# less

To display a file using the LESS application, use the **less** EXEC command.

**less** *file_name*

## Syntax Description

| | |
|---|---|
| *file_name* | Name of the file to be displayed. |

## Defaults

No default behavior or values

## Command Modes

EXEC

## Usage Guidelines

LESS is an application that displays text files a page at a time. You can use LESS to view the contents of a file, but not edit it. LESS offers some additional features when compared to conventional text file viewer applications such as type. These features are as follows:

- Backward movement—LESS allows you to move backward in the displayed text. Press **k**, **Ctrl-k**, **y**, or **Ctrl-y** to move backward. See the summary of LESS commands for more details; to view the summary, press **h** or **H** while displaying a file in LESS.
- Searching and highlighting—LESS allows you to search for text in the file that you are viewing. You can search forward and backward. LESS highlights the text that matches your search to make it easy to see where the match is.
- Multiple file support—LESS allows you to switch between different files, remembering your position in each file. You can also do a search that spans all the files you are working with.

## Examples

The following example displays the syslog.txt file using the LESS application:

```
ContentEngine# less syslog.txt
Jul 29 14:32:41 NO-HOSTNAME syslog_bootup_msgs: %CE-SYS-5-900001: <6>sym0:0: FAS
T-40 WIDE SCSI 80.0 MB/s ST (25.0 ns, offset 31)
Jul 29 14:32:41 NO-HOSTNAME syslog_bootup_msgs: %CE-SYS-5-900001: <4>SCSI device
 sda: 71687340 512-byte hdwr sectors (36704 MB)
Jul 29 14:32:41 NO-HOSTNAME syslog_bootup_msgs: %CE-SYS-5-900001: <6>Partition c
heck:
Jul 29 14:32:41 NO-HOSTNAME syslog_bootup_msgs: %CE-SYS-5-900001: <6> sda: sda1
sda2 sda3 sda4 < sda5 sda6 sda7 sda8 >
Jul 29 14:32:41 NO-HOSTNAME syslog_bootup_msgs: %CE-SYS-5-900001: <6>Fusion MPT
base driver 2.02.01.06
Jul 29 14:32:41 NO-HOSTNAME syslog_bootup_msgs: %CE-SYS-5-900001: <6>Copyright (
c) 1999-2002 LSI Logic Corporation
Jul 29 14:32:41 NO-HOSTNAME syslog_bootup_msgs: %CE-SYS-5-900001: <6>mptbase: 0
MPT adapters found, 0 installed.
Jul 29 14:32:41 NO-HOSTNAME syslog_bootup_msgs: %CE-SYS-5-900001: <6>Fusion MPT
SCSI Host driver 2.02.01.06
Jul 29 14:32:41 NO-HOSTNAME syslog_bootup_msgs: %CE-SYS-5-900001: <6>md: raid1 p
ersonality registered as nr 3
Jul 29 14:32:41 NO-HOSTNAME syslog_bootup_msgs: %CE-SYS-5-900001: <6>md: md driv
er 0.90.0 MAX_MD_DEVS=256, MD_SB_DISKS=27
Jul 29 14:32:41 NO-HOSTNAME syslog_bootup_msgs: %CE-SYS-5-900001: <6>md: Autodet
```

```
ecting RAID arrays.
Jul 29 14:32:41 NO-HOSTNAME syslog_bootup_msgs: %CE-SYS-5-900001: <6>md: autorun
 ...
/local/local1/syslog.txt
```

# line

To specify terminal line settings, use the **line** global configuration command. To disable terminal line settings, use the **no** form of this command.

**line console carrier-detect**

**no line console carrier-detect**

## Syntax Description

| | |
|---|---|
| **console** | Configures the console terminal line settings. |
| **carrier-detect** | Sets the device to check the carrier detect signal before writing to the console. |

## Defaults

This feature is disabled by default

## Command Modes

global configuration

## Usage Guidelines

You should enable carrier detection if you connect the Content Engine, Content Router, or Content Distribution Manager to a modem for receiving calls. If you are using a null-modem cable with no carrier detect pin, the device might appear unresponsive on the console until the carrier detect signal is asserted. To recover from a misconfiguration, you should reboot the device and set the 0x2000 bootflag to ignore the Carrier Detect (CD) setting.

## Examples

The following example shows how to specify terminal line settings:

```
ContentEngine(config)# line console carrier-detect
```

# 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

The following example shows how to view a long list of directory names:

```
CONTENTENGINE# lls
      size          time of last change             name
--------------  ------------------------          -----------
          4096  Mon Jan 10 14:02:26 2005  <DIR>    WebsenseEnterprise
          4096  Mon Jan 10 14:02:26 2005  <DIR>    Websense_config_backup
         10203  Mon Feb 28 04:24:53 2005           WsInstallLog
          4096  Wed Feb  9 00:59:48 2005  <DIR>    core_dir
          4096  Mon Jan 10 13:49:27 2005  <DIR>    crash
           382  Tue Mar  1 03:32:13 2005           crka.log
          1604  Tue Feb 22 03:55:04 2005           dbupgrade.log
          4096  Mon Jan 10 14:02:31 2005  <DIR>    downgrade
          4096  Mon Feb 28 04:17:32 2005  <DIR>    errorlog
         53248  Tue Mar  1 03:01:53 2005  <DIR>    logs
         16384  Mon Jan 10 13:49:26 2005  <DIR>    lost+found
           438  Tue Jan 11 05:37:57 2005           new_file.xml
          8192  Tue Mar  1 00:00:00 2005  <DIR>    preload_dir
          4096  Tue Mar  1 03:26:00 2005  <DIR>    sa
         40960  Tue Mar  1 03:32:15 2005  <DIR>    service_logs
          4096  Tue Feb 22 03:51:25 2005  <DIR>    smartfilter
        384802  Mon Feb 28 03:46:00 2005           syslog.txt
         16296  Mon Feb 21 04:42:12 2005           test
          4096  Mon Jan 10 14:02:24 2005  <DIR>    var
          4096  Sat Feb 12 07:15:23 2005  <DIR>    wmt_vod
CONTENTENGINE#
```

## 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*} | **disk** {**enable** | **filename** *filename* | **priority** *loglevel* | **recycle** *size*} | **facility** *facility* | **host** {*hostname* | *ip-address*} [**port** *port_num* | **priority** *loglevel* | **rate-limit** *message_rate*]}

**no logging** {**console** {**enable** | **priority**} | **disk** [**enable** | **filename** | **priority** | **recycle**] | **facility** | **host** {*hostname* | *ip-address*} [**port** *port_num* | **priority** *loglevel* | **rate-limit** *message_rate*]}

## Syntax Description

| | |
|---|---|
| **console** | Sets system logging to a console. |
| **enable** | Enables system logging to a console. |
| **priority** | Sets which priority level messages to send to a syslog file. |
| *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. |
| **disk** | Sets system logging to a disk file. |
| **enable** | Enables system logging to a disk file. |
| **filename** | Sets the name of the syslog file. |
| *filename* | Specifies the name of the syslog file. |
| **recycle** | Overwrites the *syslog.txt* when it surpasses the recycle size. |
| *size* | Size of the syslog file in bytes (1000000–50000000). |
| **facility** | Sets the 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 the system logging to a remote host. |
| *hostname* | Hostname of the remote syslog host. Specify up to four remote syslog hosts.<br>**Note** To specify more than one syslog host, use multiple command lines; specify one host per command. |
| *ip-address* | IP address of the remote syslog host. Specify up to four remote syslog hosts.<br>**Note** To specify more than one syslog host, use multiple command lines; specify one host per command. |
| **port** | (Optional) Specifies the port to be used when logging to a host. |
| *port_num* | Port to be used when logging to a host. The default port is 514. |
| **priority** | (Optional) Sets the priority level for messages when logging messages to a host. The default priority is warning. |
| *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. |
| **rate-limit** | (Optional) Sets the rate limit (in messages per second) for sending messages to a host. |
| *message_rate* | Rate limit (in messages per second) for sending messages to the host. (0–10000). Setting the rate limit to 0 disables rate limiting. |

## Defaults

Logging: on

Priority of message for console: warning

Priority of message for log file: debug

Priority of message for a host: warning

Log file: /local1/syslog.txt

Log file recycle size: 10,000,000 bytes

## Command Modes

global configuration

## Usage Guidelines

Use the **logging** command to set specific parameters of the system log file. System logging is always enabled internally on the Content Engine. The system log file is located on the sysfs partition as /local1/syslog.txt. This file contains the output from many of the ACNS components running on the Content Engine, such as authentication entries, privilege levels, administrative details, and diagnostic output during the boot process.

To view information about events that have occurred in all devices in your ACNS network, you can use the system message log feature. When a problem occurs in the ACNS network, use the system message logs to diagnose and correct such problems.

The syslog.txt file on the Content Distribution Manager contains information about events that have occurred on the Content Distribution Manager and not on the registered nodes. The messages written to the syslog.txt file depend on specific parameters of the system log file that you have set using the **logging** global configuration command. For example, a critical error message logged on a registered node does not appear in the syslog.txt file on the Content Distribution Manager because the problem never occurred on the Content Distribution Manager but occurred only on the registered node. However, such an error message will be displayed in the syslog.txt file on the registered node.

In the ACNS 5.1.x software and earlier releases, a disk failure syslog message is generated every time that a failed sector is accessed. In the ACNS 5.2 software release, support for filtering multiple syslog messages for a single failed sector on an IDE disk was added. In the ACNS 5.3 software release, support for filtering multiple syslog messages for a single failed section for SCSI disks and SATA disks was added.

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 **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.

### Configuring System Logging to Remote Syslog Hosts

The ACNS 5.1 software supported logging to only a single remote syslog host, and the following two commands were used to configure a single remote syslog host for a Content Engine:

```
ContentEngine(config)# logging host hostname
ContentEngine(config)# logging priority priority
```

In the ACNS 5.2 software and later releases, you can configure a Content Engine to send varying levels of messages to up to four remote syslog hosts. To accommodate this change, the ACNS 5.1.x software **logging host priority** *priority* global configuration command (shown above) is deprecated, and the **logging host hostname** global configuration command is extended as follows:

```
ContentEngine(config)# [no] logging host hostname [priority priority-code | port port
|rate-limit limit]
```

where the following is true:

- *hostname* is the hostname or IP address of the remote syslog host. Specify up to four remote syslog hosts. To specify more than one syslog host, use multiple command lines; specify one host per command. In the ACNS 5.1.x software and earlier releases, you could only configure a Content Engine to send messages to a single remote syslog host.
- *priority-code* is the severity level of the message that should be sent to the specified remote syslog host. The default priority-code is “warning” (level 4). Each syslog host can receive a different level of event messages.

**Note** You can achieve syslog host redundancy by configuring multiple syslog hosts on the Content Engine and assigning the same priority code to each configured syslog host (for example, assigning a priority code of "critical" level 2 to syslog host 1, syslog host 2, and syslog host 3).

- *port* is the destination port of the remote syslog host to which the Content Engine is to send the messages. The default port is port 514. In releases prior to the ACNS 5.2 software, you could not change the default port. Syslog messages were only sent to port 514 on the specified syslog host.
- *rate-limit* specifies the number of messages that are allowed to be sent to the remote syslog host per second. To limit bandwidth and other resource consumption, messages to the remote syslog host can be rate limited. If this limit is exceeded, messages to the specified remote syslog host are dropped. There is no default rate limit, and by default all syslog messages are sent to all of the configured syslog hosts. If the rate limit is exceeded, a message of the day (MOTD) will be printed for any CLI EXEC shell login.

#### Mapping syslog Priority Levels to RealProxy Error Codes

The RealProxy system generates error messages and writes them to the RealProxy log file. These error messages are captured by the caching application and passed to the system log file. A one-to-one mapping exists between the RealProxy error codes and the syslog priority levels.

**Note** For information on mapping the RealProxy error codes with syslog priority levels, see Chapter 21 of the *Cisco ACNS Software Configuration Guide for Locally Managed Deployments*.

## Examples

The following example shows that the Content Engine is configured to send messages that have a priority code of "error" (level 3) to the console:

```
ContentEngine(config)# logging console priority warnings
```

The following example shows that the Content Engine is configured to disable sending of messages that have a priority code of "error" (level 3) to the console:

```
ContentEngine(config)# no logging console warnings
```

The following example shows that the Content Engine is configured to send messages that have a priority code of "error" (level 3) to the remote syslog host that has an IP address of 172.31.2.160:

```
ContentEngine(config)# logging host 172.31.2.160 priority error
```

## Related Commands

**clear logging**
**debug**
**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

The following example displays a list of files within the current working directory:

```
ContentEngine# ls
/local1
```

The following example displays a list of files within the /local1 directory:

```
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. To undo the media file system space percentage allocation for the WMT cache and RealProxy cache, use the **no** form of this command.

**mediafs-division wmt-cache-space** *percent_space* **real-cache-space** *percent_space*

**no 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

The following example shows how to divide the media file system space between the WMT cache and the RealProxy cache:

```
ContentEngine(config)# mediafs-division wmt-cache-space 34 real-cache-space 66
```

## Related Commands

**show mediafs volumes**

# 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**

The following example shows how to create a new directory under local1:

```
ContentEngine# mkdir /local1/mydir
```

**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 that 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**

The following example shows how to create a new file:

```
ContentEngine# mkfile traceinfo
```

**Related Commands**

**lls**
**ls**
**mkdir**

# mode

To change the Fibre Channel interface operating mode, use the **mode** interface configuration command. To undo the change, use the **no** form of this command.

**mode** {**autosense** | **direct-attached** | **switched**}

**no mode** {**autosense** | **direct-attached** | **switched**}

## Syntax Description

| | |
|---|---|
| **autosense** | Sets the operation mode of the Content Engine to autosense, which allows the mode to be automatically set based on whether it is connected to a switch or a Storage Array. |
| **direct-attached** | Sets the operation mode when the Content Engine is directly connected to a Storage Array. |
| **switched** | Sets the operation mode when the Content Engine is connected to a switch. |

## Defaults

The default mode is autosense.

## Command Modes

interface configuration

## Usage Guidelines

To support Content Engines in a storage area network (SAN) environment, the Fibre Channel interface interconnects storage devices and Content Engines in a SAN. In a SAN, the storage device does not need to be directly attached to the server, and data transfer occurs at the block level rather than at the file level over a high-throughput, high-availability network. SCSI is still the storage access technology in a SAN, and the SCSI 3 standards have been designed to accommodate SANs.

In a Fibre Channel-based SAN, the SCSI commands and data are encapsulated in a Fibre Channel frame and then transported across the Fibre Channel network to the desired device. At the end device, the encapsulation is removed and the data is retrieved. Fibre Channel is capable of operating at approximately 2 Gbps.

The ACNS 5.x software supports a Fibre Channel SAN that uses one of two topologies: direct attached storage and fabric-based Fibre Channel.

In the direct attached storage (DAS) topology, the Content Engine is directly attached to the storage device over Fibre Channel media. Both the Content Engine and the storage device have Fibre Channel interfaces and an available bandwidth of 2 Gbps for data transfer. Use the **mode direct-attached** command when the Content Engine is directly connected to a Storage Array. The disadvantage of this method is that although high-bandwidth connectivity exists between the Content Engine and the storage device, the storage device is tied to a single Content Engine or server.

In a switched fabric Fibre Channel SAN, each Fibre Channel device is connected to the switch with a dedicated connection to the network. Use the **mode switched** command when the Content Engine is connected to a switch. Most switches are nonblocking; any device can potentially have full-bandwidth connectivity to any other device connected to the switch.

## Examples

The following example sets the mode of operation of the Fibre Channel slot 0/port 0 interface on a Content Engine that is directly connected to a Storage Array:

```
interface ContentEngine(config)# interface FibreChannel 0/0
ContentEngine(config-if)# mode direct-attached
```

The following example sets the mode of operation of the Fibre Channel slot 0/port 0 interface on a Content Engine that is connected to a switch:

```
ContentEngine(config)# interface FibreChannel 0/0
ContentEngine(config-if)# mode switched
```

## Related Commands

**show interface**
**show running-config**
**show startup-config**

# mtu

To set the interface maximum transmission unit (MTU) packet size, use the **mtu** interface configuration command. Use the **no** form of this command to reset the MTU packet size.

**mtu** *mtusize*

**no mtu** *mtusize*

## Syntax Description

| | |
|---|---|
| *mtusize* | MTU packet size in bytes (68–1500). |

## Defaults

The default MTU packet size for an Ethernet interface is 1500 bytes.

## Command Modes

interface configuration

## Usage Guidelines

The MTU is the largest size of IP datagram that can be transferred using a specific data link connection. Use the **mtu** command to set the maximum packet size in bytes.

## Examples

The following example sets the MTU packet size as 1500 bytes:

```
ContentEngine(config-if)# mtu 1500
```

The following example resets the MTU packet size:

```
ContentEngine(config-if)# no mtu 1500
```

## Related Commands

**show interface**
**show running-config**
**show startup-config**

# multicast

To configure multicast client options, use the **multicast** global configuration command. To disable individual options, use the **no** form of this command.

**multicast** {**accept-license-agreement** | **back-version-compatibility acns-5-0** | **enable** | **evaluate** | **fixed-carousel enable** | **license-key** *key* | **max-concurrent-jobs** *number-jobs* [**minimal-target-rate** *bits*] | **priority-weight** *0-100* | **sender-delay** *delay*}

**no multicast** {**accept-license-agreement** | **back-version-compatibility acns-5-0** | **enable** | **evaluate** | **fixed-carousel enable** | **license-key** | **max-concurrent-jobs** | **priority-weight** | **sender-delay**}

## Syntax Description

| | |
|---|---|
| **accept-license-agreement** | Accepts the multicast client license agreement. |
| **back-version-compatibility** | Sets the multicast sender to be compatible with multicast receivers running earlier versions of the ACNS software. |
| **acns-5-0** | Specifies that the multicast sender must be compatible with multicast receivers running the ACNS 5.0 software. |
| **enable** | Enables the multicast client. |
| **evaluate** | Starts or continues the 60-day evaluation period of the multicast client. |
| **fixed-carousel enable** | Enables the use of fixed carousel sending. |
| **license-key** | Requires the license key for the multicast client. |
| *key* | Multicast client license key parameters. |
| **max-concurent-jobs** | Specifies the maximum number of jobs that can be scheduled concurrently for multicast distribution. |
| *number-jobs* | Maximum number of jobs (1–50). The default is 5. |
| **minimal-target-rate** | (Optional) Specifies the minimum bandwidth that must be allotted for each concurrent job. |
| *bits* | Minimum bandwidth for each job in bits per second (bps) (51200–10485760). |
| **priority-weight** | (Optional) Specifies the percentage of multicast bandwidth that is used for priority-based scheduling. |
| *0-100* | Bandwidth percentage to be used for priority-based scheduling. (0–100). The default is 50 percent. |
| **sender-delay** | Sets the multicast sender delay time. |
| *delay* | Delay time in seconds (480–7200). |

## Defaults

**evaluate**: 60 days

**sender-delay** *delay*: 960 seconds.

**max-concurrent-jobs** *number-jobs*: 5

**minimal-target-rate** *rate*: 102400 bps

**priority-weight**: 50%

**Command Modes** global configuration

**Usage Guidelines** In the ACNS network, content is replicated through a channel distribution architecture. Content in channels can be transmitted by unicast pull, or if multicasting is enabled, by multicast push. Multicasting allows efficient distribution of content to multiple Content Engines and is useful when many end users are interested in the same content. The ACNS software supports Pragmatic General Multicast (PGM)-based multicast replication using either satellite or multicast-enabled terrestrial infrastructures.

Multicast delivery enables the distribution of streaming media by allowing different receiving devices on the IP multicast to receive a single stream of media content from the Content Engine simultaneously. This method 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 every time that this stream is requested.

This multicast delivery feature is enabled by setting up a multicast address on the Content Engine to which different devices, configured to receive content from the same channel, can subscribe. The delivering device sends content to the multicast address set up at the Content Engine, from which it becomes available to all subscribed receiving devices.

To take advantage of multicasting, all devices, including Content Engines, routers, and clients, must be multicast enabled. For this reason, multicasting is mostly used in local networks where routers can be configured for multicasting. Multicast delivery over the Internet can only be accomplished when all the devices that participate in the multicast have been enabled for multicasting.

For multicast content replication, Content Engines are grouped into multicast clouds. A multicast cloud consists of one sender Content Engine, an optional backup sender Content Engine, and at least one receiver Content Engine in a hub and spoke topology. All the Content Engines in one multicast cloud share a unique advertisement address that allows them to communicate multicast session information. The multicast cloud is then associated with one or more multicast-enabled channels.

In pull-based unicast content distribution, a unicast receiver pulls file data out of the proper forwarder (or root Content Engine) when a client requests the content. In multicast content distribution, the sender Content Engine in a multicast cloud proactively pushes content into the cloud according to a preconfigured schedule.

The receiver Content Engines listen on the advertisement IP address for information on the content to be replicated from the sender and decide whether to accept an advertisement and whether to receive the content.

The content metadata (machine-readable information that describes the characteristics of the content) must be distributed to a receiver first before the content can be replicated. The content metadata helps to define what content to retrieve, how the content will be retrieved, how recently the content has been updated, how the content is to be pre-positioned (for example, the expiration time), and so forth. The metadata is always distributed using unicast. The content, however, can be replicated using either multicast or unicast. A multicast receiver rejects the multicast sender's advertisement of a file if the proper content metadata has not arrived. During the content distribution process, both the content and its associated information describing the content, called the metadata, are distributed. A multicast receiver does not accept any multicast content unless it has already received the associated metadata for that particular content.

Use the **multicast fixed-carousel enable** command to enable fixed-carousel sending. Carousel sending, which refers to the multicast retransmission of content, allows receivers who join a multicast group after a distribution has ended or who miss some content to receive the content without requiring a unicast transmission. By default, the Content Engine uses intelligent carousel sending, which means that the

retransmission is guided by feedback from the content receivers. Late-joining receivers or receivers that missed some content send a negative acknowledgement (NACK) to the sender for any files that were not received.

Fixed-carousel sending causes the content to be sent without depending on any receiver feedback. When this feature is enabled, the Content Engine continuously retransmits the content after waiting for the time specified by the **sender-delay** option. You can use the **fixed-carousel** option when sending the content to receivers using a release of the ACNS software earlier than Release 5.1, which do not send NACKs to the sender. This configuration is allowed only for the primary sender and is not supported for a Content Engine configured as a backup sender. Use the **no multicast fixed-carousel enable** command to disable fixed-carousel sending and to restore the default configuration.

Use the **multicast max-concurrent-jobs** command option to set the maximum number of objects that can be scheduled concurrently for multicast distribution. When networks are reliable or the size of files being multicast is small, we recommend that you set the maximum number of concurrent objects to 50. However, when networks are unreliable or the size of files being multicast is large, a smaller number of concurrent objects (for example, five) is recommended. The default maximum number of jobs is five. The **minimal-target-rate** option sets the minimum bandwidth that must be allotted per object in bits per second (bps). The **minimal-target-rate** option is optional; if left unconfigured, the default value of 102400 bps is used.

The **priority-weight** option allows you to change the percentage of multicast bandwidth that is used for priority-based scheduling. By default, 50 percent of the bandwidth is allocated for a priority-based queue and 50 percent is allocated for a time-based, first in, first out (FIFO) queue. The time-based queue allows the system to process lower-priority traffic in a timely way even when it frequently receives large, high-priority requests.

#### Multicast Sender Delay

The multicast sender delay interval is the amount of time before each multicast transmission begins. A period of delay before the actual multicast transmission begins is required to allow the content metadata time to propagate to the receiver Content Engine. The metadata contains the content file and configuration information that is necessary for the successful transmission of content files. The sender delay parameter is used to configure an extra delay before a multicast transmission can begin.

When configuring the sender delay interval, you must take into account that the content metadata must first propagate to the receiver before the multicast transmission can begin. During a multicast session, a receiver Content Engine sends out periodic requests for files that it has not received. The sender retransmits files only as requested by the receiver Content Engine. A multicast receiver will reject a multicast sender’s advertisement of a file if the associated content metadata has not arrived. The sender delay option allows you to configure enough time for the metadata to propagate to the receiver, and avoid having the receiver reject the multicast sender's advertisement of a file.

To configure the sender delay interval, use the multicast sender-delay global configuration command on a sender Content Engine. The **sender-delay** option controls the length of time that the multicast sender must wait for its associated metadata to propagate to the multicast receivers. The default delay time value is 960 seconds, the minimum is 480 seconds, and the maximum is 7200 seconds. You can configure the duration of the delay based on your expectation or best guess of the amount of time required for the metadata to be propagated. The system takes this user-configured sender delay value and delays the multicast transmission for a period defined by the sender delay.

**Note** The sender delay interval cannot be configured using the Content Distribution Manager GUI. You must configure the sender delay interval using the CLI of the sender Content Engine.

### Multicast License Key

Before you can create a multicast cloud, you must have a multicast distribution license key (purchased from Cisco Systems) and Content Engines that are enabled for multicasting. These multicast-enabled Content Engines can then be assigned as sender and receiver Content Engines when you configure the multicast cloud.

Content Engines for multicasting must be assigned to the multicast cloud, which in turn is assigned to multicast-enabled channels. Also, you need to assign individual Content Engine senders and receivers of the cloud to the particular multicast-enabled channel. You must do this additional step even though the multicast cloud is associated with the channel.

**Note** You must assign the multicast cloud to a channel first, and then assign the individual Content Engines to the channel.

Use the **no multicast license-key** command to uninstall a license key if it is no longer needed on the device because the multicast licensed product feature is not needed. After you uninstall a license key on one device, you can use the key on another device if that device supports the multicast license key.

**Note** You must disable the multicast feature using the **no multicast enable** command before you uninstall the multicast license key.

## Examples

The following example shows how to accept the multicast distribution license for a Content Engine:

```
ContentEngine# configure
ContentEngine(config)# multicast accept-license-agreement
```

The following example shows how to enter the multicast distribution license key for a Content Engine:

```
ContentEngine(config)# multicast license-key 123456789
```

The following example shows how to enable multicasting on a Content Engine:

```
ContentEngine(config)# multicast enable
```

## Related Commands

**show multicast**

# multicast connectivity-test

To generate multicast packets and test connectivity through multicast routers, use the **multicast connectivity-test** EXEC command.

**multicast connectivity-test ce** *ip_addresses* **multicast-address** *ip_address* [**duration** *duration* | **packet-count** *packets*] [**FEC-size** *fec_bytes* **max-transfer-rate** *bandwidth* **message-length** *message_bytes* **output-style** {**error** | **detail**} **port** *port_number* **time-to-live** *ttl*]

**multicast connectivity-test cloud-id** *id* [**duration** *duration* | **packet-count** *packets* **message-length** *message_bytes* **multicast-address** *ip_address* **output-style** {**detail** | **error**} **port** *port_number*]

**multicast connectivity-test cloud-name** *name* [**duration** *duration* | **packet-count** *packets* **message-length** *message_bytes* **multicast-address** *ip_address* **output-style** {**error** | **detail**} **port** *port_number*]

**multicast connectivity-test receive** *ip_address*

**multicast connectivity-test send** *ip_address ttl_time*

## Syntax Description

| | |
|---|---|
| **ce** | Sends Pragmatic General Multicast (PGM) packets to the receiver Content Engines. |
| *ip_addresses* | IP addresses of the receiver Content Engines. You can specify a maximum of 20 receiver Content Engines. |
| **multicast-address** | Configures the multicast IP address to be used for sending PGM packets. When you use the **cloud-id** or **cloud-name** option, you can choose not to specify the multicast address. When the multicast address is not specified, the advertising IP address specified in the multicast cloud configuration is used as the multicast address. |
| *ip_address* | Multicast IP address to be used for sending PGM packets. |
| **duration** | (Optional) Configures the length of time over which PGM packets are to be sent to receiver Content Engines. This is the default option for **multicast-address**. |
| *duration* | Number of seconds that PGM packets are sent to receiver Content Engines (30–3600). The default is 180 seconds. |
| **packet-count** | (Optional) Configures the number of PGM packets to be sent to receiver Content Engines. |
| *packets* | Number of PGM packets to be sent to each receiver Content Engine (50–4096). The default is 100. |
| **FEC-size** | (Optional) Configures the appending of forward error correction (FEC) redundancy bytes. For more information, see the "Multicast Forward Error Correction" section on page 2-321. |
| *fec_bytes* | Number of FEC redundancy bytes to be appended (8–128). The values must be powers of 2 (for example, 8, 16, 32, or 64). The default is 8. |
| **max-transfer-rate** | (Optional) Configures the maximum bandwidth that can be used for this multicast transmission. |
| *bandwidth* | Maximum bandwidth value (56–1000000). The default is 128 kbps. |

| | |
|---|---|
| **message-length** | (Optional) Configures the number of bytes per PGM packet. |
| *message_bytes* | Number of bytes allowed for PGM packet (256–4096). The default is 1024. |
| **output-style** | (Optional) Configures how the output should be displayed. |
| **error** | Sets the device to display only the errors encountered for each receiver Content Engine. |
| **detail** | Sets the device to display detailed reports for each receiver Content Engine. This is the default option for **output-style**. |
| **port** | (Optional) Sets the port on the receiver Content Engines to which the PGM packets are to be sent. |
| *port_number* | Port on the receiver Content Engines to which the PGM multicast data is to be sent (1025–65535). The default port is 7000. |
| **time-to-live** | (Optional) Configures the maximum number of hops permitted for PGM packets before they expire on the network. For each hop, the original specified TTL is decremented by 1. When the TTL reaches 0, PGM packets expire and are no longer forwarded through the network. |
| *ttl* | Maximum number of hops allowed for PGM packets before they are discarded. (1–255). The default is 255. |
| **cloud-id** | Specifies the multicast cloud identifier. |
| *id* | Identifier for the multicast cloud (0–4294967295). |
| **cloud-name** | Specifies the name of the multicast cloud. |
| *name* | Name of the multicast cloud. |
| **receive** | Receives PGM packets from the specified multicast address. |
| *ip_address* | Multicast IP address of one or more receivers of PGM packets. |
| **send** | Sends PGM packets to the specified multicast address. |
| *ip_address* | Multicast IP address to be used for sending PGM packets. |
| *ttl_time* | Time To Live for multicast packets (1–255). |

## Defaults

**duration** *duration*: 180 seconds

**packet-count** *packets*: 100

**FEC-size** *fec_bytes*: 8

**max-transfer-rate** *bandwidth*: 128 kbps

**message-length** *message_bytes*: 1024

**port** *port_number*: 7000

**time-to-live** *ttl*: 255

**message-length** *message_bytes*: 1024

## Command Modes

EXEC

## Usage Guidelines

You can use the **multicast connectivity-test** command to test multicast connectivity within the ACNS network. The **multicast connectivity-test** command options run pgmrategen (the PGM packet generation application) and pgmratemon (the PGM packet receiver application) in the background to test multicast connectivity. These applications use the PGM protocol, which allows a receiver to report lost data and to request retransmission from the sender. With PGM, the sender multicasts sequenced data packets, and the receivers reply with unicast negative acknowledgments (NACKs) when data packets are missing from the expected sequence. Network elements forward the NACKs to the multicast sender and confirm each hop by multicasting a NACK confirmation on the interface on which the NACK was received.

### Multicast Forward Error Correction

Forward error correction (FEC) is a type of data encoding that protects transmissions against errors, without requiring retransmission. The FEC number denotes the number of packets that will be encoded into one FEC transmission group. A higher FEC number means that the transmission group size is larger. The multicast may be more error-resistant, but there will also be more computational overhead on the multicast sender and receivers. No bandwidth overhead is related to FEC.

In the ACNS 5.x software, the FEC default value is 8. If the multicast sender device is a high-end Content Engine model, such as a CE-7325, you can set this number higher to improve multicast reliability when your network connectivity has a high uniform loss rate. However, we do not recommend that you set this number beyond 64, because it may place too much of a load on all the receiver Content Engines.

You can also set proactive FEC using the PGM configuration file (a text file accessible from the Content Engine CLI in the /local1/multicast-expert-config/ directory). Proactive FEC is the number of extra packets that the multicast sender proactively sends out for every FEC number of data packets. The proactive FEC default value is 0. You can set the proactive FEC number higher for better multicast reliability. For example, you can set 2 proactive packets for every 16 FEC packets at the expense of 12.5 percent traffic overhead (2 divided by 16).

Proactive FEC is an additional reliability measure above and beyond that of normal FEC. Although normal FEC does not incur bandwidth overhead, proactive FEC does use bandwidth overhead. Proactive FEC primarily protects the multicast from uniform losses. For example, if the network has a uniform loss rate of 15 percent, then a proactive FEC of 2 extra packets for every 16 FEC packets (a 12.5 percent bandwidth overhead) cuts the effective loss rate down to 2.5 percent. Most network losses are not completely uniform. Still, during bursts, proactive FEC undercuts the effective burst loss rate. For example, if the burst loss rate is 20 percent while the average loss rate is 2 percent, with proactive FEC at 12.5 percent, the receiver Content Engines experience a burst loss rate of 7.5 percent and an average loss rate near 0 percent.

### Testing Multicast Connectivity in ACNS Networks

The **multicast connectivity-test** command options allow you to test multicast connectivity in the ACNS networks. To test multicast connectivity in a small group of Content Engines, use the **ce** option. You can specify a maximum of 20 multicast receiver Content Engines using this option. Use the **cloud-id** or **cloud-name** options to test multicast connectivity to receiver Content Engines grouped into multicast clouds in the ACNS network. There is no limit to the number of Content Engines you can have in a multicast cloud.

The **ce**, **cloud-id**, or **cloud-name** options initiate the following sequence of events:

1. When you specify the IP addresses of the receiver Content Engines, multicast cloud ID, or multicast cloud name, you initiate the multicast connectivity test.

   **Note** When you use the **ce** option, you must specify the multicast address. You do not need to specify the multicast address with the **cloud-id** or **cloud-name** options because the advertising IP address specified in the multicast cloud configuration is used as the multicast address.

2. The multicast sender initiates RPC calls to the receiver Content Engines, which prompt them to listen for the PGM multicast data on the default port or the port specified in the **multicast connectivity-test** command.
3. The Content Engine displays the following:
   - A list of receiver Content Engines that failed to respond to the RPC calls
   - Warning messages if multicast is not enabled on any receiver Content Engine
4. The multicast sender starts sending PGM packets to the specified multicast address. The receiver Content Engine keeps updating the session statistics for each packet received.

   If you interrupt the test by pressing **Ctrl-C**, the multicast sender sends a notification to all receiver Content Engines to stop listening and displays the information obtained so far.
5. After the multicast session is completed or the transmission has timed out, the receiver Content Engine sends the statistics to the multicast sender using an RPC call.

   **Note** If the multicast receiver Content Engine times out, it sends a "no packet received" error message to the multicast sender if it has not received any PGM packets for 60 seconds. Similarly, the multicast sender waits for 60 seconds for a response from the receiver Content Engine before timing out.

6. The multicast sender displays the statistics obtained for the session.
7. The multicast sender repeats the test with the receiver Content Engines and generates a consolidated report.

#### Using the ce Option

Use the **multicast connectivity-test ce ipaddress multicast-address** command to test multicast connectivity to the Content Engines with the IP addresses 10.77.155.171, 10.77.155.175, 10.77.155.179, using the multicast IP address 239.10.1.11.

The ACNS software uses the default values for all the optional parameters.

#### Using the cloud-id or cloud-name Option

You can use the **cloud-id** or **cloud-name** options to test multicast connectivity in large networks where a number of Content Engines are grouped under multicast clouds.

**Note** You cannot specify values for the optional parameters **FEC-size** and **max-transfer-rate** when the **cloud-id** or **cloud-name** options are used. The values for these two parameters are taken from the multicast cloud configuration.

Use the **multicast connectivity-test cloud-name** command to test the multicast connectivity to the receiver Content Engines in the multicast cloud mcloud1.

The multicast address is not specified in this example. The ACNS software uses the advertisement IP address specified in the multicast cloud configuration as the multicast address. Because no values were specified for the options, the ACNS software uses the default values for all the optional parameters.

#### multicast connectivity-test send Command

To test the multicast connectivity to the receiver Content Engines listening to a specific multicast address, use the **multicast connectivity-test send** command. This command runs the pgmrategen application, which continuously sends PGM packets to the specified multicast IP address. After you enter this command, the system displays the percentage of packets that have been multicast and stops sending packets when the packets sent reaches 100 percent. Press **Ctrl-C** to interrupt the PGM application and return to the EXEC prompt.

To determine the number of network elements through which the packet can pass before reaching the receiver, specify the Time To Live (TTL), which can vary between 1 and 255.

#### multicast connectivity-test receive Command

To receive PGM packets, use the **multicast connectivity-test receive** command. Entering this command runs the pgmratemon application, which listens for the PGM multicast data transmitted from a PGM sender on the specified multicast IP address. When a packet is received, the pgrmratemon application lists the packet size and bandwidth.

To test the multicast connectivity between two Content Engines, use the same multicast IP address for both send and receive. The pgmratemon application terminates by itself after a default period of 3 minutes and returns to the EXEC prompt. You can press **Ctrl-C** to terminate the pgmratemon application and return to the EXEC prompt.

Table 2-15 shows the **multicast connectivity-test** command options supported in the ACNS networks where Content Engines run various versions of the ACNS 5.x software.

*Table 2-15 Options Supported for the multicast connectivity-test Command in ACNS 5.x Software*

| ACNS Network Composition | Command Options Supported | Usage Notes |
|---|---|---|
| Multicast sender and receiver Content Engines running the ACNS 5.2 software and later releases | **multicast connectivity-test ce**<br>**multicast connectivity-test cloud-id**<br>**multicast connectivity-test cloud-name**<br>**multicast connectivity-test send**<br>**multicast connectivity-test receive** | You can use all options in the **multicast connectivity-test** command, because both multicast receiver and sender Content Engines are running the ACNS 5.2 software and later releases. |
| Multicast sender Content Engine running the ACNS 5.2 software and later releases and multicast receiver Content Engines running the ACNS 5.1 software | **multicast connectivity-test send**<br>**multicast connectivity-test receive** | Because the **ce**, **cloud-id**, and **cloud-name** options are not supported in the ACNS 5.1 software, you can use only the **multicast connectivity-test send** and **multicast connectivity-test receive** commands.<br>If you use the **ce**, **cloud-id**, or **cloud-name** options to initiate the multicast connectivity test, the multicast sender displays a remote procedure call (RPC) failure error message because the multicast receiver Content Engine running the ACNS 5.1 software does not support listening to RPC notifications from multicast senders.<br>The multicast sender displays an RPC failure error message if multicast is disabled on the receiver Content Engine. |
| Multicast sender Content Engine running the ACNS 5.1 software and multicast receiver Content Engines running the ACNS 5.2 software and later releases | **multicast connectivity-test send**<br>**multicast connectivity-test receive** | You can use only the **multicast connectivity-test send** and **multicast connectivity-test receive** EXEC commands, because the multicast sender Content Engine running the ACNS 5.1 software does not support the other command options.<br>To test multicast connectivity, you must start the multicast receiver in the Content Engine running the ACNS 5.2 software and later releases first, and then initiate the multicast connectivity test using the **multicast connectivity-test send** EXEC command. |

*Table 2-15 Options Supported for the multicast connectivity-test Command in ACNS 5.x Software (continued)*

| ACNS Network Composition | Command Options Supported | Usage Notes |
|---|---|---|
| Multicast sender Content Engine running the ACNS 5.0 software and multicast receiver Content Engines running the ACNS 5.2 software and later releases | **pgmrategen**<br>**pgmratemon** | Because the ACNS 5.0 software does not use the **multicast connectivity-test** commands, you must use the **pgmrategen** EXEC command to send PGM packets to multicast receiver Content Engines. Use the **pgmratemon** EXEC command or **multicast connectivity-test receive** EXEC command to start the multicast receiver on the Content Engine running the ACNS 5.2 software and later releases. |
| Multicast sender Content Engine running the ACNS 5.2 software and later releases and multicast receiver Content Engines running the ACNS 5.0 software | **pgmrategen**<br>**pgmratemon** | Use the **pgmrategen** EXEC command or **multicast connectivity-test send** EXEC command to send PGM packets to multicast receiver Content Engines. Use the **pgmratemon** EXEC command to start the multicast receiver on the Content Engine running the ACNS 5.0 software. |

## Examples

The following example shows the output of the **multicast connectivity-test ce** command for a multicast IP address of 239.1.1.1 and receiver Content Engine IP addresses 10.43.27.2 and 10.43.27.4:

```
ContentEngine# multicast connectivity-test ce 10.43.27.2 10.43.27.4 multicast-address
239.1.1.1
Connecting to Receiver : 10.43.27.2
Starting PGM Receiver on the CE 10.43.27.2
Connecting to Receiver : 10.43.27.4
Starting PGM Receiver on the CE 10.43.27.4
Packet Generation thread has started. It will start sending packets after PGMReceivers
have been started
Time Elapsed : 180 seconds
PGM Sender has finished sending packets. Awaiting Receiver response
Will wait for 60 seconds...

Configuration
-------------
Multicast Address : 239.1.1.1
Port : 7000
Max Rate to send PGM Packets : 128 kbps
Time to live for multicast packets : 255
Forwarder Error Correction Size : 8

Detailed Report
----------------

Receiver IP : 10.43.27.2
Duration : 180 seconds
No of Packets received : 1134
Packet Length : 1024 Bytes
Minimum BW : 5.714 KBps
Maximum BW : 7 KBps
Average BW : 6.291 KBps
NAK Count : 0


Receiver IP : 10.43.27.4
```

```
Duration : 180 seconds
No of Packets received : 1134
Packet Length : 1024 Bytes
Minimum BW : 5.691 KBps
Maximum BW : 7 KBps
Average BW : 6.298 KBps
NAK Count : 0


Summary Report
--------------
Total number of receivers : 2
No: of receivers which received Packets : 2
No: of receivers which did not receive Packets : 0
No: of RPC calls failures : 0
No: of Other Errors obtained from Receivers : 0
```

The following example shows all the optional parameters and default values:

```
ContentEngine# multicast connectivity-test ce 10.77.155.171 10.77.155.175 10.77.155.179
multicast-address 239.10.1.11 duration 180 FEC-size 8 max-transfer-rate 128
message-length 1024 output-style detail port 7000 time-to-live 255
```

The following example shows the **multicast connectivity-test cloud-name command** with all the optional parameters and default values:

```
ContentEngine# multicast connectivity-test cloud-name mcloud1 duration 180 message-length
1024 output-style detail port 7000
```

The following example shows the output of the **multicast connectivity-test cloud-name** command for the multicast cloud Mcloud1. Because the multicast address was not specified, the ACNS software uses the advertisement IP address 239.1.1.1, specified in the multicast cloud configuration, as the multicast IP address. PGM packets are sent to the two receiver Content Engines (IP addresses 10.43.27.2 and 10.43.27.4) that make up the multicast cloud Mcloud1.

```
ContentEngine# multicast connectivity-test cloud-name Mcloud1
Connecting to Receiver : 10.43.27.2
Starting PGM Receiver on the CE 10.43.27.2
Connecting to Receiver : 10.43.27.4
Starting PGM Receiver on the CE 10.43.27.4
Packet Generation thread has started. It will start sending packets after PGMReceivers
have been started
Time Elapsed : 180 seconds
PGM Sender has finished sending packets. Awaiting Receiver response
Will wait for 60 seconds...

Configuration
-------------
Multicast Address : 239.1.1.1
Port : 7000
Max Rate to send PGM Packets : 10000 kbps
Time to live for multicast packets : 255
Forwarder Error Correction Size : 16

Detailed Report
---------------

Receiver IP : 10.43.27.4
Duration : 180 seconds
No of Packets received : 1139
Packet Length : 1024 Bytes
Minimum BW : 4.903 KBps
Maximum BW : 7 KBps
Average BW : 6.296 KBps
```

```
NAK Count : 0


Receiver IP : 10.43.27.2
Duration : 180 seconds
No of Packets received : 1139
Packet Length : 1024 Bytes
Minimum BW : 5.641 KBps
Maximum BW : 7 KBps
Average BW : 6.319 KBps
NAK Count : 0


Summary Report
--------------
Total number of receivers : 2
No: of receivers which received Packets : 2
No: of receivers which did not receive Packets : 0
No: of RPC calls failures : 0
No: of Other Errors obtained from Receivers : 0
```

The following example shows the output of the **multicast connectivity-test send** command for a multicast IP address of 239.1.1.1 and TTL of 255:

```
ContentEngine# multicast connectivity-test send 239.1.1.1 255
Starting pgmrategen ....
pgmrategen is already running.  Exiting previous instance
Sending 1024 messages of 1024 bytes (1024 Kbytes)
PGM rate is 1024 Kbps
Progress: 99%
Total time 159.924 seconds, 6.40302 KBps, 52.4536 Kbps
```

The following example shows the output of the **multicast connectivity-test receive** command:

```
ContentEngine# multicast connectivity-test receive 239.1.1.1
Starting pgmratemon ....
This CE is not configured as Multicast receiver in any cloud
Configuring this CE as Satellite mode receiver

Press ^C to abort or wait for 3 mins to exit....

Multicasting PGM multicast data to SmartPGM receivers on multicast address 239.1

Sending 1024 messages of 1024 bytes (1024 Kbytes)
PGM rate is 1024 Kbps
Progress: 99%
Total time 8.39756 seconds, 121.94 KBps, 998.934 Kbps

Exiting....
Exiting....
Stopping pgmratemon
ContentEngine#
```

# network-filesystem client (EXEC)

To instruct the NAS share to preempt ownership to this Content Engine in error conditions, use the **network-filesystem client** EXEC command.

**network-filesystem client** {**cifs** {*hostname* | *ip-address*} *directory* **username** *name* **password** *password* {**domain** *domain* **preempt** | **preempt**} | **nfs** {*hostname* | *ip-address*} *directory* **preempt**}

## Syntax Description

| | |
|---|---|
| **cifs** | Configures the Common Internet File System (CIFS) file system. |
| *hostname* | Hostname of the network-attached storage (NAS) device. |
| *ip-address* | IP address of the NAS device. |
| *directory* | Share name (CIFS or NFS). |
| **username** | Sets the username for the CIFS client that is authenticated to allow access to the CIFS server. |
| *name* | Username for the CIFS client that is authenticated to allow access to the CIFS server. |
| **password** | Sets the password associated with the user who is authenticated for access to the CIFS server. |
| *password* | Password associated with the user who is authenticated for access to the CIFS server. |
| **domain** | (Optional) Sets the domain name used for domain-based authentication to allow access to the CIFS server. |
| *domain* | (Optional) Domain name used for domain-based authentication to allow access to the CIFS server. |
| **preempt** | Preempts the NAS share ownership to this Content Engine from another Content Engine. |
| **nfs** | Configures the NFS file system. |

## Defaults

No default behavior or values

## Command Modes

EXEC

## Usage Guidelines

When you have a NAS device attached to a Content Engine, errors may occur in the following scenarios:

- When a NAS device is removed from a Content Engine without being properly detached through the CLI or Content Distribution Manager GUI. Later, when the NAS device is attached to another Content Engine, the NAS attachment fails with the error message:

  ```
  This NAS share is being used by another CE: <CE-name> from date <date>
  ```

  and you are prompted to use the **network-filesystem client preempt** EXEC command before reattempting to attach the NAS.

- If the Content Engine hardware has been replaced and the Content Engine mistakenly believes its NAS share is being used by another Content Engine.

You can use the **network-filesystem client** EXEC command **preempt** option to preempt the NAS share from the mistaken other Content Engine and recover from the two error conditions mentioned above. The **preempt** option does nothing if the Content Engine itself already owns the NAS share.

## Examples

The following example shows how to preempt the NAS share from another Content Engine:

```
ContentEngine# network-filesystem client nfs 172.16.162.44 pub/cecdn1/ preempt
```

The following example shows how to preempt the NAS share from another Content Engine where a password is associated with the user, who is authenticated for access to the CIFS server:

```
ContentEngine# network-filesystem client cifs 172.16.162.55 /pub/cemedia1/ username
cifs_ceusr password a#2as$ domain MYGROUP preempt
```

## Related Commands

**network-filesystem client** (global configuration mode)

# network-filesystem client (global configuration)

To extend Content Engine storage to remote file systems with the Common Internet File System (CIFS) or Network File System (NFS) protocols, use the **network-filesystem client** global configuration command. To remove the configuration, use the **no** form of this command.

**network-filesystem client** {**cifs** {*hostname* | *ip-address*} *directory* {**cdnfs** | **mediafs**} **reserved-disk-space** *space* **username** *name* **password** *password* [**domain** *domain*] | **nfs** {*hostname* | *ip-address*} *directory* {**cdnfs** | **mediafs**} **reserved-disk-space** *space*}

**no network-filesystem client** {**cifs** {*hostname* | *ip-address*} *directory* {**cdnfs** | **mediafs**} **reserved-disk-space** *space* **username** *name* **password** *password* [**domain** *domain*] | **nfs** {*hostname* | *ip-address*} *directory* {**cdnfs** | **mediafs**} **reserved-disk-space** *space*}

## Syntax Description

| | |
|---|---|
| **cifs** | Configures the Common Internet File System (CIFS) file system. |
| *hostname* | Hostname of the network-attached storage (NAS) device. |
| *ip-address* | IP address of the NAS device. |
| *directory* | Share name (CIFS or Network File System [NFS]). |
| **cdnfs** | Configures the file system used for the pre-positioned content. |
| **mediafs** | Configures the file system used for the on-demand streaming content. |
| **reserved-disk-space** | Configures disk space assigned to the ACNS device. The reserved space should be a fraction of the size of the physical CIFS or NFS share. |
| *space* | Space assigned to the ACNS Content Engine device, expressed as an integer followed by MB for megabytes or GB for gigabytes.<br>**Note** A minimum disk space of 1 GB must be reserved on the NAS server if a NAS file system is to be attached to the Content Engine. |
| **username** | Sets the username for the CIFS client that is authenticated to allow access to the CIFS server. |
| *name* | Username for the CIFS client that is authenticated to allow access to the CIFS server. |
| **password** | Sets the password associated with the user who is authenticated for access to the CIFS server. |
| *password* | Password associated with the user who is authenticated for access to the CIFS server. |
| **domain** | (Optional) Sets the domain name used for domain-based authentication to allow access to the CIFS server. |
| *domain* | (Optional) Domain name used for domain-based authentication to allow access to the CIFS server. |
| **nfs** | Configures the NFS file system. |

## Defaults

No default behavior or values

## Command Modes

global configuration

## Usage Guidelines

In the ACNS software, a Content Engine is allowed to attach to network-attached storage (NAS) devices to extend the storage capacity of the Content Engine. The Content Engine may act as an NFS client, a CIFS client, or both when communicating with NAS devices. The NFS or CIFS servers supported include UNIX-like NFS servers and Microsoft Windows servers. The Content Engine is not intended to function as a NAS storage device.

The **network-filesystem client** command lets the Content Engine use a NAS device instead of a Cisco Storage Array to extend the storage capacity of the Content Engine. NAS devices are free from many of the physical limitations (such as the length of the cable) associated with SCSI storage arrays. The performance of NAS devices is slower than with a Storage Array and depends on the available bandwidth and latency of the network connecting the Content Engine and the NAS device.

The content can be pre-positioned or requested on demand from the NAS device and is stored in an exclusive partition (or share) on the Content Engine. The NAS may store files in other partitions (or shares) used by other applications, but the Content Engine will not serve content from those partitions. This feature supports the cdnfs and mediafs but does not support the cfs.

**Note** When acting as an NFS client, the Content Engine reads and writes to the NAS share as the user root. Content Engines request NFS file access using a root identity; therefore, the NFS server must be configured to map the remote root user to a user ID with sufficient read/write privileges on the server. If the NFS server assigns some other username to the remote user, make sure that the Content Engine has adequate read and write privileges on the NAS share.

Remove all unrelated content in the share before the share is attached as network-attached storage, although the share does not have to be empty. Once a share is attached to the Content Engine as network-attached storage, do not modify the share content or use the share for other purposes. Do not create new subdirectories under the share or put the content in any subdirectories that may exist.

Before a NAS share is attached to a Content Engine, you should remove unrelated content in that share. However, you are not required to empty the NAS before attaching to a Content Engine. To remove unrelated content from the NAS share, the NAS server administrator must log on to the NAS server and remove the files directly.

**Note** Do not export subdirectories under a share that has already been assigned to a Content Engine.

### acquisition-distribution database-cleanup Command

The **acquisition-distribution database-cleanup** command is useful if a NAS cdnfs volume is inaccessible when it is being detached, or if the acquisition distribution database synchronization process failed when the NAS cdnfs volume is being detached. The process may fail if the NAS cdnfs is inaccessible, or if the process is interrupted. At that time, you can run the **acquisition-distribution database-cleanup** command after the NAS cdnfs has been detached.

### cdnfs cleanup start Command

The **cdnfs cleanup start** command cleans up the unsubscribed content in the cdnfs. For example, if Content Engine 1 has an attached NAS cdnfs with content previously collected from Content Engine 2, the **cdnfs cleanup start** command can possibly remove part of this content.

**Caution** Be careful when using the **cdnfs cleanup start** command because this command cleans up the unsubscribed content in the cdnfs.

### Handling a NAS Device Failure

The ACNS software uses the NAS health prober to monitor the NAS online status. If a NAS share is offline for a relatively long time (approximately 5 minutes for cdnfs and 10 minutes for mediafs), the NAS health prober assigns a failed status to the NAS share, and the ACNS 5.1 network stops using it. After the NAS share comes back online after a failure, the system automatically begins to use the NAS share again.

The NAS health prober also monitors whether a NAS share has been preempted by another Content Engine. If the NAS share has been preempted, the NAS health prober automatically detaches the preempted share.

### Detaching NAS Shares

If you detach a NAS share from a Content Engine, the acquisition and distribution subsystem needs to synchronize its database record with the on-disk pre-positioned content. The synchronization can be done right before the NAS share is detached from the Content Engine or anytime after the NAS share is detached. You must perform acquisition and distribution pre-detaching synchronization when a NAS cdnfs is to be detached. Use the **no network-filesystem client** EXEC command to detach the NAS share. If acquisition and distribution pre-detaching synchronization fails for whatever reason, post-detaching synchronization can be performed with the **acquisition-distribution database-cleanup** EXEC command.

The time needed for synchronization before detaching the NAS share is proportional to the number of pre-positioned objects in the NAS share. The time needed for synchronization after detaching the NAS share is proportional to the number of pre-positioned objects on local disks (an estimate is 1 hour per 10,000 objects). If you only need to detach a NAS share temporarily, then synchronization might not be necessary.

**Note** Synchronization could take hours or longer if thousands of pre-positioned files are on the Content Engine. We recommend that you run this task during off-peak hours. You should run and stop normal acquisition and distribution jobs during synchronization. Streaming services are not affected when this task is running.

## Examples

The following example configures the Content Engine to serve content from the directory mydir on the NFS server my-nfs-server.mydomain.com. This example also configures the Content Engine to use a cdnfs partition with a reserved space of 10 gigabytes.

```
ContentEngine(config)# network-filesystem client nfs my-nfs-server.mydomain.com /mydir
cdnfs reserved-disk-space 10GB
```

The following example configures the Content Engine to serve content from the directory mydir on the CIFS server my-cifs-server.mydomain.com:

```
ContentEngine(config)# network-filesystem client cifs my-cifs-server.mydomain.com /mydir
mediafs reserved-disk-space 20GB username ce_cifsuser password ce_cifspswd
```

The preceding example configures the Content Engine to use a mediafs partition with a reserved space of 20 gigabytes. The username used to authenticate the Content Engine as a CIFS client is *ce_cifsuser* and the password is *ce_cifspswd*. Use the **show disks network-attached EXEC** command to determine if the NAS attachment succeeded.

If a NAS cdnfs volume is inaccessible when it is detached, the following example shows how to synchronize the acquisition and distribution on the Content Engine:

```
ContentEngine(config)# acquisition-distribution database-cleanup
```

The following example shows how to clean up the unsubscribed content:

```
ContentEngine(config)# cdnfs cleanup start
```

**Caution** If a NAS cdnfs share contains content distributed from another Content Engine, entering the **cdnfs cleanup start** command may remove that content if the content has not been recognized by the local Content Engine. Do not run this command if you do not want this content to be removed.

**Related Commands**

**acquisition-distribution database-cleanup**
**cdnfs cleanup**
**show disks details**
**show disks network-attached**
**show network-filesystem client**
**show statistics cdnfs**
**show statistics mediafs**

# network-filesystem server

To configure the Content Engine to serve pre-positioned content using the Windows file sharing protocol (CIFS), use the **network-filesystem server** global configuration command. Use the **no** form of this command to remove the configuration.

**network-filesystem server** {**cifs** | **samba**} {**authentication-mode** {**ce-user** | **cifs-user** | **ldap-server** | **password-server** | **public**} | **enable** | **max-connections** *number* | **share-web-site** *site-name* [**protect-auth-content** | **share-auth-content** | **share-name** *string*]}

**no network-filesystem server** {**cifs** | **samba**} {**authentication-mode** {**ce-user** | **cifs-user** | **ldap-server** | **password-server** | **public**} | **enable** | **max-connections** *number* | **share-web-site** *site-name* [**protect-auth-content** | **share-auth-content** | **share-name** *string*]}

## Syntax Description

| | |
|---|---|
| **cifs** | Configures a Windows file-sharing server that uses the Common Internet File System (CIFS). |
| **samba** | Specifies the deprecated keyword maintained for backward compatibility with ACNS software, Release 5.0. |
| **authentication-mode** | Configures the Windows file-sharing server authentication mode. |
| **ce-user** | Specifies the standard Content Engine authentication (default). Allows the user to log in with the regular Content Engine user account and password for accessing the Windows file server. |
| **cifs-user** | Specifies both Content Engine authentication and Windows authentication for users configured using the **username user cifs-password** command. |
| **ldap-server** | Specifies the LDAP server for Windows file sharing authentication. |
| **password-server** | Specifies a Windows domain controller or other centralized NTLM password server for authentication. |
| **public** | Allows any user with IP access to the Content Engine access to the unprotected content. No access is allowed to the protected content. |
| **enable** | Enables the Windows file server. |
| **max-connections** | Sets the maximum number of concurrent connections to the Windows file server. |
| *number* | Maximum number of concurrent connections. |
| **share-web-site** | Limits access to specified pre-positioned websites on the cdnfs. |
| *site-name* | Fully qualified domain name of a website or first-level component. |
| **protect-auth-content** | (Optional) Prevents access to the protected content for the specified website. |
| **share-auth-content** | (Optional) Allows access to the protected content for the specified website. |
| **share-name** | (Optional) Configures a CIFS share name. |
| *string* | CIFS share name. The share name must be less than or equal to 64 characters, and contain only alphanumeric characters and hyphen (-), underscore (_), or period (.). For more information, see the "Customizing the CIFS Share Name" section on page 2-336. |

**Defaults**

Windows file sharing is disabled by default.

**max-connections** *number*: 16.

**cifs server authentication mode**: ce-user.

**protect-auth-content**: do not share.

**Command Modes**

global configuration

**Usage Guidelines**

The Windows file-sharing protocol is called either the Server Message Block (SMB) or the Common Internet File System (CIFS) protocol. The Linux implementation of the Windows file-sharing system is called Samba. In the documentation for the ACNS software, Release 5.2 and later releases, the Windows file server includes the Samba server functionality.

The Windows file server is supported only on the Content Engines; the Windows file server is not supported on the Content Distribution Managers on the Content Routers.

The Windows client and file server feature in the ACNS software provides parity with the existing file-sharing service available in the ACNS software, Release 4.x.

In the ACNS software, use the **network-filesystem server cifs enable** global configuration command to enable the Windows file server, which is disabled by default. This command starts the server to serve Windows client requests for file sharing using the SMB protocol. Once the server is enabled, clients can browse the pre-positioned content on the Content Engine. Use the **no** form of this command to disable the Windows file server.

You can authenticate users who need to access the Windows file server by using one of the following four authentication modes:

- ce-user
- cifs-user
- password-server
- public
- ldap-server

When using the **ce-user** option, the user logs in with the regular Content Engine password for accessing the Windows file server. You need to configure the Windows client to send an unencrypted password, and depending on your network configuration and security policy, this setting may not be acceptable. Use the **authentication local** global configuration command to set the authentication mode for the Content Engine when using the **ce-user** option. Use the **username** global configuration command to add a user to the list of valid Windows file server users.

**Note** Authentication modes other than public and cifs-user require that the CIFS client (typically, a Windows PC or a Windows laptop) send the user's password in clear text. This setting is not the default for recent versions of Windows (including Windows 2000 and Windows XP). It may only be feasible for users to use public or cifs-user modes.

Configuring the Windows client to send an unencrypted password is not required if you use the **cifs-user** option. However, users will have to enter two passwords to access the Windows file server: the Windows password and the Content Engine password. It is possible that the user's password on the Content Engine (configured using the **username user cifs-password** command) may differ from the password on other

servers; these passwords need to be synchronized by the administrator of the ACNS network. In cases where many users need to access the content, only public authentication mode is feasible. If only the administrator can access the content (for example, to check if the content is fully replicated), then you may need to configure cifs-user authentication mode.

Use the **password-server** option to use a Windows domain controller or other centralized NTLM password server for authentication. Before using this option, identify the server with the **ntlm server domain** *name* and the **ntlm server host** *hostname* global configuration commands. This option enables the integrated NTLM authentication mode, which means that the Content Engine relies on the password database stored on an NTLM password server to authenticate users. Password servers are Windows domain controllers. Specifying this option defines the domain or workgroup in which the Content Engine and Windows clients are configured. It also defines the server that stores the password database.

Use the **public** option to allow any user with IP access to the Content Engine access to the unprotected content.

To view the status of the Windows file server, use the **show network-filesystem server cifs** EXEC command. This command displays information about the current status of the Windows file server (enabled or disabled) and a list of the files that are being shared.

**Caution** A security problem has been found in versions of Samba up to and including release 2.2.8a. An anonymous user could exploit this vulnerability to gain root access on the target machine. This problem exists with the Windows file server enabled in the releases prior to the ACNS software 5.3. To eliminate this problem, upgrade to the ACNS software, Release 5.3 and later releases, or disable the Windows file server.

#### Customizing the CIFS Share Name

In the ACNS 5.3 software, a new option has been added to the **network-filesystem server cifs share-web-site** *string* global configuration command that allows you to change the CIFS share name. To configure a new CIFS share name, use the **share-name** option.

You can use this command to configure the CIFS server in the ACNS 5.0, 5.1, and 5.2 software; however, if you configure the **share-name** option on a Content Engine that is using the pre-ACNS 5.3 software, the content in the shared directory is shared using the UNC path, and the customized share name is disregarded as follows:

```
CE(config)# network-filesystem server cifs share-web-site www.cisco.com share-name cco
```

When using this same command for a Content Engine running ACNS 5.1 software, the following UNC path is used to share the index.html directory:

\\ce\www.cisco.com\index.html

When using this same command for a Content Engine running ACNS 5.3 software, the following UNC path is used to share the index.html directory:

\\ce\cco\index.html

## Examples

The following example shows how the **network-filesystem server cifs enable** global configuration command enables the Windows file server on Content Engines when the cdnfs is configured and enabled:

```
ContentEngine(config)# network-filesystem server cifs enable
```

Use the **username** global configuration command to add a user to the list of valid Samba users. Users have the option of providing a clear-text Windows sharing password or an encrypted Samba sharing password.

```
ContentEngine(config)# username user1 cifs-password ?

  0     Specifies clear-text Windows sharing password (default)
  1     Specifies type 1 encrypted samba password
  WORD  User Windows sharing password (clear text)
```

The following example adds the username gif to the list of Samba users. The password specified in clear text is saved in the running and startup configurations in an encrypted form.

```
ContentEngine(config)# username gid cifs-password 0 l8m3
```

## Related Commands

**ntlm server**
**show network-filesystem server**
**username**

# no (global configuration)

To undo a global configuration command or set its defaults, use the **no** form of a global configuration command.

**no** *command*

**Note** The commands you can use with an ACNS device (including the **no** form of each command) vary based on whether the device is configured as a Content Distribution Manager, Content Engine, or Content Router. See Table 2-1 on page 2-2 to identify the commands available for a specific device.

## Syntax Description

| | |
|---|---|
| *command* | Specifies the command type; see the "Usage Guidelines" section for valid values. |

## Defaults

No default behavior or values

## Command Modes

global configuration

## Usage Guidelines

Valid values for *command* are as follows:

| | |
|---|---|
| **aaa** | Configures authentication, authorization, and accounting (AAA). |
| **access-lists** | Configures access control list entries. |
| **acquirer** | Configures acquisition parameters. |
| **asset** | Configures the asset tag name string. |
| **authentication** | Configures the authentication. |
| **auto-register** | Configures the autoregistration with the Content Distribution Manager. |
| **bandwidth** | Configures bandwidth controls. |
| **bitrate** | Configures the bit rate. |
| **bypass** | Configures bypass. |
| **cdm** | Configures the Content Distribution Manager settings. |
| **cdp** | Configures Cisco Discovery Protocol (CDP). |
| **clock** | Configures the time-of-day clock. |
| **cms** | Configures the Centralized Management System (CMS). |
| **device** | Configures the device mode. |
| **dns** | Configures the Content Engine DNS cache. |
| **error-handling** | Customizes how the Content Engine should handle errors. |
| **exception** | Configures exceptions. |
| **exec-timeout** | Configures the exec timeout. |
| **external-ip** | Configures up to eight external (NAT) IP addresses. |

| | |
|---|---|
| **ftp** | Configures FTP caching-related parameters. |
| **gui-server** | Configures the GUI server. |
| **help** | Configures the assistance for the command-line interface. |
| **hostname** | Configures the system's network name. |
| **http** | Configures HTTP-related parameters. |
| **https** | Configures HTTPS-related parameters. |
| **icap** | Configures the ICAP feature for the HTTP protocol. |
| **icp** | Configures Internet Cache Protocol (ICP) parameters. |
| **inetd** | Configures FTP, RCP, and TFTP services. |
| **interface** | Configures a Fast Ethernet or Gigabit Ethernet interface. |
| **ip** | Configures IP parameters. |
| **ip access-list** | Configures IP access lists. |
| **kernel** | Enables access to the kernel debugger. |
| **ldap** | Configures Lightweight Directory Access Protocol (LDAP) parameters. |
| **logging** | Configures the system logging (syslog). |
| **mediafs-division** | Configures the media file system space allocation for the Windows Media Technologies (WMT) cache and the RealProxy cache. |
| **multicast** | Configures multicast client parameters. |
| **network-filesystem** | Configures the network file system server and client. |
| **ntlm** | Configures Windows NT LAN Manager (NTLM) parameters. |
| **ntp** | Configures the Network Time Protocol (NTP). |
| **offline-operation** | Configures the offline service operation. |
| **pace** | Configures the HTTP pacing bandwidth and bit rate. |
| **port-channel** | Configures port-channel global options. |
| **pre-load** | Configures the content preloading. |
| **primary-interface** | Configures a primary interface. |
| **proxy-auto-config** | Configures the browser proxy autoconfiguration feature. |
| **proxy-protocols** | Configures proxy protocols-related parameters. |
| **radius-server** | Configures RADIUS server authentication. |
| **rtsp** | Configures Real-Time Streaming Protocol (RTSP)-related parameters. |
| **rule** | Configures the Rules Template. |
| **snmp-server** | Configures the SNMP server. |
| **sshd** | Configures the Secure Shell (SSH) service. |
| **ssh-key-generate** | Generates the Secure Shell host key. |
| **tacacs** | Configures TACACS+ authentication. |
| **tcp** | Configures global TCP parameters. |
| **telnet enable** | Configures Telnet services. |
| **tftp-server** | Configures a TFTP server. |
| **transaction-logs** | Configures the transaction logging. |
| **tvout** | Configures a TV-out service. |
| **url-filter** | Configures the URL filtering. |
| **username** | Establishes username authentication. |

| | |
|---|---|
| **wccp** | Configures the Web Cache Communication Protocol (WCCP). |
| **websense-server** | Configures local Websense server parameters. |
| **wmt** | Configures Windows Media Technologies parameters (WMT). |

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

The following example enables WCCP Version 2 on a Content Engine:

```
ContentEngine(config)# wccp version 2
```

The following example disables WCCP Version 2 on a Content Engine:

```
ContentEngine(config)# no wccp version 2
```

# no (interface configuration)

To negate a Fast Ethernet or Gigabit Ethernet interface configuration command or set its defaults, use the **no** interface configuration command.

**no interface** {**FastEthernet** *slot/port* **| GigabitEthernet** *slot/port*} [**autosense | bandwidth** {**10 | 100 | 1000**} **| cdp enable | channel-group** {**1 | 2**} **| fullduplex | halfduplex | ip** {**access-group** {*ip-access list* | *accesslist_name*} {**in | out**} **| address** {*ip_address netmask* [**secondary**] **| dhcp** [**client-id** *id* **hostname** *name* **| hostname** *name* **client-id** *id*]}} **| mtu** *mtusize* **| shutdown | standby** *grpnumber* {**errors** *maxerrors* **| ip** *ip-address netmask* **| priority** *priority*}]

To negate a Fibre Channel or port-channel interface configuration command or set its defaults, use the **no** interface configuration command.

**no** {**FibreChannel** *slot/port* [**mode** {**autosense | direct-attached | switched**} **| speed** {**1 | 2 | autosense**}] **| PortChannel** {**1 | 2**} [**ip** {**access-group** {*ip-access list* | *accesslist_name*} {**in | out**} **| address** *ip-address netmask*} **| shutdown**]}

## Syntax Description

| | |
|---|---|
| **FastEthernet** | Selects a Fast Ethernet interface to configure. |
| *slot/port* | Slot and port number for the selected interface. The slot range is 0–3; the port range is 0–3. The slot number and port number are separated with a forward slash character (/). |
| **GigabitEthernet** | Selects a Gigabit Ethernet interface to configure. |
| *slot/port* | Slot and port number for the selected interface. The slot range is 0–2; the port range is 0–3. The slot number and port number are separated with a forward slash character (/). |
| **autosense** | (Optional) Sets the interface to autosense. |
| **bandwidth** | (Optional) Sets the bandwidth of a specified interface. |
| **10** | Sets the bandwidth of the interface to 10 megabits per second (Mbps). |
| **100** | Sets the bandwidth of the interface to 100 Mbps. |
| **1000** | Sets the bandwidth of the interface to 1000 Mbps. This option is not available on all ports and is the same as autosense. |
| **cdp** | (Optional) Configures Cisco Discovery Protocol on the specified interface. |
| **enable** | Enables the Cisco Discovery Protocol on the specified interface. |
| **channel-group** | (Optional) Configures the EtherChannel group. |
| **1** | Assigns the interface's EtherChannel to group 1. |
| **2** | Assigns the interface's EtherChannel to group 2. |
| **fullduplex** | (Optional) Sets the interface to full-duplex operation. |
| **halfduplex** | (Optional) Sets the interface to half-duplex operation. |
| **ip** | (Optional) Enables IP configuration commands for the interface. |
| **access-group** | Configures access control for IP packets on this interface using an access control list (ACL). |
| *ip-access list* | Numeric identifier that identifies the ACL to apply to the current interface. For standard ACLs, the valid range is 1–99; for extended ACLs, the valid range is 100–199. |

| | |
|---|---|
| *accesslist_name* | Alphanumeric identifier of up to 30 characters, beginning with a letter that identifies the ACL to apply to the current interface. |
| **in** | Applies the specified ACL to inbound packets on the current interface. |
| **out** | Applies the specified ACL to outbound packets on the current interface. |
| **address** | (Optional) Sets the interface IP address and subnet mask. |
| *ip-address* | IP address of the interface. |
| *netmask* | Netmask of the interface. |
| **secondary** | (Optional) Makes this IP address a secondary address. |
| **dhcp** | (Optional) Sets the IP address to that negotiated over the Dynamic Host Configuration Protocol (DHCP). |
| **client-id** | (Optional) Specifies the client identifier. |
| *id* | Client identifier. |
| **hostname** | (Optional) Specifies the hostname. |
| *name* | Hostname. |
| **mtu** | (Optional) Sets the interface maximum transmission unit (MTU) size. |
| *mtusize* | MTU size in bytes (68–1500). |
| **shutdown** | (Optional) Shuts down this interface. |
| **standby** | (Optional) 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). |
| **FibreChannel** | Selects the Fibre Channel interface to configure. |
| *slot/port* | Slot and port number for the selected interface. The slot range is 0–0; the port range is 0–3. The slot number and port number are separated with a forward slash character (/). |
| **mode** | (Optional) Sets the Fibre Channel interface operation mode. |
| **autosense** | Sets the operation mode of the Content Engine to autosense. |
| **direct-attached** | Sets the operation mode when the Content Engine is directly connected to a storage array. |
| **switched** | Sets the operation mode when the Content Engine is connected to a switch. |
| **speed** | (Optional) Sets the Fibre Channel interface speed. |
| **1** | Sets the Fibre Channel interface speed to 1 gigabit per second (Gbps). |
| **2** | Sets the Fibre Channel interface speed to 2 Gbps. |
| **autosense** | Sets the Fibre Channel to automatically sense the interface speed. |
| **PortChannel** | Selects the EtherChannel of the interfaces to configure. |
| **1** | Sets the port-channel interface number to 1. |
| **2** | Sets the port-channel interface number to 2. |

| | |
|---|---|
| **ip** | (Optional) Enables IP configuration commands for the interface. |
| **access-group** | Configures access control for IP packets on this interface using an access control list (ACL). |
| *ip-access list* | Numeric identifier that identifies the ACL to apply to the current interface. For standard ACLs, the valid range is 1–99; for extended ACLs, the valid range is 100–199. |
| *accesslist_name* | Alphanumeric identifier of up to 30 characters, beginning with a letter that identifies the ACL to apply to the current interface. |
| **in** | Applies the specified ACL to inbound packets on the current interface. |
| **out** | Applies the specified ACL to outbound packets on the current interface. |
| **address** | Sets the interface IP address. |
| *ip-address* | IP address of this interface. |
| *netmask* | Netmask of this interface. |
| **shutdown** | (Optional) Shuts down this interface. |

## Defaults

No default behavior or values

## Command Modes

interface configuration

## Examples

The following example disables autosense on an interface:

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

## Related Commands

**interface**
**show interface**
**show running-config**
**show startup-config**

# ntlm

To configure Microsoft Windows NT LAN Manager (NTLM) parameters, use the **ntlm** global configuration command. To disable individual options, use the **no** form of this command.

**ntlm allow-domain** {**domain** *domain-name* | **enable**}

**ntlm basic-auth enable**

**ntlm server ad-group-search** {**enable** | **enum-user** {**domain** *domain-name* | **password** *pass* | **username** *user-name*} | **gc-server** {**host** {*hostname* | *ip-address*} | **port** *port-num* | **scheme** {**fail-over** | **load-balanced**}} | **groupname-attribute** *group-name* | **ldap-referral** {**enable** | **limit** *level_number*} | **ldap-search-port** *port-num* | **membership-attribute** *member* | **user-objectclass** *object* | **username-attribute** *user*}

**ntlm server connection-retry** *number*

**ntlm server connection-timeout** *time*

**ntlm server domain** *name*

**ntlm server enable**

**ntlm server host** {*hostname* | *ip-address*}

**ntlm server scheme** {**fail-over** | **load-balanced**}

**no ntlm server** {**allow-domain** {**domain** *domain-name* | **enable**} | **basic-auth enable** | **ad-group-search** {**enable** | **enum-user** {**domain** *domain-name* | **password** | **username** *user-name*} | **gc-server** {**host** {*hostname* | *ip-address*} | **port** *port-num* | **scheme** {**fail-over** | **load-balanced**}} | **groupname-attribute** *group-name* | **ldap-referral** {**enable** | **limit** *level_number*} | **ldap-search-port** *port-num* | **membership-attribute** *member* | **user-objectclass** *object* | **username-attribute** *user*} | **server connection-retry** *number* | **server connection-timeout** *time* | **domain** *name* | **enable** | **host** {*hostname* | *ip-address*} | **scheme** [**fail-over** | **load-balanced**]}

## Syntax Description

| | |
|---|---|
| **allow-domain** | Configures the domain allowed to perform NTLM HTTP request authentication with the Content Engine. For more information, see the "Configuring a List of Allowed Domains List for NTLM HTTP Request Authentication" section on page 2-350. |
| **domain** | Specifies the names of the domains that are allowed to perform NTLM HTTP request authentication with the Content Engine. |
| *domain-name* | Names of the domains that are allowed to perform NTLM HTTP request authentication with the Content Engine. A domain list can contain up to 32 domain names. |
| **enable** | Enables the use of a list of domains allowed to perform NTLM HTTP request authentication with the Content Engine. |
| **basic-auth** | Configures the Content Engine to send a basic authentication response header with an NTLM authentication header to the client browser. |
| **enable** | Enables the Content Engine to send a basic authentication response header with an NTLM authentication header to the client browser. |

| | |
|---|---|
| **server** | Configures NTLM server-related parameters. |
| **ad-group-search** | Configures Active Directory group search options. For more information, see the "Configuring Content Engines for Active Directory Group Searches" section on page 2-351. |
| **enable** | Enables the Active Directory group search. |
| **enum-user** | Configures the enumeration user parameters. |
| **domain** | Specifies the enumeration user domain. |
| *domain-name* | Enumeration user domain name. |
| **password** | Specifies the enumeration user password. |
| *pass* | Enumeration user password. |
| **username** | Specifies the enumeration user username. |
| *user-name* | Enumeration user username. |
| **gc-server** | Specifies the global catalog server parameters. |
| **host** | Specifies the global catalog server hostname or IP address. |
| *host-name* | Global catalog hostname. |
| *ip-address* | Global catalog IP address. |
| **domain** | Specifies the host domain name for this global catalog server. |
| *domain-name* | Host domain name. |
| **port** | Specifies the global catalog server port (the default is 3268). |
| *port-num* | Port for the global catalog server (1–65535). |
| **scheme** | Sets the scheme to be used for the host list. |
| **fail-over** | Specifies that failover should occur between hosts. |
| **load-balanced** | Specifies that round-robin load balancing should occur between hosts. |
| **groupname-attribute** | Specifies the group name attribute in the Active Directory (the default is cn). |
| *group-name* | Group name attribute in the Active Directory. |
| **ldap-referral** | Configures LDAP referral handling parameters. For more information, see the "LDAP Referral Feature" section on page 2-352. |
| **enable** | Enables the Content Engine to handle LDAP referrals. |
| **limit** | Specifies the referral limit for NTLM nested group searches. |
| *level_number* | Depth level of nested referrals that the Content Engine must search for an LDAP query made to the Active Directory server. |
| **ldap-search-port** | Specifies the LDAP search server port (the default is 389). |
| *port-num* | LDAP search server port (1–65535). |
| **membership-attribute** | Specifies the membership attribute in the Active Directory (the default is memberOf). |
| *member* | Membership attribute in the Active Directory. |
| **user-objectclass** | Specifies the object class for the user object (the default is user). |
| *object* | Object class for the user object. |
| **username-attribute** | Specifies the username attribute in the Active Directory (the default is sAMAccountName). |
| *user* | Username attribute in the Active Directory. |

| | |
|---|---|
| **connection-retry** | Specifies the maximum number of attempts to connect to the server (the default is 2). |
| *number* | Maximum number of attempts to connect to the server. |
| **connection-timeout** | Specifies the number of seconds to wait while connecting to the server (the default is 5 seconds). |
| *time* | Time to wait connecting to the server. |
| **domain** | Specifies the NTLM domain name. |
| *name* | Name of the NTLM domain. |
| **enable** | Enables the NTLM authentication. |
| **host** | Configures the NTLM NT controller name or IP address. |
| *hostname* | NTLM NT controller name. |
| *ip-address* | Host IP address. |

## Defaults

**gc-server port**: 3268

**groupname-attribute**: cn

**ldap-search-server port**: 389

**membership-attribute**: memberOf

**user-objectclass**: user

**username-attribute**: sAMAcountName

**connection-retry**: 2

**connection-timeout**: 5 seconds

## 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 hostname or address as primary or secondary. Use this command to identify the password server before enabling Common Internet File System (CIFS) authentication using the **password-server** option with the **network-filesystem server** global configuration command.

**Note** NTLM support on the Content Engine includes the following three types of support: (1) NTLM end-to-end authentication support, (2) NTLM authentication of HTTP requests, and (3) NTLM group information query for authorization purposes. The ACNS 5.x software supports NTLM Version 1 in the case of HTTP request authentication. The ACNS 5.x software supports NTLM Version 1 and Version 2 for end-to-end authentication.

Windows NT LAN Manager (NTLM) is the authentication protocol that is used by Microsoft's browsers (Internet Explorer), proxies, and web servers (IIS). The NTLM protocol, which is a challenge-response-based protocol, can be used to authenticate and block user access to the Internet. The main advantage of using NTLM for HTTP request authentication is that NTLM sends the password in an encrypted format to the server that originated the authentication challenge.

Typically, enterprises are already using NTLM to enforce access control to information that is stored on their intranet sites. Additionally, enterprises want to protect Internet browsing but not have to prompt their end users for usernames and passwords. NTLM provides this authentication scheme through Microsoft Internet Explorer and domain controllers (DCs). Content Engines support NTLM HTTP request authentication in order to support both of these models. A client (web browser) attempts to perform NTLM HTTP request authentication with the Content Engine in order to be allowed to use the Content Engine (the HTTP proxy server) to access the requested content.

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 turn 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 as follows:

- 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.

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 security settings on the Internet options 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 lease expires.

In the ACNS 5.2 software and later releases, the following enhancements were made for NTLM HTTP request authentication:

- Support for up to eight NTLM servers for HTTP request authentication—Ability to configure the Content Engine to use up to eight NTLM servers for HTTP request authentication for load-balancing purposes. The ACNS software, Release 5.1.x and earlier releases supported failover only. For more information, see the "NTLM Load Balancing for HTTP Request Authentication" section on page 2-349.
- Support for up to eight Global Catalog servers for Active Directory group searches—Ability to configure the Content Engine to use up to eight Global Catalog servers for Active Directory group searches. See the "Configuring Content Engines for Active Directory Group Searches" section on page 2-351.

**Note** The order of server configuration determines the order of load balancing or failover. For example, if a failover is enabled, then the first server configured (Server 1) is the primary server and is sent all of the requests first. The last server configured (Server 8) is the last server that the Content Engine contacts. If load balancing is enabled, only the first request is sent to the first configured server (Server 1), after which round-robin is used among the remaining servers (for example, the second request is sent to Server 2, and the third request is sent to Server 3).

- Changes to the Active Directory group search feature—LDAP queries are sent to the same Active Directory server that is assigned to perform the authentication unless the LDAP query fails. If the query fails, the Content Engine sends the authorization request to the next configured server (the Content Engine only tries one more server).
- If the NTLM nested group search feature is enabled, you do not need to configure the **ldap-search-server host** global configuration command. The Content Engine automatically uses the IP address of the configured NTLM server to send the LDAP queries.
- New **scheme** command option for NTLM servers—A **scheme** option was added to the **ntlm server** and **ntlm server ad-group-search gc-server** global configuration commands. This option allows you to specify the scheme (load balancing or failover) that is to be used among the configured NTLM or Global Catalog Servers. The default scheme is failover. Use the **ntlm server scheme** global configuration command to specify the scheme for the NTLM servers for HTTP request authentication. Use the **ntlm server ad-group-search gc-server scheme** global configuration command to change the scheme for the Global Catalog Servers for Active Directory group searches.
- Polling thread—Once one of the configured NTLM or Global Catalog Servers is marked as dead, it is removed from the load-balancing or failover farm to prevent the Content Engine from directing incoming requests to it. The Content Engine periodically polls the dead server (every 30 seconds). If the Content Engine receives a response from the server, it adds the server back into the load-balancing or failover farm.
- Authentication method controls for NTLM—Ability to enable or disable the Content Engine from sending a basic authentication response header along with an NTLM authentication header. For more information, see the "Configuring the Authentication Method Control for NTLM HTTP Request Authentication" section on page 2-351.
- Support for no default NTLM domain—If the client does not supply a domain name in the request authentication credential and there is no default domain configured on the Content Engine, then an authentication error is returned to the client. A predetermined error page that contains text indicating the reason for the error is sent to the client. This feature is also referred to as the no domain configuration feature.

**Note** The no domain configuration feature is supported only with browsers that do not support NTLM (for example, Netscape 7.1 and earlier browsers [Netscape 7.2 and later browsers support NTLM]). For the Netscape browser, the user must specify the domain if the Content Engine does not have an NTLM default domain configured; otherwise, the client receives an error message. For the Netscape browser, the domain can be supplied only as part of the username in the format domain\username. Browsers that do support NTLM, such as Internet Explorer, always include a domain name in the authentication credentials that originate from either the user being prompted to specify the credentials or from the domain that was used to log in the user on to the desktop.

- Configurable allow domain list—Ability to specify the list of domains that are allowed to perform NTLM HTTP request authentication with the Content Engine. For more information, see the "Configuring a List of Allowed Domains List for NTLM HTTP Request Authentication" section on page 2-350.

You can use the **transaction-logs log-window-domain** global configuration command to configure the Content Engine to send the username and domain name to the transaction log. The Windows domain name that is used for NTLM authentication appears in the username field of the transaction log. The username appears in the format domain\username in those formats that contain usernames that are in Extended Squid-style or custom format using the *%u* format token.

#### NTLM Load Balancing for HTTP Request Authentication

In the ACNS software, Release 4.x to Release 5.1, you needed to configure one primary domain controller for HTTP request authentication and a secondary domain controller for failover. However, in large-scale networks, if all the traffic passes through the Content Engine, even though the Content Engine authentication cache can help reduce the load on the domain controller, it may still be impractical to have a single domain controller handle authentication queries from all of the end users.

In the ACNS 5.2 software and later releases, you can use load balancing between domain controllers. With the ACNS 5.2 software and later releases, you can configure a maximum of eight servers (domain controllers) for load balancing and failover. The order of server configuration determines the order of load balancing or failover.

When you select load balancing, the requests are passed in a round-robin to the domain controllers. For example, if you have *n* servers (domain controllers), the first request goes to Server 1, the second request is sent to Server 2, the *n*th request is sent to Server *n*, and the (*n*+1)th request is sent to Server 1. If Server 1 fails, the Content Engine attempts to send the request to the next configured server that is alive. However, failover to the next alive server occurs only once. For example, if Server 2 goes down when handling request 1, then request 1 does not fail over again.

If you enable load balancing and the server information is changed during run time, the change is picked up at run time without disrupting the service. The configuration of each configured NTLM or Global Catalog Server is available through the **show ntlm** EXEC command. You can see the statistics about the total number of requests that go through the servers when you enter the **show statistics ntlm** EXEC command. To see the statistics about the requests that go through each domain controller, enter the **show statistics ntlm** EXEC command.

**Note** If you enable the Active Directory nested group search, only the servers in the same domain are supported. If you do not enable the Active Directory nested group search, the servers in multiple domains are supported if the servers have a trusted relationship.

#### Configuring the Content Engine to Use NTLM Servers for HTTP Request Authentication

You can use the Content Engine GUI or the CLI to configure a Content Engine to use external NTLM servers for HTTP request authentication.

In the ACNS software, Release 5.1.x and earlier releases, you explicitly designated a primary NTLM server and a secondary NTLM server by using the **primary** and **secondary** options of the **ntlm server host** global configuration command.

In the ACNS software, Release 5.2 and later releases, you can configure a Content Engine to use up to eight NTLM servers for HTTP request authentication. The order of the server configuration determines the order of load balancing or failover. For example, if you enable the failover, then the first server configured (Server 1 that has an IP address of 172.16.10.10) is the primary server and is sent all of the requests first. The last server configured (Server 3 that has the IP address of 172.16.10.14) is the last

server that the Content Engine contacts. If you enable the load balancing, only the first request is sent to the first configured server (Server 1), after which round-robin is used among the remaining servers (for example, the second request is sent to Server 2, and the third request is sent to Server 3).

**Note** In the ACNS 5.2 software, the **ntlm server host primary** option and the **ntlm server host secondary** options were removed because up to eight servers are now supported. In the ACNS 5.2 software, the **ntlm server host scheme load-balanced** option was added.

You can use the Content Engine GUI or the CLI to configure the Content Engine to use up to eight NTLM servers for HTTP request authentication.

From the Content Engine GUI, choose **Caching > NTLM** to access the NTLM window. Use the NTLM window to specify NTLM server settings on the Content Engine, and click **Update.** For more information about the fields on the NTLM window, click the **HELP** button in the window.

From the Content Engine CLI, use the **ntlm server** global configuration command.

#### Configuring a List of Allowed Domains List for NTLM HTTP Request Authentication

In the ACNS 5.1.x software, you were required to specify the name of the Windows NT domain that the end user was to be authenticated against. This specification was referred to as the default NTLM domain name.

In the ACNS software, Release 5.2 and later releases, you are not required to specify a name for the default domain. If the client does not supply a domain name in the request authentication credential and there is no default domain configured on the Content Engine (the **ntlm server domain** global configuration command was not used), then an authentication error message is returned to the client. A predetermined error page that contains the text indicating the reason for the error is sent to the client.

In the ACNS 5.2 software and later releases, you can specify a list of domains that are allowed to perform NTLM HTTP request authentication with the Content Engine. This capability allows you to limit the domains that can perform NTLM HTTP request authentication with the Content Engine. This feature, which is called allowed domain, is enabled on the Content Engine. If the supplied domain credential does not match any of the domains in the allowed domain list, then the HTTP request authentication fails and the client is sent an error message.

To support the allowed domain feature, the following Content Engine CLI commands are available in the ACNS 5.2 software and later releases:

- **ntlm allow-domain enable**—Enables the allowed domain list feature on the Content Engine. By default, the allow domain feature is disabled.
- **no ntlm allow-domain enable**—Disables the allowed domain list feature on the Content Engine.
- **ntlm allow-domain domain** *domain-name*—Defines the names of the domains that are allowed to perform NTLM HTTP request authentication with the Content Engine. A domain list can contain a maximum of 32 domain names.

If the allowed domain list feature is enabled, then this feature works as follows:

- If the client's domain credential matches any domain in the configured domain list, the Content Engine performs NTLM HTTP request authentication for this content request. A case-insensitive comparison is used to check whether the specified domain is listed in the allowed domain list.

- If the client's domain credential does not match any domain in the configured domain list or there are no domains configured on the allowed domain list, the Content Engine denies this content request and sends the client a 407 or 401 authentication error message. The 407 or 401 authentication message has a specific predetermined error page that contains the text indicating the reason for the error.

#### Configuring the Authentication Method Control for NTLM HTTP Request Authentication

By default, the Content Engine (the HTTP proxy server) always sends a basic authentication response header with an NTLM authentication header to the client browser. This default behavior enables the client to be authenticated with the Content Engine even if the client browser does not support the NTLM protocol, as is the case with the Netscape browser. (Internet Explorer supports the NTLM protocol.)

Because basic authentication transmits user credential information in clear text format, it is less secure than NTLM authentication. For security purposes, you may want to configure the Content Engine to not send a basic authentication response header with an NTLM authentication header.

In the ACNS 5.2 software and later releases, you can configure the authentication method control for NTLM HTTP request authentication. The authentication method control feature allows you to enable or disable the Content Engine from sending a basic authentication response header with an NTLM authentication header. To support this feature, the following Content Engine CLI commands were added:

- **ntlm basic-auth enable**—Configures the Content Engine to send a basic authentication response header with an NTLM authentication header to the client browser.
- **no ntlm basic-auth enable**—Configures the Content Engine to not send the basic authentication response header with an NTLM authentication header, or to not honor it in a request.

If you do not want the client browser to be able to use the basic authentication method between the client and the Content Engine for NTLM HTTP request authentication because it is a less secure method than NTLM, then disable the NTLM basic authentication feature on a Content Engine.

To disable the NTLM basic authentication feature on a Content Engine, enter the **no ntlm basic-auth enable** global configuration command.

If the Content Engine is configured to not send the basic authentication header to the client and the client does not support NTLM authentication (for example, Netscape browsers only support basic authentication), then the client cannot continue with this HTTP request. The client browser behavior is browser-dependent; for example, some browsers may retry the request over a certain period of time.

#### Configuring Content Engines for Active Directory Group Searches

In the ACNS software releases prior to 5.1, the Content Engine supported only local groups within a global group for NTLM group-based authorization. To ensure interoperability of the Content Engine NTLM group authentication support with the Microsoft Active Directory database, the ACNS software, Release 5.1 and later releases support static groups.

In the ACNS software, Release 5.1 and later releases, you can retrieve nested group names using an LDAP recursive search and apply all the access lists configured for the nested groups. When you use nested groups with Active Directory servers, the policies configured for parent groups are automatically applied to members in subgroups.

**Note** There are three kinds of groups in an Active Directory: universal, global, and domain local.

To perform a recursive query, an enumeration user's credentials must be provided to query the primary domain controller for a complete list of group names. An enumeration user is an account defined on the Content Engine to allow the Content Engine to perform a search on an Active Directory server. This enumeration user needs to have read privileges throughout the whole directory.

Use the **ntlm server ad-group-search gc-server host domain** *domain-name* global configuration command to specify the host domain name (for example, abc1.local) for the configured Global Catalog Server.

In the ACNS 5.2 software and later releases, you can use the **ldap-search-port** option to the **ntlm server ad-group-search** global configuration command.

Use the **ldap-search-port** option to specify the LDAP port for group information retrieval. The default is port 389. This option configures the LDAP search server port for all of the configured Active Directory domain controllers.

**Note** The **ldap-search-port** option replaces the ACNS 5.1.x software **ldap-search-server port** option.

In the ACNS 5.2 software and later releases, you can use the **scheme** option with the **ntlm server ad-group-search gc-server** global configuration command to specify whether the configured Global Catalog servers are to be used for load balancing or failover.

#### LDAP Referral Feature

In the ACNS 5.2 software, NTLM load balancing was added, which makes cross-domain authorization a more common deployment scenario. In the ACNS 5.3 software release, support for LDAP referral handling was added.

Support of LDAP referral enables the ACNS software to retrieve authorization information for a user who does not belong to the same domain as the configured Active Directory domain controller but does belong to a trusted domain. When the Active Directory domain controller receives an LDAP query for a user who is not in its own domain, but is in a trusted domain, it sends back an LDAP referral URL to the Content Engine. If the LDAP referral support is enabled on a Content Engine, the Content Engine retrieves the information about the referred server in the referral URL and contacts the server to request the user's authorization information.

Support for LDAP referral provides the following capabilities:

- Support of Active Directory trusted domain user authorization
- Support of LDAP referral for NTLM nested group searches
- Ability to configure the LDAP nesting referral level
- Ability to configure Active Directory domain controllers from multiple domains for NTLM load balancing

**Note** The ability to configure Active Directory domain controllers from multiple domains requires that the multiple domains are in a trusted relationship. You cannot perform authentication/authorization correctly if you configure the multiple domain controllers from different nontrusted domains.

To support LDAP referral handling, the **ntlm server ad-group-search ldap-referral** global configuration command was added in the ACNS 5.3 software release.

By default, LDAP referral is disabled on the Content Engine. To enable this feature, enter the **ntlm server ad-group-search ldap-referral enable** global configuration command. After enabling LDAP referral on the Content Engine, you can disable it later by entering the **no ntlm server ad-group-search ldap-referral enable** command.

You can use the **ldap-referral limit** option of the **ntlm server ad-group-search ldap-referral** command to specify the nested referral limit for NTLM nested group searches. By default, five nested referrals are allowed for an NTLM nested group search. Valid values are from 1 to 10.

Although the results of a first-level search can contain the results that the Content Engine is searching for, Active Directory servers tend to return multiple nested referral URLs, which causes additional, unnecessary round trips to the Active Directory server. You can reduce the referral limit to a smaller number if you are sure that the first few level search responses will contain the desired search result because of your directory structure.

For example, if the search result is contained in the first-level search response, you can configure the referral limit to 1 for performance purposes. By setting the referral limit to 1, the Content Engine only follows one referral URL to contact the correct domain controller (Domain Controller A) and does not follow the additional, unnecessary referral URLs that are generated from Domain Controller A with the search result.

Enter the **show ntlm** EXEC command to display the currently configured NTLM parameters on the Content Engine. The command output includes such information as whether LDAP referral is enabled (for example, the command output shows "AD LDAP referral chasing: Enabled"), and the current referral limit (for example, the command output shows "AD LDAP referral chasing limit: 8").

Use the **ntlm server ad-group-search** global configuration command to configure the Content Engine to support Active Directory group searches.

When you enable Active Directory search groups, you must configure the access list with the correct domain name. The group name should look as follows:

*DNS domain name\group name*

The LDAP queries are sent to the same Active Directory server that is assigned to perform authentication unless the LDAP query fails. If the LDAP query fails, the authorization request fails over to the next configured server. If the NTLM service or the LDAP service on the Active Directory server is not accessible, the Content Engine considers the Active Directory server as nonfunctional.

## Examples

The following example configures a Content Engine for NTLM request authentication:

```
ContentEngine(config)# ntlm server enable
ContentEngine(config)# ntlm server domain cisco_abc
ContentEngine(config)# ntlm server host 172.16.10.10
ContentEngine(config)# ntlm server host 172.16.10.12
```

The following example shows how to configure the ACL that must be configured to enable Active Directory search groups:

```
ContentEngine(config)# access-lists 300 permit groupname mydomain.local\univ11_sec
ContentEngine(config)# access-lists enable
```

The following example shows the commands required to enable and implement Active Directory search groups:

```
ContentEngine(config)# ntlm server host 10.77.157.163
ContentEngine(config)# ntlm server domain cache
ContentEngine(config)# ntlm server enable
ContentEngine(config)# ntlm server domain cache
ContentEngine ntlm server ad-group-search enum-user username administrator
ContentEngine(config)# ntlm server domain cache
ContentEngine ntlm server ad-group-search enum-user password ***
ContentEngine(config)# ntlm server ad-group-search enum-user domain cache.acns
ContentEngine(config)# ntlm server ad-group-search gc-server host 10.77.157.213 domain
acns primary
```

```
ContentEngine(config)# ntlm server ad-group-search ldap-search-server host 10.77.157.163
primary
ContentEngine(config)# ntlm server ad-group-search enable
```

The following example shows the output of the **show ntlm** command when the Active Directory search groups option has been enabled:

```
ContentEngine# show ntlm
NTLM parameters:
        NTLM Hosts:
                10.77.157.213
                10.77.157.131
                scheme:                 Fail-over
        State:          Disabled
        Basic Auth:     Enabled
        Default domain: acns
        Connection Timeout:     5
        Connection Retries:     2
        Allow Domains:          Disabled
        Allow Domains List:
                None
        AD group search is enabled
        Enumeration User:
                Username:       user1
                Password:       ****
                Domain:         dom1
        LDAP Port:506
        Global Catalog Servers:
                172.16.30.45    , Domain: gcdom
                Scheme:                 Fail-over
                Port:           3268
        User objectclass:       user
        Username attribute:     sAMAccountName
        Groupname attribute:    cn
        Membership attribute:   memberOf
        AD LDAP referral chasing:       Disabled
        AD LDAP referral chasing limit: 5
```

This list of configured NTLM servers is referred to as the host list.

The following example shows how to specify the maximum number of times that the Content Engine is to attempt to connect to one of the configured NTLM servers:

```
ContentEngine(config)# ntlm server connection-retry 3
```

The default is two attempts. Valid values are from one to three attempts. After the specified number of attempts is exceeded, the Content Engine stops attempting to connect to the NTLM server and attempts to connect to the next configured server on the host list. In the example, this value is set to 3.

The following example shows how to specify how long the Content Engine should wait for a response from the NTLM server to which it is attempting to connect:

```
ContentEngine(config)# ntlm server connection-timeout 10
```

This timeout is for one connection attempt. If the specified amount of time is exceeded, the Content Engine gives up the connection and attempts to connect to the same server up to the specified number of times (the number of retries specified with the **ntlm server connection-retry** global configuration command) before the Content Engine attempts to connect to the next server. The default is 5 seconds. Valid values are from 1 to 20 seconds. In the example, this timeout is set to 10 seconds.

In the following example, the Content Engine is configured to use the configured servers for load balancing:

```
ContentEngine(config)# ntlm server scheme load-balanced
```

When load balancing is enabled, only the first request is sent to the first configured server, after which round-robin is used among the remaining configured servers. (When failover is enabled, the Content Engine sends all the requests to the first configured server.)

The following example shows that the Content Engine is configured to use the Global Catalog Server that has the host domain name of abc1.local:

```
ContentEngine(config)# ntlm server ad-group-search gc-server host 10.77.157.213 domain
abc1.local
```

The following example shows how to specify the ldap search port option:

```
ContentEngine(config)# ntlm server ad-group-search ?
  enable                Enable Active Directory group search
  enum-user             Configure enumeration user parameters
  gc-server             Configure global catalog server parameters
  groupname-attribute   groupname attribute in Active Directory, default is cn
  ldap-search-port      Specify LDAP search server port, default 389
  membership-attribute  membership attribute in Active Directory, default is
                        memberOf
  user-objectclass      objectclass for user object, default is user
  username-attribute    username attribute in Active Directory, default is
                        sAMAccountName
```

The following example shows how to use the scheme option:

```
ContentEngine(config)# ntlm server ad-group-search gc-server ?
  host    Specify global catalog server address
  port    Specify global catalog server port, default 3268
  scheme  Scheme to use for the host list
```

The following example shows how to specify that one instead of five nested referrals are allowed for NTLM nested group searches:

```
ContentEngine(config)# ntlm server ad-group-search ldap-referral limit 1
```

The following example shows how to use the Content Engine CLI to configure a Content Engine to use the maximum number of servers (eight NTLM servers) to load balance HTTP authentication requests by specifying the hostname or IP address of each NTLM server that you want the Content Engine to use for HTTP request authentication:

```
ContentEngine(config)# ntlm server host 172.16.10.10
ContentEngine(config)# ntlm server host 172.16.10.12
ContentEngine(config)# ntlm server host 172.16.10.14
ContentEngine(config)# ntlm server host 172.16.10.16
ContentEngine(config)# ntlm server host 172.16.10.18
ContentEngine(config)# ntlm server host 172.16.10.20
ContentEngine(config)# ntlm server host 172.16.10.22
ContentEngine(config)# ntlm server host 172.16.10.24
```

## Related Commands

**authentication**
**show ntlm**

# ntp

To configure the Network Time Protocol (NTP) server 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** {*ip-address* | *hostname*} [*ip-addresses* | *hostnames*]

**no ntp server** {*ip-address* | *hostname*} [*ip-addresses* | *hostnames*]

## Syntax Description

| | |
|---|---|
| **server** | Sets the NTP server IP address. |
| *ip-address* | NTP server IP address. |
| *hostname* | NTP server hostname. |
| *ip-addresses* | (Optional) IP address of the time server providing the clock synchronization (maximum of four). |
| *hostnames* | (Optional) Hostname of the time server providing the clock synchronization (maximum of four). |

## Defaults

No default behavior or values.

## Command Modes

global configuration

## Usage Guidelines

Use this command to synchronize the Content Engine clock with the specified NTP server. The **ntp server** command enables NTP servers for timekeeping purposes and is the only way to synchronize the system clock with a time server in the ACNS 5.x software releases.

When you synchronize the Content Distribution Manager clock with an NTP server, there is a possibility of all devices registered with the Content Distribution Manager being shown as offline and then reverted to online status. This situation can occur when synchronization with the NTP server sets the Content Distribution Manager clock forward in time by an interval greater than at least two polling intervals or when the software clock on the Content Distribution Manager is changed by a similar value using the **clock** EXEC command. The Content Distribution Manager determines the status of devices in the ACNS network depending on when it was last contacted by the devices for a getUpdate request. If you set the Content Distribution Manager clock ahead in time, you have added that amount of time to the period since the Content Distribution Manager received the last getUpdate request. However, it is only a transient effect. Once the devices contact the Content Distribution Manager for their next getUpdate request after the clock setting change, the Content Distribution Manager GUI reports the status of all devices correctly.

## Examples

The following example configures the IP address of the time server providing the clock synchronization:

```
ContentEngine(config)# ntp 172.16.22.44
```

The following example resets the time server providing the clock synchronization:

```
ContentEngine(config)# no ntp 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 hostname. |
| *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 **ntpdate** command synchronizes the software clock with the hardware clock.

## Examples

The following example sets the software clock of the Content Engine using an NTP server:

```
ContentEngine# ntpdate 10.11.23.40
```

## Related Commands

**clock set**
**show clock**

# offline-operation

To enable offline operation if external network links are disrupted, use the **offline-operation** global configuration command. To disable offline operation, use the **no** form of this command.

**offline-operation enable**

**no offline-operation enable**

## Syntax Description

| | |
|---|---|
| **enable** | Enables offline operation. |

## Defaults

No default behavior or values

## Command Modes

global configuration

## Usage Guidelines

You must enter the **offline-operations enable** command so that your users can continue to access the preloaded and cached content.

Even though network access is disrupted, the ACNS software continues its attempts to make connections to remote servers, execute DNS lookups, and validate the content. The **offline-operations enable** command allows users to access the cache for browsing when there are no external links available to connect them to the origin server.

However, while offline operations are enabled, these conditions apply:

- No attempt is made to make external connections, even if the links later become available.
- If configured to do so, an outgoing proxy is not used even if it is still available.
- Only those DNS lookups stored in the HTTP proxy DNS cache are executed.
- Only HTTP and FTP-over-HTTP content can be accessed.
- The cache can still serve *proxy.pac* files.
- The ACNS network content that requires an if-modified-since authentication request is not served.
- The content cannot be validated and is delivered, if available, regardless of whether it is fresh or stale.
- A 503 Service Unavailable error is returned to the client if an object is not available.
- Only the HTTP proxy is affected.

## Examples

The following example enables offline operation on a Content Engine:

```
ContentEngine(config)# offline-operation enable
```

The following example disables offline operation on a Content Engine:

```
ContentEngine(config)# no offline-operation enable
```

# pgmrategen

To start the pgmrategen application, which sends PGM packets to the specified multicast address, use the **pgmrategen** EXEC command.

**pgmrategen** *ip_address ttl_time*

## Syntax Description

| | |
|---|---|
| *ip_address* | Multicast IP address where PGM packets are sent. |
| *ttl_time* | Time To Live for multicast packet (1–255). |

## Defaults

No default behavior or values

## Command Modes

EXEC

## Usage Guidelines

The **pgmrategen** *ipaddress ttl_time* EXEC command continuously sends PGM packets onto the specified multicast IP address as a background application. The TTL value for each packet can vary between 1 and 255, depending on the number of network elements that the packet must pass through before reaching the multicast address. The **pgmrategen** command displays the percentage of packets that have been multicast. When the percentage reaches 100, the sending of data packets stops. Because the PGM sender application runs in the background, once the progress reaches 100 percent, the application displays “Stopping pgmrategen .....” in the CLI and returns to the EXEC prompt.

To interrupt the **pgmrategen** command and return to the EXEC prompt, press **Ctrl-C**.

## Examples

The following example shows the output of the **pgmrategen** command to a multicast IP address of 239.2.2.5 and with a TTL of 255:

```
ContentEngine# pgmrategen 239.2.2.5 255
Starting pgmrategen ....

Press ^C to abort PGM Packet Generation...

Sending 1024 messages of 1420 bytes (1420 Kbytes)
PGM rate is 1024 Kbps
Progress: 5%

Progress: 99%
Total time 65.4611 seconds, 21.6923 KBps, 177.703 Kbps

PGM sender 15027:100:0a0101150000 going away, data 1024->1454080, rx 5->7100/5->7100
CPU Usage: 0.120 seconds = 0.110 user + 0.010 sys, 0.183% CPU
Stopping pgmrategen ....
ContentEngine#
```

**Related Commands**

**multicast connectivity-test receive**
**multicast connectivity-test send**
**pgmratemon**

# pgmratemon

To start the pgmratemon application, which receives PGM packets at the specified multicast address, use the **pgmratemon** EXEC command.

**pgmratemon** *ip_address*

## Syntax Description

| | |
|---|---|
| *ip_address* | Multicast IP address where PGM packets are received. |

## Defaults

No default behavior or values

## Command Modes

EXEC

## Usage Guidelines

The **pgmratemon** *ip_address* EXEC command starts the pgmratemon (PGM receiver) application in the background. This application listens for PGM multicast data transmitted from a PGM sender onto the specified multicast IP address. For testing the multicast connectivity between two Content Engines, the same multicast IP address must be used both for sending and for receiving data packets. Whenever the PGM receiver application receives a packet, it prints a line of output containing the packet size and bandwidth. The pgmratemon application terminates by itself if it does not receive any PGM packets for 3 minutes.

To interrupt the **pgmratemon** command and return to the EXEC prompt, press **Ctrl-C**.

## Examples

The following example shows the output of the **pgmratemon** command:

```
ContentEngine# pgmratemon 239.2.2.5
Starting pgmratemon ....
This CE is not configured as Multicast receiver in any cloud
Configuring this CE as Satellite mode receiver

Press ^C to abort or wait for 3 mins to exit....

Listening for PGM multicast data from a SmartPGM sender
on multicast address 224.2.2.5.

Press Ctrl-C to abort. Or wait for 3 minutes to exit.

Waiting for PGM multicast data.

[15027:100:0a0101150000], [15027:100:0a0101150000] 2.27 secs, 3.667 KBps, 30.04 Kbps
[15027:100:0a0101150000] 1.03 secs, 6.724 KBps, 55.08 Kbps
[15027:100:0a0101150000] 1.02 secs, 8.169 KBps, 66.92 Kbps
[15027:100:0a0101150000] 1 secs, 30.49 KBps, 249.8 Kbps
[15027:100:0a0101150000] 1 secs, 61.01 KBps, 499.8 Kbps
[15027:100:0a0101150000] 1 secs, 90.09 KBps, 738 Kbps
[15027:100:0a0101150000] 1 secs, 116.5 KBps, 954.2 Kbps
[15027:100:0a0101150000] 1 secs, 122 KBps, 999.6 Kbps
[15027:100:0a0101150000] 1 secs, 124.8 KBps, 1.022 Mbps
[15027:100:0a0101150000] 1 secs, 123.4 KBps, 1.011 Mbps
[15027:100:0a0101150000] 1 secs, 119.3 KBps, 976.9 Kbps
```

```
[15027:100:0a0101150000] 1 secs, 120.6 KBps, 988.3 Kbps
[15027:100:0a0101150000] 1 secs, 119.3 KBps, 976.9 Kbps
Reaping inactive session [15027:100:0a0101150000]

Stopping pgmratemon
ContentEngine#
```

**Related Commands**

**multicast connectivity-test receive**
**multicast connectivity-test send**
**pgmrategen**

# ping

To send echo packets for diagnosing basic network connectivity on networks, use the **ping** EXEC command.

**ping** {*hostname* | *ip-address*}

## Syntax Description

| | |
|---|---|
| *hostname* | Hostname 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 that 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**.

Following are sample results of the **ping** command:

- Normal response—The normal response occurs in 1 to 10 seconds, depending on network traffic.
- Destination does not respond—If the host does not respond, a “no answer from host” appears in 10 seconds.
- Destination unreachable—The gateway for this destination indicates that the destination is unreachable.
- Network or host unreachable—The Content Engine found no corresponding entry in the route table.

## Examples

The following example shows how to test the basic network connectivity with a host:

```
ContentEngine# ping 172.19.131.189
PING 172.19.131.189 (172.19.131.189) from 10.1.1.21 : 56(84) bytes of
data.
64 bytes from 172.19.131.189: icmp_seq=0 ttl=249 time=613 usec
64 bytes from 172.19.131.189: icmp_seq=1 ttl=249 time=485 usec
64 bytes from 172.19.131.189: icmp_seq=2 ttl=249 time=494 usec
64 bytes from 172.19.131.189: icmp_seq=3 ttl=249 time=510 usec
64 bytes from 172.19.131.189: icmp_seq=4 ttl=249 time=493 usec

--- 172.19.131.189 ping statistics ---
5 packets transmitted, 5 packets received, 0% packet loss
round-trip min/avg/max/mdev = 0.485/0.519/0.613/0.047 ms
ContentEngine#
```

## Related Commands

**multicast connectivity-test**

# port-channel

To configure the port-channel load-balancing options, use the **port-channel** global configuration command. To disable load-balancing options, use the **no** form of this command.

**port-channel load-balance** {**dst-ip** | **dst-mac** | **round-robin**}

**no port-channel load-balance**

## Syntax Description

| | |
|---|---|
| **load-balance** | Configures the load-balancing method. |
| **dst-ip** | Specifies the load-balancing method using destination IP addresses. |
| **dst-mac** | Specifies the load-balancing method using destination Media Access Control (MAC) addresses. |
| **round-robin** | Specifies the load-balancing method using round-robin sequential, cyclical resource allocation. |

## Defaults

Round-robin is the default load-balancing method.

## Command Modes

global configuration

## Usage Guidelines

The **port-channel load-balance** command configures one of three load-balancing algorithms and provides flexibility in choosing interfaces when an Ethernet frame is sent. The **round-robin** option allows evenly balanced usage of identical network interfaces in a channel group. Because this command takes effect globally, if two channel groups are configured, they must use the same load-balancing option.

Because the ACNS software normally initiates IP packets or Ethernet frames, it does not support hashing based on the source IP address and source MAC address. The ACNS 5.x software adds the round-robin option, which is the default load-balancing algorithm, to evenly distribute traffic among several identical network interfaces.

## Examples

The following example shows how to configure the round-robin load-balancing method on a Content Engine:

```
ContentEngine(config)# port-channel load-balance round-robin
```

## Related Commands

**interface**

# pre-load

To configure the Content Engine to fetch and preload the content, use the **pre-load** global configuration command. To disable individual options, use the **no** form of this command.

**pre-load concurrent-requests** *number*

**pre-load depth-level-default** *level_number*

**pre-load dscp** {**set-dscp** *dscp-packets* | **set-tos** *tos-packets*}

**pre-load enable**

**pre-load fetch** {**directory** *dir_names* | **domain** *domain_names* | **suffix** *suffix_names*}

**pre-load max-bandwidth** *bandwidth*

**pre-load no-fetch** {**directory** *dir_names* | **domain** *domain_names* | **suffix** *suffix_names*}

**pre-load resume**

**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** | **depth-level-default** | **dscp** {**set-dscp** *dscp-packets* | **set-tos** *tos-packets*} | **enable** | **fetch** {**directory** *dir_names* | **domain** *domain_names* | **suffix** *suffix_names*} | **max-bandwidth** | **no-fetch** {**directory** *dir_names* | **domain** *domain_names* | **suffix** *suffix_names*} | **resume** | **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**}

## Syntax Description

| | |
|---|---|
| **concurrent-requests** | Configures the maximum number of concurrent requests. |
| *number* | Number of concurrent requests (1–30). The default is 10. |
| **depth-level-default** | Configures the default depth level. |
| *level_number* | Depth level of URL download (0–20). The default is 3.<br>**Note** Setting the depth level default to 0 is useful if you have specified URLs in preload.txt files and you do not want the Content Engine to try to preload other URLs. |
| **set-dscp** | Configures differentiated services code point (DSCP) values. |
| *dscp-packets* | DSCP values; see Table 2-16 on page 2-371 for valid values. |
| **set-tos** | Configures the Type of Service (ToS) values. |
| *tos-packets* | ToS value; see Table 2-17 on page 2-372 for valid values. |
| **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. |
| **max-bandwidth** | Configures the maximum bandwidth in kbps. |
| *bandwidth* | Maximum bandwidth for the preload process (1–1000000). The default is no limitation. |
| **no-fetch** | Configures the filter for the objects that should not be fetched. |
| **directory** | Configures the directories to be excluded. |
| *dir_names* | List of directory names to be excluded, separated by spaces. |
| **domain** | Configures the domains to be excluded. |
| *domain_names* | List of domain names to be excluded, separated by spaces. |
| **suffix** | Configures the suffixes to be excluded. |
| *suffix_names* | List of suffixes to be excluded, separated by spaces. |
| **resume** | Continues from the previous preload operation. |
| **schedule** | Configures the schedule time for preload. |
| **every-day** | Configures the preloads in intervals of 1 day. |
| **start-time** | (Optional) Sets the 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 the preload end time. The default is the time until all objects have been downloaded. |
| *time* | (Optional) Time of day to end the preload (00:00–23:59 in hh:mm format). |
| **every-hour** | Configures the preloads in intervals of 1 hour or less. |
| **start-time** | (Optional) Sets the preload start time. The default is 0. |
| *time* | (Optional) Minute of the hour to start the preload (0–59). |
| **end-time** | (Optional) Sets the preload end time. The default is the time until all objects have been downloaded. |
| *time* | (Optional) Minute of the hour to end the preload (0–59). |
| **every-week** | Configures the preloads in intervals of 1 week or less. |
| *days of week* | Adds one or more weekdays.<br>**Fri** Every Friday.<br>**Mon** Every Monday.<br>**Sat** Every Saturday.<br>**Sun** Every Sunday.<br>**Thu** Every Thursday.<br>**Tue** Every Tuesday.<br>**Wed** Every Wednesday. |
| **start-time** | (Optional) Sets the 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 the preload end time. The default is the time until all objects have been downloaded. |
| *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 the URL list file path. |
| *path* | Path of the file containing the URL list or a URL. |

## Defaults

**concurrent-requests** *number*: 10

**depth-level-default** *level_number*: 3

**max-bandwidth**: no limitation

**every-day**: default

**every-day start-time** *time*: 00:00

**every-hour start-time** *time*: 0

**every-week 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

The preloaded content is the content that is retrieved and stored on a Content Engine because the administrator of that Content Engine scheduled a retrieval of specific content in anticipation of user requests for that content. You can initiate the content preloading by configuring the Content Engine to create a cache request for all the content located at the origin web server that stores the primary content.

You can specify bandwidth limits for the preload process to ensure that the bandwidth consumption does not exceed the specified bandwidth limits during the preload process. During the preload process, the Content Engine scans websites several link levels down for content, retrieves the specified content, and stores it locally for future requests. At a specified time, the Content Engine scans several levels of websites to verify that its content is still current, and it updates any content that has changed.

The ACNS 5.x software can read a file of URLs and preload the specified URL content on the Content Engine. The content that can be preloaded on a Content Engine includes HTTP URLs and FTP-over-HTTP URLs. This URL list is referred to as the preload URL list file.

**Note** All configured HTTP and FTP-over-HTTP parameters and rules apply to the preloaded objects.

The ACNS 5.1.1 software and later releases support the preloading of NTLM authenticated objects. This feature allows NTLM authenticated objects (authenticated objects that reside on the servers that authenticate NTLM only) to be preloaded on a Content Engine.

An entry in a URL list file has the following format:

*URL* [*depth*] [*domain-name*:*host-name*:*host-domain-name*]

where

*hostname* and *host-domain-name* can be null. However, the *domain name* is required if NTLM credentials have been configured. (The separator is required.)

```
http//www.cisco.com 3 apac::
```

If NTLM-related information is not present in the preload URL list file entry, the authentication scheme reverts to basic authentication.

By default, the Content Engine does not cache basic and NTLM authenticated objects. To enable a Content Engine to fetch specific objects and cache these objects that are authenticated with any authentication scheme (basic authentication or NTLM authentication), enter the **http cache-authenticated all** global configuration command.

To configure the Content Engine to cache only NTLM authenticated objects, enter the **http cache-authenticated ntlm** global configuration command. The cached objects are tagged as "NTLM protected" so that subsequent requests for these same objects are subjected to authentication before the Content Engine can serve the content to the client.

Before you preload WMT streaming media files on the Content Engine, you must enable WMT on your Content Engine. If you used the Setup utility to configure WMT caching on the Content Engine, then WMT is already enabled on the Content Engine. Otherwise, see the "Enabling WMT on the Content Engine" section for instructions on how to use the Content Engine CLI (instead of the Setup utility) to enable Windows Media services on a Content Engine before you enable the preloading of Windows Media streaming files for this Content Engine.

The preloading feature is enabled using the **pre-load enable** command. The content preloading can then be scheduled with the **pre-load schedule** command or triggered immediately with the **pre-load force** EXEC command.

You can configure the maximum number of preload processes to run at the same time using the **pre-load concurrent-requests** option. 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.

#### Creating a Preload URL List File

The preload URL list file lists the URLs (HTTP and FTP-over-HTTP) to be preloaded on the Content Engine. This file is maintained by the administrator and must be created on a remote system. This file can be transferred to the Content Engine for preloading access, or the file can be accessed from the remote server.

You can specify the path of this file by using the **pre-load url-list-file path** global configuration command. You can access this list with a frequency by using the **pre-load schedule** command.

**Note** In the **pre-load url-list-file** *path* global configuration command, the value for *path* can be a URL or a local file path.

You can place the list of URLs in a file on a local disk. You can also use the **mkdir** EXEC command to make a subdirectory that contains the preload URL list file. For instance, the **mkdir /local1/preload-directory** command creates a subdirectory called *preload-directory* on the local disk /local1.

Each URL in the preload URL list file has an optional depth parameter. The depth parameter specifies how many levels down the preloading is performed. For example, entering http://www.espn.com 3 means that you will download http://www.espn.com and all content three levels down. If you do not specify the depth level, then the preload depth level default of 3 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>
```

If you want to preload the authenticated content to a Content Engine, you must write the URL list file entry as follows:

```
http://username:password@www.authenticatedsite.com/ depth level
```

In the ACNS 5.1.5 software and earlier releases, when you configured a preload URL list file through the Content Engine CLI, the **pre-load url-list-file** global configuration command only had the HTTP or FTP option. There was no mechanism in place to fetch the preload URL list file securely.

In the ACNS 5.1.5 software, the ability to fetch the preload URL file over HTTPS was added. If a preload URL list file contains usernames and passwords, organizations can fetch the preload URL list file over HTTPS. The actual preloading of HTTPS links is not supported; only the downloading of the preload URL list file through the HTTPS protocol is supported.

#### Scheduling Content Preloading

To configure the Content Engine to preload the specific content for a future time, use the **pre-load schedule** global configuration command. The Content Engine accesses the specified preload URL list file with a frequency set by the specified preloading schedule (set through the **pre-load schedule** global configuration command).

The default start time for the preloading operation is 00:00 (the start of the day). If you do not specify the end time, the preload operation is completed after all the objects have been downloaded. If you wish to change this default, follow these guidelines:

a. To specify the start and end times for daily or weekly preloads, use *hh*:*mm* (where *hh* indicates the hour and *mm* indicates the minutes; an example is 01:00). For hourly preloads, use *mm* to specify the start and end times.

b. To specify the start time and end times for hourly preloads, the start time should be 0 and the end time should be 59. For daily and weekly preloads, the start time should be from 0 to 23, and the end time should be from 0 to 59. If you do not specify the end-time option, the preload operation continues until it is completed.

   To configure a preload on more than one day of the week, use the **pre-load schedule every-week** global configuration command.

#### Resuming Content Preloading

If content preloading is not completed before the scheduled end time, you can resume the preloading process to capture the intended content using the **pre-load resume** global configuration command. Using this command allows you to resume downloading from the breakpoint of the previous preload, instead of starting again from the very beginning of the URL list file.

**Note** If you do not enter the **pre-load resume** command on the Content Engine and you abort the content preloading before the scheduled end time, the next scheduled content preloading starts from the beginning of the URL list file.

### Bandwidth Control

You can configure a maximum bandwidth for the preloading process using the **pre-load max-bandwidth** command. Previous versions of the ACNS software did not allow for bandwidth control, so the user had no way to ensure that bandwidth consumption during preloading did not exceed the user-specified bandwidth limits. The ACNS 5.x software also allows for preloading of WMT streaming media files that may have different bit rates. You can also control WMT bandwidth using the **bandwidth wmt outgoing** and **bandwidth incoming** global configuration commands.

### ToS and DSCP

Setting the Type of Service (ToS) or differentiated services code point (DSCP) is called packet marking, which allows you to partition network data into multiple priority levels or types of service. You can 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.

The ACNS 5.x software includes ToS or DSCP support for HTTP and FTP, preload traffic. Because content preloading is initiated by the Content Engine and not by the requesting client when a connection is made to an origin server, ToS or DSCP on the traffic going toward the server must be set before contact is made with the origin server. Use the **pre-load dscp** command to set the ToS value as well as the DSCP code point for all preload traffic.

**Note** Using the **pre-load dscp** command takes precedence over any use of the Rules Template configuration commands involving DSCP server configurations.

Valid values for *dscp-packets* are listed in Table 2-16.

***Table 2-16 dscp-packets Values***

| Value or Keyword | Description[1] |
|---|---|
| 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). |

*Table 2-16 dscp-packets Values (continued)*

| Value or Keyword | Description[1] |
|---|---|
| **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 the default DSCP (000000). |
| **ef** | Sets packets with EF DSCP (101110). |

1. The number in parentheses denotes the DSCP value for each per-hop behavior keyword.

Valid values for *tos-packets* are listed in Table 2-17.

*Table 2-17 tos-packets Values*

| Value, Precedence, or ToS Name | Description[1] |
|---|---|
| 0–127 | Sets the 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 maximum reliable ToS (2). |
| **max-throughput** | Sets packets with maximum throughput ToS (4). |
| **min-delay** | Sets packets with minimum delay ToS (8). |
| **min-monetary-cost** | Sets packets with minimum 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). |

1. The number in parentheses denotes the ToS value for each IP precedence or ToS name setting.

## Examples

The following example enables the preload feature:

```
ContentEngine(config)# pre-load enable
```

The following example specifies the local pathname of the preload URL list file:

```
ContentEngine(config)# pre-load url-list-file /local1/myurllist
```

The following example specifies the FTP server from which a URL list file is accessed:

```
ContentEngine(config)# pre-load url-list-file ftp://ftpserver/ftpdirectory/urllist.txt
```

The following example specifies the depth level for URL retrieval at 4:

```
ContentEngine(config)# pre-load depth-level-default 4
```

The following example creates a filter for the objects to be excluded:

```
ContentEngine(config)# pre-load no-fetch suffix .mil .su .ca
```

The following example specifies a filter for the domain to be fetched:

```
ContentEngine(config)# pre-load fetch domain cisco.com
```

The following example specifies that other domains in an HTML page should be traversed (by default, other domains in an HTML page are not traversed):

```
ContentEngine(config)# pre-load traverse-other-domains
```

The following example specifies the maximum number of concurrent connections:

```
ContentEngine(config)# pre-load concurrent-requests 5
```

The following example specifies a daily interval for scheduling the preload:

```
ContentEngine(config)# pre-load schedule every-day start-time 01:00 end-time 02:00
```

The following example specifies an hourly interval for scheduling the preload:

```
ContentEngine(config)# pre-load schedule every-hour start-time 8 end-time 20
```

The **pre-load schedule every-week** option permits configuring a preload on more than one day of the week. The following example specifies a biweekly interval for scheduling the preload:

```
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 (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 example allows the Content Engine to resume the content preloading from the URL where content preloading was terminated:

```
ContentEngine(config)# pre-load resume
```

The following examples show the preload-related **show** commands. The first example shows the statistics of the preloading operation while content preloading is taking place.

```
ContentEngine# show statistics pre-load
Statistics of last Preloading operation
---------------------------------------

Preloading is in progress.
List of preloaded URLs are in /local1/preload_dir/downloaded_urls.

1811 objects downloaded, 35506616 bytes transferred.
```

The following example shows the statistics of the preloading operation after the content preloading process is disabled using the **no pre-load enable** command. The **show statistics pre-load** command is also used to show the statistics of the successful content preloading operation.

```
ContentEngine# show statistics pre-load
Statistics of last Preloading operation
---------------------------------------
```

```
Preloading was initiated by force.
Preloading started at Thu Jun 20 06:59:54 2002
Preloading ended   at Thu Jun 20 06:59:55 2002
List of preloaded URLs are in /local1/preload_dir/latest_preloaded_objects.
Preload errlog is /local1/preload_dir/latest_preload_error.

Number of invalid entries in URL list file  =          0
Total number of preloaded objects           =          67
Total number of preloaded bytes             =          28356
```

The following example shows how to set the ToS support to normal:

```
ContentEngine(config)# pre-load set-tos normal
```

The following example shows how to set the maximum bandwidth for content preloading at 50000 kbps:

```
ContentEngine(config)# pre-load max-bandwidth 50000
```

The following example shows the status of the current preloading operation after the **pre-load set-tos** and **pre-load max-bandwidth** commands have been used:

```
ContentEngine# show pre-load
Preloading is enabled
Number of concurrent sessions: 10
Depth level: 4
URL List File: /local1/url.txt
DSCP: set-tos normal
Max Bandwidth: 50000 Kbps
Previous preloading operation will be continued.
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
```

## 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** EXEC 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

The following 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 ACNS network, use the **primary-interface** global configuration command. Use the **no** form of the command to remove the configured primary interface.

**primary-interface** {**FastEthernet** *0-3*/*port* | **GigabitEthernet** *1-2*/*port* | **PortChannel** *1-2* | **Standby** *group_num*}

**no primary-interface** {**FastEthernet** *0-3*/*port* | **GigabitEthernet** *1-2*/*port* | **PortChannel** *1-2* | **Standby** *group_num*}

## Syntax Description

| | |
|---|---|
| **FastEthernet** | Selects a Fast Ethernet interface as the ACNS network primary interface. |
| *0-3/* | Fast Ethernet slot numbers 0, 1, 2, or 3. |
| *port* | Port number of the Fast Ethernet interface. |
| **GigabitEthernet** | Selects a Gigabit Ethernet interface as the ACNS network primary interface. |
| *1-2/* | Gigabit Ethernet slot numbers 1 or 2. |
| *port* | Port number of the Gigabit Ethernet interface. |
| **PortChannel** | Selects a port-channel interface as the ACNS network primary interface. |
| *1-2* | Port channel number 1 or 2. |
| **Standby** | Selects a standby group as the ACNS network primary interface. |
| *group_num* | Standby group number. |

## 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 and port-channel interfaces. Primary interface configuration is required for the proper functioning of Centralized Management System (CMS). After devices are registered to the Content Distribution Manager, the Content Distribution Manager uses the configured primary interface to communicated with the registered devices.

You cannot enable the ACNS network without specifying the primary interface. Also, you must have chosen the primary interface before you enable the CMS. The primary interface can be changed without disabling the ACNS network. The primary interface specifies the default route for an interface. To change the primary interface, choose a different interface as the primary interface.

In the ACNS 5.2 software and later releases, you can select a standby interface as the primary interface (you can enter the **primary-interface Standby** *group_num* command) to specify a standby group as the primary interface on a Content Engine.

## Command Modes

global configuration

## Usage Guidelines

The **primary-interface** global configuration command allows the administrator to specify the primary interface for the ACNS network.

The primary interface can be changed without disabling the ACNS network. To change the primary interface, reenter the command string and specify a different interface.

**Note** If you use the **restore factory-default preserve basic-config** command, the configuration for the primary interface is not preserved. On an ACNS 5.x device, if you want to reenable the ACNS network after using the **restore factory-default preserve basic-config** command**,** make sure to reconfigure the primary interface after the factory defaults are restored.

## Examples

The following example shows how to specify the Fast Ethernet slot 0 port 0 as the primary interface on a Content Engine model CE-7320:

```
CE-7320(config)# primary-interface FastEthernet 0/0
```

The following example shows how to specify the Fast Ethernet slot 0 port 1 as the primary interface on a Content Engine:

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

# proxy-auto-config (EXEC)

To download the proxy autoconfiguration (PAC) file from an FTP server, use the **proxy-auto-config** 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* | Hostname of the FTP server. |
| *ftp-ip-address* | IP address of the FTP server. |
| *remotedir* | Directory on the FTP server where the .pac file is located. |
| *pacfile* | Filename of the remote PAC file. |

## Defaults

No default behavior or values

## Command Modes

EXEC

## Usage Guidelines

A browser obtains proxy IP address and port configuration information from the 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.

**Note** You must configure disks /local1 or /local2 as a sysfs volume before downloading the autoconfiguration file to either of these two disk locations.

The Microsoft Internet Explorer and Netscape browsers support the proxy autoconfiguration feature. You must manually configure the browser for automatic proxy configuration.

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

The following example shows how to download an autoconfiguration file from an FTP server to the Content Engine:

```
ContentEngine# proxy-auto-config download 172.16.10.10 remotedirname proxy.pac
```

The following example shows the URL that you enter in the browser's automatic proxy configuration URL field:

```
http://CCNScache-ipaddress:portnumber/proxy.pac
```

**Note** Use a port number specified by the proxy incoming settings for configuring proxy incoming ports. For instance, if port 8080 is specified, then use 8080 as your port number in the example shown.

**Related Commands** **proxy-auto-config** (global configuration mode)
**show proxy-auto-config**

# proxy-auto-config (global configuration)

To enable the proxy autoconfiguration (PAC) feature, use the **proxy-auto-config** global configuration command. 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

Proxy autoconfiguration allows the web browser to obtain proxy information from a special script stored on the server. The browser obtains proxy IP address and port configuration information from the 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 PAC file feature, enter the **proxy-auto-config enable** global configuration command. Each time that you download a new autoconfiguration file to the Content Engine, enter the **no proxy-auto-config enable** and then a **proxy-auto-config enable** command.

PAC is supported by the Microsoft Internet Explorer and Netscape Communicator browsers. The browser must be manually configured for automatic proxy configuration.

## Examples

The following example shows how browser autoconfiguration is enabled on the Content Engine:

```
ContentEngine(config)# proxy-auto-config enable
```

The following example shows the URL that you enter in the browser's 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, hostname, 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** | **reset**}

**no proxy-protocols** {**outgoing-proxy exclude** {**enable** | **list** *word*} | **transparent**}

## Syntax Description

| | |
|---|---|
| **outgoing-proxy exclude** | Sets the global outgoing proxy exclude criteria. |
| **enable** | Enables global outgoing proxy exceptions. |
| **list** | Sets the global outgoing proxy exclude list. |
| *word* | Domain names, hostnames, 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. |
| **reset** | Resets the incoming connection. |

## Defaults

No default behavior or values

## Command Modes

global configuration

## Usage Guidelines

The **proxy-protocols outgoing-proxy exclude list** option allows the administrator to specify a single domain name, hostname, or IP address to be globally excluded from proxy forwarding. The domain name is entered as an ASCII string. The wildcard character asterisk (*) can be used for an IP address (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.

The **proxy-protocols transparent reset** option resets the incoming connection. The requested objects are not delivered.

**Note** The ACNS 5.5 software and later versions support MMS-over-HTTP only. You can designate an outgoing HTTP proxy server for streaming media in the MMS format by using the **wmt proxy outgoing http host** command to allow the forwarding of MMS data over HTTP to a standard 8080 proxy port.

## 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

The following example shows how to view the present working directory:

```
ContentEngine# pwd
/local1
```

## Related Commands

**cd**
**dir**
**lls**
**ls**

# radius-server

To configure RADIUS authentication parameters, use the r**adius-server** global configuration command. 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* **location** *url*} | **retransmit** *retries* | **timeout** *seconds*}

**no radius-server** {**enable** | **host** {*hostname* | *hostipaddr*} | **key** | **redirect** {**enable** | **message** *reply* **location** *url*} | **retransmit** | **timeout**}

## Syntax Description

| | |
|---|---|
| **enable** | Enables HTTP RADIUS authentication. |
| **host** | Specifies a RADIUS server. |
| *hostname* | Hostname of the RADIUS server. |
| *hostipaddr* | IP address of the RADIUS server. |
| **auth-port** | (Optional) Sets the UDP port for the RADIUS authentication server. |
| *port* | UDP port number (1–65535). The default is 1645. |
| **key** | Specifies the encryption key shared with the RADIUS servers. |
| *keyword* | Text of the shared key (15 characters maximum). |
| **redirect** | Redirects the response if an authentication request fails. |
| **enable** | Enables the redirect feature. |
| **message** | Replies with an authentication failure message. |
| *reply* | Reply message text string (24 characters maximum). |
| **location** | Sets the HTML page location, for example, http://www.cisco.com. |
| *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

**auth-port** *port*: UDP port 1645

**retransmit** *retries*: 2

**timeout** *seconds*: 5

## Command Modes

global configuration

## Usage Guidelines

RADIUS is a client/server authentication and authorization access protocol used by an ACNS network device to authenticate users attempting to connect to a network device. The ACNS network device functions as a client, passing user information to one or more RADIUS servers. The ACNS network

device permits or denies network access to a user based on the response that it receives from one or more RADIUS servers. RADIUS uses the User Datagram Protocol (UDP) for transport between the RADIUS client and server.

You can configure a RADIUS key on the client and server. If you configure a key on the client, it must be the same as the one configured on the RADIUS servers. The RADIUS clients and servers use the key to encrypt all RADIUS packets transmitted. If you do not configure a RADIUS key, packets are not encrypted. The key itself is never transmitted over the network.

**Note** For more information about how the RADIUS protocol operates, see RFC 2138, *Remote Authentication Dial In User Service (RADIUS)*.

RADIUS authentication usually occurs in these instances:

- Administrative login authentication—When an administrator first logs in to the Content Engine to configure the Content Engine for monitoring, configuration, or troubleshooting purposes. For more information, see the "Enabling and Disabling Administrative Login Authentication and Authorization Through RADIUS" section on page 2-385.
- HTTP request authentication—When an end user sends a service request that requires privileged access to content that is served by the Content Engine. For more information, see the "Configuring RADIUS Authentication of HTTP Requests" section on page 2-386.

RADIUS authentication is disabled by default. You can enable RADIUS authentication and other authentication methods at the same time. You can also specify which method to use first.

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.

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 following **rule** command is relevant to RADIUS authentication only if the **redirect** option has been configured.

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.

#### Enabling and Disabling Administrative Login Authentication and Authorization Through RADIUS

When configuring a Content Engine to use RADIUS to authenticate and authorize administrative login requests, follow these guidelines:

- By default, RADIUS authentication and authorization is disabled on a Content Engine.
- Before enabling RADIUS authentication on the Content Engine, you must specify at least one RADIUS server for the Content Engine to use.
- You can enable RADIUS authentication and other authentication methods at the same time. You can specify which method to use first using the **primary** keyword. When local authentication is disabled, if you disable all other authentication methods, local authentication is reenabled automatically.
- You can use the Content Engine GUI or the CLI to enable RADIUS authentication and authorization on a Content Engine.

**Tip** From the Content Engine GUI, choose **System > Authentication**. Use the displayed Authentication Configuration window.

To use the Content Engine CLI to enable RADIUS authentication and authorization on a Content Engine, enable RADIUS authentication for normal login mode by entering the **authentication login radius** global configuration command as follows:

```
ContentEngine(config)# authentication login radius enable [primary] [secondary]
[tertiary]
```

Use the **authentication configuration radius** global configuration command to enable RADIUS authorization as follows:

```
ContentEngine(config)# authentication configuration radius enable [primary] [secondary]
[tertiary]
```

**Note** To disable RADIUS authentication and authorization on a Content Engine, use the **no** form of the **authentication** global configuration command (for example, use the **no authentication login radius enable** command to disable RADIUS authentication).

#### Configuring RADIUS Authentication of HTTP Requests

To configure RADIUS authentication for HTTP requests on a Content Engine, configure the RADIUS server settings on the Content Engine and enable RADIUS authentication for HTTP requests on the Content Engine using the **radius-server** global configuration command.

## 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. You can verify the configuration with the **show radius-server** and **show rule all** commands.

```
ContentEngine(config)# radius-server enable
ContentEngine(config)# radius-server host 172.16.90.121
ContentEngine(config)# radius-server key myradiuskey
ContentEngine(config)# rule action no-auth pattern-list 2
ContentEngine(config)# rule pattern-list 2 domain mydomain.net
ContentEngine(config)# rule enable


ContentEngine# show radius-server
Radius Configuration:
---------------------
Radius Authentication is on
    Timeout       = 5
    Retransmit    = 3
    Key           = ****
    Servers
    -------
    IP 172.16.90.121 Port =   1645    State: ENABLED

ContentEngine# 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:

```
ContentEngine(config)# no radius-server enable
```

The following example shows how to force the Content Engine to try RADIUS authentication first (before using TACACS+ authentication):

```
ContentEngine(config)# authentication login radius enable primary
```

The following example shows how to force the Content Engine to try RADIUS authorization first (before using TACACS+ authorization):

```
ContentEngine(config)# authentication configuration radius enable primary
```

## Related Commands

**debug authentication http-request**
**rule**
**show radius**

# reload

To halt and perform a cold restart on the Content Engine, use the **reload** EXEC command.

**reload** [**force**]

## Syntax Description

| | |
|---|---|
| **force** | (Optional) Forces a reboot without further prompting. |

## Defaults

No default behavior or values

## Command Modes

EXEC

## Usage Guidelines

To reboot the Content Engine, use the **reload** command. If the current running configuration is different from the startup configuration and if the configuration changes are not saved to flash memory, you are prompted to save the current running configuration parameters to the startup configuration. If WCCP is not enabled on the Content Engine, any open connections are dropped after you enter this command. If WCCP is enabled on the Content Engine, the Content Engine performs a proper shutdown of WCCP after you enter the **reload** command to prevent broken TCP connections. The Content Engine does not reboot until either all connections have been serviced or the maximum wait time (specified with the **wccp shutdown max-wait** command [by default, 120 seconds]) has elapsed for WCCP Version 2.

During a proper shutdown of WCCP, 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 properly shut down.

You cannot shut down an individual WCCP service on a particular port (for example, you cannot shut down the reverse proxy service on port 80) on a Content Engine; you must shut down WCCP on the Content Engine. After WCCP is shut down on the Content Engine, the Content Engine still preserves its WCCP configuration settings and still services proxy-style requests (for example, HTTP requests that the Content Engine receives directly from a client browser).

To save any file system contents to disk from memory before a restart, use the **cache synchronize** command.

## Examples

The following example reloads the Content Engine after you have saved the configuration changes. The Content Engine waits for the configured period of time before it shuts down WCCP Version 2. A countdown message appears, indicating how many seconds remain before WCCP will be shut down on the Content Engine.

```
CONTENTENGINE# reload
System configuration has been modified. Save?[yes]:yes
Proceed with reload?[confirm]yes
Proceed with clean WCCP shutdown?[confirm]yes
Waiting (1 seconds) for WCCP shutdown. Press ^C to skip shutdownn
WCCP clean shutdown wait time exceeded
Shutting down all services, will timeout in 15 minutes.
reload in progress .....
```

The following example forces a reboot on the Content Engine:

```
ContentEngine# reload force
```

## Related Commands

**cache synchronize**
**write**
**write erase**

# rename

To rename a file on the 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

The following example renames a file named errlog.txt as old_errlog.txt:

```
ContentEngine# rename errlog.txt old_errlog.txt
```

## Related Commands

**cpfile**

# restore

To restore the device to its manufactured default status, removing the user data from the disk and flash memory, use the **restore** EXEC command. This command erases all existing content on the device; however, your network settings are preserved and the device is accessible through Telnet and Secure Shell (SSH) after it reboots.

**restore factory-default** [**preserve basic-config**]

## Syntax Description

| | |
|---|---|
| **factory-default** | Resets the device configuration and data to their manufactured default status. |
| **preserve** | (Optional) Preserves certain configurations and data on the device. |
| **basic-config** | Selects basic network configurations. |

## 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. You need to enter this command from the root directory, or else the following error message is displayed:

```
CONTENTENGINE# restore factory-default

Need to cd to / before issuing this command

Command aborted.
CONTENTENGINE#
```

Be sure to back up the Content Distribution Manager database and copy the backup file to a safe location that is separate from that of the Content Distribution Manager, or change over from the primary to a standby Content Distribution Manager before you use the **restore factory-default** command on your primary Content Distribution Manager. The primary Content Distribution Manager operation must be halted before proceeding with backup and restore commands.

**Caution** This command erases user-specified configuration information stored in the flash image and removes the data on the disk, the user-defined disk partitions, and the entire Content Distribution Manager database. User-defined disk partitions that are removed include the sysfs, cfs, mediafs, and cdnfs partitions. The configuration being removed includes the starting configuration of the device.

By removing the Content Distribution Manager database, all configuration records for the entire ACNS network are deleted. If you do not have a valid backup file or a standby Content Distribution Manager, you must use the **cms deregister force** command and reregister every Content Engine and Content Router after you have reconfigured the Content Distribution Manager, because all previously configured data is lost.

If you used your standby Content Distribution Manager to store the database while you reconfigured the primary, you can simply register the former primary as a new standby Content Distribution Manager.

If you created a backup file while you configured the primary Content Distribution Manager, you can copy the backup file to this newly reconfigured Content Distribution Manager and use the **cms database restore** command.

**Caution** If you upgraded your software after you received your software recovery CD-ROM, using the CD-ROM software images may downgrade your system.

Cisco ACNS software consists of three basic components:

- Disk-based software
- Flash-based software
- Hardware platform cookie (stored in flash memory)

All of these components must be correctly installed for Cisco ACNS software to work properly.

**Note** For information on the types of software images provided and the options available from the software recovery CD-ROM installer menu, see Chapter 9 of the *Cisco ACNS Software Upgrade and Maintenance Guide*.

## Examples

The following two examples show the results of using the **restore factory-default** and **restore factory-default preserve basic-config** commands. Because configuration parameters and data are lost, prompts are given before initiating the restore operation to ensure that you want to proceed.

**Note** If you use the **restore factory-default preserve basic-config** command, the configuration for the primary interface is not preserved. On a device running the ACNS 5.x software, if you want to reenable the ACNS network after using the **restore factory-default preserve basic-config** command, make sure to reconfigure the primary interface after the factory defaults have been restored.

```
ContentDistributionManager# restore factory-default
This command will wipe out all of data on the disks
and wipe out ACNS CLI configurations you have ever made.
If the box is in evaluation period of certain product,
the evaluation process will not be affected though.

It is highly recommended that you stop all active services
before this command is run.

Are you sure you want to go ahead?[yes/no]

ContentDistributionManager# restore factory-default preserve basic-config
This command will wipe out all of data on the disks
and all of ACNS CLI configurations except basic network
configurations for keeping the device online.
The to-be-preserved configurations are network interfaces,
default gateway, domain name, name server and hostname.
If the box is in evaluation period of certain product,
the evaluation process will not be affected.
```

```
It is highly recommended that you stop all active services
before this command is run.

Are you sure you want to go ahead?[yes/no]
```

**Note** You can enter basic configuration parameters (such as the IP address, hostname, and name server) at this point or later through entries in the command-line interface.

The following example shows that entering the **show disk** command after the **restore** command verifies that the **restore** command has removed data from the partitioned file systems (sysfs, cfs, mediafs, and cdnfs):

```
ContentEngine# show disk

SYSFS         0.0GB         0.0%
CFS           0.0GB         0.0%
MEDIAFS       0.0GB         0.0%
CDNFS        0.0GB          0.0%
FREE         29.9GB       100.0%
```

Because flash memory configurations were removed after the **restore** command was used, the **show startup-config** command does not return any flash memory data. The **show running-config** command returns the default running configurations.

The **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

**cms database backup**
**cms database restore**
**show disks**
**show rtsp server real-subscriber**
**show running-config**
**show startup-config**
**show wmt**

# rmdir

To delete a directory, use the **rmdir** EXEC command.

**rmdir** *directory*

## Syntax Description

| | |
|---|---|
| *directory* | Name of the directory that 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 removes only empty directories.

## Examples

The following example removes the oldfiles directory under /local1:

```
ContentEngine# rmdir /local1/oldfiles
```

## Related Commands

**lls**
**ls**
**mkdir**

# rtsp (EXEC)

To restore the RealProxy or RealSubscriber to its default configuration, use the **rtsp real-proxy default-configuration** or **rtsp real-subscriber default-configuration** EXEC command. RealProxy and RealSubscriber each use an XML-based configuration file. Changes made from the RealProxy or RealSubscriber GUI are saved to the respective configuration file.

**rtsp** {**real-proxy restore factory-default** | **real-subscriber restore factory-default**}

## Syntax Description

| | |
|---|---|
| **real-proxy** | Restores the RealProxy configuration to the defaults. |
| **restore factory-default** | Restores RealProxy or RealSubscriber configuration files and databases to the factory default. |
| **real-subscriber** | Restores the RealSubscriber configuration to the defaults. |

## Defaults

No default behavior or values

## Command Modes

EXEC

## Usage Guidelines

Restoring the RTSP RealProxy or RealSubscriber to its default configuration will overwrite its current RTSP configuration. Attempting to enter either of these commands generates a warning message to inform you that current configurations will be lost if you proceed.

## Examples

The following example restores the RealProxy configuration to the defaults:

```
ContentEngine# rtsp real-proxy restore factory-default
User would lose the current real proxy configuration. Do you want to proceed? yes
Restart RealProxy to load the factory defaults and configuration.
```

The following example restores the RealSubscriber configuration to the defaults:

```
ContentEngine# rtsp real-subscriber restore factory-default
User would lose the current real subscriber configuration. Do you want to proceed? yes
Restart Real Subscriber to load the factory defaults and configuration.
```

## Related Commands

**show rtsp**