpsd)
  2164     S      0  0.3    6533120         990 (librarian)
  2167     S      0  0.4    7110656        1144 (libaux)
  2170     S      0  0.3    5955584         863 (mapper)
  2178     S      0  0.3    6135808         927 (cache)
  2179     S      0  0.3    6287360         948 (router)
```

```
  2180     S      0  0.3    5955584         926 (fill)
  2183     S      0  0.3    5832704         852 (remotereq)
  2185     S      0  0.3    8269824         873 (videosvr)
  2188     S      0  0.4    7651328        1196 (contentsvr)
  2189     S      0  0.3    6103040         953 (routeraux)
  2190     S      0  0.4   10272768        1075 (dfcontrolsvr)
  2226     S      0  0.1    3559424         504 (smbd)
  2228     S      0  0.0    2084864         247 (nmbd)
  2973     Z      0  0.0          0           0 (cache)
  8446     S      0  0.1    2506752         327 (httpsd)
  8447     S      0  0.0    1421312         116 (gcache)
 18173     S      0  0.0    1220608         132 (in.telnetd)
 18174     S  34816  0.0    1736704         238 (login)
 18175     S  34816  0.0    2162688         184 (admin-shell)
 18176     S      0  0.3    4980736         847 (parser_server)
 19426     S      0  0.1    2551808         350 (httpsd)
 19427     S      0  0.1    2576384         354 (httpsd)
 19456     Z      0  0.0          0           0 (cache)
 19503     Z      0  0.0          0           0 (crond)
 19515     S  34816  0.0    1163264         109 (more)
 19516     S  34816  0.0    1941504         168 (exec_show_tech-)
 19554     R  34816  0.1    2277376         266 (exec_show_proce)

------------------ system memory --------------------

Total physical memory    :    1026312 KB
Total free memory        :     474692 KB
Total memory shared      :          0 KB
Total buffer memory      :       5100 KB
Total cached memory      :     464040 KB

------------------ interfaces ---------------------

Interface type: FastEthernet  Slot: 0  Port: 0
Type:Ethernet
Ethernet address:00:05:32:02:DD:74
Internet address:172.16.5.234
Broadcast address:172.16.5.255
Netmask:255.255.255.0
Maximum Transfer Unit Size:1500
Metric:1
Packets Received: 513241
Input Errors: 0
Input Packets Dropped: 0
Input Packets Overruns: 0
Input Packets Frames: 0
Packet Sent: 153970
Output Errors: 0
Output Packets Dropped: 0
Output Packets Overruns: 0
Output Packets Carrier: 0
Output Queue Length:100
Collisions: 0
Interrupts:9
Flags:UP BROADCAST RUNNING MULTICAST
Mode:autoselect, full-duplex, 100baseTX
```

show telnet

To display Telnet services configuration, use the **show telnet** EXEC command.

show telnet

Syntax Description

This command has no arguments or keywords.

Defaults

The default value is enabled.

Command Modes

EXEC

Examples

```
Console# show telnet
telnet service is enabled
```

Related Commands

telnet enable

exec-timeout

show tftp-server

To display the Trivial File Transfer Protocol (TFTP) server configuration, use the **show tftp-server** EXEC command.

show tftp-server

Syntax Description

This command has no arguments or keywords.

Defaults

No default behavior or values

Command Modes

EXEC

Examples

```
Console# show tftp-server

        == TFTPD Directory List ==
           /local1/public
```

Related Commands

tftp-server

show transaction-logging

To display the transaction log configuration settings and list of archived transaction log files, use the **show transaction-logging** EXEC command.

show transaction-logging

Syntax Description

This command has no arguments or keywords.

Defaults

No default behavior or values

Command Modes

EXEC

Usage Guidelines

Use the **show transaction-logging** command to display the current settings for the transaction logging feature and list of archived transaction log files.

Examples

```
ContentEngine# show transaction-logging
Transaction log configuration:
---------------------------------------
Logging is enabled.
Logging of ecdn internal communication is disabled.
End user identity is visible.
File markers are enabled.
Archive interval:every-day every 1 hour
Maximum size of archive file:2000000 KB
Log File format is squid.

Exporting files to ftp servers is enabled.
File compression is disabled.
Export interval:every-day every 1 hour

ftp-server       username        directory
10.1.1.1         mylogin         /ftpdirectory

HTTP Caching Proxy Transaction Log File Info
  Working Log file - size :213
                     age:255
  Archive Log file - celog_10.1.1.1_20020131_000000.txt    size:285162
  Archive Log file - celog_10.1.1.1_20020131_010000.txt    size:235218
  Archive Log file - celog_10.1.1.1_20020131_020000.txt    size:186460
  Archive Log file - celog_10.1.1.1_20020131_030000.txt    size:319997
  Archive Log file - celog_10.1.1.1_20020131_040000.txt    size:426165
  Archive Log file - celog_10.1.1.1_20020131_050000.txt    size:215609
  WMT MMS Caching Proxy/Server Transaction Log File Info
  Working Log file - size :541
                     age:225
```

```
Archive Log file - mms_export_10.1.1.1_20020131_000006    size:37622
Archive Log file - mms_export_10.1.1.1_20020131_010011    size:23500
Archive Log file - mms_export_10.1.1.1_20020131_020019    size:48562
Archive Log file - mms_export_10.1.1.1_20020131_030044    size:36821
Archive Log file - mms_export_10.1.1.1_20020131_040050    size:57315
Archive Log file - mms_export_10.1.1.1_20020131_050014    size:28346
```

Related Commands

clear transaction-log

show statistics transaction-logs

transaction-log force

transaction-logs

show trusted-hosts

To display the name of the trusted host, use the **show trusted-hosts** EXEC command.

show trusted-hosts

Syntax Description

This command has no arguments or keywords.

Defaults

No default behavior or values

Command Modes

EXEC

Examples

```
Console# show trusted-hosts

Trusted Host checking: ON
10.33.123.2/C_Medici
192.168.111.1/Procrustes
```

Related Commands

trusted-host

show url-filter

To display URL filter configuration information, use the **show url-filter** EXEC command.

show url-filter

Syntax Description

This command has no arguments or keywords.

Defaults

URL filtering is disabled by default.

Command Modes

EXEC

Examples

```
Console# show url-filter
URL filtering is DISABLED

Local list configurations
=================================
Good-list file name : /local1/good.list
Bad-list file name : /local1/bad.list
Custom message directory :

Websense server configuration
=================================
Websense server IP     : <none>
Websense server port   : 15868
Websense server timeout: 20 (in seconds)
Websense allow mode is ENABLED

N2H2 server configuration
=============================
N2H2 server IP       : <none>
N2H2 server port     : 4005
N2H2 server timeout  : 5 (in seconds)
N2H2 allow mode is ENABLED
```

Related Commands

url-filter

url-filter local-list-reload

clear statistics url-filter N2H2

clear statistics url-filter websense

show url-filter

show statistics url-filter N2H2

show statistics url-filter websense

no url-filter

debug url-filter N2H2

debug url-filter websense

show user

To display user information for a particular user, use the **show user** EXEC command.

show user username *name*

Syntax Description

username	Displays username keyword.
name	Username.

Defaults

No default behavior or values

Command Modes

EXEC

Examples

```
ContentEngine# show user username bwhidney
Username            : bwhidney
Uid                 : 5013
Password            : bSzyydQbSb
Privilege           : super user
```

Related Commands

show users

show users

To display users, use the **show users** EXEC command.

show users {**administrative** | **request-authenticated**}

Syntax Description

administrative	Lists users with administrative privileges.
request-authenticated	Lists users authenticated by HTTP request.

Defaults

No default behavior or values

Command Modes

EXEC

Examples

```
ContentEngine# show users administrative
           UID USERNAME
           0 admin

ContentEngine# show users request-authenticated
           USERNAME   MODE
```

The following example shows the output if no users are authenticated by HTTP request.

```
ContentEngine# show users request-authenticated
There are no users authenticated by HTTP request
```

Related Commands

show user

show version

To display version information about your Content Engine software, use the **show version** EXEC command.

show version

Syntax Description

This command has no arguments or keywords.

Defaults

No default behavior or values

Command Modes

EXEC

Examples

```
ContentEngine# show version
Application and Content Networking Software (ACNS)
Copyright (c) 1999-2001 by Cisco Systems, Inc.
Application and Content Networking Software Release 4.1.0 (build b14 Jan  3 2002
)
Version: ce507-4.1.0

Compiled 13:10:04 Jan  3 2002 by acme
Compile Time Options: PP

System was restarted on Sat Jan  5 01:37:41 2002.
The system has been up for 22 hours, 55 minutes, 46 seconds.
```

show wccp

To display WCCP information, use the **show wccp** EXEC command.

show wccp content-engines

show wccp flows {**custom-web-cache** | **media-cache** | **reverse-proxy** | **web-cache** | **wmt**} [**summary**]

show wccp gre

show wccp modules

show wccp port-list

show wccp routers

show wccp services [**detail**]

show wccp slowstart {**custom-web-cache** | **media-cache** | **reverse-proxy** | **web-cache** | **wmt**}

show wccp status

no show wccp {**content-engines** | **flows** {**custom-web-cache** | **media-cache** | **reverse-proxy** | **web-cache** | **wmt**} [**summary**] | **gre** | **modules** | **port-list** | **routers** | **services** [**detail**] | **slowstart** {**custom-web-cache** | **media-cache** | **reverse-proxy** | **web-cache** | **wmt**} | **status**}

Syntax Description

content-engines	Displays WCCP Content Engine information.
flows	Displays WCCP packet flow count by bucket.
custom-web-cache	Displays custom web caching service packet flows.
media-cache	Displays media caching service packet flows.
reverse-proxy	Displays reverse proxy web caching service packet flows.
web-cache	Displays standard web caching service packet flows.
wmt	Displays WMT caching service packet flows.
summary	(Optional) Displays summary information.
gre	Displays WCCP Generic Routing Encapsulation (GRE).
modules	Displays running status of WCCP registered modules.
port-list	Displays running status of WCCP port lists.
routers	Displays WCCP router list.
services	Displays WCCP services configured.
detail	(Optional) Displays detail of services.
slowstart	Displays WCCP slow start state for the selected service.
status	Displays version of WCCP that is enabled and running.

Defaults

No default behavior or values

Command Modes

EXEC

Examples

```
ContentEngine# show wccp gre
Transparent GRE packets received:           0
Transparent non-GRE packets received:       0
Transparent non-GRE packets passed through: 0
Total packets accepted:                     0
Invalid packets received:                   0
Packets received with invalid service:      0
Packets received on a disabled service:     0
Packets received too small:                 0
Packets dropped due to zero TTL:            0
Packets dropped due to bad buckets:         0
Packets dropped due to no redirect address: 0
Connections bypassed due to load:           0
Packets sent back to router:                0
Packets sent to another CE:                 0
GRE fragments redirected:                   0
Packets failed GRE encapsulation:           0
Packets dropped due to invalid fwd method:  0
Packets dropped due to insufficient memory: 0

ContentEngine# show wccp routers
Routers Seeing this Content Engine
                Router Id          Sent To
                10.0.0.0         10.1.1.1
        Routers not Seeing this Cache Engine
                10.1.1.1
        Routers Notified of but not Configured
                -NONE-
        Multicast Addresses Configured
                -NONE-

Router Information for Service: Reverse-Proxy
        Routers Seeing this Content Engine
                Router Id          Sent To
                10.0.0.0         10.1.1.1
        Routers not Seeing this Content Engine
                10.1.1.1
        Routers Notified of but not Configured
                -NONE-
        Multicast Addresses Configured
                -NONE-
```

show wmt

To display Windows Media Technologies (WMT) configuration and license information, use the **show wmt** EXEC command.

show wmt [**license-agreement** | **proxy**]

Syntax Description

license-agreement	(Optional) Displays WMT end user license agreement
proxy	(Optional) Displays WMT proxy mode configuration.

Defaults

No default behavior or values

Command Modes

EXEC

Examples

```
ContentEngine# show wmt
WMT version: cdm4630-001.000

WMT not enabled
WMT disallowed client protocols: none
WMT end user license agreement accepted
WMT license key not installed
WMT evaluation enabled. Estimated 16 days 22 hours left for evaluation.
WMT incoming port: 1755
WMT max sessions configured: 2500
WMT max sessions platform limit: 2500
WMT max sessions used: 2500 sessions
WMT max bandwidth: 15000 Kbits/sec
WMT max bit rate allowed per stream has no limit
WMT cache enabled
WMT cache max-obj-size: 1024
WMT debug level: 0
WMT L4 switch not enabled
WMT debug client ip not set
WMT debug server ip not set
WMT/REAL cache space partition: wmt 70%, real 30%

ContentEngine# show wmt proxy
Incoming Proxy-Mode:
Configured proxy mode WMT on port: 1755
WMT Transparent Proxy (WCCP):
  Not configured.
WMT Transparent Proxy (L4 Switch):
  Not configured.
```

shutdown

To shut down a specific hardware interface, use the **shutdown** interface configuration command. To restore an interface to operation, use the **no** form of this command.

shutdown

no shutdown

Syntax Description

This command has no arguments or keywords.

Defaults

No default behavior or values

Command Modes

Interface configuration

Usage Guidelines

See the "interface" section on page 2-92 for alternative syntax.

Examples

```
ContentEngine(config-if)# shutdown
ContentEngine(config-if)# no autosense
```

snmp-server community

To enable the SNMP agent and set up the community access string to permit access to the SNMP agent, use the **snmp-server community** global configuration command. Use the **no** form of this command to disable the SNMP agent and to remove the previously configured community string.

snmp-server community *string* [**group** | **rw**]

no snmp-server community *string* [**group** | **rw**]

Syntax Description

string	Community string that acts like a password and permits access to the SNMP agent.
group	(Optional) Specifies group to which the community string belongs.
rw	(Optional) Enables read-write access to this community string.

Defaults

The SNMP agent is disabled and a community string is not configured. When configured, an SNMP community string by default permits read-only access to all objects.

Command Modes

Global configuration

Examples

The following example enables the SNMP agent and assigns the community string comaccess to SNMP.

```
ContentEngine(config)# snmp-server community comaccess
```

The following example disables the SNMP agent and removes the previously defined community string.

```
ContentEngine(config)# no snmp-server community
```

Related Commands

show snmp

snmp-server contact

snmp-server enable traps

snmp-server group

snmp-server host

snmp-server location

snmp-server notify

snmp-server user

snmp-server view

snmp-server contact

To set the system contact (sysContact) string, use the **snmp-server contact** global configuration command. Use the **no** form of this command to remove the system contact information.

snmp-server contact *line*

no snmp-server contact

Syntax Description

contact	Specifies text for MIB-II object sysContact.
line	Identification of the contact person for this managed node.

Command Modes

Global configuration

Defaults

No system contact string is set.

Usage Guidelines

The system contact string is the value stored in the MIB-II system group sysContact object.

Examples

This is an example of a system contact string.

```
ContentEngine(config)# snmp-server contact Dial System Operator at beeper # 27345

ContentEngine(config)# no snmp-server contact
```

Related Commands

show snmp

snmp-server community

snmp-server enable traps

snmp-server group

snmp-server host

snmp-server location

snmp-server notify

snmp-server user

snmp-server view

snmp-server enable

To enable the Content Engine to send SNMP traps, use the **snmp-server enable traps** global configuration command. Use the **no** form of this command to disable all SNMP traps or only SNMP authentication traps.

snmp-server enable traps [**config** | **content-engine** {**disk-fail** | **disk-read** | **disk-write** | **overload-bypass** | **transaction-log**} | **entity** | **snmp** [**authentication** | **cold-start**]]

no snmp-server enable traps [**config** | **content-engine** {**disk-fail** | **disk-read** | **disk-write** | **overload-bypass** | **transaction-log**} | **entity** | **snmp** [**authentication** | **cold-start**]]

Syntax Description

config	(Optional) Enables CiscoConfigManEvent traps.
content-engine	(Optional) Enables SNMP Content Engine traps.
disk-fail	Enables disk failure error trap.
disk-read	Enables disk read error trap.
disk-write	Enables disk write error trap.
overload-bypass	Enables WCCP overload bypass error trap.
transaction-log	Enables transaction log write error trap.
entity	(Optional) Enables SNMP entity traps.
snmp	(Optional) Enables SNMP specific traps.
authentication	(Optional) Enables authentication trap.
cold-start	(Optional) Enables cold start trap.

Defaults

This command is disabled by default. No traps are enabled.

Command Modes

Global configuration

Usage Guidelines

If you do not enter an **snmp-server enable traps** command, no traps are sent. To configure traps, you must enter the **snmp-server enable traps** command.

The **snmp-server enable traps** command is used in conjunction with the **snmp-server host** command. Use the **snmp-server host** command to specify which host or hosts receive SNMP traps. To send traps, you must configure at least one **snmp-server host** command.

For a host to receive a trap, both the **snmp-server enable traps** command and the **snmp-server host** command for that host must be enabled.

In addition, SNMP must be enabled with the **snmp-server community** command.

To disable the sending of the MIB-II SNMP authentication trap, you must enter the command **no snmp-server enable traps snmp authentication**.

Examples

The following example enables the Content Engine to send all traps to the host 172.31.2.160 using the community string public.

```
ContentEngine(config)# snmp-server enable traps
ContentEngine(config)# snmp-server host 172.31.2.160 public
```

The following example disables all traps.

```
ContentEngine(config)# no snmp-server enable traps
```

Related Commands

show snmp

snmp-server host

snmp-server community

snmp-server contact

snmp-server group

snmp-server host

snmp-server location

snmp-server notify

snmp-server user

snmp-server view

snmp-server group

To define a user security model group, use the **snmp-server group** global configuration command. Use the **no** form of this command to remove the specified group.

snmp-server group *name* {**v1** [**notify** *name* | **read** *name* | **write** *name*] | **v2c** [**notify** *name* | **read** *name* | **write** *name*] | **v3** {**auth** [**notify** *name* | **read** *name* | **write** *name*] | **noauth** [**notify** *name* | **read** *name* | **write** *name*] | **priv** *name* [**notify** *name* | **read** *name* | **write** *name*]}}

no snmp-server group *name* {**v1** [**notify** *name* | **read** *name* | **write** *name*] | **v2c** [**notify** *name* | **read** *name* | **write** *name*] | **v3** {**auth** [**notify** *name* | **read** *name* | **write** *name*] | **noauth** [**notify** *name* | **read** *name* | **write** *name*] | **priv** *name* [**notify** *name* | **read** *name* | **write** *name*]}}

Syntax Description

name	Name of group.
v1	Specifies the group using the Version 1 Security Model.
notify	(Optional) Specifies a notify view for the group.
name	Notify view name.
read	(Optional) Specifies a read view for the group.
name	Read view name.
write	(Optional) Specifies a write view for the group.
name	Write view name.
v2c	Specifies the group using the Version 2c Security Model.
v3	Specifies the group using the User Security Model (SNMPv3).
auth	Specifies the group using the AuthNoPriv Security Level.
noauth	Specifies the group using the noAuthNoPriv Security Level.
priv	Specifies the group using the AuthPriv Security Level.

Defaults

The default is that no user security model group is defined.

Command Modes

Global configuration

Usage Guidelines

Select one of three SNMP security model groups: Version 1 (**v1**) Security Model, Version 2c (**v2c**) Security Model, or the User Security Model (**v3** or SNMPv3). Optionally, you then specify a notify, read, or write view for the group for the particular security model chosen. The **v3** option allows you to specify the group using one of three security levels: **auth** (AuthNoPriv Security Level), **noauth** (noAuthNoPriv Security Level), or **priv** (AuthPriv Security Level).

Examples

```
ContentEngine(config)# snmp-server group acme v1 notify mymib
```

Related Commands

show snmp
snmp-server community
snmp-server contact
snmp-server enable traps
snmp-server host
snmp-server location
snmp-server notify
snmp-server user
snmp-server view

snmp-server host

To specify the recipient of a host SNMP trap operation, use the **snmp-server host** global configuration command. Use the **no** form of this command to remove the specified host.

snmp-server host {*hostname* | *ip-address*} *communitystring username* [**v2c** [**retry** *number* | **timeout** *seconds*] | **v3** {**auth** [**retry** *number* | **timeout** *seconds*] | **noauth** [**retry** *number* | **timeout** *seconds*] | **priv** [**retry** *number* | **timeout** *seconds*]}]

no snmp-server host {*hostname* | *ip-address*} *communitystring username* [**v2c** [**retry** *number* | **timeout** *seconds*] | **v3** {**auth** [**retry** *number* | **timeout** *seconds*] | **noauth** [**retry** *number* | **timeout** *seconds*] | **priv** [**retry** *number* | **timeout** *seconds*]}]

Syntax Description

hostname	Host name of SNMP trap host.
ip-address	IP address of SNMP trap host.
communitystring	Password-like community string sent with the trap operation.
username	Username.
v2c	Specifies the Version 2c Security Model.
retry	Sets the count for the number of retries for the inform request. (The default is 2 tries.)
number	Number of retries of the inform request (1–10).
timeout	Sets the timeout for the inform request (1–1000). (The default is 15 seconds.)
seconds	Timeout value in seconds.
v3	Specifies the User Security Model (SNMPv3).
auth	Sends notification using the AuthNoPriv Security Level.
noauth	Sends notification using the noAuthNoPriv Security Level.
priv	Sends notification using the AuthPriv Security Level.

Defaults

This command is disabled by default. No traps are sent. The version of the SNMP protocol used to send the traps is SNMP Version 1.

retry number: 2 retries

timeout: 15 seconds

Command Modes

Global configuration

Usage Guidelines

If you do not enter an **snmp-server host** command, no traps are sent. To configure the Content Engine to send SNMP traps, you must enter at least one **snmp-server host** command. To enable multiple hosts, you must issue a separate **snmp-server host** command for each host. The maximum number of **snmp-server host** commands is four.

When multiple **snmp-server host** commands are given for the same host, the community string in the last command is used.

The **snmp-server host** command is used in conjunction with the **snmp-server enable traps** command to enable SNMP traps.

In addition, SNMP must be enabled with the **snmp-server community** command.

Examples

The following example sends the SNMP traps defined in RFC 1157 to the host specified by the IP address 172.16.2.160. The community string is comaccess.

```
ContentEngine(config)# snmp-server enable traps
ContentEngine(config)# snmp-server host 172.16.2.160 comaccess
```

The following example removes the host 172.16.2.160 from the SNMP trap recipient list.

```
ContentEngine(config)# no snmp-server host 172.16.2.160
```

Related Commands

show snmp

snmp-server community

snmp-server contact

snmp-server enable traps

snmp-server group

snmp-server location

snmp-server notify

snmp-server user

snmp-server view

snmp-server location

To set the SNMP system location string, use the **snmp-server location** global configuration command. Use the **no** form of this command to remove the location string.

snmp-server location *line*

no snmp-server location *line*

Syntax Description

location	Specifies text for MIB-II object sysLocation.
line	String that describes the physical location of this node.

Defaults

No system location string is set.

Command Modes

Global configuration

Usage Guidelines

The system location string is the value stored in the MIB-II system group system location object. You can see the system location string with the **show snmp** EXEC command.

Examples

This is an example of a system location string.

```
ContentEngine(config)# snmp-server location Building 3/Room 214
```

Related Commands

show snmp

snmp-server community

snmp-server contact

snmp-server enable traps

snmp-server group

snmp-server host

snmp-server notify

snmp-server user

snmp-server view

snmp-server notify inform

To configure the SNMP notify inform request, use the **snmp-server notify inform** global configuration command. Use the **no** form of this command to return the setting to the default value.

snmp-server notify inform

no snmp-server notify inform

Syntax Description

This command has no arguments or keywords.

Defaults

No default behavior or values

Command Modes

Global configuration

Usage Guidelines

If you do not issue the **snmp-server notify inform** command, the default is an SNMP trap request.

Examples

```
ContentEngine(config)# snmp-server notify inform
```

Related Commands

show snmp
snmp-server community
snmp-server contact
snmp-server enable traps
snmp-server group
snmp-server host
snmp-server location
snmp-server user
snmp-server view

snmp-server user

To define a user who can access the SNMP engine, use the **snmp-server user** global configuration command. Use the **no** form of this command to remove access.

snmp-server user *name group* [**auth** {**md5** *password* [**priv** *password*] | **sha** *password* [**priv** *password*]} | **remote** *octetstring* [**auth** {**md5** *password* [**priv** *password*] | **sha** *password* [**priv** *password*]}]

no snmp-server user *name group* [**auth** {**md5** *password* [**priv** *password*] | **sha** *password* [**priv** *password*]} | **remote** *octetstring* [**auth** {**md5** *password* [**priv** *password*] | **sha** *password* [**priv** *password*]}]

Syntax Description

name	Name of user.
group	Defines the group to which the user belongs.
auth	Configures user authentication parameters.
md5	Configures HMAC MD5 authentication algorithm.
password	HMAC MD5 user authentication password.
priv	Configures authentication parameters for the packet.
password	HMAC MD5 user private password.
sha	Configures HMAC SHA authentication algorithm.
password	HMAC SHA authentication password.
remote	Specifies engine identity of remote SNMP entity to which the user belongs.

Defaults

No default behavior or values

Command Modes

Global configuration

Examples

```
Content Engine# snmp-server user acme admin
```

Related Commands

show snmp

snmp-server community

snmp-server contact

snmp-server enable

snmp-server group

snmp-server host

snmp-server location

snmp-server notify

snmp-server view

snmp-server view

To define a Version 2 SNMP (SNMPv2) MIB view, use the **snmp-server view** global configuration command. Use the **no** form of this command to undefine the MIB view.

snmp-server view *viewname familyname* {**excluded** | **included**}

no snmp-server view *viewname familyname* {**excluded** | **included**}

Syntax Description

viewname	Name of MIB view.
familyname	MIB view family name.
excluded	Excludes MIB family from the view.
included	Includes MIB family from the view.

Defaults

No default behavior or values

Command Modes

Global configuration

Examples

```
Content Engine# snmp-server view contentview ciscoContentEngineMIB included
```

Related Commands

show snmp

snmp-server community

snmp-server contact

snmp-server enable

snmp-server group

snmp-server host

snmp-server location

snmp-server notify

snmp-server user

ssh-key-generate

To generate the Secure Shell (SSH) host key, use the **ssh-key-generate** command in EXEC mode.

ssh-key-generate [**key-length** *length*]

Syntax Description

key-length	(Optional) Configures the length of the SSH key.
length	Specifies the number of bits to create an SSH key (512–2048).

Defaults

key-length *length*: 1024 bits

Command Modes

EXEC

Usage Guidelines

Before you enable the **sshd** command, use the **ssh-key-generate** command to generate a private and a public host key, which the client programs use to verify server's identity.

When a user runs an SSH client and logs in to the Content Engine, the public key for the SSH daemon running on the Content Engine is recorded in the client machine *known_hosts* file in the user's home directory. If the Content Engine administrator subsequently regenerates the host key by issuing the **ssh-key-generate** command, the user must delete the old public key entry associated with the Content Engine in the *known_hosts* file before running the SSH client program to log in to the Content Engine. When the user runs the SSH client program after deleting the old entry, the *known_hosts* file is updated with the new SSH public key for the Content Engine.

Examples

```
This example generates an SSH public key, then enables the SSH daemon.
Console(config)# ssh-key-generate
Ssh host key generated successfully
Saving the host key to box ...
Host key saved successfully
Console(config)# sshd enable
Starting ssh daemon ...
Ssh daemon started successfully
```

Related Commands

sshd

sshd

To enable the Secure Shell (SSH) daemon, use the **sshd** command in global configuration mode. Use the **no** form of the command to disable SSH.

sshd {**enable** | **password-guesses** *number* | **timeout** *seconds*}

no sshd {**enable** | **password-guesses** *number* | **timeout** *seconds*}

Syntax Description

enable	Enables the SSH feature.
password-guesses	Configures allowable password guesses per connection.
number	Maximum number of incorrect password guesses allowable (1–99). (The default is 3.)
timeout	Configures SSH login grace time.
seconds	SSH login grace time value in seconds (1–99999). (The default is 300.)

Defaults

password-guesses *number*: 3 guesses

timeout *seconds*: 300 seconds

Command Modes

Global configuration

Usage Guidelines

SSH enables login access to the Content Engine through a secure and encrypted channel. SSH consists of a server and a client program. Like Telnet, you can use the client program to remotely log on to a machine that is running the SSH server, but unlike Telnet, messages transported between the client and the server are encrypted. The functionality of SSH includes user authentication, message encryption, and message authentication.

Before you enable the **sshd** command, use the **ssh-key-generate** command to generate a private and a public host key, which the client programs use to verify server's identity.

Examples

```
Console(config)# sshd enable
Console(config)# sshd password-guesses 4
Console(config)# sshd timeout 20
```

Related Commands

ssh-key-generate

show sshd

no sshd enable

standby

To configure an interface to be a backup for another interface, use the **standby** command in interface configuration mode. Use the **no** form of the command to restore the default configuration of the interface.

standby *group_number* {**errors** *max_errors* | **ip** *ip-address netmask* | **priority** *priority_level*}

no standby *group_number* {**errors** *max_errors* | **ip** *ip-address netmask* | **priority** *priority_level*}

Syntax Description

group_number	Standby group number (1–4).
errors	Sets the maximum number of errors allowed on this interface.
max_errors	Maximum number of errors (0–4294967295).
ip	Sets the IP address of a standby group.
ip-address	IP address of a standby group.
netmask	Netmask of a standby group.
priority	Sets the priority of an interface for the standby group.
priority_level	Priority level number (0–4294967295).

Defaults

There are no standby interfaces by default. The **errors** option is disabled by default.

Command Modes

Interface configuration

Usage Guidelines

When an active network interface fails (because of cable trouble, Layer 2 switch failure, high error count, and so forth), and that interface is part of a standby group, a standby interface can become active and take the load off the failed interface.

To configure standby interfaces, interfaces are logically assigned to standby groups. The following rules define the standby group relationships:

- A standby group comprises two or more interfaces.
- An interface can belong to more than one standby group, and can thus act as a standby for more than one interface.
- The maximum number of standby groups on a Content Engine is four.
- Each interface is assigned a unique IP address, and each standby group is assigned a unique standby IP address, shared by all members of the group.
- Each interface in a standby group is assigned a priority. The operational interface with the highest priority in a standby group is the active interface. Only the active interface uses the group IP address.
- If the active interface fails, the operational interface in its standby group that is assigned the next highest priority becomes active.

- If all the members of a standby group fail and then one recovers, the ACNS 4.0 software brings up the standby group on the operational interface.
- The priority of an interface in a standby group can be changed at runtime. The member interface that has the highest priority after this change becomes the new active interface (the default action is to preempt the currently active interface if an interface with higher priority exists).

The maximum number of errors allowed on the active interface before the interface is shut down and the standby is brought up is configured with the **errors** option, which is disabled by default.

Examples

This example configures three interfaces to be part of the same standby group, with interface 3/0 as the active interface.

```
Console(config)# interface fastEthernet 3/0 standby 1 ip 172.16.10.10 255.255.254.0
Console(config)# interface fastEthernet 3/1 standby 1 ip 172.16.10.10 255.255.254.0
Console(config)# interface fastEthernet 3/2 standby 1 ip 172.16.10.10 255.255.254.0
Console(config)# interface fastEthernet 3/0 standby 1 priority 300
Console(config)# interface fastEthernet 3/1 standby 1 priority 200
Console(config)# interface fastEthernet 3/2 standby 1 priority 100
Console(config)# interface fastEthernet 3/0 standby 1 errors 10000
Console(config)# interface fastEthernet 3/1 standby 1 errors 10000
Console(config)# interface fastEthernet 3/2 standby 1 errors 10000

Console# show standby
Standby Group:1
IP address: 172.16.10.10, netmask: 255.255.254.0
Maximum errors allowed on the active interface: 10000
        Member interfaces:
                FastEthernet 3/0        priority: 300
                FastEthernet 3/1        priority: 200
                FastEthernet 3/2        priority: 100

        Active interface: FastEthernet 3/0
```

Related Commands

show standby

sysfs

To maintain the system file system, use the **sysfs** command.

sysfs check *partition_name*

sysfs format *partition_name*

sysfs mount *partition_name* {**local1** | **local2**}

sysfs repair *partition_name*

sysfs sync

sysfs unmount {**local1** | **local2**}

Syntax Description

check	Checks a file system.
partition_name	Disk and partition name (for example, disk00/00 or disk00/01).
format	Erases and creates a file system on a disk device.
mount	Mounts a disk or volume file system.
local1	Mounts to /local1.
local2	Mounts to /local2.
repair	Checks and repairs a sysfs file system.
sync	Synchronizes all **sysfs** commands running on the system.
unmount	Unmounts a sysfs partition.
local1	Unmounts a sysfs mounted at /local1.
local2	Unmounts a sysfs mounted at /local2.

Defaults

No default behavior or values

Command Modes

EXEC

Usage Guidelines

A sysfs formatted volume must be mounted when you use transaction logs.

A file system is automatically repaired when it is mounted.

Examples

The following example creates a sysfs partition on the first disk on the SCSI bus, formats the partition, and mounts the volume /local1.

```
ContentEngine# disk erase-all-partitions disk00
ContentEngine# disk partition disk00/00 50% sysfs
ContentEngine# sysfs format disk00/00
ContentEngine# sysfs mount disk00/00 local1
```

tacacs

To configure TACACS+ server parameters, use the **tacacs** command in global configuration mode. Use the **no** form of the command to disable individual options.

tacacs {**key** *keyword* | **retransmit** *retries* | **server** {*hostname* | *ip-address*} [**primary**] | **timeout** *seconds*}

no tacacs {**key** *keyword* | **retransmit** *retries* | **server** {*hostname* | *ip-address*} [**primary**] | **timeout** *seconds*}

Syntax Description

key	Sets security word.
keyword	Keyword. An empty string is the default.
retransmit	Sets the number of times that requests are retransmitted to a server.
retries	Number of attempts allowed (1–3). The default is two retry attempts.
server	Sets a server address.
hostname	Host name of TACACS+ server.
ip-address	IP address of TACACS+ server.
primary	(Optional) Sets the server as primary.
timeout	Sets the number of seconds to wait before a request to a server is timed out.
seconds	Timeout in seconds (1–20). The default is 5 seconds.

Defaults

keyword: none (empty string)

timeout *seconds*: 5

retries: 2

Command Modes

Global configuration

Usage Guidelines

The TACACS database validates users before they gain access to a Content Engine. TACACS+ is derived from the United States Department of Defense (RFC 1492) and is used by Cisco Systems as an additional control of nonprivileged and privileged mode access. This release supports TACACS+ only and not TACACS or Extended TACACS.

TACACS+ provides both authentication and authorization options. Authentication or login is the action of identifying and validating a user. It verifies a username with the password. Authorization or configuration is the action of determining what a user is allowed to do. To configure TACACS+, use the **authentication** and **tacacs** commands.

The Users GUI page or **user** global configuration commands provide a way to add, delete, or modify usernames, passwords, and access privileges in the local database. The TACACS+ remote database can also be used to maintain login and configuration privileges for administrative users. The **tacacs** command or the TACACS+ GUI page allows you to configure the network parameters required to access the remote database.

Use the **tacacs key** command specifies the TACACS+ key, used to encrypt the packets transmitted to the server. This key must be the same as the one specified on the server daemon. The maximum number of characters in the key should not exceed 99 printable ASCII characters (except tabs). An empty key string is the default. All leading spaces are ignored; spaces within and at the end of the key string are not ignored. Double quotes are not required even if there are spaces in the key, unless the quotes themselves are part of the key.

Login and configuration privileges can be obtained from both the local database or the TACACS+ remote database. If both databases are enabled, then both databases are queried. If the user data cannot be found in the first database queried, then the second database is tried. When the **primary** keyword is entered for TACACS+ login or configuration authentication, the TACACS+ database is queried first, and the local database is queried second. If the TACACS+ database is not designated as the primary, and both the local and the TACACS+ databases are enabled, the local database is queried first. If both the local and TACACS+ databases are disabled (**no authentication**), the Content Engine verifies that both are disabled and if so, sets the Content Engine to the default state.

One primary and two backup TACACS+ servers can be configured; authentication is attempted on the primary server first, then on the others in the order in which they were configured. The primary server is the first server configured unless another is explicitly specified as primary with the **tacacs server** *hostname* **primary** command.

The **tacacs timeout** is the number of seconds the Content Engine waits before declaring a timeout on a request to a particular TACACS+ server. The range is from 1 to 20 seconds with 5 seconds as the default. The number of times the Content Engine repeats a retry-timeout cycle before trying the next TACACS+ server is specified by the **tacacs retransmit** command. The default is two retry attempts.

Three unsuccessful login attempts are permitted. TACACS+ logins may appear to take more time than local logins depending on the number of TACACS+ servers and the configured timeout and retry values.

Examples

This example configures the key used in encrypting packets.

```
Console(config)# tacacs key human789
```

This example configures the host named spearhead as the primary TACACS+ server.

```
Console(config)# tacacs server spearhead primary
```

This example sets the timeout interval for the TACACS+ server.

```
Console(config)# tacacs timeout 10
```

This example sets the number of times authentication requests are retried (retransmitted) after a timeout.

```
Console(config)# tacacs retransmit 3
```

Related Commands

authentication

show authentication

show statistics authentication

show tacacs

tcp

To configure Transmission Control Protocol (TCP) parameters, use the **tcp** global configuration command. To disable TCP parameters, use the **no** form of this command.

tcp client-mss *maxsegsize*

tcp client-receive-buffer *kbytes*

tcp client-rw-timeout *seconds*

tcp client-satellite

tcp client-send-buffer *kbytes*

tcp cwnd-base *segments*

tcp ecdn enable

tcp increase-xmit-timer-value *value*

tcp init-ss-threshold *value*

tcp keepalive-probe-cnt *count*

tcp keepalive-probe-interval *seconds*

tcp keepalive-timeout *seconds*

tcp server-mss *maxsegsize*

tcp server-receive-buffer *kbytes*

tcp server-rw-timeout *seconds*

tcp server-satellite

tcp server-send-buffer *kbytes*

tcp type-of-service enable

no tcp {**client-mss** *maxsegsize* | **client-receive-buffer** *kbytes* | **client-rw-timeout** *seconds* | **client-satellite** | **client-send-buffer** *kbytes* | **cwnd-base** *segments* | **ecdn enable** | **increase-xmit-timer-value** *value* | **init-ss-threshold** *value* | **keepalive-probe-cnt** *count* | **keepalive-probe-interval** *seconds* | **keepalive-timeout** *seconds* | **server-mss** *maxsegsize* | **server-receive-buffer** *kbytes* | **server-rw-timeout** *seconds* | **server-satellite** | **server-send-buffer** *kbytes* | **type-of-service enable**}

Syntax Description

client-mss	Sets client TCP maximum segment size.
maxsegsize	Maximum segment size in bytes (512–1460).
client-receive-buffer	Sets client connections receive buffer size.
kbytes	Receive buffer size in kilobytes (1–1024).
client-rw-timeout	Sets client connection's read/write timeout.

seconds	Timeout in seconds (1–3600).
client-satellite	Sets client TCP compliance to RFC 1323 standard.
client-send-buffer	Sets client connection's send buffer size.
kbytes	Send buffer size in kilobytes (8–1024).
cwnd-base	Sets initial send congestion window in segments.
segments	Initial send congestion window segments (1-10).
tcp ecdn enable	Enables TCP ECDN.
increase-xmit-timer-value	Increases default retransmit time multiple.
value	Retransmit multiplier (1–3).
init-ss-threshold	Sets initial slow start threshold value.
value	Slow start threshold value.
keepalive-probe-cnt	Sets TCP keepalive probe counts.
count	Number of probe counts (1–10).
keepalive-probe-interval	Sets TCP keepalive probe interval.
seconds	Keepalive probe interval in seconds (1–300).
keepalive-timeout	Sets TCP keepalive timeout.
seconds	Keepalive timeout in seconds (1 to 3600).
server-mss	Sets server TCP maximum segment size.
maxsegsize	Maximum segment size in bytes (512–1460).
server-receive-buffer	Sets server connection's receive buffer size.
kbytes	Receive buffer size in kilobytes (1–1024).
server-rw-timeout	Sets server connection's read/write timeout.
seconds	Read/write timeout in seconds (1–3600).
server-satellite	Sets server TCP compliance to RFC 1323 standard.
server-send-buffer	Sets server connection's send buffer size.
kbytes	Send buffer size in kilobytes (1–1024).
type-of-service enable	Sets TCP type of service to match client's type of service.

Defaults

tcp server maximum segment size: 1432 bytes

tcp client maximum segment size: 1432 bytes

tcp server-receive-buffer: 8 kilobytes

tcp client-receive-buffer: 8 kilobytes

tcp server-rw-timeout: 120 seconds

tcp client-rw-timeout: 30 seconds

tcp server-send-buffer: 8 kilobytes

tcp client-send-buffer: 8 kilobytes

tcp keepalive-probe-cnt: 4

tcp keepalive-probe-interval: 75 seconds

tcp keepalive-timeout: 300 seconds

tcp server-satellite (RFC 1323): disabled

tcp client-satellite (RFC 1323): disabled

tcp type of service: disabled

Command Modes

Global configuration

Usage Guidelines

In nearly all environments, the default TCP setting is adequate.

Examples

```
ContentEngine(config)# tcp client-receive-buffer 100
ContentEngine(config)# no tcp client-receive-buffer
```

Related Commands

show tcp

telnet enable

To enable Telnet, use the **telnet** EXEC command.

telnet enable

no telnet enable

Syntax Description

This command has no arguments or keywords.

Defaults

No default behavior or values

Command Modes

Global configuration

Usage Guidelines

Use this terminal emulation protocol for remote terminal connection.

Examples

```
ContentEngine(config)# telnet enable
```

Related Commands

show telnet

terminal

To set the number of lines displayed in the console window, or to display the current console debug command output, use the **terminal** EXEC command.

terminal {**length** *lines* | **monitor** [**disable**]}

Syntax Description

length	Sets the number of lines displayed on the terminal screen.
lines	Number of lines on the screen displayed before pausing (0–512). Enter 0 for no pausing.
monitor	Copies debug output to the current terminal.
disable	(Optional) Turns off monitoring at this terminal.

Defaults

Default is 24 lines.

Command Modes

EXEC

Usage Guidelines

When 0 is entered as the *lines* parameter, output to the screen does not pause. For all nonzero values of *lines*, the -More- prompt is displayed when the number of output lines matches the specified *lines* number. The -More- prompt is considered a line of output. To view the next screen, press the **Spacebar**. To view one line at a time, press the **Enter** key. To exit the **show** command output, press the **Esc** key or any other key.

The **terminal monitor** command allows a Telnet session to display the output of the debug commands that appear on the console. The monitoring continues until the Telnet session is terminated.

Examples

The following example sets the number of lines to display to 20.

```
ContentEngine(config)# terminal length 20
```

The following example sets the number of lines to the default of 24.

```
ContentEngine(config)# no terminal length
```

The following example configures the terminal for no pausing.

```
ContentEngine(config)# terminal length 0
```

Related Commands

All **show** commands

tftp-server

To set the Trivial File Transfer Protocol (TFTP) server directory, use the **tftp-server** global configuration command.

tftp-server dir *directory*

no tftp-server dir *directory*

Syntax Description

dir	Sets the TFTP server directory.
directory	Path name of the TFTP server.

Defaults

No default behavior values

Command Modes

Global configuration

Usage Guidelines

Use **tftp-server** to allow files to be transferred from one server to another over a network without the use of client authentication.

Examples

```
Console(config)# tftp-server dir /mypath
```

Related Commands

show tftp-server

transaction-log force

To force the archive or export of the transaction log, use the **transaction-log force** EXEC command.

transaction-log force {**archive** | **export**}

Syntax Description

archive	Forces the archive of the *working.log* file.
export	Forces the archived files to be exported to an FTP server.

Defaults

No default behavior or values

Command Modes

EXEC

Usage Guidelines

The **transaction-log force archive** command causes the transaction log *working.log* file to be archived to the Content Engine hard disk following the next transaction. This command has the same effect as the **clear transaction-log** command.

The **transaction-log force export** command causes the transaction log to be exported to an FTP server designated by the **transaction-logs export ftp-server** command.

The force commands do not change the configured or default schedule for archive or export of transaction log files. If the archive interval is configured in seconds or the export interval is configured in minutes, the forced archive or export interval period is restarted after the force operation.

If a scheduled archive or export job is in progress when a corresponding force command is entered, the force command has no effect. If a force command is in progress when an archive or export job is scheduled to run, the force operation is completed and the archive or export is rescheduled for the next configured interval.

Examples

```
ContentEngine# transaction-log force archive
ContentEngine# transaction-log force export
```

Related Commands

transaction-logs

clear statistics transaction-logs

clear transaction-log

show statistics transaction-logs

show transaction-logging

transaction-logs

To enable transaction logs, use the **transaction-logs** command in global configuration mode. To disable transaction logs, use the **no** form of this command.

transaction-logs archive interval every-day {**at** *hour:minute* | **every** *hour*}

transaction-logs archive interval every-hour {**at** *minute* | **every** *minute*}

transaction-logs archive interval *second*

transaction-logs archive max-file-size *filesize*

transaction-logs ecdn enable

transaction-logs enable

transaction-logs export compress

transaction-logs export enable

transaction-logs export ftp-server {*hostname* | *servipaddrs*} *login passw directory*

transaction-logs export interval every-day {**at** *hour:minute* | **every** *hour}*

transaction-logs export interval every-hour {**at** *minute* | **every** *minute*}

transaction-logs export interval every-week on *weekday* [**at** *hour:minute*]

transaction-logs export interval *minute*

transaction-logs file-marker

transaction-logs format {**apache** | **extended-squid** | **squid**}

transaction-logs sanitize

no transaction-logs {**archive** {**interval** {**every-day** {**at** *time* | **every** *hour*} | **every-hour** {**at** *minute* | **every** *minute*} | *second*}} | **max-file-size** *filesize*} | **ecdn enable** | **enable** | **export** {**compress** | **enable** | **ftp-server** {*hostname* | *servipaddrs*} *login passw directory* | **interval** {**every-hour** {**at** *minute* | **every** *minute*} | **every-day** {**at** *hour:minute* | **every** *hour*} | **every-week on** *weekday* [**at** *hour:minute*] | *minute*}} | **file-marker** | **format** {**apache** | **extended-squid** | **squid**} | **sanitize**

Syntax Description

archive	Configures archive parameters.
interval	Determines how frequently the archive file is to be saved.
every-day	Archives using frequencies of 1 day or less.
at *hour:minute*	Hour and minute (hh:mm) in local time for daily archive.
every *hour*	Number of hours for daily file archive.
every-hour	Archives using frequencies of 1 hour or less.
at *minute*	Minute alignment for the hourly archive (0–59).
every *minute*	Minute within the hour for file archive (0–59).

second	Number of seconds at which to archive (120–604800).
max-file-size	Sets maximum archive file size.
filesize	Maximum archive file size in kilobytes (1000–2000000).
ecdn	Sets Enterprise CDN (E-CDN) logging features.
enable	Enables logging of E-CDN internal communication.
enable	Enables transaction log feature.
export	Configures file export parameters.
enable	Enables the exporting of log files at the specified interval.
compress	Compresses the archived files in the gzip format before exporting.
ftp-server	Sets FTP server to receive exported archived files.
hostname	Host name of target FTP server.
servipaddrs	IP address of target FTP server.
login	User login to target FTP server.
passw	User password to target FTP server.
directory	Target directory for exported files on FTP server.
interval	Determines how frequently the file is to be exported.
every-hour	Exports files using intervals of 1 hour or less.
at *minute*	Minute (0–59) within the hour for exporting files.
every *minute*	Interval in minutes (0–59).
every-day	Exports files using daily intervals.
at *hour:minute*	Hour and minute (hh:mm) in local time for daily file export.
every *hour*	Number of hours for daily file export.
every-week	Exports files using weekly intervals.
on *weekday*	Days of the week for file export (Mon, Tue, Wed, Thu, Fri, Sat, Sun).
at *hour:minute*	(Optional) Hour and minute (hh:mm) for weekly file export. The default is 00:00.
minute	Number of minutes in the interval at which to export a file (1–10080).
file-marker	Adds statements to transaction log indicating the file beginning and end.
format	Sets the format to use for the *working.log* file.
apache	Configures Apache Common Log format (CLF).
extended-squid	Configures the Extended Squid log format.
squid	Configures the Squid native log format (Squid–1.1 *access.log* format).
sanitize	Writes user IP addresses in log file as 0.0.0.0.

Defaults

archive: disabled

ecdn: disabled, even if transaction logs is enabled

enable: disabled

export compress: disabled

export: disabled

file-marker: disabled

sanitize: disabled

archive interval: every day, every 1 hour

archive max-file-size: 2,000,000 kilobytes

export interval: every day, every 1 hour

format: Squid native log format

Command Modes

Global configuration

Usage Guidelines

Enable transaction log recording with the **transaction-logs enable** command. The transactions that are logged include HTTP and WMT Microsoft Media Server (MMS) caching proxy server transactions. When enabled, daemons create a *working.log* file in the */local1/logs/* on the sysfs volume for HTTP transactions and a separate *working.log* file in */local1/logs/export* for WMT transactions.

The **transaction-logs ecdn enable** command enables E-CDN internal communication logging to the transaction log and counts as an HTTP statistic when transaction logging is enabled. E-CDN internal communication logging is not recorded to the transaction log or counted as an HTTP statistic in the default mode, even if transaction logging is enabled. The E-CDN internal communication is typically HTTP POST traffic for CDN topology, CDN routing, device monitoring, and other related activities.

After an interval specified by the **transaction-logs archive interval** command, the *working.log* file is renamed as an archive file. Only transactions subsequent to the archiving event are recorded in a new *working.log* file. The transaction log archive file-naming conventions are shown in Table 2-9. The Content Engine default archive interval is every day, every 1 hour.

Use the **transaction-logs archive max-file-size** command to specify the maximum size of an archive file. The *working.log* file is archived when it attains the maximum file size if this size is reached before the configured archive interval time.

Use the **transaction-logs file-marker** option to mark the beginning and end of archive files. By examining the file markers of an exported archive file, the administrator can determine whether the FTP process transferred the entire file. The file markers are in the form of dummy transaction entries that are written in the configured log format. The following example shows the start and end dummy transactions in the default native Squid log format.

- 970599034.130 0 0.0.0.0 TCP_MISS/000 0 NONE TRANSLOG_FILE_START - NONE/- -
- 970599440.130 0 0.0.0.0 TCP_MISS/000 0 NONE TRANSLOG_FILE_END - NONE/- -

Use the **format** option to format the log files for either native Squid, Extended Squid formats, or Apache Common Log Format (CLF). The Extended Squid log format uses the RFC 981 field of the Squid log format for the username. The extended Squid format logs the associated username for authentication for each record in the log file, if available, and is used for billing purposes.

Use the **sanitized** option to disguise the IP address of clients in the transaction log file. The default is not sanitized. A sanitized transaction log disguises the network identity of a client by changing the IP address in the transaction logs to 0.0.0.0. The **no** form disables the sanitize feature.

The transaction log archive and export functions are configured with the following commands:

- The **transaction-logs archive interval** global configuration command allows the administrator to specify when the *working.log* file is archived.
- The **transaction-logs export interval** global configuration command allows the administrator to specify when the archived transaction logs are exported.

The following limitations apply:

- When the interval is scheduled in units of hours, the value must divide evenly into 24. For example, the interval can be every 4 hours, but not every 5 hours.
- When the interval is scheduled in units of minutes, the value must divide evenly into 60.
- Only the more common choices of minutes are supported. For example, the interval can be 5 minutes or 10 minutes, but not 6.
- The selection of interval alignment is limited. If an interval is configured for every 4 hours, it will align with midnight. It cannot align with 12:30 or with 7 a.m.
- The feature does not support different intervals within a 24-hour period. For example, it does not support an interval that is hourly during regular business hours, and then every 4 hours during the night.

```
ContentEngine(config)# transaction-logs archive interval every-day
  at        Specify the time at which to archive each day
  every     Specify the interval in hours. It will align with midnight

ContentEngine(config)# transaction-logs archive interval every-day at
<0-23>:  Time of day at which to archive (hh:mm)

ContentEngine(config)# transaction-logs archive interval every-day every
<1-24>  Interval in hours: {1, 2, 3, 4, 6, 8, 12 or 24}
```

Transaction Log Archive File-Naming Convention

The archive transaction log file is named as follows for HTTP and WMT caching:

```
celog_10.1.118.5_20001228_235959.txt

mms_export_10.1.118.5_20001228_235959
```

If the **export compress** feature is enabled when the file is exported, then the file extension will be *.gz* as shown in the following example.

```
celog_10.1.118.5_20001228_235959.txt.gz

mms_export_10.1.118.5_20001228_235959.gz
```

Table 2-9 describes the name elements.

Table 2-9 Description of Archive Log Name Elements

Sample of Element	Description
celog_	HTTP caching proxy server archive file.
mms_export_	WMT caching proxy server archive file.
10.1.118.5_	IP address of the Content Engine creating the archive file.
20001228_	Date on which archive file was created (yyyy/mm/dd).
235959	Time when archive file was created (hh/mm/ss).

Compressing Archive Files

The **transaction-logs export compress** option compresses an archive into a gzip file format before exporting it. Compressing the archive file uses less disk space on both the Content Engine and the FTP export server. The compressed file uses less bandwidth when transferred. The archive filename of the compressed file has the extension *.gz*.

Exporting Transaction Logs to External FTP Servers

The **transaction-logs export ftp-server** option can support up to four FTP servers. To export transaction logs, you must first enable the feature and configure the FTP server parameters. The following information is required for each target FTP server:

- Server IP address or the host name

  The Content Engine translates the host name with a DNS lookup and then stores the IP address in the configuration.

- FTP user login and user password
- Path of the directory where transferred files are written

  Use a fully qualified path or a relative path for the user login. The user must have write permission to the directory.

Use the **no** form of the **transaction-logs export enable** command to disable the entire transaction-logs feature while retaining the rest of the configuration.

Receiving a Permanent Error from the External FTP Server

A permanent error (Permanent Negative Completion Reply, RFC 959) occurs when the FTP command to the server cannot be accepted, and the action does not take place. Permanent errors can be caused by invalid user logins, invalid user passwords, and attempts to access directories with insufficient permissions.

When an FTP server returns a permanent error to the Content Engine, the export is retried at 10-minute intervals or sooner if the configured export interval is sooner. If the error is a result of a misconfiguration of the **transaction-logs export ftp server** command, then you must reenter the Content Engine parameters to clear the error condition. The **show statistics transaction-logs** command displays the status of logging attempts to export servers.

In the following example, an invalid user login parameter was included in the **transaction-logs export ftp-server** command. The **show statistics transaction-logs** command shows that the Content Engine failed to export archive files.

```
ContentEngine# show statistics transaction-logs
Transaction Log Export Statistics:

Server:172.16.10.5
      Initial Attempts:1
      Initial Successes:0
      Initial Open Failures:0
      Initial Put Failures:0
      Retry Attempts:0
      Retry Successes:0
      Retry Open Failures:0
      Retry Put Failures:0
      Authentication Failures:1
      Invalid Server Directory Failures:0
```

To correct the misconfiguration, you must reenter the **transaction-logs export ftp-server** parameters.

```
ContentEngine(config)# transaction-logs export ftp-server 10.1.1.1 goodlogin pass
/ftpdirectory
```

The **transaction-logs format** command has three options: **squid**, **extended-squid**, and **apache**. The default format is **squid**.

```
ContentEngine(config)# transaction-logs format
  apache Apache Common Log format
  extended-squid  Extended Squid log format
  squid           Squid log format
ContentEngine(config)# transaction-logs format apache
ContentEngine(config)# no transaction-logs format
  <cr>
ContentEngine(config)# transaction-logs format extended-squid
  <cr>
```

Examples

In this example, an FTP server is configured.

```
ContentEngine(config)# transaction-logs export ftp-server 10.1.1.1 mylogin mypasswd
/ftpdirectory

ContentEngine(config)# transaction-logs export ftp-server myhostname mylogin mypasswd
/ftpdirectory
```

To delete an FTP server, use the **no** form of the command.

```
ContentEngine(config)# no transaction-logs export ftp-server 10.1.1.1
ContentEngine(config)# no transaction-logs export ftp-server myhostname
```

Use the **no** form of the command to disable the entire transaction log export feature while retaining the rest of the configuration.

```
ContentEngine(config)# no transaction-logs export enable
```

To change a username, password, or directory, reenter the entire line.

```
ContentEngine(config)# transaction-logs export ftp-server 10.1.1.1 mynewname mynewpass
/newftpdirectory
```

The **show transaction-logging** command displays information on exported log files.

```
ContentEngine# show transaction-logging
Transaction log configuration:
--------------------------------------
Logging is enabled.
Logging of ecdn internal communication is disabled.
End user identity is visible.
File markers are disabled.
Archive interval: every-day every 1 hour.
Maximum size of archive file: 2000000 KB

Exporting files to ftp servers is enabled.
File compression is disabled.
Export interval: every-day every 1 hour.

ftp-server           username        directory
10.1.1.1              mylogin        /ftpdirectory
10.2.2.2              mylogin        /ftpdirectory
 Working Log file - size: 103
                    age: 0
```

Note For security reasons, passwords are never displayed.

Configuring Intervals Between 1 Hour and 1 Day

The archive or export interval can be set for once a day with a specific time stamp. It can also be set for hour frequencies that align with midnight. For example, every 4 hours means archiving occurs at 00:00, 04:00, 08:00, 12:00, or 16:00. It is not possible to archive at half-hour intervals such as 00:30, 04:30, or 08:30. The following intervals are acceptable: `1, 2, 3, 4, 6, 8, 12, and 24`

```
ContentEngine(config)# transaction-logs archive interval
  <120-604800>  Interval time in seconds <120 - 604800 seconds>
  every-day     Archive using frequencies of 1 day or less
  every-hour    Archive using frequencies of 1 hour or less
  every-week    Archive one or more times a week
ContentEngine(config)# transaction-logs archive interval every-day
  at     Specify the local time at which to archive each day
  every  Specify the interval in hours. It will align with midnight
ContentEngine(config)# transaction-logs archive interval every-day every
  1   Hourly
  12  Every 12 hours
  2   Every 2 hours
  24  Every 24 hours
  3   Every 3 hours
  4   Every 4 hours
  6   Every 6 hours
  8   Every 8 hours
```

This example shows the command to archive every 4 hours aligned with midnight local time (that is, 00:00, 04:00, 08:00, 12:00, 16:00, 20:00).

```
ContentEngine(config)# transaction-logs archive interval every-day every 4
```

This example shows the command to export once a day at midnight local time.

```
ContentEngine(config)# transaction-logs export interval every-day every 24
```

Configuring Intervals of 1 Hour or Less

The interval can be set for once an hour with a minute alignment. It can also be set for frequencies of less than an hour; these frequencies will align with the top of the hour. That is, every 5 minutes means archiving will occur at 17:00, 17:05, and 17:10.

```
ContentEngine(config)# transaction-logs archive interval every-hour
  at        Specify the time at which to archive each day
  every     Specify interval in minutes. It will align with top of the hour

ContentEngine(config)# transaction-logs archive interval every-hour at
  <0-59>  Specify the minute alignment for the hourly archive
ContentEngine(config)# transaction-logs archive interval every-hour every
  <2-30>  Interval in minutes: {2, 5, 10, 15, 20, 30}
```

Configuring Export Interval on Specific Days

The export interval can be set for specific days of the week at a specific time. One or more days can be specified. The default time is midnight.

The administrator must be aware that archived logs are automatically deleted when free disk space is low. It is important to select an export interval that exports files frequently enough so that files are not automatically removed prior to export. The following example shows an export interval of every Monday, Wednesday, Friday, and Saturday at 2:00 a.m.

```
ContentEngine(config)# transaction-logs export interval mon wed fri sat at 02:00
```

Related Commands

clear transaction-log

show transaction-logging

show statistics transaction-logs

transaction-log force

trusted-host

To enable trusted host, use the **trusted-host** global configuration command. To disable trusted hosts, use the **no** form of this command.

trusted-host {*hostname* | *ip-address* | **domain-lookup**}

no trusted-host {**domain-lookup**}

Syntax Description

hostname	Host name of trusted host.
ip-address	IP address of trusted host.
domain-lookup	Checking trusted host.

Command Modes

Global configuration

Usage Guidelines

To allow reception of files (for example, remote copy protocol) from specified hosts, these hosts must be identified using the **trusted-host** *hostname* command. You must first enable this feature with the **trusted-host domain-lookup** command.

Examples

```
Console(config)# trusted-host domain-lookup

Console(config)# trusted-host 172.31.90.33

Console(config)# no trusted-host domain-lookup
```

Related Commands

show trusted-host

type

To display a file, use the **type** EXEC command.

type *filename*

Syntax Description		
	filename	Name of file.

Defaults

No default behavior or values

Command Modes

EXEC

Usage Guidelines

Use this command to display the contents of a file within any Content Engine file directory. This command may be used to monitor features such as transaction logging or system logging (syslog).

Examples

```
ContentEngine# type /local1/syslog.txt
```

Related Commands

cpfile

dir

lls

ls

mkfile

type-tail

To view a specified number of lines of the end of a log file or to view the end of the file continuously as new lines are added to the file, use the **type-tail** command in EXEC mode.

type-tail *filename* [*line* | **follow**]

Syntax Description

filename	File to be examined.
line	(Optional) Number of lines at the end of the file to be displayed (1–65535).
follow	(Optional) Displays the end of the file continuously as new lines are added to the file.

Defaults

10 lines shown

Command Modes

EXEC

Usage Guidelines

This command allows you to monitor a log file by letting you view the end of the file. You can specify the number of lines at the end of the file that you want to view, or you can follow the last line of the file as it continues to log new information. To stop the last line from continuously scrolling as with the **follow** command, use the key sequence **Ctrl-C**.

Examples

This example shows the list of log files in the /local1 directory.

```
stream-ce# ls /local1
core_dir
errloglive
errorlog
logs
lost+found
medialogs
service_logs
syslog.txt
```

This example displays the last ten lines of the syslog.txt file. In this example, the number of lines to display is not specifed; however, ten lines is the default.

```
stream-ce# type-tail /local1/syslog.txt
Oct  8 21:49:15 stream-ce syslog:(26830)TRCE:input_serv.c:83-> select_with
return 0, ready = 0
Oct  8 21:49:15 stream-ce syslog:(26832)TRCE:al_master.c:246-> select_with
return 0, ready = 0
Oct  8 21:49:15 stream-ce syslog:(26832)TRCE:in_mms.c:1747-> tv = NULL
Oct  8 21:49:17 stream-ce syslog:(26830)TRCE:input_serv.c:83-> select_with
return 0, ready = 0
Oct  8 21:49:17 stream-ce syslog:(26832)TRCE:al_master.c:246-> select_with
return 0, ready = 0
Oct  8 21:49:17 stream-ce syslog:(26832)TRCE:in_mms.c:1747-> tv = NULL
Oct  8 21:49:19 stream-ce syslog:(26830)TRCE:input_serv.c:83-> select_with
return 0, ready = 0
```

```
Oct  8 21:49:19 stream-ce syslog:(26832)TRCE:al_master.c:246-> select_with
return 0, ready = 0
Oct  8 21:49:19 stream-ce syslog:(26832)TRCE:in_mms.c:1747-> tv = NULL
Oct  8 21:49:21 stream-ce syslog:(26830)TRCE:input_serv.c:83-> select_with
return 0, ready = 0
```

This example displays the last 20 lines of the syslog.text file.

```
stream-ce# type-tail /local1/syslog.txt 20
Oct  8 21:49:11 stream-ce syslog:(26832)TRCE:al_master.c:246-> select_with
return 0, ready = 0
Oct  8 21:49:11 stream-ce syslog:(26832)TRCE:in_mms.c:1747-> tv = NULL
Oct  8 21:49:13 stream-ce syslog:(26830)TRCE:input_serv.c:83-> select_with
return 0, ready = 0
Oct  8 21:49:13 stream-ce syslog:(26832)TRCE:al_master.c:246-> select_with
return 0, ready = 0
Oct  8 21:49:13 stream-ce syslog:(26832)TRCE:in_mms.c:1747-> tv = NULL
Oct  8 21:49:15 stream-ce syslog:(26830)TRCE:input_serv.c:83-> select_with
return 0, ready = 0
Oct  8 21:49:15 stream-ce syslog:(26832)TRCE:al_master.c:246-> select_with
return 0, ready = 0
Oct  8 21:49:15 stream-ce syslog:(26832)TRCE:in_mms.c:1747-> tv = NULL
Oct  8 21:49:17 stream-ce syslog:(26830)TRCE:input_serv.c:83-> select_with
return 0, ready = 0
Oct  8 21:49:17 stream-ce syslog:(26832)TRCE:al_master.c:246-> select_with
return 0, ready = 0
Oct  8 21:49:17 stream-ce syslog:(26832)TRCE:in_mms.c:1747-> tv = NULL
Oct  8 21:49:19 stream-ce syslog:(26830)TRCE:input_serv.c:83-> select_with
return 0, ready = 0
Oct  8 21:49:19 stream-ce syslog:(26832)TRCE:al_master.c:246-> select_with
return 0, ready = 0
Oct  8 21:49:19 stream-ce syslog:(26832)TRCE:in_mms.c:1747-> tv = NULL
Oct  8 21:49:21 stream-ce syslog:(26830)TRCE:input_serv.c:83-> select_with
return 0, ready = 0
Oct  8 21:49:21 stream-ce syslog:(26832)TRCE:al_master.c:246-> select_with
return 0, ready = 0
Oct  8 21:49:21 stream-ce syslog:(26832)TRCE:in_mms.c:1747-> tv = NULL
Oct  8 21:49:23 stream-ce syslog:(26830)TRCE:input_serv.c:83-> select_with
return 0, ready = 0
Oct  8 21:49:23 stream-ce syslog:(26832)TRCE:al_master.c:246-> select_with
return 0, ready = 0
Oct  8 21:49:23 stream-ce syslog:(26832)TRCE:in_mms.c:1747-> tv = NULL
```

This example follows the file as it grows.

```
stream-ce# type-tail /local1/syslog.txt ?
  <1-65535>  The numbers of lines from end
  follow     Follow the file as it grows
  <cr>
stream-ce# type-tail /local1/syslog.txt follow
Oct  8 21:49:39 stream-ce syslog:(26832)TRCE:in_mms.c:1747-> tv = NULL
Oct  8 21:49:41 stream-ce syslog:(26830)TRCE:input_serv.c:83-> select_with
return 0, ready = 0
Oct  8 21:49:41 stream-ce syslog:(26832)TRCE:al_master.c:246-> select_with
return 0, ready = 0
Oct  8 21:49:41 stream-ce syslog:(26832)TRCE:in_mms.c:1747-> tv = NULL
Oct  8 21:49:43 stream-ce syslog:(26830)TRCE:input_serv.c:83-> select_with
return 0, ready = 0
Oct  8 21:49:43 stream-ce syslog:(26832)TRCE:al_master.c:246-> select_with
return 0, ready = 0
Oct  8 21:49:43 stream-ce syslog:(26832)TRCE:in_mms.c:1747-> tv = NULL
Oct  8 21:49:45 stream-ce syslog:(26830)TRCE:input_serv.c:83-> select_with
return 0, ready = 0
Oct  8 21:49:45 stream-ce syslog:(26832)TRCE:al_master.c:246-> select_with
```

```
return 0, ready = 0
Oct  8 21:49:45 stream-ce syslog:(26832)TRCE:in_mms.c:1747-> tv = NULL
Oct  8 21:49:47 stream-ce syslog:(26830)TRCE:input_serv.c:83-> select_with
return 0, ready = 0
Oct  8 21:49:47 stream-ce syslog:(26832)TRCE:al_master.c:246-> select_with
return 0, ready = 0
Oct  8 21:49:47 stream-ce syslog:(26832)TRCE:in_mms.c:1747-> tv = NULL
Oct  8 21:49:49 stream-ce syslog:(26830)TRCE:input_serv.c:83-> select_with
return 0, ready = 0
Oct  8 21:49:49 stream-ce syslog:(26832)TRCE:al_master.c:246-> select_with
return 0, ready = 0
Oct  8 21:49:49 stream-ce syslog:(26832)TRCE:in_mms.c:1747-> tv = NULL
```

undebug

To disable debugging functions, use the **undebug** EXEC command. Also see the **debug** EXEC command.

undebug

Syntax Description

This command has no arguments or keywords.

Command Modes

EXEC

Usage Guidelines

It is recommended that **debug** commands be used only at the direction of Cisco Systems technical support personnel.

Related Commands

debug

no debug

show debug

url-filter

To configure URL filtering, use the **url-filter** command in global configuration mode. Use the **no** form of this command to disable selected options.

url-filter bad-sites-deny {**enable** | *filename*}

url-filter custom-message *dirname*

url-filter good-sites-allow {**enable** | *filename*}

url-filter N2H2 {**allowmode enable** | **enable** | **server** {*hostname* | *ip-address*} [**port** *portnum* [**timeout** *seconds*]]}

url-filter smartfilter enable

url-filter websense {**allowmode enable** | **enable** | **server** {*hostname* | *ip-address*} [**port** *portnum* [**timeout** *seconds*]]}

no url-filter bad-sites-deny {**enable** | *filename*} | **custom-message** *dirname* | **good-sites-allow** {**enable** | *filename*} | **N2H2** {**allowmode enable** | **enable** | **server** {*hostname* | *ip-address*} [**port** *portnum* [**timeout** *seconds*]]} | **smartfilter enable** | **websense** {**allowmode enable** | **enable** | **server** {*hostname* | *ip-address*} [**port** *portnum* [**timeout** *seconds*]]}

Syntax Description

bad-sites-deny	Configures the bad sites list for access denial.
enable	Enables bad list URL filtering.
filename	Name of the file that contains the bad sites list.
custom-message	Sends customized access-denied message. Specifies the directory that contains the block.html file.
dirname	Name of the directory that contains the block.html file.
good-sites-allow	Configures the good sites list for allowing access.
enable	Allows good list URLs.
filename	Name of the file that contains the good sites list.
N2H2	Configures the N2H2 server to determine URL access policy.
allowmode enable	Enables access to a site if the N2H2 server does not respond.
enable	Enables N2H2 filtering.
server	Configures N2H2 server parameters.
hostname	Host name of the N2H2 server.
ip-address	IP address of the N2H2 server.
port	(Optional) Establishes the N2H2 server port number.
portnum	Port on which to send the N2H2 requests (1–65535).
timeout	(Optional) Configures the maximum time to wait for a response from the N2H2 server.
seconds	Timeout value in seconds (0–20). The default is 5 seconds.
smartfilter	Configures SmartFilter URL filtering.
enable	Enables SmartFilter filtering.

websense	Configures Websense parameters.
allowmode	Allows access to a site if the Websense server does not respond.
enable	Enables allow mode.
enable	Enables Websense filtering.
server	Specifies the Websense server.
hostname	Host name of the Websense server.
ip-address	IP address of the Websense server.
port	(Optional) Establishes the Websense server port number.
portnum	Port on which to send the Websense requests (1–65535).
timeout	(Optional) Configures the maximum time to wait for a response from the Websense server.
seconds	Timeout value in seconds (0–240).

Defaults

url-filter N2H2 allowmode: enabled

url-filter websense allowmode: enabled

N2H2 server port *port_num*: 4005

N2H2 sever timeout *seconds*: 5

websense server port *port_num*: 15868

websense server timeout *seconds*: 20

smartfilter: disabled

Command Modes

Global configuration

Usage Guidelines

The URL filtering feature allows the Content Engine to control client access to websites in any of the following ways:

- Deny access to URLs specified in a list.
- Permit access only to URLs specified in a list.
- Direct traffic to an N2H2 server for filtering.
- Direct traffic to a Websense Enterprise server for filtering.

Note To ensure that URL filtering applies to every URL that passes through the Content Engine, disable all bypass features. Bypass features **error-handling transparent** and **bypass load** are enabled initially by default and must be disabled manually. For error handling, use the **error-handling send-cache- error** or **error-handling reset-connection** command instead.

Only one form of URL filtering can be active at any one time. The URL filtering feature existed in Cache software 2.x releases. The URL filtering feature in ACNS 4.x software differs from the URL feature in other releases as follows:

- An **enable** command option now exists for the **good-sites-allow** and **bad-sites-deny** options.
- URL list filenames and customized blocking message directory name are now specified in the command-line interface (CLI).
- The **url-filter local-list-reload** command now dynamically refreshes a local URL list.
- The command **bad-sites-block** has been changed to **bad-sites-deny**.

URL Filtering with URL Lists

You can configure the Content Engine to deny client requests for URLs that are listed in a badurl.lst file, or configure it to fulfill only requests for URLs in a goodurl.lst file.

Follow these steps to deny requests for specific URLs:

Step 1 Create a plain text file named badurl.lst and enter the URLs that you want to block. The list of URLs in the badurl.lst file must be written in the form www.domain.com and delimited with carriage returns.

Step 2 Copy the badurl.lst file to the /local1 sysfs directory of the Content Engine.

Note We recommend creating a separate directory under local1 to hold the bad and good lists. For example, /local1/filtered_urls.

Step 3 Use the **url-filter bad-sites-deny** command to point to the bad URL list.

```
Console(config)# url-filter bad-sites-deny local/local1/badurl.lst
```

Step 4 Use the **url-filter bad-sites-deny enable** command to actively deny the bad URLs.

```
Console(config)# url-filter bad-sites-deny enable
```

Use the **no** form of this command to disable URL blocking.

```
Console(config)# no url-filter bad-sites-deny enable
```

Follow these steps to permit specific URLs to the exclusion of all other URLs:

Step 1 Create a plain text file named goodurl.lst.

In this file, enter the URLs that you want to exclusively allow. The list of URLs in the goodurl.lst file must be written in the form www.domain.com and delimited with carriage returns.

Step 2 Copy the goodurl.lst file to the /local1 sysfs directory of the Content Engine.

Note We recommend creating a separate directory under local1 to hold the bad and good lists. For example, /local1/filtered_urls.

Step 3 Use the **url-filter good-sites-allow** command to point to the goodurl.lst file.

```
Console(config)# url-filter good-sites-allow local/local1/goodurl.lst
```

Step 4 Use the **url-filter good-sites-allow enable** command to actively permit only the good URLs.

```
Console(config)# url-filter good-sites-allow enable
```

Use the **no** form of this command to disable the permission of only good URLs.

```
Console(config)# no url-filter good-sites-allow enable
```

Note When you update the badurl.lst or goodurl.lst file, use the **url-filter local-list-reload** EXEC command to recopy the URL list file to the Content Engine.

Custom Blocking Messages

The Content Engine with ACNS 4.x software can be configured to return a customized blocking message to the client. The custom message must be an administrator-created HTML page named block.html. Make sure to copy all embedded graphics associated with the custom message HTML page to the same directory that contains the block.html file. To enable the customized blocking message, use the **url-filter custom-message** command and specify the directory name.

To disable the custom message, use the **no url-filter custom-message** command.

The **url-filter custom-message** command can be enabled and disabled without affecting the **good-list** and **bad-list** configuration.

Note Do not use local1 or local2 as directories. Create a separate directory under local1 or local2 for holding the custom message file.

In the block.html file, objects (such as .gif, .jpeg, and so on) must be referenced with the string /content/engine/blocking/url, as shown in the example below.

The following is an example of a block.html file:

```
<TITLE>Cisco Content Engine example customized message for url-filtering</TITLE>
<p>
<H1>
<CENTER><B><I><BLINK>
<FONT COLOR="#800000">P</FONT>
<FONT COLOR="#FF00FF">R</FONT>
<FONT COLOR="#00FFFF">A</FONT>
<FONT COLOR="#FFFF00">D</FONT>
<FONT COLOR="#800000">E</FONT>
<FONT COLOR="#FF00FF">E</FONT>
<FONT COLOR="#00FFFF">P</FONT>
<FONT COLOR="#FF8040">'</FONT>
<FONT COLOR="#FFFF00">S</FONT>
</BLINK>
<FONT COLOR ="#0080FF">Blocked Page</FONT>
</I></B></CENTER>
</H1>
<p>
<p>
<IMG src="/content/engine/blocking/url/my.gif">
<p>
This page is blocked by the Content Engine.
<p>
```

To disable the **custom-message** option without disabling URL filtering, enter the URL filtering command without the **custom-message** option (for example, **url-filter good-sites-allow**).

RADIUS and URL Filtering

When both RADIUS authentication and URL filtering are enabled on the Content Engine, users can be configured to bypass URL filtering using the Filter-Id user attribute in the RADIUS server database.

The following is an example of a userFilter-Id attribute entry in the RADIUS server database.

```
test        Password = "test"
            Service-Type = Framed-User,
            Filter-Id = "No-Web-Blocking"
```

The Filter-Id attribute is defined as either No-Web-Blocking or Yes-Web-Blocking. If blocking is not specified, Yes-Web-Blocking is the default RADIUS filter. Yes-Web-Blocking means that the request is subject to URL filtering and No-Web-Blocking means that the request is not subject to URL filtering.

URL Filtering with the N2H2 Server

To effectively enforce Internet usage policies, Internet access needs to be managed. By combining the N2H2 database with fully managed solutions and plug-ins, N2H2 is able to effectively perform URL filtering. The Content Engine can use an N2H2 Enterprise server as a filtering engine and enforce the filtering policy configured on the N2H2 server. Refer to the *N2H2 Internet Filtering Administrator's Guide* for further information on N2H2 filtering configuration and policies. The following commands are used to configure N2H2 feature on the Content Engine.

The command **url-filter N2H2 server** *ip-address* [**port** *1-65535* [**timeout** *1-20*]] is available at privilege level 0 and configures the Content Engine to query the N2H2 server. The optional port field specifies the port on the N2H2 server to which the Content Engine sends IFP requests. The default port number is 4005. The optional timeout (in seconds) specifies how long the Content Engine waits for an IFP response from the N2H2 server. The default timeout is 5 seconds. If there is an error in the connection, two request retries are attempted before it fails.

This command does not verify whether or not an N2H2 server is accessible at the specified IP address. The configuration can be changed while N2H2 is enabled, and the Content Engine accepts the new configuration during run time.

The **url-filter N2H2 enable** command is available at privilege level 0 and enables N2H2 URL filtering. If the URL filter is already enabled with N2H2 or other filtering schemes, the enable command fails. Even if the server IP address is not configured, the command is accepted, but filtering does not take effect until the N2H2 server is properly configured.

The **url-filter N2H2 allowmode enable** command is available at privilege level 0. Allow mode is a way to handle HTTP requests when N2H2 filtering is enabled and the Content Engine has difficulty communicating with the N2H2 server. When the Content Engine fails to receive responses from the N2H2 server with the **allowmode enable option** set, it allows all traffic to pass through it and proceeds with normal HTTP processing. With the **no url-filter N2H2 allowmode enable command** (disabling the N2H2 allow mode), the Content Engine blocks all HTTP traffic attempts that are routed to it. Allow mode is disabled by default and can be configured with or without enabling N2H2; it is independent of the N2H2 server configuration. The Content Engine can accept the new configuration for allow mode even if N2H2 filtering is already enabled.

To ensure that all traffic is filtered by the N2H2 server with a cluster of Content Engines, make sure to use the **url-filter N2H2 enable** command on each Content Engine in the cluster.

URL Filtering with the Websense Enterprise Server

The Content Engine can use a Websense Enterprise server as a filtering engine and enforce the filtering policy configured on the Websense server. Refer to the *Websense Administrator's Guide* located at http://www.websense.com for further information on Websense configuration instructions and filtering policies.

Before you enable Websense URL filtering on the Content Engine, configure the Websense server IP address or host name and optionally enter the **port** number and maximum **timeout** parameter. The **timeout** option sets the maximum amount of time that the Content Engine will wait for a Websense response. The timeout default is 20 seconds. The **port** option specifies the port number on which the server will intercept requests from the Content Engine (the default port is 15868). Use the **no url-filter websense server** command to disable Websense URL filtering. Enable Websense URL filtering by entering the **url-filter websense enable** command.

The **url-filter websense allowmode enable** command permits the Content Engine to fulfill the client request after a Websense server timeout. The Websense server returns its own blocking message.

To use Websense URL filtering with a cluster of Content Engines, make sure to configure the **url-filter websense server** command on each Content Engine in the cluster to ensure that all traffic is filtered.

URL Filtering with SmartFilter Software

SmartFilter software provides employee Internet management (EIM) functionality with proxy servers, firewalls, and caching appliances. SmartFilter software filtering capability may be enabled by entering the **url-filter smartfilter enable** command. To disable SmartFilter software filtering, enter **no url-filter smartfilter enable**. To ensure that all traffic is filtered by SmartFilter software with a cluster of Content Engines, make sure to apply the **url-filter smartfilter enable** command on each Content Engine in the cluster. For more information on configuring SmartFilter software, refer to the *SmartFilter for Cisco Content Engine User's Guide, Release 3.0.2*.

Examples

To block list URLs, enter this command.

```
Console(config)# url-filter bad-sites-deny badurl.lst
```

To disable URL blocking, use the **no** form of this command.

```
Console(config)# no url-filter bad-sites-deny enable
Console(config)# no url-filter good-sites-allow enable
```

To enable a custom message, first specify the directory in which the block.html file is located and then enter the **enable** command.

```
Console(config)# url-filter custom-message /local1/url_dir
Console(config)# url-filter custom-message enable
```

To configure a Content Engine to use N2H2 URL filtering with IP address 172.16.22.10, port 4008, and a 6-second timeout, enter this command.

```
Console(config)# url-filter N2H2 server 172.16.22.10 port 4008 timeout 6
```

To configure a Content Engine to use Websense URL filtering with IP address 172.16.11.22, port 15900, and a 4-second timeout, enter this command.

```
Console(config)# url-filter websense server 172.16.11.22 port 15900 timeout 4
```

To enable a Content Engine to use SmartFilter URL filtering, enter this command.

```
Console(config)# url-filter smartfilter enable
starting smartfilter
```

Related Commands

clear statistics url-filter websense
clear statistics url-filter N2H2
debug url-filter websense
debug url-filter N2H2
show url-filter
show statistics url-filter websense
show statistics url-filter N2H2
url-filter local-list-reload

url-filter local-list-reload

To reload new good site or bad site lists on the Content Engine when the **url-filter** feature is enabled, use the **url-filter local-list-reload** command in EXEC mode.

url-filter local-list-reload

Syntax Description

local-list-reload	Reloads the lists of bad and good URLs when the **url-filter** global configuration command is enabled.

Defaults

No default behavior or values

Command Modes

EXEC

Usage Guidelines

Use the **url-filter local-list-reload** command to reload to the latest good sites or bad sites lists that you created or edited using the **url-filter** command.

Examples

```
Console# url-filter local-list-reload
```

Related Commands

url-filter

username

To establish username authentication, use the **username** global configuration command.

username *name* {**password** {**0** *word* | **1** *word* | *word*} | **privilege** {**0-0** | **15-15** | **200-300**}}

Syntax Description

name	Username.
password	Establishes password.
0	Specifies clear-text password.
1	Specifies type 1 encrypted password.
word	User password (clear text).
privilege	Sets user privilege level.
0-0	Sets user privilege (0-0) to normal user.
15-15	Sets user privilege (15-15) to superuser.
200-300	Reserved for system use.

Defaults

The **password** value is set to 0 (clear-text) by default.

Command Modes

Global configuration

Usage Guidelines

The **username** global configuration command changes the password and privilege level for existing user accounts.

Examples

This example demonstrates how passwords and privilege levels are reconfigured.

```
ContentEngine# show user username abeddoe
Uid                 : 2003
Username            : abeddoe
Password            : ghQ.GyGhP96K6
Privilege           : normal user
ContentEngine# show user username bwhidney
Uid                 : 2002
Username            : bwhidney
Password            : bhlohlbIwAMOk
Privilege           : normal user
ContentEngine(config)# username bwhidney password 1 victoria
ContentEngine(config)# username abeddoe privilege 15
User's privilege changed to super user (=15)
ContentEngine# show user username abeddoe
Uid                 : 2003
Username            : abeddoe
Password            : ghQ.GyGhP96K6
Privilege           : super user
```

```
ContentEngine# show user username bwhidney
Uid                 : 2002
Username            : bwhidney
Password            : mhYWYw.7P1Ld6
Privilege           : normal user
```

Related Commands

show user

show users

user

wccp custom-web-cache

To enable the Content Engine to accept redirected HTTP traffic on a port other than 80, use the **wccp custom-web-cache** command. To disable custom web caching, use the **no** form of the command.

wccp custom-web-cache {**router-list-num** *num* **port** *port* [[**hash-destination-ip**] [**hash-destination-port**] [**hash-source-ip**] [**hash-source-port**] [**l2-redirect**] [**password** *key*] [**weight** *percentage*]]}

no wccp custom-web-cache {**router-list-num** *num* **port** *port* [[**hash-destination-ip**] [**hash-destination-port**] [**hash-source-ip**] [**hash-source-port**] [**l2-redirect**] [**password** *key*] [**weight** *percentage*]]}

Syntax Description

router-list-num	Sets router list number.
num	Router list number (1–8).
port	Sets port number.
port	Port list number (1–65535).
hash-destination-ip	(Optional) Defines load-balancing hash of destination IP (default).
hash-destination-port	(Optional) Defines load-balancing hash of destination port.
hash-source-ip	(Optional) Defines load-balancing hash of source IP.
hash-source-port	(Optional) Defines load-balancing hash of source port.
l2-redirect	(Optional) Packet forwarding by Layer 2 redirect.
password	(Optional) Sets authentication password.
key	WCCP service password key.
weight	(Optional) Sets weight percentage for load balancing.
percentage	Percentage value (0–100).

Defaults

No default behavior or values

Command Modes

Global configuration

Usage Guidelines

The **wccp custom-web-cache** command can configure the Content Engine to automatically establish WCCP Version 2 redirection services with a Cisco router on a user-specified port number and then perform transparent web caching for all HTTP requests over that port while port 80 transparent web caching continues without interruption. For custom web caching, service 98 must be enabled on the routers. WCCP Version 1 does not support custom web caching.

Transparent caching on ports other than port 80 can be performed by the Content Engine when WCCP is not enabled or when client browsers have previously been configured to use a legacy proxy server. See the **http proxy** global configuration command for further information.

The **l2-redirect** option permits the Content Engine to receive transparently redirected traffic from a WCCP Version 2-enabled switch or router if the Content Engine has a Layer 2 connection with the device, and the device is configured for Layer 2 redirection.

The **weight** parameter represents a percentage of load redirected to the Content Engine cluster (for example, a Content Engine with a weight of 30 receives 30 percent of the total load). If the total of all weight parameters in the Content Engine cluster exceeds 100, the percentage load for each Content Engine is recalculated as the percentage that its weight parameter represents of the combined total.

Examples

The following example shows the configuration for starting custom web caching on interface 3 of a WCCP Version 2-enabled router.

```
router(config): ip wccp 98

[Output not shown]

router(config-if): ip interface 3
router(config-if): ip web-cache 98 redirect out

[Output not shown]
```

The following example shows the configuration on the Content Engine.

```
ContentEngine(config)# wccp custom-web-cache router-list-num 5 port 82 weight 30 password
Allied hash-destination-ip hash-source-port

ContentEngine(config)# no wccp custom-web-cache

ContentEngine(config)# http proxy outgoing ans.allied.com 82 no-local-domain

ContentEngine# show running-config
Building configuration...
Current configuration:
!
....
!
http proxy outgoing 192.168.200.68 82 no-local-domain
!
wccp router-list 5 10.1.1.1
wccp custom-web-cache router-list 5 port 82 weight 30 password Allied hash-destination-ip
hash-source-port
wccp home-router 10.1.1.2
wccp version 2
!
end
```

Related Commands

wccp web-cache

http proxy incoming

http proxy outgoing

wccp flow-redirect

To enable WCCP flow redirection, use the **flow-redirect enable** global configuration command. Use the **no** form of the command to disable flow redirection.

wccp flow-redirect enable

no wccp flow-redirect enable

Syntax Description

enable	Enables flow redirection.

Defaults

Enabled

Command Modes

Global configuration

Usage Guidelines

This command works with WCCP Version 2 only. The flow protection feature is designed to keep the TCP flow intact as well as to not overwhelm Content Engines when they come up or are reassigned new traffic. This feature also has a slow start mechanism whereby the Content Engines try to take a load appropriate for their capacity.

Examples

```
ContentEngine(config)# wccp flow-redirect enable
```

Related Commands

wccp slow-start enable

wccp home-router

To configure a WCCP Version 1 router IP address, use the **wccp home-router** global configuration command. To disable this function, use the **no** form of this command.

wccp home-router *ip-address*

no wccp home-router *ip-address*

Syntax Description

ip-address	Home router IP address.

Defaults

Disabled

Command Modes

Global configuration

Usage Guidelines

To use WCCP Version 1 with the Content Engine, you must also point the Content Engine to a designated home router. Use the **wccp home-router** *ip-address* command to do this. This may also be the address of the IP default gateway.

Make sure that WCCP Version 1 is enabled on the router.

Examples

```
ContentEngine(config)# wccp home-router 172.16.65.243
ContentEngine(config)# no wccp home-router 172.16.65.243
```

Related Commands

show wccp routers

wccp version 1

wccp media-cache

To enable the Content Engine to accept redirected RTSP traffic, use the **wccp media-cache** global configuration command. To disable media caching, use the **no** form of the command.

wccp media-cache {**router-list-num** *num* [[**l2-redirect**] [**password** *key*] [**weight** *percentage*]]}

no wccp media-cache {**router-list-num** *num* [[**l2-redirect**] [**password** *key*] [**weight** *percentage*]]}

Syntax Description

router-list-num	Sets router list number.
num	Router list number (1–8).
l2-redirect	(Optional) Packet forwarding by Layer 2 redirect.
password	(Optional) Sets authentication password.
key	WCCP service password key.
weight	(Optional) Sets weight percentage for load balancing.
percentage	Percentage value (0–100).

Defaults

The default for the **weight** option is 100; that is, 100 percent of RTSP traffic is redirected to the Content Engine.

Command Modes

Global configuration

Usage Guidelines

The **wccp media-cache** command can configure the Content Engine to automatically establish WCCP Version 2 redirection services with a list of Cisco routers and then perform transparent web caching for all RTSP requests on port 554.

The **L2-redirect** option permits the Content Engine to receive transparently redirected traffic from a WCCP Version 2-enabled switch or router if the Content Engine has a Layer 2 connection with the device, and the device is configured for Layer 2 redirection.

The **weight** parameter represents a percentage of load redirected to the Content Engine cluster (for example, a Content Engine with a weight of 30 receives 30 percent of the total load). If the total of all weight parameters in the Content Engine cluster exceeds 100, the percentage load for each Content Engine is recalculated as the percentage that its weight parameter represents of the combined total.

Examples

This is an example of configuration to enable transparent redirection of RTSP traffic to the Content Engine running RealProxy. It is assumed that RealProxy has been installed and configured, and that 100 percent of the RTSP traffic is redirected to the Content Engine (default of the **weight** option).

```
ContentEngine(config)# rtsp proxy media-real enable
ContentEngine(config)# wccp router-list 1 172.16.25.25 172.16.25.24
ContentEngine(config)# wccp media-cache router-list-num 1
ContentEngine(config)#
```

On the router side, Ethernet0 is the outbound interface to the Internet.

```
router(config)# ip wccp 80
router(config)# interface Ethernet 0
router(interface)# ip wccp 80 redirect out
```

Related Commands

wccp web-cache

rtsp proxy incoming

wccp port-list

To associate ports with specific WCCP Version 2 dynamic services, use the **wccp port-list** global configuration command.

wccp port-list *listnum portnum*

no wccp port-list *listnum portnum*

Syntax Description

listnum	Port list number (1–8).
portnum	Port number (1–65535). Up to eight ports per list number are allowed.

Defaults

No default behavior or values

Command Modes

Global configuration

Usage Guidelines

Up to eight port numbers can be included in a single port list. The port list is referenced by the **wccp service-number** command that configures a specific WCCP Version 2 dynamic service (90–97) to operate on the listed ports.

Examples

In the following example, ports 10, 200, 3000, 110, 220, 330, 440, and 40000 are included in port list 3.

```
ContentEngine(config)# wccp port-list 3 10 200 3000 110 220 330 440 40000
```

Related Commands

wccp service-number

wccp reverse-proxy

To enable WCCP Version 2 reverse proxy service, use the **wccp reverse-proxy** global configuration command. To disable this function, use the **no** form of this command.

wccp reverse-proxy {**router-list-num** *num* [[**l2-redirect**] [**password** *key*] [**weight** *percentage*]]}

no wccp reverse-proxy {**router-list-num** *num* [[**l2-redirect**] [**password** *key*] [**weight** *percentage*]]}

Syntax Description

router-list-num	Sets router list number.
num	Router list number (1–8).
l2-redirect	(Optional) Packet forwarding by Layer 2 redirect.
password	(Optional) Sets authentication password.
key	WCCP service password key.
weight	(Optional) Sets weight percentage for load balancing.
percentage	Percentage value (0–100).

Defaults

Disabled.

Command Modes

Global configuration

Usage Guidelines

This command applies only to WCCP Version 2.

You must configure the **wccp router-list** command before you use this command. The routers in the list must have WCCP reverse proxy service enabled (service 99). Refer to the *Cisco Cache Software Configuration Guide, Release 3.1.1* for information on configuring reverse proxy service on the router.

By default, the router does load balancing across the various Content Engines in a cluster based on the destination IP address (for example, web server IP address). When WCCP reverse proxy is enabled, the router does load balancing in a cluster based on the source IP address (for example, the client's browser IP address).

To enable the use of a password for a secure reverse proxy cache within a cluster, use the **wccp reverse-proxy password** *key* command to be sure to enable all other Content Engines and routers within the cluster with the same password.

The **L2-redirect** option permits the Content Engine to receive transparently redirected traffic from a WCCP Version 2-enabled switch or router if the Content Engine has a Layer 2 connection with the device, and the device is configured for Layer 2 redirection.

The **weight** parameter represents a percentage of the total load redirected to the Content Engine in a cluster (for example, a Content Engine with a weight of 30 receives 30 percent of the total load). If the total of all weight parameters in a Content Engine cluster exceeds 100, the percentage load for each Content Engine is recalculated as the percentage that its weight parameter represents of the combined total.

Examples

```
ContentEngine(config)# wccp reverse-proxy router-list-num 8 password mysecret weight 100

ContentEngine(config)# no wccp reverse-proxy
```

Related Commands

show wccp content-engines

show wccp services

wccp router-list

wccp version 2

wccp router-list

To configure a router list for WCCP Version 2, use the **wccp router-list** global configuration command. To disable this function, use the **no** form of this command.

wccp router-list *number ip-address*

no wccp router-list *number ip-address*

Syntax Description

number	Router list number (1–8).
ip-address	IP address of router to add to list.

Defaults

Disabled

Command Modes

Global configuration

Usage Guidelines

Use this command to configure various router lists for use with WCCP Version 2 services. For example, you can specify one router list for WCCP Version 2 web cache service and another list for reverse proxy at the same time without having to reconfigure groups of routers or caches. You can add up to 8 router lists and up to 32 IP addresses per list.

Examples

```
ContentEngine(config)# wccp router-list 7 172.31.68.98
ContentEngine(config)# no wccp router-list 7 172.31.68.98
```

Related Commands

wccp reverse-proxy

wccp web-cache

wccp version 2

wccp service-number

To enable up to eight dynamic WCCP redirection services on the Content Engine, use the **wccp service-number** global configuration command. The services must also be configured on the router running WCCP Version 2.

wccp service-number *servnumber* {**router-list-num** *num* **port** *port* **application** {**cache** | **streaming**} [[**hash-destination-ip**] [**hash-desination-port**] [**hash-source-ip**] [**hash-source-port**] [**l2-redirect**] [**password** *key*] [**weight** *percentage*]]}

no wccp service-number *servnumber* {**router-list-num** *num* **port** *port* **application** {**cache** | **streaming**} [[**hash-destination-ip**] [**hash-desination-port**] [**hash-source-ip**] [**hash-source-port**] [**l2-redirect**] [**password** *key*] [**weight** *percentage*]]}

Syntax Description

servnumber	WCCP Version 2 service number (90–97).
router-list-num	Sets router list number.
num	Router list number (1–8).
port	Sets port number.
port	Port list number (1–65535).
application	Specifies the application (caching or streaming media).
cache	Directs traffic to the caching application.
streaming	Directs traffic to the streaming media application.
hash-destination-ip	(Optional) Defines load-balancing hash of destination IP address (the default).
hash-destination-port	(Optional) Defines load-balancing hash of destination port.
hash-source-ip	(Optional) Defines load-balancing hash of source IP address.
hash-source-port	(Optional) Defines load-balancing hash of source port.
l2-redirect	(Optional) Packet forwarding by Layer 2 redirect.
password	(Optional) Sets authentication password.
key	WCCP service password key.
weight	(Optional) Sets weight percentage for load balancing.
percentage	Percentage value (0–100).

Defaults

If a load-balancing hash is defined, the destination IP address is the default.

Command Modes

Global configuration

Usage Guidelines

The **application cache** option redirects traffic to the Content Engine cache processes. The **application streaming** option redirects traffic to the Content Engine media processes.

Proxy Mode

The Content Engine supports up to eight incoming ports each for File Transfer Protocol (FTP), HTTPS, and HTTP proxy modes. The RTSP proxy currently permits only one proxy port. The incoming proxy ports can be the same ports that are used by the transparent-mode services. The incoming proxy ports can be changed without stopping any WCCP services running on the Content Engine or on other Content Engines in the farm.

The Content Engine parses requests received on a port to determine the protocol to be serviced. If the Content Engine is not configured to support a received protocol, the proxy server returns an error. For example, if port 8080 is configured to run an HTTP and HTTPS proxy service, an FTP request coming to this port is rejected.

Some TCP ports are reserved for system or network services and should not be used for proxying services in transparent mode or in proxy mode. If more than eight ports are required, the administrator can configure multiple custom WCCP services. Intercepted FTP, HTTP, and HTTPS requests addressed to other proxy servers (received on transparent-mode ports) are serviced according to the **proxy-protocols transparent** command parameters.

Transparent Mode

Transparent and proxy mode requests can be distinguished by comparison of the destination IP address of the request and the IP address of the Content Engine. A nonmatching IP address indicates that the request has been redirected and is therefore transparent. The style of the URL within the request may be proxy-style or server-style (that is, a server-style URL does not include the protocol and host name). In general, transparent requests have a server-style URL, but proxy style ones may also be received, for example, when the Content Engine is intercepting a request destined for a proxy. If a server-style URL is received, only HTTP is supported. If a proxy-style URL is received, all of the protocols enabled on the Content Engine are supported.

The **wccp service-number** command can enable up to eight WCCP redirection services on a Content Engine, provided that the services are also configured on the router. There are eight new dynamic WCCP services (90 to 97).

Each **wccp service-number** command specifies a router list, single port list (containing up to eight ports), application type, hash parameters, password, and weight. With eight custom services using a maximum number of eight ports each, the maximum number of ports that can be specified for transparent redirection is 64.

The legacy custom web cache and reverse proxy services (service numbers 98 and 99) can be configured with only one port each. If only one legacy service is configured, the total maximum number of transparent redirection ports is 57. If both legacy services are configured, the maximum port total is 50.

All ports receiving HTTP that are configured as members of the same WCCP service share the following characteristics:

- They have the same hash parameters as configured with the **wccp service-number** command.
- The service on individual ports cannot be stopped or started individually (WCCP Version 2 restriction).

With Content Engines in a farm, the following restrictions apply:

- All Content Engines that use the same WCCP service are required to configure the same list of ports and the same hash parameters.
- A Content Engine that tries to join the farm with the same WCCP service using a different list of ports or different hash parameters is rejected by the router.
- To change the port list for a particular WCCP service, WCCP service must be stopped on all involved Content Engines and then all must be restarted with the new parameters.

The Content Engine WCCP implementation currently allows global settings that apply to all WCCP services, such as healing parameters, slow start, and others. The multiple service model does not change that, and the settings in question remain global for the whole WCCP system.

Modifying Configurations

For proxy-mode and transparent-mode commands, issuing a new command replaces the old one. In proxy mode, a **no** command that specifies the protocol and no ports disables the service for that protocol. To add or remove ports in proxy mode, issue a new command that specifies all the ports to be used. Ports can also be removed by a **no** command with a list of ports to remove. A **no** command that specifies only some of the configured ports removes these ports from the list, and the service continues to run on the remaining ports. For example, if HTTPS is received on 8080, 8081, and 82, the **no https proxy incoming 8081** command disables port 8081 but permits the HTTPS proxy service to continue on ports 8080 and 82.

In transparent mode, to add or remove ports for a WCCP service, modify the port list or create a new port list for the WCCP service. In transparent mode, a **no** command that specifies the WCCP service number disables the service.

To use the **l2-redirect** hashing option, the Content Engine must be directly connected at Layer 2 to a switch or router that supports accelerated hardware switching.

Examples

In this example, WCCP dynamic service 90 is configured with router list 1 and port list 1. Port 8080 is the only element in port list 1.

```
ContentEngine(config)# wccp 90 router-list-num 1 port-list-number 1 hash-source-ip
hash-destination-port

ContentEngine(config)# wccp port-list 1 8080
```

In this example, the Content Engine is configured to accept HTTP and HTTPS proxy requests on ports 81, 8080, and 8081.

```
ContentEngine(config)# http proxy incoming 81 8080 8081
ContentEngine(config)# https proxy incoming 81 8080 8081
```

Related Commands

ftp proxy incoming

https proxy incoming

http proxy incoming

proxy-protocols

rtsp proxy incoming

show https proxy

show http proxy

show services

show wccp services

wccp shutdown

To set the maximum time interval after which the Content Engine will perform a clean shutdown, use the **wccp shutdown** global configuration command. To disable the clean shutdown, use the **no** form of the command.

wccp shutdown max-wait *seconds*

no wccp shutdown max-wait

Syntax Description

max-wait	Sets the clean shutdown time interval.
seconds	Time in seconds (0–86400). The default is 120 seconds.

Defaults

The maximum time interval before a clean shutdown is 120 seconds by default.

Command Modes

Global configuration

Usage Guidelines

To prevent broken TCP connections, the Content Engine performs a clean shutdown of WCCP after a **reload** or **wccp version** command is issued. The Content Engine does not reboot until either all connections have been serviced or the configured **max-wait** interval has elapsed.

During a clean shutdown, the Content Engine continues to service the flows it is handling but starts to bypass new flows. When the number of flows goes down to zero, the Content Engine takes itself out of the cluster by having its buckets reassigned to other Content Engines by the lead Content Engine. TCP connections can still be broken if the Content Engine crashes or is rebooted without WCCP being cleanly shut down. The clean shutdown can be aborted while in progress.

Examples

```
ContentEngine(config)# wccp shutdown max-wait 4999
```

Related Commands

wccp version

wccp slow-start

wccp flow-redirect

wccp slow-start

To enable the slow start capability of the Cisco Cache software on the Content Engine, use the **wccp slow-start enable** global configuration command. To disable slow start capability, use the **no** form of this command.

wccp slow-start enable

no wccp slow-start enable

Syntax Description

enable	Enables WCCP slow start.

Defaults

Enabled

Command Modes

Global configuration

Usage Guidelines

Within a cluster of Content Engines, TCP connections are redirected to other Content Engines as units are added or removed. A Content Engine can be overloaded if it is reassigned new traffic too quickly or introduced abruptly into a fat pipe.

WCCP slow start performs the following tasks to prevent a Content Engine from being overwhelmed when it comes online or is reassigned new traffic:

- TCP flow protection when WCCP 2 is enabled and a Content Engine is introduced into the cluster
- TCP flow protection when WCCP 2 is disabled and a Content Engine is leaving the cluster
- Load assignment to the Content Engine in slow increments rather than a full load at bootup

Slow start is applicable only in the following cases:

- Initial bootup when there is no Content Engine yet present in the server farm
- When a new Content Engine is added to a cluster that is not handling the full load; for example, when there are some buckets that are being shed by the cluster

In all other cases slow start is not necessary and all the Content Engines can be assigned their share of the buckets right away.

Examples

```
ContentEngine# wccp slow-start enable
ContentEngine# no wccp slow-start enable
```

Related Commands

wccp flow-redirect

wccp version

To specify the version of WCCP that the Content Engine should use, enter the **wccp version** global configuration command. Use the **no** form of the command to disable the currently running version.

wccp version {**1** | **2**}

no wccp version {**1** | **2**}

Syntax Description

1	WCCP Version 1.
2	WCCP Version 2.

Defaults

No default behavior or values

Command Modes

Global configuration

Usage Guidelines

Both WCCP versions allow transparent caching of web content. For a detailed description of both versions, refer to the *Cisco Cache Software Configuration Guide.* It is not necessary to disable WCCP Version 1 before enabling WCCP Version 2, and vice versa. Be sure the routers used in the WCCP environment are running a software version that supports the WCCP version configured on the Content Engine.

When operating with WCCP Version 2, the Content Engine performs a clean shutdown after a **reload**, **wccp version 1**, or **no wccp version 2** command is executed. A clean shutdown prevents broken TCP connections.

Examples

```
ContentEngine(config)# no wccp version 1
ContentEngine(config)# wccp version 2
```

Related Commands

wccp home-router

wccp web-cache

To instruct the router to run the web cache service with WCCP Version 2, use the **wccp web-cache** global configuration command. To disable this function, use the **no** form of this command.

wccp web-cache {**router-list-num** *num* [[**l2-redirect**] [**password** *key*] [**weight** *percentage*]]}

no wccp web-cache {**router-list-num** *num* [[**l2-redirect**] [**password** *key*] [**weight** *percentage*]]}

Syntax Description

router-list-num	Sets router list number.
num	Router list number (1–8).
l2-redirect	(Optional) Packet forwarding by Layer 2 redirect.
password	(Optional) Sets authentication password.
key	WCCP service password key.
weight	(Optional) Sets weight percentage for load balancing.
percentage	Percentage value (0–100).

Defaults

The default is no WCCP web cache.

Command Modes

Global configuration

Usage Guidelines

Use this command to enable web cache service with WCCP Version 2. With web cache service, the router balances the traffic load within a Content Engine cluster based on the destination IP address (for example, web server IP address).

You must set the **wccp router-list** command before you use this command.

Both **weight** and **password** are optional and may be used together or separately.

To enable the use of a password for a secure web cache cluster, use the **password** *key* option and be sure to enable all other Content Engines and routers within the cluster with the same password.

The **l2-redirect** option permits the Content Engine to receive transparently redirected traffic from a WCCP Version 2-enabled switch or router if the Content Engine has a Layer 2 connection with the device, and the device is configured for Layer 2 redirection.

The **weight** parameter represents a percentage of the total load redirected to the Content Engine (for example, a Content Engine with a weight of 30 receives 30 percent of the total load). If the total of all weight parameters in a Content Engine cluster exceeds 100, the percentage load for each Content Engine is recalculated as the percentage that its weight parameter represents of the combined total.

Examples

```
ContentEngine(config)# wccp web-cache router-list-num 1

ContentEngine(config)# no wccp web-cache
```

Related Commands

show wccp content-engines

show wccp routers

show wccp status

wccp version 2

wccp wmt

To instruct the router to run the web cache service with WCCP Windows Media Technologies (WMT), use the **wccp wmt** global configuration command. To disable this function, use the **no** form of this command.

wccp wmt {**router-list-num** *num* [[**l2-redirect**] [**password** *key*] [**weight** *percentage*]]}

no wccp wmt {**router-list-num** *num* [[**l2-redirect**] [**password** *key*] [**weight** *percentage*]]}

Syntax Description

router-list-num	Specifies router list number.
num	Router list number (1–8).
l2-redirect	(Optional) Packet forwarding by Layer 2 redirect.
password	(Optional) Specifies authentication password.
key	WCCP service password key.
weight	(Optional) Sets weight percentage for load balancing.
percentage	Percentage value (0–100).

Defaults

The default is no WCCP WMT web cache service.

Command Modes

Global configuration

Usage Guidelines

Set the **router-list-num** *num* command option to assign the routers.

The **l2-redirect** option permits the Content Engine to receive transparently redirected traffic from a WCCP Version 2-enabled switch or router if the Content Engine has a Layer 2 connection with the device, and the device is configured for Layer 2 redirection.

Both **weight** and **password** are optional and may be used in combination or separately.

To enable the use of a password for a secure web cache cluster, use the **password** *key* option and enable all other Content Engines and routers within the cluster with the same key.

The **weight** parameter is the percentage of the total load redirected to the Content Engine. If the total percentage of every weight parameter in a Content Engine cluster exceeds 100, the percentage for each load is recalculated as a percentage that its weight parameter represents of the combined total.

Examples

In the following examples, the **wccp wmt router-list-num** command sets the router to run the web cache service with assigned router list 2 and sets Layer 2 redirect, authentication password key, and cluster load balancing weight percentages.

```
ContentEngine(config)# wccp wmt router-list-num 2

ContentEngine(config)# wccp wmt router-list-num 2 password cisco

ContentEngine(config)# wccp wmt router-list-num 2 l2-redirect

ContentEngine(config)# wccp wmt router-list-num 2 weight 45
```

Related Commands

show wccp content-engines

show wccp routers

show wccp status

wccp version 2

whoami

To display the username of the current user, use the **whoami** EXEC command.

whoami

Syntax Description

This command has no arguments or keywords.

Defaults

No default behavior or values

Command Modes

EXEC

Usage Guidelines

Use this command to display the username of the current user.

Examples

```
ContentEngine# whoami
admin
```

Related Commands

pwd

wmt

To configure Windows Media Technologies (WMT), use the **wmt** command in global configuration mode. Use the **no** form of this command to negate these actions.

wmt accept-license-agreement

wmt broadcast {**alias-name** *name* **source** *url*}

wmt cache {**enable** | **max-obj-size** *size*}

wmt disallowed-client-protocols [**HTTP** | **TCP** | **UDP**]

wmt enable

wmt evaluate

wmt incoming *number*

wmt l4-switch {**enable**}

wmt license-key *key*

wmt max-bandwidth *size*

wmt max-bitrate *bit_rate*

wmt max-current-sessions *number*

wmt multicast {**schedule-start** *name minute hour day month* | **station-configuration** *name dest_addr dest_port media_source* [**play-forever**]}

no wmt {**accept-license-agreement** | **broadcast** {**alias-name** *name* **source** *url*} | **cache** {**enable** | **max-obj-size** *size*} | **disallowed-client-protocols** [**HTTP** | **TCP** | **UDP**] | **enable** | **evaluate** | **incoming** *number* | **l4-switch** {**enable**} | **license-key** *key* | **max-bandwidth** *size* | **max-bitrate** *bit_rate* | **max-current-sessions** *number* | **multicast** {**schedule-start** *name minute hour day month* | **station-configuration** *name dest_addr dest_port media_source* [**play-forever**]}}

Syntax Description

accept-license-agreement	Acknowledges and accepts the end user license agreement (EULA). Although the **no** form of the command is available from the CLI, it simply prints an error message that the EULA acceptance cannot be revoked.
broadcast	Configures a live broadcast.
alias-name	Configures the broadcast alias name.
name	Broadcast alias name.
source	Specifies broadcast source URL.
url	Broadcast source URL.
cache	Configures WMT cache.
enable	Enables the WMT media cache.
max-obj-size	Sets the maximum size of the object to be cached.
size	Object size in megabytes (1–1000000).

disallowed-client-protocols	Specifies disallowed WMT client protocols.
HTTP	(Optional) Disallows HTTP (streaming over http://).
TCP	(Optional) Disallows TCP (mmst://).
UDP	(Optional) Disallows UDP (mmsu://).
enable	Enables the WMT server.
evaluate	Starts or continues a 30-day evaluation of WMT.
incoming	Configures incoming WMT requests.
number	Port number to listen for requests (1–65535).
l4-switch	Configures Layer 4 switch interoperability for WMT.
enable	Enables Layer 4 switch interoperability for WMT.
license-key	Sets the WMT license key. Although the **no** form of the command is available from the CLI, it simply prints an error message that the license key cannot be removed.
key	License key.
max-bandwidth	Sets the maximum aggregate bandwidth limitation.
size	Maximum bandwidth in kilobits per second (1–1000000).
max-bitrate	Sets the maximum stream bit rate that can be served to a client.
bit_rate	Maximim bit rate per stream in kilobits per second (0–4294967295). The default is 0 (no limit).
max-concurrent-sessions	Configures the maximum number of unicast clients that can be served concurrently.
number	Limit for incoming unicast requests; this limit is subject to physical resources on the platform (1–2500).
multicast	Configures multicasting and scheduling.
schedule-start	Configures an automatic start schedule.
name	Multicast station name.
minute	Start time minute (0–59).
hour	Start time hour (0–23).
day	Start time day (1–31).
month	Start time month (1–12).
station-configuration	Configures multicast stations.
name	Multicast station name.
dest_addr	Multicast station destination IP address.
dest_port	Multicast station destination port (1–65535).
media_source	Multicast station media source.
play-forever	(Optional) Configures the stream to loop and restart. The default plays the stream once and stops.

Defaults

accept-license-agreement: not accepted

wmt enable: not enabled

max-bitrate: 0 (no bit rate limit)

max-object-size: 1GB

multicast station-configuration: play stream once and then stop.

Command Modes

Global configuration

Usage Guidelines

Based on the capabilities and the limitations of your network, a Content Engine can receive and deliver WMT streaming content through IP multicast. This multicast feature enables you to distribute streaming media efficiently by allowing different devices on the IP multicast to receive a single stream of media content simultaneously. This can save significant network bandwidth consumption because a single stream is sent to many devices, rather than sending a single stream to a single device one at a time whenever this streaming media is requested.

To distribute streaming media, set up a multicast address in the Content Engine to which different devices, each configured to receive content from the same channel, can subscribe. The delivering device sends content to this multicast address, from which it becomes available to all subscribed receiving devices.

Use the **wmt multicast** {**schedule-start** *name minute hour day month* | **station-configuration** *name dest_addr dest_port media_source* [**play-forever**]} command to enable WMT multicasting on the Content Engine. The **schedule-start** *name minute hour day month* option creates a scheduling option to allow the Content Engine to start a multicast at a specified time. This option only works if you have preconfigured a multicast station.

Note You must enable WMT on the Content Engine before you can use the **wmt multicast** command.

When **wmt enable** is invoked while E-CDN is enabled, as shown in the following example, the following message appears.

```
ContentEngine(config)# wmt enable

WMT server can only use up to 100Kbps of bandwidth initially. Please go to ECDN bandwidth
page in the CDM GUI to adjust the max bandwidth for WMT server.

ContentEngine# show wmt
WMT version: ce590-001.000

WMT enabled
WMT end user license agreement accepted
WMT license key installed
```

The **station-configuration** *name dest_addr dest_port media_source* option specifies a multicast station name, IP address, port number, and media source for the multicast station created. One station needs a multicast IP address. You must enter a valid class D IP multicast address in the range 224.0.0.0 through 239.255.255.255, except for the reserved IP ranges based on RFC 1700 and related documents as follows:

- 224.0.0.0 through 224.0.6.255
- 224.0.13.0 through 224.0.13.255
- 224.1.0.0 through 224.2.255.255
- 232.0.0.0 through 232.255.255.255

Note You must choose a multicast IP address that does not conflict internally within the same multicast-enabled network configuration. This multicast IP address is not related to the IP address of the Content Engine.

The allowable multicast port range defined by the *dest_port* option is 1 through 65535. However, the multicast-enabled network may impose certain restrictions on your choice of port number. Normally, port numbers below 1024 should be avoided, but the Content Engine does not enforce any restrictions.

The *media_source* option determines the source of the multicast. The source can be any valid WMT URL. For instance, if you can play the URL on your Windows Media player, then you can define this URL as the source of your multicast.

The Content Engine can support the following multicast scenarios:

- Multicast in and Unicast out
- Multicast in and Multicast out
- Unicast in and Multicast out

Unicast in and Multicast out

The unicast input can be from a video on demand (VOD) publishing point, a live unicast publishing point, an encoder, or a streaming media source from a local disk. The asf header obtained from the unicast input and a configured multicast-out IP address are used to create the multicast description .nsc file. Clients use this easily accessible file to subscribe and request the multicast.

Enabling WMT Multicasting in the Unicast in and Multicast-out Scenario

To enable WMT multicasting in this scenario with CLI commands, follow these steps:

Step 1 Enable WMT multicasting and configure a multicast station on the Content Engine in global configuration mode with the **wmt multicast station-configuration** command.

```
ContentEngine(config)# wmt multicast station-configuration test1 233.33.33.33 6666
mms://sourceIPaddress/source.asf play-forever
```

In this example, a station named *test1* that acts as the multicast source file is configured. Its class D IP address is 233.33.33.33 and the multicast port is 3333. The **play-forever** option is used. This option automatically restarts the stream from the beginning once the end of the multicast has been reached.

Note This source file can be located on any WMT server, including a Windows server, or the Content Engine, assuming that the source device and the requesting clients are on the same subnet. In the case of the Content Engine, pre-positioned media files should be stored in the /local1/wmt_vod directory. In this scenario, the media source is represented by mms://CEIPaddress/wmt_vod/sourcefile.nsc.

Step 2 Start the multicast from the source file, and use the **wmt** command in EXEC mode.

```
ContentEngine# wmt multicast-station start test1
```

Step 3 Open your WMT player and choose **File > Open URL**, and enter the following:

```
http://CEIPaddress/test1.nsc
```

Step 4 Click **OK**.

The WMT player should connect to the MMS media source specified in Step 1.

Multicast in and Multicast out

In this multicasting scenario, another description file *.nsc is created that is accessible through multicast out to clients. The clients use this file to subscribe and request the multicast.

Note The initial delivery of the requested file is through unicast out. However, once a single client has access to this file, other clients can join the multicast group and receive the same content.

Multicast in and Unicast out

In this scenario a unicast-out publishing point is created to deliver the incoming live stream to requesting clients.

Enabling WMT Multicasting in the Multicast in and Unicast out Scenario

To enable WMT multicasting in this scenario with CLI commands, following these steps:

Step 1 Enable WMT multicasting and configure a broadcasting station on the Content Engine in global configuration mode with the **wmt broadcast** command:

```
ContentEngine(config)# wmt broadcast alias-name unicast-station source
http://172.16.30.31/station.nsc
```

In this step a unicast station with an alias name *unicast-station* is configured with a multicast source station.nsc file.

Step 2 Open your WMT player and choose **File > Open URL**. Enter the following URL:

```
mms://CEIPaddress/unicast-station
```

Click **OK**. The WMT player should connect to the MMS media source specified in Step 1.

Examples

This example shows the commands required to enable the WMT server in ACNS software. All three commands are required.

```
ContentEngine(config)# wmt license-key WMT_LICENSE_KEY_590

ContentEngine(config)# wmt accept-license-agreement

ContentEngine(config)# wmt enable
```

Related Commands

show wmt

show wmt license-agreement

show running-config

show tech-support

show statistics wmt

show wccp-services

show wccp-flows

clear statistics wmt proxy

clear statistics wmt server

clear wmt proxy cache

wmt (EXEC mode)

wmt

To start or stop Windows Media Technologies (WMT) multicast stations, use the **wmt** command in EXEC mode.

wmt {**multicast-station** {**start** *name* | **stop** *name*} | **test-command**}

Syntax Description

multicast-station	Sets the WMT multicast stations to start or stop.
start	Starts a WMT multicast station.
name	Name of the WMT multicast station to be started.
stop	Stops a WMT multicast station.
name	Name of the WMT multicast station to be stopped.
test-command	Demonstrates WMT functionality.

Defaults

No default behavior or values

Command Modes

EXEC

Usage Guidelines

Use this command to start or stop particular WMT multicast stations or to demonstrate WMT functionality using the **test-command** option.

Examples

The three following examples demonstrate the **start**, **stop** and **test-command** options on the multicast station named acme.

```
ContentEngine# wmt multicast-station start acme

ContentEngine# wmt multicast-station stop acme

ContentEngine# wmt test-command
```

Related Commands

multicast-client

show multicast-client

show ntp

show statistics wmt

wmt multicast

write

To write running configurations to memory or to a terminal session, use the **write** EXEC command.

write [**erase** | **memory** | **terminal**]

Syntax Description

erase	(Optional) Erases startup configuration from NVRAM.
memory	(Optional) Writes the configuration to NVRAM. This is the default.
terminal	(Optional) Writes the configuration to a terminal session.

Defaults

The configuration is written to NVRAM by default.

Command Modes

EXEC

Usage Guidelines

Use this command to either save running configurations to NVRAM or erase memory configurations. Following a **write erase** command, no configuration is held in memory, and a prompt for configuration specifics occurs after you reboot the Content Engine.

Use the **write terminal** command to display the current running configuration in the terminal session window. The equivalent command is **show running-config**.

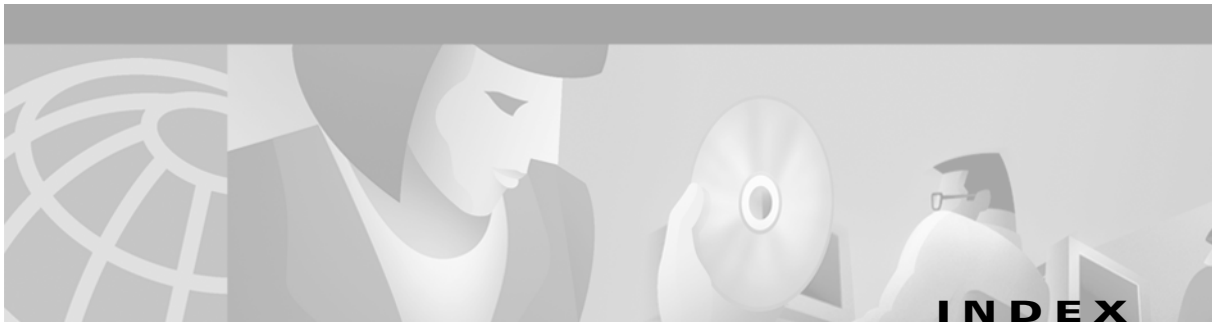
Examples

```
ContentEngine# write
```

Related Commands

copy running-config startup-config

show running-config



Symbols

A

B

C

D

E

F

G

H

I

K

L

M

N

O

P

R

S

T

U

V

W