rtsp (global configuration)

To configure the Content Engine to receive and send Real-Time Streaming Protocol (RTSP) broadcasts, use the **rtsp** global configuration command. To disable individual options, use the **no** form of this command.

rtsp advanced {**bypass-gateway** {**cisco-streaming-engine** | **media-real** | **real-subscriber**} **max-initial-setup-delay** *time_delay* | **max-request-rate** *number*}

rtsp {**ip-address** *ip-address* | **l4-switch enable** | **port incoming** *port* | **proxy media-real** {**accept-license-agreement** | **enable** | **evaluate** | **license-key** *keyword*}}

rtsp server real-subscriber {**accept-license-agreement** | **enable** | **evaluate** | **license-key** *keyword*}

rtsp server cisco-streaming-engine {**broadcast** {**id** *id* **source** {**source-rtsp** *rtsp-url* | **source-udp** *url srce-ip-addr recv-ip-addr recv-port-list-num*} **track-count** *1-4* **destination** {**destination-pull** | **destination-udp** *ttl dest-ip-addr dest-list-num*} | **port-list** *list-num port-num* [*port-num*]} | **enable**}

no rtsp {**advanced** {**bypass-gateway** {**cisco-streaming-engine** | **media-real** | **real-subscriber**} | **max-initial-setup-delay** *time_delay* | **max-request-rate** *number*} | **ip-address** | **l4-switch enable** | **port incoming** *port* | **proxy media-real** {**accept-license-agreement** | **enable** | **evaluate** | **license-key** *keyword*} | **server** {**real-subscriber** {**accept-licence-agreement** | **enable** | **evaluate** | **license-key**}} | **cisco-streaming-engine** {**broadcast** {**id** *id* **source** {**source-rtsp** *rtsp-url* | **source-udp** *url srce-ip-addr recv-ip-addr recv-port-list-num*} **track-count** *1-4* **destination** {**destination-pull** | **destination-udp** *ttl dest-ip-addr dest-list-num*} | **port-list** *list-num port-num* [*port-num*]} | **enable**}}

Syntax Description

advanced	Configures the advanced options for the RTSP gateway.
bypass-gateway	Configures the RTSP requests for which bypassing the RTSP gateway is allowed.
cisco-streaming-engine	Enables bypassing the RTSP gateway for requests intended for the Cisco Streaming Engine.
media-real	Enables bypassing the RTSP gateway for requests intended for RealProxy.
real-subscriber	Enables bypassing the RTSP gateway for requests intended for RealSubscriber.
max-initial-setup-delay	Sets the maximum delay allowed in seconds between TCP accept and the first RTSP message from the client.
time_delay	Maximum time delay allowed in seconds (0–2147483647). The default is 10.
max-request-rate	Sets the maximum number of incoming requests allowed by the RTSP gateway per second.
number	Maximum number of incoming requests allowed by the RTSP gateway per second. The default is 40.
ip-address	Configures the IP address for RTSP-based content distributors.
ip-address	IP address for RTSP-based content distributors.

l4-switch	Configures the Layer 4 switch interoperability with RTSP.
enable	Enables the Layer 4 switch interoperability.
port	Configures an RTSP port to listen for requests.
incoming	Sets the listener port for incoming RTSP requests.
port	Port number for incoming requests (1–65535). The default is 554.
proxy	Configures RTSP proxy parameters.
media-real	Configures the RealMedia cache.
accept-license-agreement	Accepts the RealProxy license.The license agreement can be viewed with the **show rtsp proxy media-real license-agreement** command.
enable	Enables RealProxy.
evaluate	Starts or continues the 60-day evaluation period of RealProxy.
license-key	Sets the required license key for the RealProxy.
keyword	RealProxy keyword string.
real-subscriber	Configures the RealSubscriber parameters.
accept-license-agreement	Accepts the RealSubscriber license. The license agreement can be viewed with the **show rtsp server real-subscriber license-agreement** command.
enable	Enables RealSubscriber.
evaluate	Starts or continues the 60-day evaluation period of RealSubscriber.
license-key	Sets the required license key for RealSubscriber.
keyword	RealSubscriber keyword string.
server	Configures RTSP server parameters.
cisco-streaming-engine	Enables the Cisco Streaming Engine parameters.
broadcast	Configures live streaming with the Cisco Streaming Engine.
id	Configures a specific live streaming program.
id	Alphanumeric identifier for the live streaming program.
source	Configures information about the source of the live streaming content.
source-rtsp	Specifies an external server providing RTSP streaming access.
rtsp-url	RTSP URL required to access the streaming server.
source-udp	Specifies an external server sending unicast or multicast streaming content over UDP.
url	HTTP URL required to access the Session Description Protocol (SDP) file for the stream.
srce-ip-addr	IP address of the external source of the streaming content.
recv-ip-addr	IP address used to receive the stream. Use a multicast address when receiving multicast streams. Use **0.0.0.0** when receiving a push (proactively streaming) unicast transmission.
recv-port-list-num	Numeric identifier for the list of ports.
track-count	Specifies the number of tracks in the streaming content. This number should match the number of ports configured in the port list.
1-4	Number of tracks.
destination	Configures information about the destination of the live streaming content.
destination-pull	Specifies an RTSP unicast on-demand access point.
destination-udp	Specifies a multicast UDP destination.

ttl	Time To Live for multicast live streaming packets.
dest-ip-addr	IP address for the multicast transmission.
dest-list-num	Numeric identifier for the list of ports.
port-list	Specifies an ordered list of ports to be used for configuring the live streaming content.
list-num	Numeric identifier for the list of ports.
port-num	(Optional) Port numbers to be used for live streaming.
enable	Enables the Cisco Streaming Engine.

Defaults

The RTSP proxy is disabled by default.

max-initial-setup-delay *time_delay*: 10 seconds

max-request-rate *number*: 40 requests

port incoming *port*: 554

Command Modes

global configuration

Usage Guidelines

RTSP is a standard Internet streaming control protocol (RFC 2326). It is an application-level protocol that controls the delivery of data with real-time properties, such as video and audio. Apple QuickTime, Real Networks, and the Cisco Streaming Engine use RTSP as the streaming control protocol.

Note Pre-positioned content requests are only accepted and served on the RTSP gateway (for both RealMedia and the Cisco Streaming Engine) default port number 554. If the default RTSP port number is changed to any other port number on the Content Engine when the Content Engine is waiting for RTSP pre-positioned content from the origin server, the Content Engine is unable to serve the content.

Note All cached RealMedia content is deleted from the Content Engine mediafs cache when you upgrade a Content Engine from any ACNS software release to the ACNS 5.1.9 software and later releases or the ACNS 5.2.1 software and later releases. This deletion occurs because the metadata file formats have been changed in these releases and affect the way that the cached RealMedia streaming file is interpreted.

Note The RealServer configuration that is performed through the RealServer administrator GUI is deleted when you upgrade the Content Engine to the ACNS 5.2.5 software and later releases.

Live Streaming with the Cisco Streaming Engine

The ACNS software supports live streaming content with many kinds of network topologies and deployment scenarios. This feature allows the integration of streaming content from Cisco IP/TV Servers and QuickTime live broadcast servers with the ACNS network. Support for broadcast of playlists is included (except for Content Engines at the network edge), allowing you to convert one or more disk files into a playlist and to send them out through simulated live streaming.

Use the **rtsp server cisco-streaming-engine broadcast port-list** command to define the UDP ports that you want the Content Engine to use for sending and receiving the streaming content. Use the **rtsp server cisco-streaming-engine broadcast id** command to define a specific configuration. Three different source types are supported and two different destination types, for a total of six different configurations, as shown below:

- Push (proactively streaming) UDP unicast source:
  - Push UDP multicast destination
  - Client-initiated RTSP unicast destination
- Client-initiated (pull) RTSP unicast source:
  - Push UDP multicast destination
  - Client-initiated RTSP unicast destination
- Push UDP multicast source:
  - Push UDP multicast destination
  - Client-initiated RTSP unicast destination

A UDP source is an external streaming server, such as the Cisco IP/TV Broadcast Server, that is sending the stream through UDP to the Content Engine. You must correctly configure the streaming server, with all other necessary components in the network, so that it is sending out the stream regardless of whether a Content Engine or other host is actually tuned into that stream. The UDP source can be unicast or multicast.

When you use the **destination-udp** option, the live program proactively sends the stream to the destination using UDP. The stream can be accessed by a live program with a UDP source. Currently, only a multicast UDP destination is supported. The UDP destination automatically provides the RTSP request access point.

An RTSP source is a fully qualified RTSP URL that references an external streaming server, such as a parent Content Engine, which provides the corresponding RTSP request point. When you use the **destination-pull** option, the live program relays its source to a published RTSP program, which can be accessed using an RTSP URL that corresponds to the RTSP source.

For UDP streams, the HTTP or FTP URL provides access to the SDP file for the stream, which must be published by the external streaming server. In a typical scenario, the external streaming source is sending two streaming tracks (audio and video) to the Content Engine on two different ports.

When the destination is UDP multicast, this configuration automatically provides the RTSP access point.

RTSP Gateway

The RTSP gateway is a process that runs on the Content Engine. The RTSP gateway accepts an RTSP request and performs the initial RTSP handshake with RTSP-based clients (for example, RealMedia clients and Windows Media 9 players) on behalf of the back-end RTSP servers (for example, the RealProxy server and the WMT RTSP server) that are running on the Content Engine.

Note On Content Engines that are running the ACNS 5.2 software and earlier releases, RealProxy is the only back-end RTSP server that can be enabled on a Content Engine.

In the ACNS 5.3 software release, the RTSP gateway was expanded to enable it to switch and tunnel RTSP requests from Windows Media 9 players to a WMT RTSP-based server. You can enable RealProxy or the WMT RTSP-based server as a back-end RTSP server on Content Engines that are running the ACNS 5.3 software and later releases. For Content Engines that are registered with a Content Distribution Manager, you can also enable RealSubscriber and Cisco Streaming Engine as back-end RTSP servers that run on the Content Engine.

For every RTSP request, the RTSP gateway examines the following properties of the request:

- The URL and its position in the UNS
- The user agent
- The IP address of the final destination
- The media type

See the *Cisco ACNS Software Configuration Guide for Locally Managed Deployments* for the possible actions that the RTSP gateway can take on an RTSP streaming request.

On Content Engines that are running the ACNS 5.2 software and earlier releases, the RealProxy server is the only back-end RTSP server that can be enabled on the Content Engine. In the ACNS 5.3 software and later releases, you can enable the RealProxy server or the WMT RTSP server as back-end RTSP servers on Content Engines.

After the successful completion of uniformity checks, the RTSP gateway tunnels the request to the appropriate back-end RTSP server that is running on the Content Engine. The RTSP gateway can tunnel the request to RealProxy, RealSubscriber, or the Cisco Streaming Engine on the Content Engine, depending on the requested media type, the back-end RTSP servers that are currently enabled on the Content Engine, and the media player that is requesting the content.

Note For Content Engines, RealProxy is the only back-end RTSP server that can be enabled on the Content Engine. For Content Engines that are registered with a Content Distribution Manager, you can also enable RealSubscriber and Cisco Streaming Engine as back-end RTSP servers that run on the Content Engine.

After the RTSP gateway tunnels the request to a particular back-end RTSP server that is running on the Content Engine and the back-end server and the client negotiate the UDP ports, the RTSP gateway continues with RTSP message passing (SETUP). When the RTSP client issues a PLAY request, the streaming server starts streaming the data to the client over UDP.

Based on the properties of the incoming request, including user agent, final destination, and media file type, the RTSP gateway performs the following tasks with Content Engines:

- Forwards the incoming request to the appropriate back-end RTSP server (the RealProxy server or the WMT RTSP server) that is running on the Content Engine as follows:
  - Forwards requests to the RealProxy server if the client is a RealMedia player. The Content Engine uses RealNetworks' proprietary RTSP as the protocol to serve the content to the media player.
  - Forwards requests to the WMT RTSP server if the client is a Windows Media 9 player. The Content Engine uses the IETF standard RTSP protocol and proprietary Microsoft extensions to server the content to Windows Media 9 players.

- Redirects the incoming request
- Rejects the incoming request

If the Content Engine is registered with a Content Distribution Manager, the RTSP gateway also redirects the incoming requests to other content distributors (for example, RealSubscriber or Cisco Streaming Engine) that are configured on the Content Engine.

Network Address Translation (NAT) is designed for IP address simplification and conservation because it enables private IP internetworks that use nonregistered IP addresses to connect to the Internet. NAT operates on a router, usually connecting two networks together, and translates the private addresses in the internal network into legal addresses before packets are forwarded onto another network. As part of this functionality, NAT can be configured to advertise only one external address for the entire network. This configuration provides additional security, effectively hiding the entire internal network from the world behind that address. NAT has the dual functionality of security and address conservation and is typically implemented in remote access environments.

Note If the Content Engine is behind a NAT-enabled router, you must specify the IP address of the RTSP gateway that is running on the Content Engine. By default, no IP address is specified.

Default RTSP Gateway Settings

The RTSP gateway is automatically enabled on the Content Engine and cannot be disabled with a command. Table 2-18 lists the default settings for the RTSP gateway.

Table 2-18 Default Settings for the RTSP Gateway

RTSP Gateway Setting	Default Setting
IP address of RTSP gateway	Not specified
Incoming RTSP port	Port 554
Incoming RTSP request rate	40 requests per second
Layer 4 switching	Not enabled
WCCP transparent interception	Not configured
Maximum initial setup delay	10 seconds
Maximum request rate	40 requests per second

By default, the RTSP gateway is always enabled on a Content Engine and cannot be disabled by entering a CLI command. As Table 2-18 shows, the RTSP gateway has a set of default settings. You only need to change these default settings under the following conditions:

- You want to configure the RTSP gateway to listen for incoming RTSP requests on a port other than the default port (port 554).
- The Content Engine is behind a NAT-enabled router. In this case, you must specify the IP address of the RTSP gateway. By default, an IP address for the RTSP gateway is not specified.

Bypassing the RTSP Gateway

In the ACNS software, Release 5.1 and earlier releases, clients were not allowed to send content requests directly to streaming servers. The ACNS 5.3 software release and later releases allow client requests to bypass the RTSP gateway.

Use the **rtsp advanced bypass-gateway cisco-streaming-engine** global configuration command to enable sending content requests directly to the Cisco Streaming Engine bypassing the RTSP gateway.

Note When the RTSP gateway is bypassed for a streaming server (Cisco Streaming Engine, RealProxy, or RealSubscriber), the Content Engine does not perform the uniformity checks such as UNS, rules, or URL filtering for an incoming request directed to that streaming server.

Configuring RealSubscriber

The ACNS 5.x software includes RealServer Version 9.0.2.855 as an optional component that is used as a streaming media engine. When RealServer software is configured for subscriber-only mode, it is referred to as RealSubscriber.

Removing the RealSubscriber License Key

The **no rtsp server real-subscriber license-key** command can be used to uninstall a license key if it is no longer needed on the device because the RealSubscriber licensed product is not needed. After a license key is uninstalled on one device, it can be used on another device if that device supports the RealSubscriber license key.

Note You must disable the RealSubscriber feature by entering the **no rtsp server real-subscriber enable** command before the RealSubscriber license key is uninstalled.

RealSubscriber Publisher Licensing

RealServer, which runs on the Content Engine, is referred to as RealSubscriber and uses the RealSubscriber license file from RealNetworks, Inc. RealPublisher is a RealServer that runs on a device other than the Content Engine and uses the RealPublisher license file.

To add licenses, use the RealNetworks distributed licensing feature. This licensing feature allows you to configure a Content Engine RealServer so that it can obtain additional licenses from a RealServer that is acting as a publisher. You configure the Content Engine RealServer with the publisher’s IP address (or hostname) and administration port using the RealSubscriber graphical user interface. You then enter the IP address and port information of every Content Engine in the publisher RealServer user interface.

Note For information on enabling RealProxy on the Content Engine, see Chapter 8 of the *Cisco ACNS Software Configuration Guide for Locally Managed Deployments*.

Note You must configure disk space to include mediafs storage with the **disk config** command before you can cache streaming media using RealProxy. You may also want to configure the percentage of mediafs space allocated to RTSP traffic with the **mediafs-division** global configuration command.

Removing the RealProxy Streaming Media Caching License Key

The **no rtsp proxy media-real license-key** command can be used to uninstall a license key if it is no longer needed on the device because the RealProxy licensed product feature is not needed. After a license key is uninstalled on one device, it can be used on another device if that device supports the RealProxy license key.

Note You must disable the RealProxy feature by entering the **no rtsp proxy media-real enable** command before the RealProxy license key is uninstalled.

Enabling Transparent Caching of RTSP Traffic Using WCCP-Enabled Routers

The RealMedia RTSP redirection service (service 80) is a WCCP Version 2 standard media caching service that supports the transparent redirection of RTSP requests from RealMedia players. This WCCP service permits WCCP Version 2-enabled routers to redirect RTSP requests from RealMedia players transparently to a single port (port 554) on a Content Engine. After receiving a redirected RTSP request, the Content Engine checks if whether it has a cached copy of the requested content. If it has the cached copy, the Content Engine sends the requested content to the RealMedia player. Otherwise, the Content Engine retrieves the requested content from the origin streaming server, caches a copy locally if RealMedia transparent caching is enabled on the Content Engine, and sends the requested content to the RealMedia player.

During transparent caching, the user's network traffic flows through the WCCP-enabled router rather than through the Content Engine to access streaming media. Before enabling transparent caching, make sure that you have fulfilled the following requirements:

- The Content Engine is running the ACNS 4.1 software and later releases.
- RealProxy software is installed with mediafs partitions mounted.
- You have the RealNetworks, Inc. license key.
- You have the IP addresses of the RealProxy and routers.

Note For information on the procedure to enable transparent redirection of RTSP traffic to the RealProxy, see Chapter 8 of the *Cisco ACNS Software Configuration Guide for Locally Managed Deployments*.

Enabling Conventional Proxy Caching of RTSP Traffic

During conventional proxy caching, the user media player is pointed to the Content Engine rather than to a WCCP-enabled router to access the streaming media. Before enabling conventional proxy caching, make sure that you have fulfilled the following requirements:

- The Content Engine is running the ACNS 4.1 software and later releases.
- RealProxy software is installed with mediafs partitions mounted.
- You have the RealNetworks, Inc., license key.
- You have the IP address of the RealProxy

Note For information on configuring the Content Engine to service RealPlayer clients with RealProxy on the Content Engine, see Chapter 4 of the *Cisco ACNS Software Configuration Guide for Locally Managed Deployments*.

Disabling RealMedia Caching

The RealProxy player comes with a streaming media cache of its own for the replication of on-demand content. However, an administrator can disable caching for the following various reasons:

- To collect more accurate data regarding cache hits or misses
- To prevent delivery of stale data
- To prevent access to personalized content by other users

To prevent caching of all material from all servers and RealProxy, follow these steps:

Note The administrator, usernames, and all associated passwords configured on the Content Engine are not the same as the ones contained in the RealProxy authentication database. Therefore, not all Content Engine users have access to the RealSystem Administrator GUI.

Streaming On-Demand Clips

All on-demand clips are automatically available to the Content Engine. If there is content served by your RealServer that you do not want to be cached, then disable RealMedia caching.

Unicasting, Splitting, and Multicasting

Live clips are not available to caching software. RealProxy will still proxy the live broadcasts for clients. RealServer acts as a source for live splitting, and RealProxy acts as a splitter. Live splitting allows the Content Engine to receive a unicast live stream and send it to multiple destinations on request.

Access Control

If a client requests a cached stream, RealProxy sends the request to the source RealServer for permission before allowing the client to play the stream. If RealServer denies the request, RealProxy does not allow the client to receive the stream.

You can block a single RealProxy from caching the material served by your RealServer by creating an access control rule from the RealSystem Administrator GUI that prohibits the IP address of that RealProxy from connecting to your RealServer. You can also restrict access to the content based on the number of players and bandwidth.

RealProxy cannot cache live broadcasts, because no actual downloadable file is available to cache. However, RealProxy includes an ability to share live streams among clients and reduce the bandwidth required from a transmitter. RealServer and RealProxy communicate through live splitting. RealServer is preconfigured to act as a transmitter, and RealProxy is automatically set up to act as a receiver.

Note A description of the Real-Time Streaming Protocol (RTSP) is available as IETF RFC 2326.

Examples

The following examples show the various scenarios supported using the Cisco Streaming Engine for live streaming support.

In the first example, the source is UDP unicast, with the server pushing the content:

```
ContentEngine2(config)# rtsp server cisco-streaming-engine broadcast port-list 1 21112
21114
ContentEngine2(config)# rtsp server cisco-streaming-engine broadcast id p1 source
source-udp http://172.19.227.80/17902.sdp 172.19.227.78 0.0.0.0 1 track-count 2
destination destination-udp 1 239.2.2.101 1
```

The ID of the stream is p1. The destination is UDP multicast, which automatically provides the RTSP access point. Users can access the stream using the following URL:

rtsp://ContentEngine2.cisco.com/p1.sdp.

The following example shows that the source is a client-initiated RTSP unicast. The destination is UDP multicast, as in the previous example.

```
ContentEngine1(config)# rtsp server cisco-streaming-engine broadcast port-list 2 22222
22224
```

```
ContentEngine1(config)# rtsp server cisco-streaming-engine broadcast id p2 source
source-rtsp rtsp://172.31.150.162:550/cse_live/ucast/p1.sdp track-count 2 destination
destination-udp 2 239.2.2.102 2
```

The following example shows that the source is a push UDP unicast, as in the first example. The destination is a client-initiated RTSP unicast. This configuration is useful in situations where you do not want push UDP streaming to waste network bandwidth.

```
ContentEngine2(config)# rtsp server cisco-streaming-engine broadcast port-list 3 23332
23334
ContentEngine2(config)# rtsp server cisco-streaming-engine broadcast id p3 source
source-udp http://172.19.227.80/3861.sdp 172.19.227.78 0.0.0.0 3 track-count 2 destination
destination-pull
```

The following example shows that the source is a client-initiated RTSP unicast, which is the same as in the second example. The destination is a client-initiated RTSP unicast.

```
ContentEngine1(config)# rtsp server cisco-streaming-engine broadcast id p4 source
source-rtsp rtsp://172.31.150.162:550/cse_live/ucast/p2.sdp track-count 2 destination
destination-pull
```

The following example shows that the source is a push UDP multicast, and the receiving IP address is a multicast IP address, which means that the Content Engine is receiving the stream from the same external source as in the first example, but the server is now sending a multicast to the network instead of sending a unicast to the Content Engine. The destination is a client-initiated RTSP access point only, as in the third and fourth examples.

```
ContentEngine2(config)# rtsp server cisco-streaming-engine broadcast port-list 5 25552
25554
ContentEngine2(config)# rtsp server cisco-streaming-engine broadcast id p5 source
source-udp http://172.19.227.80/17902.sdp 172.19.227.78 239.2.2.240 5 track-count 2
destination destination-pull
```

The following example has the same source as in the previous example, but the destination is a push multicast over UDP. The RTSP access point is still provided.

```
ContentEngine2(config)# rtsp server cisco-streaming-engine broadcast port-list 6 26662
26664
ContentEngine2(config)# rtsp server cisco-streaming-engine broadcast id p6 source
source-udp http://172.19.227.80/17902.sdp 172.19.227.78 239.2.2.240 6 track-count 2
destination destination-udp 239.2.2.102 2
```

The **show statistics rtsp proxy media-real requests** command displays RTSP request statistics.

Table 2-19 describes the fields shown in the **show statistics rtsp proxy media-real requests** display.

Table 2-19 show statistics rtsp proxy media-real requests Field Descriptions

Field	Description
Media Cache Statistics - Requests	
Total Received Requests	Total number of requests served.
Demand Cache Hit	Total number of cache hits.
Demand Cache Miss	Total number of cache misses
Demand Pass-Through	Total number demand pass-through connections.
Live Split	Total number of live-split connections.
Live Pass-Through	Total number of live pass-through connections.

The **show statistics rtsp proxy media-real savings** command displays RTSP savings statistics.

Table 2-20 describes the fields shown in the **show statistics rtsp proxy media-real savings** display.

Table 2-20 show statistics rtsp proxy media-real savings Field Descriptions

Field	Description
Media Cache Statistics - Savings	
Total	Total number of requests served.
Hits	Total number of cache hits.
Miss	Total number of cache misses
Bytes	Number of bytes bytes delivered.
Savings	Savings incurred by caching.

Related Commands

show rtsp

rule

To set the rules by which the Content Engine filters HTTP, HTTPS, and RTSP traffic, use the **rule** global configuration command. To disable individual options, use the **no** form of this command.

The general **rule** command is as follows:

rule {**action** *action-type* **pattern-list** *list_num* [**protocol** {**all** | *protocol-type*}] | **enable** | **pattern-list** *list_num pattern-type*}

The specific **rule** commands are as follows:

rule action allow pattern-list *list_num* [**protocol** {**all** | **http** | **https** | **rtsp**}]

rule action append-username-header pattern-list *list_num* [**protocol** {**all** | **http** | **https**}]

rule action block pattern-list *list_num* [**protocol** {**all** | **http** | **https** | **rtsp**}]

rule action cache-cookie pattern-list *list_num* [**protocol** {**all** | **http** | **https** | **rtsp**}]

rule action cache-non-cacheable ttl {**days** *days* **pattern-list** *list_num* [**protocol** {**all** | **http** | **https**}] | **hours** *hours* **pattern-list** *list_num* [**protocol** {**all** | **http** | **https**}] | **minutes** *minutes* **pattern-list** *list_num* [**protocol** {**all** | **http** | **https**}] | **seconds** *seconds* **pattern-list** *list_num* [**protocol** {**all** | **http** | **https**}]}

rule action cache-only pattern-list *list_num* [**protocol** {**all** | **http** | **https**}]

rule action dscp client cache-hit {**match-server pattern-list** *list_num* [**protocol** {**all** | **http** | **https**}] | **set-dscp** *dscpvalue* | **set-tos** *tosvalue*}

rule action dscp client cache-miss {**match-server pattern-list** *list_num* [**protocol** {**all** | **http** | **https**}] | **set-dscp** *dscpvalue* | **set-tos** *tosvalue*}

rule action dscp server {**match-client pattern-list** *list_num* [**protocol** {**all** | **http** | **https**}] | **set-dscp** *dscpvalue* | **set-tos** *tosvalue*}

rule action freshness-factor *exp_time* **pattern-list** *list_num* [**protocol** {**all** | **http** | **https**}]

rule action generate-url-signature {**include-client-src-ip** | **key-id-owner** *1-32* {**key-id-number** *1-16*} {**pattern-list** *1-512*} [**protocol** {**all** | **http**}]}

rule action insert-no-cache pattern-list *list_num* [**protocol** {**all** | **http** | **https**}]

rule action no-auth pattern-list *list_num* [**protocol** {**all** | **http** | **https** | **rtsp**}]

rule action no-cache pattern-list *list_num* [**protocol** {**all** | **http** | **https**}]

rule action no-persistent-connection {**all** | **client-only** | **server-only**} **pattern-list** *list_num* [**protocol** {**all** | **http** | **https**}]

rule action no-proxy pattern-list *list_num* [**protocol** {**all** | **http** | **https**}]

rule action no-url-filtering pattern-list *list_num* [**protocol** {**all** | **http** | **https**}]

rule action redirect *url* **pattern-list** *list_num* [**protocol** {**all** | **http** | **https** | **rtsp**}]

rule action redirect-url-for-cdn pattern-list *list_num* [**protocol** {**all** | **http** | **https** | **rtsp**}]

rule action refresh pattern-list *list_num* [**protocol** {**all** | **http** | **https**}]

rule action reset pattern-list *list_num* [**protocol** {**all** | **http** | **https** | **rtsp**}]

rule action rewrite pattern-list *list_num* [**protocol** {**all** | **http** | **https** | **rtsp**}]

rule action use-dns-server {*hostname* | *ip-address*} **pattern-list** *list_num* [**protocol** {**all** | **http** | **https**}]

rule action use-icap-service *service-name* **pattern-list** *list_num* [**protocol** {**all** | **http** | **https**}]

rule action use-proxy {*hostname* | *ip-address*} *port* {**failover pattern-list** | **pattern-list**} *list_num* [**protocol** {**all** | **http** | **https**}]

rule action use-server {*hostname* | *ip-address*} *port* **pattern-list** *list_num* [**protocol** {**all** | **http** | **https**}]

rule action use-xforward-clt-ip pattern-list *list_num* [**protocol** {**all** | **http** | **https**}]

rule dns-resolve each-request

rule enable

rule pattern-list *list-num* **domain** *dn_regexp*

rule pattern-list *list-num* **dst-ip** *d_ipaddress d_subnet*

rule pattern-list *list-num* **dst-port** *port*

rule pattern-list *list-num* **groupname** *group_name*

rule pattern-list *list-num* **groupname-regex** *group_name_regexp*

rule pattern-list *list-num* **group-type** {**and** | **or**}

rule pattern-list *list-num* **header-field** {**referer** *ref_regexp* | **request-line** *req_regexp* | **user-agent** *ua_regexp*}

rule pattern-list *list-num* **header-field-sub** {**referer** *ref_regexp ref_sub* | **request-line** *req_regexp req_sub* | **user-agent** *ua_regexp ua_sub*}

rule pattern-list *list-num* **icap-attribute** *icap_attribute icap_value*

rule pattern-list *list-num* **mime-type** *mt_regexp*

rule pattern-list *list-num* **src-ip** *s_ipaddress s_subnet*

rule pattern-list *list-num* **url-regex** *url_regexp*

rule pattern-list *list-num* **url-regsub** *url_regexp url_sub*

rule pattern-list *list-num* **username** *user_name*

no rule action {**allow pattern-list** *list_num* [**protocol** {**all** | **http** | **https** | **rtsp**}] | **append-username-header pattern-list** *list_num* [**protocol** {**all** | **http** | **https**}] | **block pattern-list** *list_num* [**protocol** {**all** | **http** | **https** | **rtsp**}] | **cache-cookie pattern-list** *list_num* [**protocol** {**all** | **http** | **https** | **rtsp**}] | **cache-non-cacheable ttl** {**day** *days* **pattern-list** *list_num* [**protocol** {**all** | **http** | **https**}] | **hours** *hours* **pattern-list** *list_num* [**protocol** {**all** | **http** | **https**}] | **minutes** *minutes* **pattern-list** *list_num* [**protocol** {**all** | **http** | **https**}] | **seconds** *seconds* **pattern-list** *list_num* [**protocol** {**all** | **http** | **https**}]} | **cache-only pattern-list** *list_num* [**protocol** {**all** | **http** | **https**}] | **dscp client cache-hit** {**match-server pattern-list** *list_num* [**protocol** {**all** | **http** | **https**}] | **set-dscp** *dscpvalue* | **set-tos** *tosvalue*} | **dscp client cache-miss** {**match-server pattern-list** *list_num* [**protocol** {**all** | **http** | **https**}] | **set-dscp** *dscpvalue* | **set-tos** *tosvalue*} | **dscp server** {**match-client pattern-list** *list_num* [**protocol** {**all** | **http** | **https**}] | **set-dscp** *dscpvalue* | **set-tos** *tosvalue*} | **freshness-factor** *exp_time* **pattern-list** *list_num* [**protocol** {**all** | **http** | **https**}] | {**generate-url-signature** {**include-client-src-ip** | **key-id-owner** *1-32* {**key-id-number** *1-16*} {**pattern-list** *1-512*} [**protocol** {**all** | **http**}]}**insert-no-cache pattern-list** *list_num* [**protocol** {**all** | **http** | **https**}] | **no-auth pattern-list** *list_num* [**protocol** {**all** | **http** | **https** | **rtsp**}] | **no-cache pattern-list** *list_num* [**protocol** {**all** | **http** | **https**}] | **no-persistent-connection** {**all** | **client-only** | **server-only**} **pattern-list** *list_num* [**protocol** {**all** | **http** | **https**}] | **no-proxy pattern-list** *list_num* [**protocol** {**all** | **http** | **https**}] | **no-url-filtering pattern-list** *list_num* [**protocol** {**all** | **http** | **https**}] | **redirect** *url* **pattern-list** *list_num* [**protocol** {**all** | **http** | **https** | **rtsp**}] | **redirect-url-for-cdn pattern-list** *list_num* [**protocol** {**all** | **http** | **https** | **rtsp**}] | **refresh pattern-list** *list_num* [**protocol** {**all** | **http** | **https**}] | **reset pattern-list** *list_num* [**protocol** {**all** | **http** | **https** | **rtsp**}] | **rewrite pattern-list** *list_num* [**protocol** {**all** | **http** | **https** | **rtsp**}] | **use-dns-server** {*hostname* | *ip-address*} **pattern-list** *list_num* [**protocol** {**all** | **http** | **https**}] | **use-icap-service** *service-name* **pattern-list** *list_num* [**protocol** {**all** | **http** | **https**}] | **use-proxy** {*hostname* | *ip-address*} *port* {**failover pattern-list** | **pattern-list**} *list_num* [**protocol** {**all** | **http** | **https**}] | **use-server** {*hostname* | *ip-address*} *port* **pattern-list** *list_num* [**protocol** {**all** | **http** | **https**}] | **use-xforward-clt-ip pattern-list** *list_num* [**protocol** {**all** | **http** | **https**}]}

no rule enable

no rule pattern-list *list-num* {**domain** *dn_regexp* | **dst-ip** *d_ipaddress d_subnet* | **dst-port** *port* | *list-num* **groupname** *group_name* | **groupname-regex** *group_name_regexp* | **group-type** {**and** | **or**} | **header-field** {**referer** *ref_regexp* | **request-line** *req_regexp* | **user-agent** *ua_regexp*} | **header-field-sub** {**referer** *ref_regexp ref_sub* | **request-line** *req_regexp req_sub* | **user-agent** *ua_regexp ua_sub*} | **icap-attribute** *icap_attribute icap_value* | **mime-type** *mt_regexp* | **src-ip** *s_ipaddress s_subnet* | **url-regex** *url_regexp* | **url-regsub** *url_regexp url_sub* | **username** *user_name*}

Syntax Description

action	Describes the action that the rule is to take.
action-type	Types of actions that you can associate with a defined pattern list. The action-type is indicated by preceding its description with an Action in this table.
pattern-list	Configures the pattern list.
list_num	Pattern list number (1–512).
protocol	Specifies the protocol for which this rule is to be matched.
all	Matches this rule with all applicable protocols for this action.

protocol-type	Protocol types that support rule actions, namely, **http**, **https**, and **rtsp**. **Note** The term **http** traffic is used to refer to requests over HTTP including HTTP, FTP over HTTP, and HTTPS over HTTP. The Rules Template is not supported for FTP native requests.
enable	Enables rules processing on the Content Engine.
pattern-type	Types of rule patterns that you can add to a pattern list. The pattern-type is indicated by preceding its description with a Pattern type in this table.
allow	Action—Allows the incoming request that matches the pattern list. This rule action can be used in combination with reset or block actions to allow selective types of requests. Allow does not carry any meaning as a standalone action.
http	Matches this rule with the HTTP protocol.
https	Matches this rule with the HTTPS protocol.
rtsp	Matches this rule with the RTSP protocol.
append-username-header	Action—Appends the username to the request header on a cache miss in the request sent to the origin server.
block	Action—Blocks this request and allows all others.
cache-cookies	Action—Caches the request containing cookies.
cache-non-cacheable	Action—Overrides HTTP response headers and caches this object.
ttl	Specifies the Time-To-Live value of this object.
days	Specifies the Time-To-Live units in days.
days	Time-To-Live value in days (1–1825).
hours	Specifies the Time-To-Live units in hours.
hours	Time-To-Live value in hours (1–43800).
minutes	Specifies the Time-To-Live units in minutes.
minutes	Time-To-Live value in minutes (1–2628000).
seconds	Specifies the Time-To-Live units in seconds.
seconds	Time-To-Live value in seconds (1–157680000).
cache-only	Action—Caches this object depending on the HTTP response headers. Caches this object only if it is a match and is allowed to be cached by HTTP.
dscp client	Action—Configures IP Type of Service or differentiated services code point (ToS/DSCP) field responses for the client.
cache-hit	Sends responses to the client when a cache hit occurs.
match-server	Uses the original ToS or DSCP value of the server.
set-dscp	Configures differentiated services code point (DSCP) values.
dscpvalue	DSCP values; see Table 2-16 on page 2-371 for valid values.
set-tos	Configures type of service (ToS) values.
tosvalue	The ToS value; see Table 2-17 on page 2-372 for valid values.
cache-miss	Sends responses to the client when a cache miss occurs.
dscp server	Action—Configures IP Type of Service/differentiated services code point (ToS/DSCP) for outgoing responses.
match-client	Uses original ToS/DSCP value of the client.

freshness-factor	Action—Caches heuristic modifiers.
exp_time	Expiration time of object as a percentage of age (0–100).
generate-url-signature	Specifies that the Content Engine generates a signed URL that is included in the autogenerated ASX file when content routing is in use and the pattern matches.
include-client-src-ip	Specifies the client IP to be included in the signed URL.
key-id-owner	Specifies the owner of the key (1–32). The key is a shared secret string.
key-id-number	Specifies the identification number of the key (1–16).
pattern-list	Specifies the number of the pattern list (1-512). Valid patterns are domain, url-regex, or dst-ip.
protocol	(Optional) Specifies the supported protocols.
insert-no-cache	Action—Inserts a no-cache header in the response.
no-auth	Action—Does not authenticate.
no-cache	Action—Does not cache the object.
no-persistent-connection	Prevents the use of persistent connections.
all	Prevents the use of persistent connections to either clients or servers.
client-only	Prevents the use of persistent connections to clients.
server-only	Prevents the use of persistent connections to servers.
no-proxy	Action—Does not use any upstream proxy. For an example, see the "Example of no-proxy Action" section on page 2-425.
no-url-filtering	Action—Bypasses the URL filtering for certain HTTP and HTTPS requests. This feature is supported for local list URL filtering (good and good sites' lists), and Websense, SmartFilter, or N2H2 URL filtering in the ACNS 5.2.3 software and later releases. For an example, see the "Example of no-url-filtering Action" section on page 2-424.
redirect	Action—Redirects the request to the rewritten URL.
url	Redirect URL.
redirect-url-for-cdn	Action—Redirects the request to an alternative URL for the ACNS network content.
refresh	Action—Revalidates the object with the web server.
reset	Action—Issues a TCP RST.
rewrite	Action—Rewrites the original request as a specified URL and fetches the rewritten URL on a cache miss.
use-dns-server	Action—Uses a specific DNS server.
hostname	Hostname of the DNS server.
ip-address	IP address of the DNS server.
use-icap-service	Action—Uses a specific Internet Content Adaptation Protocol (ICAP) server.
service-name	Service name used for handling a request through an ICAP server.
use-proxy	Action—Makes use of a specific upstream proxy.
hostname	Hostname of the specific proxy.
ip-address	IP address of the specific proxy.
port	Port number of the specific proxy (1–65535).
failover	Causes an outgoing proxy to fail over to outgoing HTTP proxy servers.

use-server	Action—Makes use of a specific server. For an example, see the "Example of use-server Action" section on page 2-424.
use-xforward-clt-ip	Uses the client IP address in the X-forwarded header for filtering.
hostname	Hostname of the specific server.
ip-address	IP address of the specific server.
port	Port number of the specific server (1–65535).
dns-resolve	Specifies the DNS resolution of the hostname when the action use-proxy or use-proxy failover is hit.
each-request	Enables dynamic resolution of the hostname.
domain	Pattern type—Specifies the regular expression to match the domain name.
dn_regexp	Regular expression to be matched with the domain name.
dst-ip	Pattern type—Specifies the destination IP address of the request.
d_ipaddress	Destination IP address of the request.
d_subnet	Destination IP subnet mask.
dst-port	Pattern type—Specifies the destination port number.
port	Destination port number (1–65535).
groupname	Specifies the name of the group to which the user belongs.
groupname-regex	Specifies the regular expression to be matched with the name of the group (to which the user belongs).
group-type	Pattern type—Specifies whether the pattern list is an AND or OR type. The default is OR.
and	Specifies an AND pattern to the pattern list.
or	Specifies an OR pattern to the pattern list.
header-field	Pattern type—Specifies the request header field pattern.
referer	Specifies the referer request header.
ref_regexp	Regular expression to be matched with the referer request header.
request-line	Specifies the request method line.
req_regexp	Regular expression to be matched with the request method line.
user-agent	Specifies the user agent request header.
ua_regexp	Regular expression to be matched with the User Agent request header.
header-field-sub	Pattern type—Specifies the header field pattern of the request and substitute replacement pattern.
referer	Specifies the referer request header.
ref_regexp	Regular expression to be matched with the referer request header.
ref_sub	Request header regular expression replacement string.
request-line	Specifies the request method line.
req_regexp	Regular expression to be matched with the request method line.
req_sub	Request method line regular expression replacement string.
user-agent	Specifies the user agent request header.
ua_regexp	Regular expression to be matched with the User Agent request header.
ua_sub	Regular expression replacement string for the User Agent request header.

mime-type	Pattern type—Specifies the MIME type to be matched with the Content-Type HTTP header.
mt_regexp	Regular expression to be matched with the content type.
src-ip	Pattern type—Specifies the source IP address of the request.
s_ipaddress	Source IP address of the request.
s_subnet	Source IP subnet mask.
url-regex	Pattern type—Specifies the regular expression to match a substring of the URL.
url_regexp	Regular expression to be matched with the URL string.
url-regsub	Pattern type—Sets the regular expression to match the URL and replacement pattern.
url_regexp	Regular expression to be matched with the URL string.
url_sub	URL string replacement pattern.
username	Specifies the username in the HTTP request.

Defaults

The default is rule processing disabled.

The group-type pattern is OR by default.

Command Modes

global configuration

Usage Guidelines

The Rules Template allows you to specify a set of rules, each clearly identified by an action and a pattern. This feature allows you to configure a Content Engine to use specific rules to filter HTTP, HTTPS, and RTSP traffic. A common use of this feature is to configure a Content Engine to block the spread of Internet worms and viruses within an organization by checking whether a requested web page matches the pattern of a known Internet worm and if so then automatically blocking the request.

If you have enabled rules processing on a Content Engine (enabled the Rules Template feature on the Content Engine and configured rules for the Content Engine), the Content Engine checks each incoming client request to determine if a rule pattern matches the requested content. If a rule pattern matches the given request, the Content Engine uses the specified action (policy) to handle this incoming traffic.

The Content Engine can match incoming requests against the following:

- Patterns in the IP address of the client requesting the content (source IP address), including the IP address, the network mask, and the port list
- Patterns in the IP address of the origin web or media server (destination IP addresses), including the IP address, the network mask, and the port list
- Regular expression of the URL
- Regular expression of the domain portion of the URL
- MIME types of the web object that the client is requesting
- Regular expressions symbolizing domain names
- Headers that are sent in the request, including the following:
  - User-agent of the request, which indicates which client software is issuing the request

  - Referer, which indicates the web page from which the browser jumped to this link
  - Request Line, which indicates the request line itself

You can apply the policies defined in the Rules Template to the HTTP, (including FTP over HTTP) and HTTPS protocols and to the RTSP protocol for streaming media objects. Policies that can be applied include the following:

- Allowing a request to be completed
- Appending the username to the request headers
- Blocking the request
- Overriding the HTTP response header and caching the object
- Caches the object depending on the HTTP response header
- Bypassing authentication for the request
- Resetting the request
- Using a specific object freshness calculation factor
- Not caching an object
- Bypassing an upstream proxy for the request
- Redirecting the request to a different URL
- Revalidating the object with the origin server
- Rewriting the URL
- No URL filtering for the specified HTTP and HTTPS requests
- Using a specific ICAP server
- Using a specific upstream proxy
- Using a specific server for the request
- Setting ToS or DSCP in the response sent to the client
- Setting ToS or DSCP in the response sent to the server

Note To enter a question mark (?) in a rule regular expression from the command-line interface, use the escape character followed by a question mark (?). Use of the escape sequence prevents the command-line interface from displaying context-sensitive help.

Supported Rule Actions per Protocol

For the RTSP streaming protocol, the redirect and the redirect_url_for_cdn rule actions are supported for RTSP requests from RealMedia players. These two rule actions are not supported for RTSP requests from Windows Media Players. For example, Windows Media Services 9 (WMS 9) supports the following rule actions for RTSP requests: block, reset, rewrite, and allow but does not support the redirect and redirect_url_for cdn rule actions for RTSP requests. See the *Cisco ACNS Software Configuration Guide for Locally Managed Deployments* for a complete list of all rule actions per protocol that are supported by Content Engines running the ACNS 5.3 software and later releases.

Supported Action and Pattern Combinations

Not all actions support all patterns for request matching because some patterns do not make sense for some actions. See the *Cisco ACNS Software Configuration Guide for Locally Managed Deployments* for a detailed description of the supported action and pattern combinations for Content Engines.

Understanding Actions and Patterns

A *rule* is specified by an action and a pattern list. An action is performed on a request if this request matches the pattern list specified in the **rule pattern-list** command.

An *action* is something that the Content Engine performs when processing a request; for instance, an action could be blocking the request, using an alternative proxy, and so forth.

A *pattern list* defines the limits of a request; for instance, a pattern list may specify that the source IP address falls in the subnet range 172.16.*.*.

Rules can be dynamically added, displayed, or deleted from the Content Engine. The rules are preserved across reboots because they are written into persistent storage, such as NVRAM, using the appropriate CLI commands, the Content Engine GUI, or the Content Distribution Manager GUI. Only the system resources limit the number of rules that the Content Engine can support. Because rules consume resources, the more rules that you define can greatly impact how the Content Engine performs.

Note The number of actions is limited only by available resources. The maximum number of pattern lists is 512. The maximum number of patterns per action is 128. A single pattern list can contain up to 128 patterns of a particular pattern type.

Actions

The Rules Template supports the following types of actions:

- **Allow**—Allows incoming requests that match the pattern list.

  This rule action can be used in combination with reset or block actions to allow selective types of requests. Allow does not carry any meaning as a standalone action.
- **Append-username-header**—Appends the username-header on a cache miss.
- **Block**—Blocks this request and allows all others.
- **Cache-non-cacheable**—Overrides the HTTP response headers and caches the objects.

  Note The **rule action cache-non-cacheable** command cannot cache objects if the objects are authenticated. Some origin servers do not send the Last-Modified or ETag entity headers for authenticated objects. Revalidation of those authorized objects cannot be performed by the Content Engine. Those authenticated objects are served only from the origin server. If the server *does* send the Last-Modified and ETag headers for authenticated objects, then they are properly revalidated and served from the cache.

- **Cache-only**—Caches objects depending on the HTTP response headers. Caches this object only if it is a match and is allowed to be cached by HTTP. If one or more rules specify this action, an object is cached if and only if it matches at least one of the cache-only rules and passes every other caching restriction such as the object-size check and the no-cache-on-authenticated-object check. If the object does not match any of the cache-only rules, the object is not cached.
- **DSCP client**—Configures the IP ToS or DSCP code point field as follows:
  - **cache-hit**—Sets the IP ToS or DSCP code point bits for the client-side connection to the configured value for **cache-hit** responses to the client.
  - **cache-miss**—Sets the IP ToS or DSCP code point bits for the client-side connection to the configured value for **cache-miss** responses to the client.

Setting the ToS or differentiated services code point (DSCP) is called packet marking, which allows you to partition the network data into multiple priority levels or types of service. With the ACNS 5.x software, 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.

You can set specific ToS or DSCP values for the following:

– Requests from the Content Engine to the server

– Responses to the client on a cache hit

– Responses to the client on a cache miss

The ToS or DSCP may be set based on any of the policies matching the **src-ip** *s_ipaddress s_subnet*, **dst-ip** *d_ipaddress d_subnet*, **dst-port** *port*, **domain** *LINE,* **url-regex** *LINE*, or **mime-type** *LINE* options. In addition, you can configure global ToS or DSCP settings with the **ip dscp** command.

Note The Rules Template configuration takes precedence over the **ip dscp** command, and the **url-filter** command takes precedence over the **rule** command to the extent that even the **rule no-block** command is executed only if the **url-filter** command has not blocked the request.

Valid values for *dscpvalue* are listed in Table 2-21.

Table 2-21 dscpvalue 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).
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).

Table 2-21 dscpvalue Values (continued)

Value or Keyword	Description[1]
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 *tosvalue* are listed in Table 2-22.

Table 2-22 tosvalue 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.

- **DSCP server**—Configures the IP ToS or DSCP field for requests to the origin server.
- **Freshness-factor**—Determines the Time To Live if the request URL matches a specified regular expression. The **refresh** configuration takes priority over **freshness-factor** configurations.
- **Insert-no-cache**—Inserts a no-cache header in the response.
- **No-auth**—Does not authenticate.

  The **no-auth** rules result in the display of multiple authentication windows in the following scenario:

  – When the main page (for example, index.htm) is excluded from proxy authentication by using **no-auth** rules

  – When the user entry is not already included in the Content Engine authentication cache

  – When the index.htm page contains objects belonging to different domains

  The **no-auth** option permits specific login and content requests to bypass authentication and authorization features such as LDAP, RADIUS, SSH, or TACACS+.

To avoid multiple authentication windows, enter the **http avoid-multiple-auth-prompts** command in global configuration mode. Once it is configured, check the configuration with the **show http avoid-multiple-auth-prompts** command.

- **No-cache**—Does not cache this object. If both the **no-cache** and **cache-only** actions are matched, **no-cache** takes precedence.
- **No-persistent-connection**—Does not use persistent connections for client connections, server connections, or all connections**.**
- **No-proxy**—For a cache miss, does not use the configured upstream proxy but contacts the server directly.

  The **no-proxy** action is applicable when the administrator of the Content Engine has configured an outgoing proxy server for the Content Engine. The **no-proxy** action states that for requests that match the criteria, if a connection with the origin server is needed (for example, because of a cache miss), the Content Engine should not use the specified proxy server to establish the connection with the origin server.

  This rule is useful if a company has a Content Engine (CE1) at the Internet gateway to cache all Internet content and a Content Engine at each branch office (CE2, CE3, CE4). In this case, the administrator can configure CE2, CE3, and CE4 at the branch offices to use CE4 as the outgoing proxy server but set up the **no-proxy** rule for requests for corporate internal content. When CE2, CE3, and CE4 receive a client request and the requested content is not already stored in their local caches, they will process the request as follows:

  – If the client request is for Internet content, then CE2, CE3, and CE4 should use CE1 at the Internet gateway instead of going to the origin server directly.

  – If the client request is for corporate internal content, then CE2, CE3, and CE4 should establish a connection directly with the origin server instead of going to CE1.
- **No-url-filtering**— Bypasses URL filtering for certain HTTP and HTTPS requests. This feature is supported for local list URL filtering (good and good sites' lists), and Websense, SmartFilter, or N2H2 URL filtering in the ACNS 5.2.3 software and later releases. The **no-url-filtering** action supports the following rule patterns: src-ip, dst-ip, dst-port, domain, group-name, groupname-regex, header-field, url-regex, and username.

Note The **no-url-filtering** action supports the src-ip, dst-ip, dst-port, domain, group-name, groupname-regex, header-field, url-regex, and username rule patterns. Patterns can be ANDed or ORed by using the group-type pattern (for example, **rule pattern-list 1 group-type and**). The default is OR.

When the Content Engine receives an HTTP or HTTPS request that has foo.com as the domain, the **rule action no-url-filtering** rule is matched. The Content Engine bypasses URL filtering for that particular request as shown in the partial output of the **debug http proxy** command as follows:

```
Oct 28 12:25:12 Content Engine 3: Rule action no-url-filtering match - Bypassing
urlfiltering
```

If the **rule action no-url-filtering** rule is matched and SmartFilter URL filtering is being used instead of Websense URL filtering, the output of the **debug http proxy** command would be as follows:

```
Oct 28 12:25:12 Content Engine 3: Rule action no-url-filtering match - Bypassing
SmartFilter processing
```

When the Content Engine receives an HTTP or HTTPS request for websites other than foo.com (for requests that have www.abc.com as the domain), the **rule action no-url-filtering** rule is not matched. The Content Engine proceeds with Websense URL filtering for that particular request as shown in the partial output of the **debug http proxy** command as follows:

```
Oct 28 12:28:06 Content Engine 3: Rule action no-url-filtering not hit - Proceed with
urlfiltering
```

If the **rule action no-url-filtering** rule is not matched and SmartFilter URL filtering is being used instead of Websense URL filtering, the output of the **debug http proxy** command would be as follows:

```
Oct 28 12:25:12 Content Engine 3: Rule action no-url-filtering not hit- Proceed with
SmartFilter processing
```

Enter the **show statistics rule http action no-url-filtering** EXEC command to display statistics for the **no-url-filtering** action.

Enter the **show run**, **show statistics rule all**, and **clear statistics rule all** EXEC commands to display information about the **no-url-filtering** action.

- **Redirect-url-for-cdn**—Redirects the original request to an alternative URL for the ACNS network content. This rule action is only applicable for Content Engines that are registered with a Content Distribution Manager; it is not applicable to standalone Content Engines.
- **Redirect**—Redirects the original request to a specified URL. Redirect is relevant to the RADIUS server only if the RADIUS server has been configured for **redirect**.
- **Refresh**—For a cache hit, forces an object freshness check with the server.
- **Reset**—Issues a TCP RST. This reset request is useful when resetting Code Red or Nimda virus requests.
- **Rewrite**—Rewrites the original request as a specified URL. The Content Engine searches for the rewritten URL in the cache, and then on a cache miss, fetches the rewritten URL and returns the object transparently to the client. You should use a **redirect** rule instead of a **rewrite** rule because of possible performance impacts. The reason for the performance impact is that, for a **redirect** rule, the Content Engine sends a 302 (Found) message to the client with the new redirect URL. The client issues a separate request to the redirected URL. However, for a **rewrite** action, the original request URL is rewritten as the specified URL. The URL rewrite could change the domain name of the URL, which necessitates a DNS lookup to find the destination IP address of the new rewritten server to which the request must be sent. The original IP address derived from the WCCP redirect packet cannot be used. This action may result in a DNS lookup for the new rewritten server, which would lower the transactions per second (TPS).
- **Use-dns-server**—Uses the DNS specified server.
- **Use-icap-service**—Uses a specified ICAP server.
- **Use-proxy**—For a cache miss, uses a specific upstream proxy. Specify the upstream proxy IP address (or domain name) and port number. The **use-proxy failover** rule is similar to the **use-proxy** rule, except that if the connection attempt on the configured outgoing proxy fails, the requests fail over to the outgoing proxies configured with the HTTP proxy outgoing configuration. The rule requests use the HTTP proxy outgoing **origin-server** option, if it is configured.

  The HTTP failover does not apply if the destination is on the exclude list. When in transparent mode, the setting for the original proxy takes precedence.

  If both **no-proxy** and **use-proxy** are matched, **no-proxy** takes precedence.
- **Use-server**—Sends server-style HTTP requests from the Content Engine to the specified IP address and port on a cache miss.

Among **use-server**, **no-proxy**, and **use-proxy** rules, the **use-server** rule is the first rule to be checked. If it results in a rule miss, **no-proxy** and **use-proxy** rules are executed in succession (**use-proxy** is not checked if a **no-proxy** rule matches).

If a rule is configured with a fully qualified domain name (FQDN) and a request is received with the partial domain name in transparent mode, the rule fails to be executed, because the FQDN is not in the request URL. In transparent mode, if a request is destined for a particular domain (for which a domain rule is configured) and does not contain the Host header, the rule pattern match fails.

For HTTP requests that match the specified criteria, if the Content Engine needs to contact the origin server (for example, if a cache miss occurs), the Content Engine does not go to the server indicated in the request to retrieve the requested object; instead, it uses a different destination. This rule action is primarily used for on-demand requests in reverse proxy deployments.

Note The **use-server** rule action applies to HTTP processing only.

- **Use-xforward-clt-ip**—Uses the client IP address in the X-Forwarded-For header for filtering.

Patterns

The Rules Template supports the following pattern types:

- **Domain**—Matches the domain name in the URL or the Host header against a regular expression. For example, .*ibm.* matches any domain name that contains the ibm substring. \.foo\.com$ matches any domain name that ends with the .foo.com substring.

Note In regular expression syntax, the dollar sign ($) metacharacter directs that a match is made only when the pattern is found at the end of a line.

- **Dst-ip**—Matches the request's destination IP address and netmask. Specify an IP address and a netmask. In proxy mode, the Content Engine does a DNS lookup to resolve the destination IP address of the HTTP request, making the response time longer, and possibly negating the benefit of setting a **dst-ip** rule. When an outgoing proxy is configured, the Content Engine forwards the cache miss requests to the outgoing proxy without examining the destination server IP address, making the **dst-ip** rule unenforceable on the first Content Engine.
- **Dst-port**—Matches the request's destination port number. Specify a port number.
- **Group-type**—Specifies whether the pattern list is an AND or OR type. The default is OR.
- **Groupname**—Matches the group name of the end user (the web client that is requesting content), who was authenticated through LDAP or NTLM, against the group name specified in the rule.

  This pattern can be applied only to request authentication for users who have been authenticated through LDAP or NTLM. This pattern supports exact string comparison and is case insensitive. The maximum length of the group name is 255 characters. Valid characters are an underscore and alphanumeric characters. If the groupname configuration in the Rules Template and the group name-based access list match, then the access list takes precedence.

Tip If you intend to use the groupname pattern, make sure that you set the correct number of maximum group entries in the authentication group cache (the **http authentication cache max-group-entries** *number* global configuration command). This number should correspond to the maximum number of groups that could be returned during authorization queries (for example, the total number of groups defined on the AAA server.) The number can be from 500 to 12000. The number of entries in the authentication group cache is dependent on the physical resources available on the Content Engine.

- **Groupname-regex**—Matches the group name of the end user (the web client that is requesting the content) against the regular expression specified in the rule. For example, use the **groupname-regex** pattern type to configure a regular expression-based policy on group names or to OR multiple group names in the same line of a single pattern list.
- **Mime-type**—Matches the MIME type of the response. Specify a MIME type string, for example, image/gif, as defined in RFC 2046, which can be found at the following URL:

  http://www.faqs.org/rfcs/rfc2046.html

  You can specify a substring; for example, you can specify “java” and have it apply to all MIME types with the java substring, such as application/x-javascript.
- **Src-ip**—Matches the request’s source IP address and netmask. Specify an IP address and a netmask.
- **URL-regex**—Matches the URL against a regular expression. The match is case insensitive. Specify a regular expression.
- **Header-field**—Matches the header field pattern of the request.

  Request header field patterns **referer**, **request-line**, and **user-agent** are supported for actions **block**, **reset**, **redirect**, and **rewrite**. The **referer** pattern is matched against the Referer header in the request, the **request-line** pattern is matched against the first line of the request, and the **user-agent** pattern is matched against the User-Agent header in the request.
- **Header-field-sub**—Matches the header field pattern of the request and substitute replacement pattern.
- **URL-regsub**—Matches the URL against a regular expression to form a new URL per pattern substitution specification for the **rewrite** and **redirect** actions. The match is case insensitive. The valid substitution index range is from 1 to 9.
- **Username**—Matches the username of the end user (the web client that is requesting content), who was authenticated through LDAP, NTLM, RADIUS, or TACACS+, against the username or usernames specified in the rule. The maximum length of the username is 255 characters for LDAP, RADIUS, or TACACS+ authentication.

  Valid characters are an underscore and alphanumeric characters. This pattern supports an exact string comparison and is case insensitive.

  To specify multiple usernames in the same line for the same pattern list, use a delimiter.

  By default, the match does not consider the domain name and matches only the username. To include the domain name and the username in the match, specify *domainname\username*.

  For NTLM authentication, the domain\username:password:NTLM string must be 50 characters or less. If this string is greater than 50 characters, the domain name is truncated and the rule username pattern is not matched. An error message is generated in the system log in this situation.

  To match all users in a particular domain, enter the following command:

  ```
  ContentEngine(config)# rule pattern-list 1 username domain domainname\*
  ```

where *domainname* is the name of the domain (for example, cisco).

If an empty string is given as a replacement pattern, the referer header is stripped. Stripping of the referer header occurs in the user-agent pattern.

Note The **rule action no-proxy**, **rule action use-proxy** *hostname port-number* **failover**, and **rule action use-proxy** commands take precedence over the **https proxy outgoing**, **http proxy outgoing**, and **ftp proxy outgoing** global configuration commands.

Among the **use-server**, **no-proxy**, and **use-proxy** rules, the **use-server** rule is the first one to be checked. If none of these rules match, the **no-proxy** and **use-proxy** rules are executed in succession (the **use-proxy** rule is not checked if there is a match with the **no-proxy** rule). If a rule is configured with a fully qualified domain name (FQDN) and a request is received with the partial domain name in transparent mode, the rule fails to be executed because the FQDN is not in the request URL. In transparent mode, if a request is destined for a particular domain (for which a domain rule is configured) and does not contain the Host header, the rule pattern match fails. If both the **no-proxy** and **use-proxy** rules are matched, the **no-proxy** rule takes precedence.

The **use-proxy-failover** rule is similar to the **use-proxy** rule, except that if the connection attempt on the configured outgoing proxy fails, the requests fail over to the outgoing proxies configured with the HTTP proxy outgoing configuration. The rule requests use the **http proxy outgoing origin-server** option if it is configured. The **use-proxy-failover** rule takes precedence over the **use-proxy** rule. If both the **no-proxy** and **use-proxy-failover** rules are matched, the **no-proxy** rule takes precedence. The HTTP failover does not apply if the destination is on the exclude list. In transparent mode, the setting for the original proxy takes precedence.

Multiple patterns can be entered on the same pattern list. If any of them matches the incoming request, the corresponding action is taken.

Actions can be applied to specific protocols or to a set of protocols. If no protocol is configured, then the specified action will be taken for all the traffic that goes through the Content Engine.

Multiple patterns for the same pattern list must be entered on different lines.

Note The Rules Template configuration takes precedence over the **ip dscp** command, and the **url-filter** command takes precedence over the **rule** command to the extent that even the **rule no-block** command is executed only if the **url-filter** command has not blocked the request.

Note For execution order of rule patterns and for more information on pattern lists, see Chapter 13 of the *Cisco ACNS Software Configuration Guide for Locally Managed Deployments*.

Rules Template Processing Considerations

Actions and patterns have a predefined order of execution. A group of rules with the same action is always executed either before or after another group of rules with a different action. The **reset**, **block**, **rewrite**, and **redirect** rule actions support the following additional patterns: **request-line**, **referer**, and **user-agent** regular expressions. The request-line regular expression matches the first line of the request. The user-agent regular expression matches the User-Agent header value of the request. The referer regular expression matches the referer header value of the request. The order is not affected by the order in which the rules are entered using CLI commands.

Allow and **block** carry the same precedence. The order of execution depends on the order of configuration between **allow** and **block** actions. Other actions always take precedence over **allow**. For example, a **reset** action always takes precedence over **allow** regardless of the order of configuration.

Among rules of the same action, a predefined execution order exists among the rule patterns, which means that within a group of rules of the same action, one group of rules with the same pattern is always executed either before or after another group of rules with a different pattern.

Among all rules of the same action and of the same rules pattern, the rules are evaluated in a Last-Entered-First-Examined fashion (the reverse of the order in which the rules were entered). This order is not affected by the order in which the rules are entered using CLI commands.

Most actions do not have any parameters. Exceptions to this are **use-server**, **freshness-factor**, and **use-proxy**,

Examples

The following example shows that the Content Engine directs a request to abc.abc.com to the proxy server that has an IP address of 2.3.4.5 because the **use-proxy 2.3.4.5 *.abc.com** rule was entered last and is evaluated first, and the request will hit a match with that rule:

```
ContentEngine(config)# rule action use-proxy 1.2.3.4 abc.abc.com
ContentEngine(config)# rule action use-proxy 2.3.4.5 *.abc.com
```

The following example shows that the Content Engine directs a request to abc.abc.com to the proxy server that has an IP address of 1.2.3.4.:

```
ContentEngine(config)# rule action use-proxy 2.3.4.5 *.abc.com
ContentEngine(config)# rule action use-proxy 1.2.3.4 abc.abc.com
```

Tip In the ACNS 5.x software, if you enter the **show rule** EXEC command on a Content Engine, the rules are displayed randomly. If you enter the **show statistics rule** EXEC command, the rules are displayed in the order in which the rule actions are executed. You can use this command to see how a Content Engine processes the rules that you define for it.

Note For the execution order of rule actions, see Chapter 13 of the *Cisco ACNS Software Configuration Guide for Locally Managed Deployments*.

The following example shows that the Content Engine is configured to replace the internal.domain.com string in a request to the server named dummy:

```
ContentEngine(config)# rule rewrite header-field referer internal.domain.com dummy
```

The following example shows that if an empty string is given as a replacement pattern, then the referer header is stripped. This rule states that all requests, which have a referer header that indicates a corporate internal server in ABCBigCorp, strip the referer field so that the outside web server does not see the name of the corporate internal server.

```
ContentEngine(config)# rule rewrite header-field referer internal.abcbigcorp.com ""
```

The following example shows that the **rule pattern-list** command is configured to add a pattern to an existing pattern list to perform an action to be defined on destination IP address 172.16.25.25 using the dst-ip pattern:

```
ContentEngine(config)# rule pattern-list 10 dst-ip 172.16.25.25 255.255.255.0
ContentEngine# show rule pattern-list 10 all
Rules Template Configuration
```

```
----------------------------
Rule Processing Enabled

Pattern-Lists :

rule pattern-list 11 dst-ip 172.16.25.25 255.255.255.0
rule pattern-list 11 domain foo.com
ContentEngine#
```

The following example shows that the **rule action block** command is configured and associated with an existing pattern list:

```
ContentEngine(config)# rule action block pattern-list 10 protocol all
ContentEngine# show rule action block
Rules Template Configuration
----------------------------
Rule Processing Enabled

Actions :

rule action block pattern-list 10 protocol all
ContentEngine#
```

The following example shows that the **rule action block** command is configured and associated with an existing pattern list, which lists as its pattern the domain yahoo.com:

```
ContentEngine(config)# rule pattern-list 10 domain yahoo.com
ContentEngine# show rule pattern-list 10 all
Rules Template Configuration
----------------------------
Rule Processing Enabled

Pattern-Lists :

rule pattern-list 10 domain yahoo.com
ContentEngine(config)# rule action block pattern-list 10 protocol all
ContentEngine# show statistics rule http action block
Rules Template Statistics
-------------------------
Rule hit count = 3   Rule: rule action block pattern-list 10 protocol all
ContentEngine#
```

In this example, the request (using HTTP) to yahoo.com was denied three times.

Example of no-url-filtering Action

The following example shows how to use the **no-url-filtering** action to bypass URL filtering with Websense URL filtering. The **rule action no-url-filtering** command is specified and then associated with a specific pattern list (pattern list 100). The **domain** pattern type is added to the pattern list 100 to configure the Content Engine to match requests that have foo.com as the domain. In this scenario, Websense URL filtering has already been configured and enabled on the Content Engine.

```
ContentEngine (config)# rule action no-url-filtering pattern-list 100
ContentEngine (config)# rule pattern-list 100 domain .*foo.com
ContentEngine (config)# rule enable
```

Example of use-server Action

The following example shows that the Content Engine is a reverse proxy for www.abcbigcorp.com and is a proxy to the rest of the Internet. The IP address for the company's website (www.abcbigcorp.com) is actually the IP address of the Content Engine, and not the company's web site server. When the Content Engine receives the request http://www.abcbigcorp.com/main.html, its normal processing would be to obtain the IP address of www.abcbigcorp.com and send the request to that IP address. However, in this case, because the IP address of www.abcbigcorp.com is the IP address of the Content Engine, the administrator needs to prevent the Content Engine from sending the request to itself.

The administrator of CE1 can configure the following rule that will instruct CE1 to send such requests (for example, cache misses) to the web server for www.abcbigcorp.com where 1.2.3.4 is the IP address of the web server for www.abcbigcorp.com:

```
CE1(config)# rule use-server 1.2.3.4 80 domain www.abcbigcorp.com
```

Example of no-proxy Action

The following example shows that the **rule action block** command blocks all patterns specified with the **rule pattern-list** command:

Note In the following examples, it is assumed that all actions and patterns apply to all protocols unless specifically stated.

```
ContentEngine(config)# rule pattern-list 12 domain \.foo.com
ContentEngine(config)# rule action block pattern-list 12
```

The following example shows that the **rule action block** command (action) blocks all patterns specified with the **rule pattern-list 12** command:

```
ContentEngine(config)# rule pattern-list 12 domain \.foo.com
ContentEngine(config)# rule pattern-list 12 dst-ip 172.16.25.25 255.255.255.0
ContentEngine(config)# rule action block pattern-list 12
ContentEngine(config)#
```

The following example prevents caching of requests that match a URL request that contains the *cgi-bin* string:

```
ContentEngine(config)# rule pattern-list 13 url-regex \.*cgi-bin.*
ContentEngine(config)# rule action no-cache pattern-list 13
ContentEngine(config)#
```

The following example shows that to delete rules, use **no** in front of the rule creation command as follows:

```
ContentEngine(config)# no rule use-proxy foo.com 8080 pattern-list 13
```

The following example sets the freshness factor for MIME-type images:

```
ContentEngine(config)# rule pattern-list 13 mime-type image/.*
ContentEngine(config)# rule action freshness-factor 75 pattern-list 13
```

The following example denies a pattern type addition to an existing pattern list. The pattern type **url-regsub** cannot be associated with the **action freshness-factor** command.

```
ContentEngine(config)# rule pattern-list 13 url-regsub http://old-domain-name
http://new-domain-name
Configured Pattern type not valid for the associated action - freshness-factor
Following pattern-types are applicable for this action
        src-ip
        dst-ip
        dst-port
        mime-type
        url-regex
        domain
ContentEngine(config)#
```

The actions that are to be taken by the rules are configured through the **rule action** commands. Patterns that are to be matched to a particular pattern that you specify are configured through **rule pattern-list** commands.

The following example shows how patterns are ANDed by configuring patterns with same pattern list number and applying that pattern list to an action:

```
ContentEngine(config)# rule action block pattern-list 1
ContentEngine(config)# rule pattern-list 1 url-regex yahoo
ContentEngine(config)# rule pattern-list 1 dst-port 80
```

The following example sets the ToS value to minimize delay for outbound requests to a specified destination IP address 10.1.1.1:

```
ContentEngine(config)# rule action dscp server set-tos min-delay protocol all
ContentEngine(config)# rule pattern-list 2 dst-ip 10.1.1.1 255.255.255.255
```

The following example sets the ToS value to minimize delay for all outbound requests:

```
ContentEngine(config)# ip dscp server set-tos min-delay
```

Using the **ip** command, the following example uses the ToS or differentiated services code point (DSCP) value that was originally sent by the server (when the object was first fetched) for all future cache hit responses for the same object:

```
ContentEngine(config)# ip dscp client cache-hit match-server
ContentEngine(config)# rule action no-cache pattern-list 3 protocol all
ContentEngine(config)# rule pattern-list 3 url-regex \.*cgi-bin.*
ContentEngine(config)# rule pattern-list 4 dst-ip 172.31.120.0 255.255.192.0
  <cr>
```

Other options of the **rule** command work similarly to the preceding examples.

The following example redirects a request for old-domain-name that has been changed to new-domain-name:

```
Cache(config)# rule action redirect http://old-domain-name/ pattern-list 1 protocol http
Cache(config)# rule pattern-list 1 url-regsub http://old-domain-name/
http://new-domain-name/
```

The following example redirects requests from an IETF site to a site that is locally mirrored:

```
Cache(config)# rule action redirect http://www.ietf.org/rfc/(.*)  pattern-list 2 protocol
http
```

The following example shows that if the request URL is http://www.ietf.org/rfc/rfc1111.txt, the Content Engine rewrites the URL as http://wwwin-eng.cisco.com/RFC/RFC/rfc1111.txt and sends a 302 Temporary Redirect response with the rewritten URL in the Location header to the client. The browser automatically initiates a request to the rewritten URL.

```
Cache(config)# rule pattern-list 2 url-regsub http://www.ietf.org/rfc/(.*)
http://wwwin-eng.cisco.com/RFC/RFC/\1
```

The following example redirects all requests for linux.org to a local server in India that is closer to where the Content Engine is located:

```
Cache(config)# rule action redirect http://linux.org/(.*) pattern-list 3
protocol http
```

The following example shows that two URLs are to be matched if the pattern is **url-regsub**. If the URLs that are given in the action configuration are invalid, a warning is displayed during the configuration of this rule. The action URL is taken when the header field patterns are configured.

```
Cache(config)# rule pattern-list 3 url-regsub http://linux.org/(.*)
http://linux.org.in/\1
```

The following example rewrites requests from an IETF site to a site that is locally mirrored:

```
Cache(config)# rule action rewrite pattern-list 5 protocol http
Cache(config)# rule pattern-list 5 url-regsub http://www.ietf.org/rfc/.*
http://wwwin-eng.cisco.com/RFC/$1
Cache(config)# rule action redirect-url-for-cdn pattern-list 1
Cache(config)# rule pattern-list 1 url-regsub xyz abc
Configured Pattern type not valid for the associated action - redirect-url-for-cdn
Following pattern-types are applicable for this action
        src-ip
        url-regex
Cache(config)# rule pattern-list 1 url-regex roti
```

The following example shows that any requests from the source IP address (src-ip) 172.16.53.88 are not authenticated:

```
ContentEngine(config)# rule enable
ContentEngine(config)# rule action no-auth pattern-list 1 protocol all
ContentEngine(config)# rule pattern-list 1 src-ip 172.16.53.88 255.255.255.255
```

Note If the ACNS 5.x software is configured for authentication and SmartFilter URL filtering, requests that are allowed to bypass authentication will also bypass the SmartFilter URL filter.

In the following example, any requests to dst-ip 172.22.73.34 are not authenticated:

```
ContentEngine(config)# rule action no-auth pattern-list 2 protocol all
ContentEngine(config)# rule pattern-list 2 dst-ip 172.22.73.34 255.255.255.255
```

In the following example, any requests with the destination port 9090 are not authenticated:

```
ContentEngine(config)# rule action no-auth pattern-list 3 protocol all
ContentEngine(config)# rule pattern-list 3 dst-port 9090
```

In the following example, any requests with cgi-bin in the URL are not authenticated:

```
ContentEngine(config)# rule action no-auth pattern-list 4 protocol all
ContentEngine(config)# rule pattern-list 4 url-regex .*cgi-bin.*
```

In the following example, any requests with cisco.com as the domain are not authenticated. (For example, requests for roti.cisco.com or badal.cisco.com are excluded from the Content Engine authentication.)

```
ContentEngine(config)# rule action no-auth pattern-list 5 protocol all
ContentEngine(config)# rule pattern-list 5 domain cisco.com
```

The following example bypasses requests with cisco.com as the domain from URL filtering:

```
CONTENTENGINE(config)# rule action no-url-filtering pattern-list 6 protocol all
CONTENTENGINE(config)# rule pattern-list 6 domain cisco.com
```

Related Commands

bypass static
clear statistics rule
http proxy outgoing
proxy-protocols outgoing exclude
show rule
show statistics rule

script

To execute a script provided by Cisco or check the script for errors, use the **script** EXEC command.

script {**check** | **execute**} *file_name*

Syntax Description

check	Checks the validity of the script.
execute	Executes the script. The script file must be a sysfs file in the current directory.
file_name	Name of the script file.

Defaults

No default behavior or values

Command Modes

EXEC

Usage Guidelines

The **script** EXEC command opens the script utility, which allows you to execute Cisco-supplied scripts or check errors in those scripts. The script utility can read standard terminal input from the user if the script you run requires inputs from the user.

Note The script utility is designed to run only Cisco-supplied scripts. You cannot execute script files that lack Cisco signatures or that have been corrupted or modified.

Examples

The following example checks for errors in the script file test_script.pl:

```
CONTENTENGINE# script check foo.script

 Script file foo.script is valid.

CONTENTENGINE#
```

setup

To configure basic configuration settings (general settings, device network settings, and disk configuration) on the Content Engine and a set of commonly used caching services, use the **setup** EXEC command. You can also use the **setup** EXEC command to complete basic configuration after upgrading to the ACNS 5.2 software and later releases.

setup

Syntax Description

This command has no arguments or keywords.

Defaults

No default behavior or values

Command Modes

EXEC

Usage Guidelines

For complete instructions on using the **setup** command, see the *Cisco ACNS Software Configuration Guide for Locally Managed Deployments*.

Examples

The following example shows the first screen of the wizard when you enter the **setup** EXEC command on a Content Engine running the ACNS software:

```
ContentEngine# setup


Here is the current profile of this device

CDN device                    : Yes

Do you want to change this (y/n) [n]:






Press the ESC key at any time to quit this session
```

show aaa accounting

To display the authentication, authorization, and accounting (AAA) configuration, use the **show aaa accounting** EXEC command.

show aaa accounting

Syntax Description

accounting	Displays the AAA accounting configuration for the following accounting types: • EXEC shell • Command (for normal users and superusers) • System

Defaults

No default behavior or values

Command Modes

EXEC

Examples

Table 2-23 describes the fields shown in the **show aaa accounting** display.

Table 2-23 show aaa accounting Field Descriptions

Field	Description
Accounting Type	Displays the AAA accounting configuration for the following types of user accounts: Exec Command level 0 Command level 15 System
Record Event(s)	Displays the configuration of the AAA accounting notice that is sent to the accounting server.
stop-only	The WAAS device sends a stop record accounting notice at the end of the specified activity or event to the TACACS+ accounting server.
start-stop	The WAAS device sends a start record accounting notice at the beginning of an event and a stop record at the end of the event to the TACACS+ accounting server. The start accounting record is sent in the background. The requested user service begins regardless of whether or not the start accounting record was acknowledged by the TACACS+ accounting server.
wait-start	The WAAS device sends both a start and a stop accounting record to the TACACS+ accounting server. However, the requested user service does not begin until the start accounting record is acknowledged. A stop accounting record is also sent.

Table 2-23 show aaa accounting Field Descriptions (continued)

Field	Description
disabled	Accounting is disabled for the specified event.
Protocol	Displays the accounting protocol that is configured.

Related Commands

aaa accounting

show access-lists

To display the access control list (ACL) configuration, use the **show access-lists** EXEC command.

show access-lists 300

Syntax Description

300	Displays group name-based access control lists.

Defaults

No default behavior or values

Command Modes

EXEC

Examples

Table 2-24 describes the fields shown in the **show access-lists 300** display.

Table 2-24 show access-lists Field Descriptions

Field	Description
Access Control List Configuration	
Access Control List is enabled	Configuration status of the access control list option.
Groupname and username-based List	Lists the group name-based access control lists.

Related Commands

access-lists 300

show acquirer

To display the acquirer information and progress of content acquisition for a specified channel number or name, use the **show acquirer** EXEC command.

show acquirer [**channels** [**channel-id** *channel-num* | **channel-name** *channel-name*] | **progress** [**channel-id** *channel-num* | **channel-name** *channel-name*] | **proxy authentication**]

Syntax Description

channels	(Optional) Displays acquirer information for channels.
channel-id	(Optional) Displays channel information from the specified channel ID.
channel-num	Channel number (0–4294967295).
channel-name	(Optional) Displays channel information from the specified channel name descriptor.
channel-name	Channel name.
progress	(Optional) Displays the acquisition progress for the specified channel.
channel-id	(Optional) Displays the acquisition progress of the specified channel ID.
channel-name	(Optional) Displays the acquisition progress of the specified channel name descriptor.
proxy	(Optional) Displays the proxy information for the acquirer.
authentication	Displays the proxy authentication details for the acquirer.

Defaults

No default behavior or values

Command Modes

EXEC

Usage Guidelines

The **show acquirer channels** command displays channel information for a channel that is specified either by ID or by name. If more than one channel with an identical name exists and each subscribes to a different website, the channel information for all channels with that specified name, for which this Content Engine is the root Content Engine, is displayed.

The **show acquirer progress** command displays information for all channels for which the Content Engine is the root Content Engine. It displays the number of acquired objects in relation to the total number of objects for both a single item or crawler jobs. When an acquisition is in progress, it displays the number of data bytes currently being downloaded in relation to the total size of the object and its URL. The **show acquirer progress** command also displays information about the authentication that allows the acquirer to access content through a transparent or nontransparent proxy server.

The **show acquirer proxy authentication** command displays the proxy authentication configuration for the acquirer if you have enabled content acquisition through a proxy server and proxy authentication is configured. Use the **acquirer proxy authentication outgoing** global configuration command to configure authentication when you enable content acquisition through a proxy server. You must first configure the proxy host and the port using the **http proxy outgoing host** global configuration command.

When you enable content acquisition through a proxy server, you can provide the proxy configuration and proxy authentication information in the manifest file. If the proxy and proxy authentication are configured in the manifest file, the **show acquirer proxy authentication** command does not display any proxy details.

Examples

The **show acquirer channels** EXEC command displays basic details about the channels for which a Content Engine is the root and for which it is acquiring data.

Table 2-25 describes the fields shown in the **show acquirer channels** display.

Table 2-25 show acquirer channels Field Descriptions

Field	Description
Acquirer information for all channels	
Channel-id	Numerical identifier for the channel.
Channel-Name	Name for the channel.
WebSite-Name	Name of the website for the channel.
Root-CE-Type	Root Content Engine type. Values are Configured or Temporary. Configured—The configured root Content Engine. Temporary—The temporary root Content Engine appointed due to root Content Engine failover.
State	Operation status of the channel. Values are Enabled or Disabled. Enabled—This channel is active. Disabled—This channel has been disabled.
Disk Quota	Total disk space (in megabytes) allocated for this channel.
Origin FQDN	Fully qualified domain name (FQDN) of the origin server that is configured for this channel.
Request Routed FQDN	FQDN to be used for Content Router redirected requests for a channel.
Channel Priority	Priority rating of the channel. Values are as follows: • 250—low priority • 500—medium priority • 750—high priority
Manifestfile-TTL	Interval in minutes at which the root Content Engine will check the manifest file for any changes.
Manifestfile-URL	URL of the manifest file used for this channel.

The **show acquirer progress** EXEC command displays the progress of the acquirer for a specified channel. If a specific channel is not mentioned, the display shows the progress for all the channels for which the Content Engine is the root.

Table 2-26 describes the fields shown in the **show acquirer progress** display.

Table 2-26 show acquirer progress Field Descriptions

Field	Description
Acquirer Progress Information for channel:*channel-id*	
Channel-id	Numerical identifier for the channel.
Channel-Name	Name for the channel.
Acquired Single Items	Total number of single items completed out of all of the single items specified in the manifest. For example, 200/301 shows that all 200 items out of a total of 301 items have been acquired.
Acquired Crawl Items	Total number of links with crawling completed out of the total crawlable items for each crawling task specified in the manifest, along with the starting URL.
Download Size (Bytes)	Current URL fetched by the acquirer for the channel, if applicable, along with the file size details.

The following example shows the output from the **show acquirer proxy authentication** command when there are no proxies configured using the **acquirer proxy authentication** global configuration command:

```
ContentEngine# show acquirer proxy authentication
No proxy authentication information configured
ContentEngine#
```

The following example shows the output from the **show acquirer proxy authentication** command after configuring the proxy using the **acquirer proxy authentication** global configuration command:

```
ContentEngine# show acquirer proxy authentication
acquirer proxy authentication outgoing 172.28.225.29 8080 admin password **** ntlm
My-Domain basic-auth-disable
acquirer proxy authentication transparent admin password **** ntlm My-Domain
basic-auth-disable
ContentEngine#
```

Related Commands

acquirer (global configuration mode)
http proxy outgoing
show statistics acquirer

show alarms

To display information on various types of alarms, their status, and history, use the **show alarms** EXEC command.

show alarms [**critical** [**detail** [**support**]] | **detail** [**support**] | **history** [[*start_num* [*end_num* [**detail** [**support**]] | **detail** [**support**]]] | **critical** [*start_num* [*end_num* [**detail** [**support**]] | **detail** [**support**]]] | **detail** [**support**] | **major** [*start_num* [*end_num* [**detail** [**support**]] | **detail** [**support**]]] | **minor** [*start_num* [*end_num* [**detail** [**support**]]] | **detail** [**support**]]] | **major** [**detail** [**support**]] | **minor** [**detail** [**support**]] | **status**]

Syntax Description

critical	(Optional) Displays critical alarm information.
detail	(Optional) Displays detailed information for each alarm.
support	(Optional) Displays additional information about each alarm.
history	(Optional) Displays information about the history of various alarms.
start_num	(Optional) Alarm number that appears first in the alarm history.
end_num	(Optional) Alarm number that appears last in the alarm history.
major	(Optional) Displays information about major alarms.
minor	(Optional) Displays information about minor alarms.
status	(Optional) Displays the status of various alarms and alarm overload settings.

Defaults

No default behavior or values.

Command Modes

EXEC

Usage Guidelines

In the ACNS 5.2 software and later releases, the Node Health Manager feature is supported. The Node Health Manager enables ACNS applications to raise alarms to draw attention in error/significant conditions. The Node Health Manager, which is the data repository for such alarms, aggregates the health and alarm information for the applications, services (for example, the cache service) and resources (for example, disk drives) that are being monitored on the Content Engine. For example, this new feature gives you a mechanism to determine if a monitored application (for example, the HTTP proxy caching service) is alive on the Content Engine. These alarms are referred to as ACNS software alarms.

The ACNS software uses SNMP to report error conditions by generating SNMP traps. In the ACNS software, the following Content Engine applications can generate an ACNS software alarm:

- Node Health Manager (Alarm overload condition and Node Manager aliveness)
- Node Manager for service failures (aliveness of monitored applications)
- System Monitor (sysmon) for disk failures

The three levels of alarms in the ACNS software are as follows:

- Critical—Alarms that affect the existing traffic through the Content Engine and are considered fatal (the Content Engine cannot recover and continue to process traffic).
- Major—Alarms that indicate a major service (for example, the cache service) has been damaged or lost. Urgent action is necessary to restore this service. However, other node components are fully functional and the existing service should be minimally impacted.
- Minor—Alarms that indicate that a condition that will not affect a service has occurred but that corrective action is required in order to prevent a serious fault from occurring.

You can configure alarms using the **snmp-server enable traps alarms** global configuration command.

Use the **show alarms critical** EXEC command to display the current critical alarms being generated by the ACNS software applications. Use the **show alarms critical detail** EXEC command to display additional details for each of the critical alarms being generated. Use the **show alarms critical detail support** EXEC command to display an explanation about the condition that triggered the alarm and how you can find out the cause of the problem. Similarly, you can use the **show alarms major** and **show alarms minor** EXEC commands to display the details of major and minor alarms.

Use the **show alarms history** EXEC command to display a history of alarms that have been raised and cleared by the ACNS software on the Content Engine. The ACNS software retains the last 100 alarm raise and clear events only.

Use the **show alarm status** EXEC command to display the status of current alarms and the Content Engine's alarm overload status and alarm overload configuration.

Examples

Table 2-27 describes the fields shown in the **show alarms history** display.

Table 2-27 show alarms history Field Descriptions

Field	Description
Op	Operation status of the alarm. Values are R–Raised or C–Cleared.
Sev	Severity of the alarm. Values are Cr–Critical, Ma–Major, or Mi–Minor.
Alarm ID	Type of event that caused the alarm.
Module/Submodule	Software module affected.
Instance	Object that this alarm event is associated with. For example, for an alarm event with the Alarm ID disk_failed, the instance would be the name of the disk that failed. The Instance field does not have pre-defined values and is application specific.

Table 2-28 describes the fields shown in the **show alarms status** display.

Table 2-28 show alarms status Field Descriptions

Field	Description
Critical Alarms	Number of critical alarms.
Major Alarms	Number of major alarms.
Minor Alarms	Number of minor alarms.
Overall Alarm Status	Aggregate status of alarms.

Table 2-28 show alarms status Field Descriptions (continued)

Field	Description
Device is NOT in alarm overload state.	Status of the device alarm overload state.
Device enters alarm overload state @ 999 alarms/sec.	Threshold number of alarms per second at which the device enters the alarm overload state.
Device exits alarm overload state @ 99 alarms/sec.	Threshold number of alarms per second at which the device exits the alarm overload state.
Overload detection is ENABLED.	Status of whether overload detection is enabled on the device.

Related Commands

alarm overload-detect
snmp-server enable traps

show arp

To display the Address Resolution Protocol (ARP) table, use the **show arp** EXEC command.

show arp

Syntax Description

This command has no arguments or keywords.

Defaults

No default behavior or values

Command Modes

EXEC

Examples

The **show arp** command displays the Internet-to-Ethernet address translation tables of the Address Resolution Protocol. Without flags, the current ARP entry for the hostname is displayed.

Table 2-29 describes the fields shown in the **show arp** display.

Table 2-29 show arp Field Descriptions

Field	Description
Protocol	Type of protocol.
Address	Ethernet address of the hostname.
Flags	Current ARP flag status.
Hardware Addr	Hardware Ethernet address given as six hexadecimal bytes separated by colons.
Type	Type of wide-area network.
Interface	Type of Ethernet interface.

show authentication

To display the authentication configuration, use the **show authentication** EXEC command.

show authentication {**http-request** | **user**}

Syntax Description

http-request	Displays the authentication configuration for HTTP requests.
user	Displays the authentication configuration for the user login to the system.

Defaults

No default behavior or values

Command Modes

EXEC

Usage Guidelines

The ACNS 5.x software caching service supports NTLM, LDAP, TACACS+, and RADIUS server HTTP request authentication. An HTTP request authenticates a user's domain and password with a preconfigured primary domain controller before allowing requests from the user to be served by the Content Engine. To display the authentication configuration for an HTTP request for an NTLM, LDAP, TACACS+, and RADIUS server, use the **show authentication http-request** command.

To display the local and TACACS+ authentication configuration for the user login, use the **show authentication user** command.

When the Content Engine authenticates a user through an NTLM, LDAP, TACACS+, or RADIUS server, a record of the authentication is stored locally. As long as the entry is stored, subsequent attempts to access restricted Internet content by the same user do not require additional server lookups.

Examples

The following sample output displays the HTTP request authentication configuration for the LDAP, RADIUS, TACACS+, and NTLM servers:

```
ContentEngine# show authentication http-request
HTTP Request Authentication via:
--------------------------------
LDAP server: disabled
RADIUS server: disabled
TACACS+ server: disabled
NTLM server: disabled
```

Table 2-30 describes the fields shown in the **show authentication user** display.

Table 2-30 show authentication user Field Descriptions

Field	Description
Login Authentication: Console/Telnet/Ftp/SSH Session	Displays which authentication service is enabled for login authentication and the configured status of the service.

Table 2-30 show authentication user Field Descriptions (continued)

Field	Description
Windows domain	Operation status of the authentication service. Values are enabled or disabled. Priority status of each authentication service. Values are primary, secondary, or tertiary.
RADIUS	
TACACS+	
Local	
Configuration Authentication: Console/Telnet/Ftp/SSH Session	Displays which authentication service is enabled for configuration authentication and the configured status of the service.
Windows domain	Operation status of the authentication service. Values are enabled or disabled. Priority status of each authentication service. Values are primary, secondary, or tertiary.
RADIUS	
TACACS+	
Local	

Related Commands

authentication configuration
authentication login
clear statistics authentication
show http authentication
show statistics authentication

show auto-register

To display the status of the automatic registration of a Content Engine or Content Router with the Content Distribution Manager, use the **show auto-register** EXEC command.

show auto-register

Syntax Description

This command has no arguments or keywords.

Defaults

No default behavior or values

Command Modes

EXEC

Examples

Table 2-31 describes the output in the **show auto-register** display.

Table 2-31 show auto-register Field Description

Field	Description
Auto registration is enabled.	Configuration status of the autoregistration feature.
Auto registration is disabled.	

Related Commands

auto-register enable

show bandwidth

To display the bandwidth allocated to a particular device, use the **show bandwidth** EXEC command.

show bandwidth [**cisco-streaming-engine** | **http** | **media-real** | **real-subscriber** | **wmt**]

Syntax Description

cisco-streaming-engine	(Optional) Displays Cisco Streaming Engine bandwidth settings.
http	(Optional) Displays HTTP bandwidth settings.
media-real	(Optional) Displays RealProxy bandwidth settings.
real-subscriber	(Optional) Displays RealSubscriber bandwidth settings.
wmt	(Optional) Displays WMT bandwidth settings.

Defaults

No default behavior or values

Command Modes

EXEC

Examples

Table 2-32 describes the fields shown in the **show bandwidth** display.

Table 2-32 show bandwidth Field Descriptions

Field	Description
MODULE	Types of application servers for which bandwidth allocation is displayed. They are as follows: • **wmt incoming** is for incoming WMT streaming content requests from end users. • **wmt outgoing** is for outgoing WMT media from Content Engines. • **real-proxy incoming** is for incoming RealProxy requests from end users. • **real-proxy outgoing** is for outgoing RealProxy traffic from Content Engines. • **real-server** is for streaming RealMedia pre-positioned content to end users. • **cisco-streaming-engine** is for streaming content in response to RTSP requests from end users. • **http** is for sending content in response to HTTP requests from end users.
Bandwidth Kbps	Maximum amount of bandwidth that you want allowed in kilobits per second (kbps) for a particular period of time.
Start Time	Time of the day for the bandwidth rate setting to begin, using a 24-hour clock in local time on the Content Engine (hh:mm).

Table 2-32 show bandwidth Field Descriptions (continued)

Field	Description
End Time	Time of the day for the bandwidth rate setting to end, using a 24-hour clock in local time on the Content Engine (hh:mm).
Default Bandwidth Kbps	Amount of default bandwidth (in kbps). The default bandwidth is the amount of bandwidth associated with each content service type when there is no scheduled bandwidth.
Max Bandwidth Kbps	Maximum bandwidth (in kbps) permitted by the system license. This bandwidth specifies the upper limit of allowable bandwidth.

Related Commands

bandwidth

show banner

To display information on various types of banners, use the **show banner** EXEC command.

show banner

Syntax Description

This command has no arguments or keywords.

Defaults

No default behavior or values.

Command Modes

EXEC

Examples

Table 2-33 describes the fields shown in the **show banner** display.

Table 2-33 show banner Field Descriptions

Field	Description
Banner is enabled.	Configuration status of the banner feature.
MOTD banner is: abc	(Message of the day) Displays the configured message of the day.
Login banner is: acb	Displays the configured login banner.
Exec banner is: abc	Displays the configured EXEC banner.

Related Commands

banner

show bitrate

To display the bit rate allocated to a particular device, use the **show bitrate** EXEC command.

show bitrate [**http** | **wmt**]

Syntax Description

http	(Optional) Displays HTTP bit-rate settings.
wmt	(Optional) Displays WMT bit-rate settings.

Defaults

No default behavior or values

Command Modes

EXEC

Examples

Table 2-34 describes the fields shown in the **show bitrate** display.

Table 2-34 show bitrate Field Descriptions

Field	Description
MODULE	Types of application servers for which the bit rate is displayed: • **http** is the maximum pacing bit rate in kilobits per second (kbps) for large files sent using the HTTP protocol. • **wmt outgoing** is the maximum bit rate per WMT stream that can be served by the Content Engine. • **wmt incoming** is the maximum bit rate per WMT stream that can be received by the Content Engine.
Default Bitrate Kbps	Bit rate associated with the application servers when the bit rate has not been configured on the Content Engine.
Configured Bitrate Kbps	Bit rate configured on the Content Engine in kbps.

Related Commands

bitrate

show bypass

To display bypass configuration information, use the **show bypass** EXEC command.

show bypass [**list** | **statistics** {**auth-traffic** | **load**} | **summary**]

Syntax Description

list	(Optional) Specifies the bypass list entries.
statistics	(Optional) Specifies the IP bypass statistics.
auth-traffic	Specifies the authenticated traffic bypass statistics.
load	Specifies the load bypass statistics.
summary	(Optional) Specifies the summary of bypass information.

Defaults

No default behavior or values

Command Modes

EXEC

Usage Guidelines

The total number of bypass entries is equal to the number of dynamic bypass entries plus the number of static bypass entries, which is equal to 4960. The maximum number of static bypass entries is 50.

Examples

Table 2-35 describes the fields shown in the **show bypass** display.

Table 2-35 show bypass Field Descriptions

Field	Description
Total number of HTTP connections bypassed	Total number of HTTP connections bypassed.
Connections bypassed due to system overload	Number of connections bypassed due to system overload.
Connections bypassed due to authentication issues	Number of connections bypassed due to authentication issues.
Connections bypassed to facilitate error transparency	Number of connections bypassed to facilitate error transparency.
Connections bypassed due to static configuration	Number of connections bypassed due to static configuration.
Total number of entries in the bypass list	Total number of entries in the bypass list.
Number of Authentication bypass entries	Number of authentication bypass entries.
Number of Error bypass entries	Number of error bypass entries.
Number of Static Configuration entries	Number of static configuration entries.

Table 2-36 describes the fields shown in the **show bypass list** display.

Table 2-36 show bypass list Field Descriptions

Field	Description
Client	IP address and port of the client. For any client with this IP address, the WAE will not process the packet, but will bypass it and send it back to the router.
Server	IP address and port of the server.
Entry type	Type of bypass list entry. The Entry type field contains one of the following values: static-config, auth-traffic, server-error, or accept. A static-config entry is a bypass list entry that is user-configured. An auth-traffic entry is a type of dynamic entry that the internal software adds automatically when the server requests authentication.

The **show bypass statistics auth-traffic** EXEC command displays authentication traffic bypass statistics.

Table 2-37 describes the fields shown in the command display.

Table 2-37 show bypass statistics auth-traffic Field Descriptions

Field	Description
Authentication Bypass Statistics	
HTTP connections bypassed due to authentication	Number of HTTP connections bypassed due to authentication.
Number of authentication bypass entries	Number of authentication bypass entries.

The **show bypass statistics load** command displays overload bypass statistics.

Table 2-38 describes the fields shown in the command display.

Table 2-38 show bypass statistics load Field Descriptions

Field	Description
Load Bypass Statistics	
Load Bypass is enabled	Configuration status for load bypass.
System bypass mode - not available	Availability status of system bypass mode.
Number of bypassed buckets not available	Number of bypassed buckets when system bypass mode is available.
Number of bypassed connections not available	Number of bypassed connections when system bypass mode is available.
Number of transitions from Bypass mode to Normal mode	Number of transitions from bypass mode to normal mode.
Number of transitions from Normal mode to Bypass mode	Number of transitions from normal mode to bypass mode.
MODULE	Name of the module.

Table 2-38 ***show bypass statistics load Field Descriptions (continued)***

Field	Description
Normal	Number of bypassed connections in normal mode.
Overload	Number of connections bypassed due to system overload.
Inundated	Number of inundated connections.
Cum Secs	Number of cumulative seconds for connections to this module.
Current State	Current state of the connection to this module.

The **show bypass summary** command displays a bypass summary that includes the number of entries in the bypass list.

Table 2-39 describes the fields shown in the command display.

Table 2-39 ***show bypass summary Field Descriptions***

Field	Description
Total number of HTTP connections bypassed	Total number of HTTP connections bypassed.
Connections bypassed due to system overload	Connections bypassed due to system overload.
Connections bypassed due to authentication issues	Connections bypassed due to authentication issues.
Connections bypassed due to facilitate error transparency	Connections bypassed due to facilitate error transparency.
Connections bypassed due to static configuration	Connections bypassed due to static configuration.
Total number of entries in the bypass list	Total number of entries in the bypass list.
Number of Authentication bypass entries	Number of authentication bypass entries.
Number of Error bypass entries	Number of error bypass entries.
Number of Static Configuration entries	Number of static configuration entries.

Related Commands

bypass
clear bypass
show bypass statistics

show cdnfs

To display ACNS network file system (cdnfs) information, use the **show cdnfs** EXEC command.

show cdnfs volumes

Syntax Description

volumes	Displays ACNS network file system volumes.

Defaults

No default behavior or values

Command Modes

EXEC

Examples

Table 2-40 describes the fields shown in the **show cdnfs volumes** display.

Table 2-40 show cdnfs volumes Field Descriptions

Field	Description
cdnfs 00–04	ACNS network file system and disk number.
nnnnnnKB	Size of the volume in kilobytes.

Related Commands

cdnfs
disk (EXEC mode)
show disks
show disks details
show statistics cdnfs

show cdn-statistics

To display Content Engine and device group statistics for the ACNS network, use the **show cdn-statistics** EXEC command. This command is available only on Content Distribution Manager devices.

show cdn-statistics {**cisco-streaming-engine** {**content-engines** | **device-group-name** *groupname* | **device-groups**} | **http** {**content-engines** | **device-group-name** *groupname* | **device-groups**} | **real-proxy** {**content-engines** | **device-group-name** *groupname* | **device-groups**} | **wmt** {**content-engines** | **device-group-name** *groupname* | **device-groups**}}

Syntax Description

cisco-streaming-engine	Displays the Cisco Streaming Engine statistics.
content-engines	Displays statistical data for each Content Engine.
device-group-name	Displays statistical data for Content Engines in the specified device group.
groupname	Name of the device group.
device-groups	Displays statistical data for each device group.
http	Displays HTTP caching statistics.
real-proxy	Displays RealProxy statistics.
wmt	Displays WMT statistics.

Defaults

No default behavior or values

Command Modes

EXEC

Examples

The **show cdn-statistics http content-engines** command displays the HTTP caching statistics for all the Content Engines registered to the Content Distribution Manager.

Table 2-41 describes the fields shown in the command display.

Table 2-41 show cdn-statistics http content-engines Field Descriptions

Field	Description
Name	Name of the Content Engine displayed.
Requests/Sec	Number of transactions handled by the Content Engine per second.
Bytes/Sec	Number of bytes sent and received that were handled by the Content Engine during the transaction.
Service Time	Amount of time used to handle the service transactions.
Hit Rate	Number of cache or cdnfs hits per second.
Updated	Time of the last update.

The **show cdn-statistics cisco-streaming-engine content-engines** EXEC command displays the streaming statistics for all the Content Engines registered to the Content Distribution Manager.

Table 2-42 describes the fields shown in the command display.

Table 2-42 show cdn-statistics cisco-streaming-engine content-engines Field Descriptions

Field Name	Description
Name	Name of the Content Engine displayed.
Average Bandwidth In	Average value of incoming bandwidth.
Average Bandwidth Out	Average value of outgoing bandwidth.
Current Bandwidth In	Current incoming bandwidth.
Current Bandwidth Out	Current outgoing bandwidth.
Total Bytes In	Total incoming bytes served.
Total Bytes Out	Total outgoing bytes served.
Total Packets In	Total number of incoming packets.
Total Packets Out	Total number of outgoing packets.
RTSP Connections	Total number of active RTSP connections.
All Connections	Total number of all the active connections.
Updated	Time of last update.

The **show cdn-statistics wmt content-engines** EXEC command displays the Windows Media streaming statistics for all the Content Engines registered to the Content Distribution Manager.

Table 2-43 describes the fields shown in the command display.

Table 2-43 show cdn-statistics wmt content-engines Field Descriptions

Field Name	Description
Name	Name of the Content Engine displayed.
Concurrent Requests	Total number of concurrent request served.
Kbits/Sec	Request served in kilobits per second.
Cache Hit Rate	Number of cache hits per second.
Updated	Time of last update.

The **show cdn-statistics real-proxy content-engines** EXEC command displays the RTSP streaming statistics for all the Content Engines registered to the Content Distribution Manager.

Table 2-44 describes the fields shown in the command display.

Table 2-44 cdn-statistics real-proxy content-engines Field Descriptions

Field Name	Description
Name	Name of the Content Engine displayed.
Requests	Total number of request served.
Bytes	Total number of bytes served.
Hit Rate	Number of cache hits per second.
Updated	Last updated time.

show cdp

To display Cisco Discovery Protocol (CDP) configuration information, use the **show cdp** EXEC command.

show cdp [**entry** *neighbor* [**protocol** | **version** [**protocol**]] | **holdtime** | **interface** [**FastEthernet** *slot/port* | **GigabitEthernet** *slot/port*] | **neighbors** [**detail** | **FastEthernet** *slot/port* [**detail**] | **GigabitEthernet** *slot/port* [**detail**]] | **run** | **timer** | **traffic**]

Syntax Description

entry	(Optional) Displays information for a specific neighbor entry.
neighbor	Name of the CDP neighbor entry.
protocol	(Optional) Displays the CDP protocol information.
version	(Optional) Displays the CDP version.
holdtime	(Optional) Displays the length of time that CDP information is held by neighbors.
interface	(Optional) Displays the interface status and configuration.
FastEthernet	(Optional) Displays the Fast Ethernet configuration.
slot/port	Fast Ethernet slot (0–3) and port number.
GigabitEthernet	(Optional) Displays the Gigabit Ethernet configuration.
slot/port	Gigabit Ethernet slot (1–2) and port number.
neighbors	(Optional) Displays CDP neighbor entries.
detail	(Optional) Displays detailed neighbor entry information.
FastEthernet	(Optional) Displays neighbor Fast Ethernet information.
slot/port	Neighbor Fast Ethernet slot (0–3) and port number.
detail	(Optional) Displays the detailed neighbor Fast Ethernet network information.
GigabitEthernet	(Optional) Displays neighbor Gigabit Ethernet information.
slot/port	Neighbor Gigabit Ethernet slot (1–2) and port number.
detail	(Optional) Displays the detailed Gigabit Ethernet neighbor network information.
run	(Optional) Displays the CDP process status.
timer	(Optional) Displays the time when CDP information is resent to neighbors.
traffic	(Optional) Displays CDP statistical information.

Defaults

No default behavior or values

Command Modes

EXEC

Examples

The **show cdp** command displays CDP information regarding how frequently CDP packets are resent to neighbors, the length of time that CDP packets are held by neighbors, the disabled status of CDP Version 2 multicast advertisements, CDP Ethernet interface ports, and general CDP traffic information.

Table 2-45 describes the fields shown in the **show cdp** display.

Table 2-45 show cdp Field Descriptions

Field	Description
Sending CDP packets every XX seconds	Interval (in seconds) between transmissions of CDP advertisements. This field is controlled by the **cdp timer** command.
Sending a holdtime value of XX seconds	Time (in seconds) that the device directs the neighbor to hold a CDP advertisement before discarding it. This field is controlled by the **cdp holdtime** command.
Sending CDPv2 advertisements is XX	Transmission status for sending CDP Version-2 type advertisements. Possible values are enabled or disabled.

Table 2-46 describes the fields shown in the **show cdp entry** *neighbor* display.

Table 2-46 show cdp entry Field Descriptions

Field	Description
Device ID	Name of the neighbor device and either the MAC address or the serial number of this device.
Entry address(es)	
IP address	IP address of the neighbor device.
CLNS address	Non-IP network address. Depends on type of neighbor.
DECnet address	Non-IP network address. Depends on type of neighbor.
Platform	Product name and number of the neighbor device.
Interface	Protocol being used by the connectivity media.
Port ID (outgoing port)	Port number of the port on the neighbor device.
Capabilities	Capability code discovered on the neighbor device. This is the type of the device listed in the CDP Neighbors table. Possible values are: R—Router T—Transparent bridge B—Source-routing bridge S—Switch H—Host I—IGMP device r—Repeater
Holdtime	Time (in seconds) that the current device will hold the CDP advertisement from a transmitting router before discarding it.
Version	Software version running on the neighbor device.

Table 2-47 describes the fields shown in the **show cdp entry** *neighbor* **protocol** display.

Table 2-47 show cdp entry protocol Field Descriptions

Field	Description
Protocol information for XX	Name or identifier of the neighbor device.
IP address	IP address of the neighbor device.
CLNS address	Non-IP network address. Depends on type of neighbor.
DECnet address	Non-IP network address. Depends on type of neighbor.

Table 2-48 describes the fields shown in the **show cdp entry** *neighbor* **version** display.

Table 2-48 show cdp entry version Field Descriptions

Field	Description
Version information for XX	Name or identifier of the neighbor device.
Software, Version	Software and version running on the neighbor device.
Copyright	Copyright information for the neighbor device.

Table 2-49 describes the field in the **show cdp holdtime** display.

Table 2-49 show cdp holdtime Field Descriptions

Field	Description
XX seconds	Time (in seconds) that the current device will hold the CDP advertisement from a transmitting router before discarding it.

Table 2-50 describes the fields shown in the **show cdp interface** display.

Table 2-50 show cdp interface Field Descriptions

Field	Description
Interface_slot/port is XX	Operation status of the CDP interface. Values are up or down.
CDP protocol is XX	Protocol being used by the connectivity media.

Table 2-51 describes the fields shown in the **show cdp neighbors** display.

Table 2-51 show cdp neighbors Field Descriptions

Field	Description
Device ID	Configured ID (name), MAC address, or serial number of the neighbor device.
Local Intrfce	(Local Interface) Protocol being used by the connectivity media.

Table 2-51 show cdp neighbors Field Descriptions (continued)

Field	Description
Holdtime	Time (in seconds) that the current device will hold the CDP advertisement from a transmitting router before discarding it.
Capability	Capability code discovered on the device. This is the type of the device listed in the CDP Neighbors table. Possible values are: R—Router T—Transparent bridge B—Source-routing bridge S—Switch H—Host I—IGMP device r—Repeater
Platform	Product number of the device.
Port ID (outgoing port)	Port number of the device.

Table 2-52 describes the fields shown in the **show cdp neighbors detail** display.

Table 2-52 show cdp neighbors detail Field Descriptions

Field	Description
Device ID	Configured ID (name), MAC address, or serial number of the neighbor device.
Entry address (es)	List of network addresses of neighbor devices.
Platform	Product name and number of the neighbor device.
Capabilities	Device type of the neighbor. This device can be a router, a bridge, a transparent bridge, a source-routing bridge, a switch, a host, an IGMP device, or a repeater.
Interface	Protocol being used by the connectivity media.
Port ID (outgoing port)	Port number of the port on the neighbor device.
Holdtime	Time (in seconds) that the current device will hold the CDP advertisement from a transmitting router before discarding it.
Version	Software version running on the neighbor device.
Copyright	Copyright information for the neighbor device.
advertisement version	Version of CDP being used for CDP advertisements.
VTP Management Domain	VLAN trunk protocol management domain. The VLAN information is distributed to all switches that are part of the same domain.
Native VLAN	VLAN to which the neighbor interface belongs.

Table 2-53 describes the field in the **show cdp run** display.

Table 2-53 Field Description for the show cdp run Command

Field	Description
CDP is XX.	Shows whether CDP is enabled or disabled.

Table 2-54 describes the field in the **show cdp timer** display.

Table 2-54 Field Description for the show cdp timer Command

Field	Description
cdp timer XX	Time when CDP information is resent to neighbors.

Table 2-55 describes the fields shown in the **show cdp traffic** display.

Table 2-55 show cdp traffic Field Descriptions

Field	Description
Total packets Output	(Total number of packets sent) Number of CDP advertisements sent by the local device. Note this value is the sum of the CDP Version 1 advertisements output and CDP Version 2 advertisements output fields.
Input	(Total number of packets received) Number of CDP advertisements received by the local device. Note this value is the sum of the CDP Version-1 advertisements input and CDP Version 2 advertisements input fields.
Hdr syntax	(Header Syntax) Number of CDP advertisements with bad headers, received by the local device.
Chksum error	(CheckSum Error) Number of times the checksum (verifying) operation failed on incoming CDP advertisements.
Encaps failed	(Encapsulations Failed) Number of times CDP failed to transmit advertisements on an interface because of a failure caused by the bridge port of the local device.
No memory	Number of times the local device did not have enough memory to store the CDP advertisements in the advertisement cache table when the device was attempting to assemble advertisement packets for transmission and parse them when receiving them.
Invalid packet	Number of invalid CDP advertisements received and sent by the local device.
Fragmented	Number of times fragments or portions of a single CDP advertisement were received by the local device instead of the complete advertisement.
CDP version 1 advertisements Output	Number of CDP Version 1 advertisements sent by the local device.
Input	Number of CDP Version 1 advertisements received by the local device.
CDP version 2 advertisements Output	Number of CDP Version 2 advertisements sent by the local device.
Input	Number of CDP Version 2 advertisements received by the local device.

Related Commands

cdp enable
clear cdp counters
clear cdp table

show cfs

To display information about the cache file system (cfs), use the **show cfs** EXEC command.

show cfs {**statistics** | **volumes**}

Syntax Description

statistics	Displays the cfs statistics.
volumes	Displays the cfs volumes.

Defaults

No default behavior or values

Command Modes

EXEC

Usage Guidelines

The cfs caches HTTP and FTP objects.

Examples

The **show cfs statistics** command summarizes logical statistics for disk 0 and disk 1.

Table 2-56 describes the fields shown in the **show cfs statistics** display.

Table 2-56 show cfs statistics Field Descriptions

Field	Description
CFS Statistics	
Volume x	Cache file system (CFS) volume based on the disk configurations. Normally, one volume per disk maps to the CFS disk partition.
Total disk space	Total amount of disk space available in bytes.
Total disk space used	Total amount of disk space used so far for caching content.
Total disk objects read	Total number of cached segments read from the disk. Large URL objects could span several segments.
Total disk objects write	Total number of cached segments written to the disk. Large URL objects could span several segments.
Total bytes of disk read	Total number of bytes read from the disk.
Total bytes of disk write	Total number of bytes written to the disk.
Disk read errors	Number of disk read errors encountered.
Disk write errors	Number of disk write errors encountered.

The **show cfs volumes** command displays different disk names and indicates whether or not a cfs partition is mounted. The cfs size is displayed in kilobytes, as shown in the following example:

```
ContentEngine# show cfs volumes
cfs 00 (disk00/04):      12707839KB       mounted
cfs 01 (disk01/00):       5652464KB       mounted
ContentEngine#
```

Related Commands

cfs
show disks
show statistics cfs

show clock

To display the system clock, use the **show clock** EXEC command.

show clock [**detail** | **standard-timezones** {**all** | **details** *timezone* | **regions** | **zones** *region-name*}]

Syntax Description

detail	(Optional) Displays detailed information; indicates the clock source (NTP) and the current summer time setting (if any).
standard-timezones	(Optional) Displays information about the standard time zones.
all	Displays all of the standard time zones (approximately 1500 time zones). Each time zone is listed on a separate line.
details	Displays detailed information for the specified time zone.
timezone	Name of the time zone.
regions	Displays the region name of all the standard time zones. All 1500 time zones are organized into directories by region.
zones	Displays the name of every time zone that is within the specified region.
region-name	Name of the region.

Defaults

No default behavior or values

Command Modes

EXEC

Usage Guidelines

The ACNS system has several predefined standard time zones. Some of these time zones have built-in summertime information while others do not. For example, if you are in an eastern region of the United States (US), you must use the US/Eastern time zone that includes summertime information and will adjust the clock automatically every April and October. There are about 1500 standard time zone names.

In the ACNS 5.2.x software and earlier releases, there was no restriction on these reserved standard time zone names. You could overload these standard names in various ways. For example, you could use the US/Pacific time zone, but enter the **clock summertime** EXEC command to define a different summertime schedule. In the ACNS 5.3 software release, strict checking was added. The **clock summertime** command is now disabled when a standard time zone is configured. You can now only configure summertime if the time zone is not a standard time zone (if the time zone is a customized zone).

In addition, in the ACNS 5.3 software release, CLI commands were added to enable you to display a list of all the standard time zones. The **show clock standard-timezones all** EXEC command enables you to browse through all standard time zones and choose from these predefined time zones. You can choose a customized name that does not conflict with the predefined names of the standard time zones. Most predefined names of the standard time zones have two components, a region name and a zone name. You can list time zones by several criteria, such as regions and zones.

Examples

Table 2-57 describes the field in the **show clock** display.

Table 2-57 show clock Field Description

Field	Description
Local time	Day of the week, month, date, time (hh:mm:ss), and year in local time relative to the UTC offset.

Table 2-58 describes the fields shown in the **show clock detail** display.

Table 2-58 show clock detail Field Descriptions

Field	Description
Local time	Local time relative to UTC.
UTC time	Universal time clock date and time.
Epoch	Number of seconds since Jan. 1, 1970.
UTC offset	UTC offset in seconds, hours, and minutes.

The following example shows an excerpt of the output from the **show clock standard-timezones all** EXEC command. As the following example shows, all of the standard time zones (approximately 1500 time zones) are listed. Each time zone is listed on a separate line.

```
ContentEngine # show clock standard-timezones all
Africa/Abidjan
Africa/Accra
Africa/Addis_Ababa
Africa/Algiers
Africa/Asmera
Africa/Bamako
Africa/Bangui
Africa/Banjul
Africa/Bissau
Africa/Blantyre
Africa/Brazzaville
Africa/Bujumbura
Africa/Casablanca
Africa/Ceuta
Africa/Conakry
Africa/Dakar
Africa/Dar_es_Salaam
Africa/Djibouti
.
.
.
```

The following example shows an excerpt of the output from the **show clock standard-timezones region** EXEC command. As the example shows, all first level time zone names or directories are listed. All 1500 time zones are organized into directories by region.

```
ContentEngine # show clock standard-timezones regions
Africa/
America/
Antarctica/
Arctic/
Asia/
Atlantic/
```

```
Australia/
Brazil/
CET
.
.
.
```

The following example shows an excerpt of the output from the **show clock standard-timezones zones** EXEC command. As the following example shows, this command lists the name of every time zone that is within the specified region (for example, the US region).

```
ContentEngine # show clock standard-timezones zones US
Alaska
Aleutian
Arizona
Central
East-Indiana
Eastern
Hawaii
Indiana-Starke
Michigan
Mountain
Pacific
Samoa
```

The following example shows an excerpt of the output from the **show clock standard-timezones details** EXEC command. This command shows details about the specified time zone (for example, the US/Eastern time zone), The command output also includes the standard offset from the Greenwich mean time (GMT).

```
ContentEngine # show clock standard-timezones details US/Eastern
US/Eastern is standard timezone.
Getting offset information (may take a while) ...
Standard offset from GMT is -300 minutes (-5 hour(s)).
It has built-in summertime.
Summer offset from GMT is -240 minutes. (-4 hour(s)).
```

Related Commands

clock set

show cms

To display the Centralized Management System (CMS) embedded database content and maintenance status and other information, use the **show cms** EXEC command.

show cms {**database** {**content** {**dump** *filename* | **text** | **xml**} | **maintenance** [**detail**]} | **info** | **processes**}

Syntax Description

database	Displays embedded database maintenance information.
content	Writes the database content to a file.
dump	Dumps all database content to a text file.
filename	Name of the file to be saved under local1 directory.
text	Writes the database content to a file in text format.
xml	Writes the database content to a file in XML format.
maintenance	Shows the current database maintenance status.
detail	(Optional) Displays database maintenance details and errors.
info	Displays CMS application information.
processes	Displays CMS application processes.

Defaults

No default behavior or values

Command Modes

EXEC

Examples

Table 2-59 describes the fields shown in the Content Distribution Manager **show cms info** display.

Table 2-59 show cms Field Descriptions for the Content Distribution Manager

Field	Description
CDN information	
Model	Model name of the device.
Node Id	Unique identifier given to the device by the Content Distribution Manager at registration, which is used to manage the device.
Device Mode	Configured mode of device used during registration.
Current CDM role	Role of the current Content Distribution Manager: Primary or Standby.
CMS services information	
Service cms_httpd is running	Status of the management service (running or not running). This field is specific to the Content Distribution Manager only.
Service cms_cdm is running	Status of the management service (running or not running). This field is specific to the Content Distribution Manager only.

Table 2-60 describes the fields shown in the Content Engine **show cms info** display.

Table 2-60 show cms Field Descriptions for the Content Engine

Field	Description
CDN information	
Model	Model name of the device.
Node Id	Unique identifier given to the device by the Content Distribution Manager at registration, which is used to manage the device.
Device Mode	Configured mode of device used during registration.
Current CDM address	Address of the Content Distribution Manager as currently configured in the **cdm ip** global configuration command. This address may differ from the registered address if a standby Content Distribution Manager is managing the device instead of the primary Content Distribution Manager with which the device is registered.
Registered with CDM	Address of the Content Distribution Manager with which the device is registered.
Status	Connection status of the device to the Content Distribution Manager. This field may contain one of 3 values: Online, Offline, or Pending.
Time of last config-sync	Time when the device management service last contacted the Content Distribution Manager for updates.

The following example writes the database content to a file in text format:

```
ContentDistributionManager# show cms database content text
Database content can be found in /local1/cms-db-12-12-2002-17:06:08:070.txt.
```

The following example writes the database content to a file in XML format:

```
ContentDistributionManager# show cms database content xml
Database content can be found in /local1/cms-db-12-12-2002-17:07:11:629.xml.
```

The following example shows the output of the **show cms database maintenance detail** on a Content Engine:

```
ContentEngine# show cms database maintenance detail
Database maintenance is not running.
Regular database maintenance is enabled.
Regular database maintenance schedule is set on Sun, Mon, Tue, Wed, Thu, Fri, Sat at 02:00
Full database maintenance is enabled.
Full database maintenance schedule is set on Sun, Mon, Tue, Wed, Thu, Fri, Sat at 04:00
Disk usage for STATE partition: Total: 1523564K, Available: 1443940K, Use: 6%

DATABASE VACUUMING DETAILS AND ERRORS
-------------------------------------
Database Vacuuming never performed or it did not complete due to error.
Latest Vacuuming status :No Error
Last Vacuum Error : No Error
Last Reindex Time : Thu Jul 15 02:02:49 2004
Latest Reindexing status :No Error
Last Reindex Error: No Error
ContentEngine#
```

Related Commands

cms database maintenance
cms enable

show content-routing

To display the Content Router simplified hybrid routing table, use the **show content-routing** EXEC command. This command is available only on the Content Router.

show content-routing {**forwarding** [**website** *fqdn*] | **routes** [**website** *fqdn* [**ip-address** *client-ip*]] | **summary** [**website** *fqdn*]}

Syntax Description

forwarding	Displays the route forwarding table for each website.
website	(Optional) Displays the route forwarding table for the specified website.
fqdn	Fully qualified domain name of the website in the route forwarding table.
routes	Displays all routes and best routes for each website.
website	(Optional) Displays all routes and best routes for the specified website.
fqdn	Fully qualified domain name of the specified website in the route.
ip-address	(Optional) Displays the route that content routing chooses for the specified website and client IP address.
client-ip	Client IP address.
summary	Displays summary statistics for all websites in the content routing table.
website	(Optional) Displays summary statistics for one website in the content routing table.
fqdn	Fully qualified domain name of the specified website in the content routing table.

Defaults

No default behavior or values

Command Modes

EXEC

Usage Guidelines

The **show content-routing** command displays the route table that drives the simplified hybrid routing algorithm on a Content Router. A *route* specifies a Content Engine that is available to serve content from a particular website to clients on a particular IP subnet.

The *forwarding table* is a fast-lookup data structure that is constructed and optimized for high-speed lookups using the best routes for a particular website.

For example, if website cdn.acme.com has the route 172.29.229.0/24 20 sanjose-ce2, this route indicates that clients on the 172.29.229.0/24 subnet should be routed to Content Engine sanjose-ce2 for content from this website. The metric value of 20 establishes the relative cost for the route. The route with the lowest metric value is least costly and is preferred.

Dynamic routing based on the Routing Information Protocol (RIP) chooses the best route to some destination host or network based on the number of hops. The shorter a route is, the better RIP rates it. Very long routes with 16 or more hops are regarded as unusable and are discarded. When the information is sent from the local routing table, the length of the route from the metric value associated with the routing table entry is computed. This metric value is set by the system administrator when configuring the route and should reflect the actual cost of using this route.

Depending on coverage zone configuration, there can be multiple Content Engines that are available to serve content to an individual client. In this case, the best route is the one with the lowest metric value. For example, if clients on subnet 172.29.229.0/24 are covered by two Content Engines: 172.29.229.0/24 20 sanjose-ce2 and 172.29.229.0/24 30 denver-ce1, the best route is Content Engine sanjose-ce2, because it has the smaller metric value. If sanjose-ce2 fails, then denver-ce1 automatically becomes the best route for this subnet.

The route table contains a separate set of routes for each website that is distributed by the ACNS network. The routes are a function of these factors:

- Assignment of Content Engines to the website's channels
- Coverage zone configuration of the ACNS network
- Whether or not the Content Router is receiving keepalive messages from individual Content Engines

Examples

The following example displays all routes in the routing table for the website cdn.acme.com:

```
ContentRouter# show content-routing routes website cdn.acme.com

----- All routes to web site: cdn.acme.com -----
172.29.224.0/24 20 seattle-ce1
172.29.224.0/24 30 denver-ce1
172.29.229.0/24 20 sanjose-ce2
172.29.229.0/24 30 denver-ce1
172.29.230.0/24 20 boston-ce2
172.29.230.0/24 30 nyc-ce1

----- Best routes to web site: cdn.acme.com -----
172.29.224.0/24 20 seattle-ce1
172.29.229.0/24 20 sanjose-ce2
172.29.230.0/24 20 boston-ce2
```

The following example displays the route that simplified hybrid routing chooses for a given website and ACNS network client:

```
ContentRouter# show content-routing routes website cdn.acme.com ip-address 172.29.230.14

Routed to CE boston-ce2
```

The following example displays the forwarding table for the website cdn.acme.com:

```
ContentRouter# show content-routing forwarding website cdn.acme.com

----- Forwarding Table for web site: cdn.acme.com -----
 fwd_memory_usage() = 3092 bytes (3084 bytes mtries)
172.29.224.0/24 seattle-ce1
172.29.229.0/24 sanjose-ce2
172.29.230.0/24 boston-ce2
```

The following example displays the forwarding table for all websites:

```
Mgt-CR-GUI# show content-routing forwarding

----- Forwarding Table for web site -----
domain: www.cr-cse.com (Origin Server: 10.77.157.36)
 fwd_memory_usage() = 1036 bytes (1028 bytes mtries)


----- Forwarding Table for web site -----
domain: Barrichello (Origin Server: www.rubens.com)
 fwd_memory_usage() = 1036 bytes (1028 bytes mtries)
```

```
----- Forwarding Table for web site -----
domain: cr-api-hutch (Origin Server: www.hutch.com)
 fwd_memory_usage() = 1036 bytes (1028 bytes mtries)


----- Forwarding Table for web site -----
domain: bkanthar.com (Origin Server: www.bkanthar.com)
 fwd_memory_usage() = 1036 bytes (1028 bytes mtries)


----- Forwarding Table for web site -----
domain: www.minne.com (Origin Server: www.cisco.org)
 fwd_memory_usage() = 1036 bytes (1028 bytes mtries)
```

The following example displays the route table summary statistics for the website cdn.acme.com:

```
ContentRouter# show content-routing summary website cdn.acme.com

domain cdn.acme.com
  Number of CEs in all routes: 6
  Number of CEs in best routes: 3
  Memory usage: 3092
  Number of nodes in forwarding table: 3
```

Related Commands

show statistics content-routing

show debugging

To display the state of each debugging option, use the **show debugging** EXEC command.

show debugging

Syntax Description

This command has no arguments or keywords.

Defaults

No default behavior or values

Usage Guidelines

This command displays only the type of debugging enabled, not the specific subset of the command.

Command Modes

EXEC

Examples

In the following example, the **debug icp client** command coupled with the **show debugging** command shows that Internet Cache Protocol (ICP) debugging is enabled:

```
ContentEngine# debug icp client
ContentEngine# show debugging
Debug icp (client) is on
```

Related Commands

debug
undebug

show device-mode

To display the configured or current mode of a device, use the **show device-mode** EXEC command.

show device-mode {**configured** | **current**}

Syntax Description

configured	Displays the configured device mode.
current	Displays the current device mode.

Defaults

No default behavior or values

Usage Guidelines

If the configured and current device modes differ, a reload is required for the configured device mode to take effect.

Command Modes

EXEC

Examples

Table 2-61 describes the field in the **show device-mode configured** display.

Table 2-61 show device-mode configured Field Description

Field	Description
Configured device mode	Device mode that has been configured, but has not yet taken effect.

Table 2-62 describes the field in the **show device-mode current** display.

Table 2-62 show device-mode current Field Description

Field	Description
Current device mode	Current mode in which the ACNS device is operating.

The following example shows how to use the **show device-mode** command to show the device mode when you change the device from a Content Engine to a Content Router using the **device mode** command:

```
Acmehost# show device-mode current
Current device mode: content-engine
Acmehost# show device-mode configured
Configured device mode: content-engine
Acmehost(config)# device mode content-router
The new configuration will take effect after a reload
Acmehost(config)# exit
Acmehost# show device-mode current
Current device mode: content-engine
Note: The configured and current device modes differ,
a reload is required for the configured device mode to
take effect.
Acmehost# show device-mode configured
Configured device mode: content-router
Note: The configured and current device modes differ,
a reload is required for the configured device mode to
take effect.
Acmehost# write memory
Acmehost# reload force
...reload...


Acmehost# show running-config
device mode content-router
!
hostname Acmehost
..

Acmehost# show device-mode configured
Configured device mode: content-router
Acmehost# show device-mode current
Current device mode: content-router
```

Related Commands

device mode

show disks

To view information about your disks, use the **show disks** EXEC command.

show disks [**configured** | **current** | **details** | **failed-sectors** [*disk_name*] | **network-attached** | **SMART-info** [**details**] | **storage-array** [**details**]]

Syntax Description

configured	(Optional) Displays new configurations after a reboot.
current	(Optional) Displays currently effective configurations.
details	(Optional) Displays currently effective configurations with more details.
failed-sectors	(Optional) Displays a list of failed sectors on the disks.
disk_name	(Optional) Name of the disk for which failed sectors are displayed (disk01, disk02, and so on).
network-attached	(Optional) Displays the status of network-attached storage (NAS) devices.
SMART-info	(Optional) Displays hard drive diagnostic information and information about impending disk failures.
details	(Optional) Displays more detailed SMART disk monitoring information.
storage-array	(Optional) Displays disk information residing on the Storage Array, if applicable.
details	(Optional) Displays a more detailed operating status of the Storage Array.

Defaults

No default behavior or values

Command Modes

EXEC

Usage Guidelines

The **show disks** command displays the names of the disks currently attached to the Content Engine. The **show disks configured** command displays the percentage or amount of disk space allocated to each file system, instead of the names of the disks, after reboot.

```
ContentEngine# show disks configured
SYSFS                 10%
CFS                   30%
MEDIAFS               30%
CDNFS                 30%
```

In the ACNS 5.1.x software and earlier releases, a disk failure syslog message is generated each 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.

In the ACNS 5.3 software and later releases, enter the **show disks failed-sectors** EXEC command to display a list of failed sectors on the Content Engine disks as follows:

```
ContentEngine# show disks failed-sectors
    List of failed sectors on disk00
    --------------------------------
    89923
    9232112
   List of failed sectors on disk01
    --------------------------------
    <None>
```

To display a list of failed sectors for a only a specific disk drive, specify the name of the disk when entering the **show disks failed-sectors** command. If there are disk failures, a message is printed notifying you about this situation when you log in to a Content Engine that is running the ACNS 5.3 software and later releases.

Proactively Monitoring Disk Health with SMART

In the ACNS 5.3 software release, the ability to proactively monitor the health of disks with Self Monitoring, Analysis, and Reporting Technology (SMART) was added. SMART provides you with hard drive diagnostic information and information about impending disk failures.

SMART is supported by most disk vendors and is a standard method used to determine the health of a disk. SMART has several read-only attributes (for example, the power-on hours attribute, the load and unload count attribute) that provide the ACNS software with information regarding the operating and environmental conditions that may indicate an impending disk failure.

To display more detailed information, enter the **show disks SMART-info details** EXEC command. The output from the **show disks SMART-info** and the **show disks SMART-info details** commands will differ based on the disk vendor and the type of drive technology (Integrated Drive Electronics [IDE], Small Computer Systems Interface [SCSI], and Serial Advanced Technology Attachment [SATA] disk drives).

Even though SMART attributes are vendor dependent, there is a common way of interpreting most SMART attributes. Each SMART attribute has a normalized current value and a threshold value. When the current value exceeds the threshold value, the disk is considered as failed. The ACNS software monitors the SMART attributes and reports any impending failure through syslog messages, SNMP traps, and alarms.

In the ACNS 5.3 software and later releases, the output from the **show tech-support** EXEC command also includes SMART information.

Filter Out Multiple Syslog Messages For a Single Failed Sectors on SCSI, IDE, and SATA Disk Drives

In the ACNS releases prior to the ACNS 5.2 release, many disk failure messages were generated when a single disk sector failed, which caused an unnecessary alarm. In the ACNS 5.2 software, changes for suppressing multiple syslog messages on a single sector failure have been implemented for IDE disk drives. In the ACNS 5.3 software, this fix has also been extended to SCSI and SATA disk drives.

Use the **show disks failed-sectors** [*diskname*] EXEC command to display a list of failed sectors on the disks.

Examples

Table 2-63 describes the fields shown in the **show disks details** display.

Table 2-63 show disks details Field Descriptions

Field	Description
Physical disk information	Lists the disks by number. WAE 7300 series appliances show information for 6 disk drives and WAE 500 and 600 series appliances show information for 2 disk drives.
disk00	Availability of the disk: Present, Not present or Not responding, Not used, or (*). **Note** Disk drives that are currently marked as bad are shown as "Not used" in the output. Future bad disk drives (drives that will not be used after the next time that the Content Engine is reloaded) are shown with an asterisk (*). Disk identification number and type. Disk size in megabytes and gigabytes.
disk01	Same type of information is shown for each disk.
Mounted filesystems	Table containing the following information:
Device	Path to the partition on the disk.
Type	Type of the file system. Values include PHYS-FS, SYSFS, CDNFS, CFS, MEDIAFS.
Size	Total size of the file system in megabytes and gigabytes.
Mount point	Mount point for the file system. For example, the mount point for SYSFS is /local/local1.
System use	Amount of disk space being used for system use.
Free	Amount of unused disk space available.

The **show disks network-attached** command shows the results of using the **network-filesystem client** command to configure a NAS device to extend the storage capacity of the Content Engine.

Table 2-64 describes the fields shown in the command display.

Table 2-64 show network-filesystem client all Field Descriptions

Field	Description
NFS: //172.19.226.234:/nfs-mediafs: FILESYSTEM	Path to the NFS partition on the disk and the type of the file system being used, such as CDNFS, CFS, or MEDIAFS.
Current status	Connection status of the NFS share. Values may be attaching or online.
Total volume size	Total size of the file system in megabytes.
Free volume space	Amount of unused disk space available in megabytes.
Volume space assigned to CE	Amount of space that is being used in the disk for the respective file system.
Used volume space	Amount of disk space being used (in megabytes) and the percentage of the total.

Table 2-64 show network-filesystem client all Field Descriptions (continued)

Field	Description
Volume space assigned to CE	Amount of disk space (in megabytes and gigabytes) being used for system use.
CIFS: //172.19.226.235:smb-cdnfs: FILESYSTEM	Path to the CIFS partition on the disk and the type of the file system being used, such as CDNFS, CFS, or MEDIAFS.
Current status	Connection status of the CIFS client.
Total volume size	Total size of the file system in megabytes.
Free volume space	Amount of unused disk space available in megabytes.
Used volume space	Amount of disk space being used (in megabytes) and the percentage of the total.
Volume space assigned to CE	Amount of disk space (in megabytes and gigabytes) being used for system use.

The following example shows how to display a list of failed sectors for disk01:

```
ContentEngine# show disks failed-sectors disk01
```

SMART support is vendor dependent; each disk vendor has a different set of supported SMART attributes. The following example shows the output from the **show disks SMART-info** EXEC command that was entered on two different Content Engines (Content Engine A and Content Engine B). These two Content Engines contain hard disks that were manufactured by different vendors.

```
ContentEngineA# show disks SMART-info
=== disk00 ===
Device: IBM      IC35L036UCD210-0 Version: S5BS
Serial number:         22222222
Device type: disk
Transport protocol: Fibre channel (FCP-2)
Local Time is: Sun Jan  2 03:14:16 2005 Etc
Device supports SMART and is Enabled
Temperature Warning Disabled or Not Supported
SMART Health Status: OK

=== disk01 ===
disk01: Not present

ContentEngineB# show disks SMART-info
Disk 01:
========
Device Model:     HITACHI_DK23BA-20
Serial Number:    111111
Firmware Version: 00E0A0D2
SMART support is: Available - device has SMART capability.
SMART support is: Enabled
SMART overall-health self-assessment test result: PASSED
Vendor Specific SMART Attributes with Thresholds:
ID# ATTRIBUTE_NAME          FLAG     VALUE WORST THRESH TYPE     WHEN_FAILED RAW_VALUE
  1 Raw_Read_Error_Rate     0x000d   100   083   050    Pre-fail     -       677
  3 Spin_Up_Time            0x0007   100   100   050    Pre-fail     -       0
  4 Start_Stop_Count        0x0032   100   100   050    Old_age      -       249
  5 Reallocated_Sector_Ct   0x0033   099   099   010    Pre-fail     -       30
  <cr>
```

Table 2-65 describes some typical fields in the **show disks SMART-info** display.

Table 2-65 show disks SMART-info Field Descriptions

Field	Description
disk00—disk05	WAE 7300 series appliances show information for 6 disk drives, and WAE 500 and 600 series appliances show information for 2 disk drives.
Device Model	Vendor number and version number of the disk.
Serial Number	Serial number for the disk.
Device type	Type of device is disk.
Transport protocol	Physical layer connector information, for example: Parallel SCSI (SPI-4).
Local time is	Day of the week, month, date, time hh:mm:ss, year, clock standard.
Device supports SMART and is Enabled	Status of SMART support: Enabled or Disabled.
Temperature Warning Enabled	Temperature warning status: Enabled or Disabled.
SMART Health Status:	Health status of the disk: OK or Failed.

Related Commands

disk (EXEC mode)

show distribution

To display the distribution information for a specified channel and to probe a remote Content Engine for the liveness of its associated channel, use the **show distribution** EXEC command.

show distribution [**channels** [**channel-id** *channel-num* | **channel-name** *channel-name*]]

show distribution [**forwarder-list** [**channel-id** *channel-num* [**detail**] | **channel-name** *channel-name* [**detail**] | **detail**]]

show distribution [**location** {**forwarder-load-weight** | **live-load-weight** | **location-leader-preference**} [**channel-id** *channel-num* | **channel-name** *channel-name*]]

show distribution [**mcast-data-receiver** [**channels** | **cloud** [**detail**]]]

show distribution [**mcast-data-sender** [**channels** | **cloud** [**detail**]]]

show distribution [**object-status** *object-url*]

show distribution [**processes**]

show distribution [**remote** *ip-address* {**metadata-sender channel-id** *channel-num* [**start-generation-id** *gen-id* **end-generation-id** *gen-id*] | **unicast-sender channel-id** *channel-num* {**cdn-url** *cdn_url* | **probe** | **relative-cdn-url** *cdn_url*}}]

show distribution [**remote traceroute** {**forwarder-next-hop channel-id** *channel-num* {**max-hop** *maxhop_num* | **trace-till-good** | **trace-till-root**} | **unicast-sender channel-id** *channel-num* {**cdn-url** *cdn-url* | **probe** | **relative-cdn-url** *cdn-url*} {**max-hop** *maxhop_num* | **trace-till-good** | **trace-till-root**}}]

Syntax Description

channels	(Optional) Displays information about the specified channel.
channel-id	(Optional) Specifies the channel ID.
channel-num	Channel number (64-bit number).
channel-name	(Optional) Specifies the channel name.
channel-name	Channel name.
forwarder-list	(Optional) Displays the forwarder lists for all channels subscribed to by the Content Engine.
channel-id	(Optional) Specifies the channel ID.
channel-num	Channel number.
detail	(Optional) Displays detailed forwarder lists for a subscribed channel ID.
channel-name	(Optional) Specifies the channel name.
channel-name	Channel name.
detail	(Optional) Displays detailed forwarder lists for a subscribed channel name.
detail	(Optional) Displays detailed forwarder lists for all channels subscribed to by the Content Engine.
location	(Optional) Displays channel routing-related parameters for the Content Engines in the location (specified by the channel ID).

forwarder-load-weight	Displays the forwarder load weight value of the Content Engines in the location (specified by the channel ID). For more information, see the "Forwarder Probability" section on page 2-481.
live-load-weight	Displays the live load weight value of the Content Engines in the location (specified by the channel ID). For more information, see the "Live Splitting Probability" section on page 2-482.
location-leader-preference	Displays the location leader preference value of the Content Engines in the location (specified by the channel ID). For more information, see the "Location Leader Preference" section on page 2-481.
channel-id	(Optional) Specifies the channel ID.
channel-num	Channel number (64-bit number).
channel-name	(Optional) Specifies the channel name.
channel-name	Channel name.
mcast-data-receiver	(Optional) Displays multicast data receiver information.
channels	(Optional) Displays the list of channels assigned to the multicast receiver.
cloud	(Optional) Displays the cloud configuration information for the multicast receiver.
detail	(Optional) Displays the detailed cloud configuration information.
mcast-data-sender	(Optional) Displays multicast data sender information.
channels	(Optional) Displays the list of channels assigned to the multicast sender.
cloud	(Optional) Displays the cloud configuration information for the multicast sender.
detail	(Optional) Displays the detailed cloud configuration information.
object-status	(Optional) Displays information on the status of a pre-positioned object.
object-url	URL of the pre-positioned object.
processes	(Optional) Displays information on distribution processes.
remote	(Optional) Displays channel information about a remote Content Engine.
ip-address	IP address of the remote Content Engine.
metadata-sender	Displays the metadata from a remote Content Engine.
channel-id	Specifies the channel ID.
channel-num	Channel number.
start-generation-id	(Optional) Specifies the beginning database value of the current version of the multicast cloud.
gen_id	Beginning database value.
end-generation-id	Specifies the ending database value of the curent version of the multicast cloud.
gen_id	Ending database value.
unicast-sender	Displays the unicast data from a remote Content Engine.
channel-id	Specifies the channel ID of the channel to which the Content Engine is assigned.
channel-num	Channel number of the channel to which the Content Engine is assigned.
cdn-url	Checks the object on a remote Content Engine using the specified URL.
cdn_url	ACNS network URL used to check the object on a remote Content Engine.
probe	Probes the remote unicast sender.

relative-cdn-url	Checks the object on a remote Content Engine using the specified URL.
cdn_url	ACNS network URL used to check the object on a remote Content Engine.
traceroute	Displays the traceroute for the channel routing status.
forwarder-next-hop	Displays the next forwarder in the path for the Content Engine. **Note** This keyword lets you display the forwarding Content Engines to the root Content Engine in a manner similar to the **traceroute** command.
unicast-sender	Displays the unicast sender for the Content Engine.
channel-id	Specifies the channel ID with which the unicast sender is associated.
channel-num	Channel number of the channel with which the unicast sender is associated.
max-hop	Displays the maximum number of hops needed to reach the unicast sender.
maxhop_num	Maximum number of hops (1–1024).
trace-till-good	Allows the device to trace the route of an object until the object is found.
trace-till-root	Allows the device to trace the route of an object until the device reaches the root Content Engine.

Defaults

No default behavior or values

Command Modes

EXEC

Usage Guidelines

Use the **show distribution remote** *ip-address* **metadata-sender channel-id** *channel-num* [**start-generation-id** *gen_id* **end-generation-id** *gen_id*] command option to retrieve the metadata from a remote Content Engine assigned to a specified channel ID. The start and end generation IDs specify the beginning and ending database values representing the current version of the multicast cloud stored in the local database.

Note Generation IDs must be greater than zero. Also, you must specify both the start and the end generation IDs, or neither ID.

The **show distribution remote** *ip-address* **unicast-sender channel-id** *channel-num* **relative-cdn-url** *cdn-url* command shows the status of the relative ACNS network URL of an object at a remote Content Engine assigned to a specified channel ID. A relative ACNS network URL is one that lacks the prefix of the protocol and hostname. For example, the relative ACNS network URL for http://www.mycompany.com/abc.def.html is abc.def.html.

Use the **show distribution remote** *ip-address* **unicast-sender channel-id** *channel-num* **probe** command to probe a remote Content Engine for the liveness of the channel to which it is assigned.

The **show distribution object-status** *object-url* command can be used to display the properties of a pre-positioned object.

You can view the location leader preference and forwarder weight for the Content Engine using the **show distribution** command. However, you can configure the location leader preference and forwarder weight only using the Content Distribution Manager GUI for each Content Engine. Default values are assumed if you do not manually configure them.

The **show distribution location forwarder-load-weight** command displays the probability of Content Engines assigned to the channel within the location being selected as a forwarder. The **show distribution location location-leader-preference** command displays the location-leader-preference value of Content Engines that are assigned to the channel within the location.

Use the **show distribution channels** command to view the forwarder for the Content Engines. If a receiver Content Engine is unable to find its forwarder Content Engine, one of the following reasons is displayed in the Status/Reason column of the output of the **show distribution channels** command:

- LLMT—The home Content Engine cannot find the forwarder because the home Content Engine has a limit on the forwarder lookup level.
- FAIL—The home Content Engine cannot find the forwarder because there is a failed Content Engine along the path within the specified forwarder lookup level.
- NGWT:—The home Content Engine cannot find the forwarder because there is a Content Engine with a negative forwarder-load-weight along the path within the specified forwarder lookup level.

Forwarder Probability

When a Content Engine (location leader) selects its forwarder from an upstream location, it uses the *forwarderLoad-weight* value configured for each Content Engine in its upstream location. The weight value of each Content Engine corresponds to the probability of the Content Engine being selected as the forwarder.

Each Content Engine generates a unique random number. When a Content Engine needs to select an inter-location forwarder, it views all the Content Engines in one remote location as a collection, with the size corresponding to their weight. It uses the generated random number to select a Content Engine as a forwarder. Content Engines with a higher weight are more likely to be selected as forwarders.

Note The *forwarderLoad-weight* value represents a probabilistic value. When a large number of children Content Engines select a forwarder from a location, the load on the forwarder represents the weight.

The load on the forwarder is the replication load per channel on the Content Engine because all Content Engines subscribed to the channel will select a forwarder to balance the load. However, as the number of Content Engines assigned to different channels might differ, a forwarder might receive requests for content from Content Engines in other channels, which increases the load. Therefore, it is possible that the total load (of replicating content) on the forwarder does not reflect the weight but depends on the number of Content Engines assigned to a channel.

For each Content Engine, you can specify the probability of each Content Engine acting as a forwarder to Content Engines from downstream locations using the Content Distribution Manager GUI. You can also specify whether certain Content Engines should never serve as a forwarder to downstream Content Engines. The specification is channel independent.

If you choose not to configure settings using the Location Leader and Forwarder Settings for Content Engine window, the channel routing algorithm uses the random number method to generate one permutation of the Content Engine ID ordered list.

Location Leader Preference

For intra-location (list of Content Engines in a Content Engine's own location) forwarder selection, the channel routing algorithm first creates an ordered list of the Content Engines based on their *location leader-preference* in a descending order. Next, the channel routing algorithm will select the first Content Engine in the ordered list as the location leader.

When multiple Content Engines have the same *location leader-preference*, the channel routing algorithm guarantees that all the Content Engines assigned to each channel in the location still generate the same ordered list, which avoids routing loops. A routing loop is a deadlock situation in which the forwarder selection among multiple Content Engines within the loop prevents the Content Engines from receiving the content from upstream locations. For example, CE1 uses CE2 as the forwarder and CE2 uses CE1 as the forwarder for the same channel.

However, for two different channels, the ordered list could be different (when multiple Content Engines have the same location leader probability value) even if the subscribed Content Engines are the same. When multiple Content Engines have the highest *location leader-preference*, although each Content Engine has an equal chance of acting as the location leader (for different channels), only one Content Engine always acts as the location leader for one particular channel. If you configure all the Content Engines in your network to have the same *location leader-preference* value, then each Content Engine has an equal chance of acting as a location leader. When each Content Engine has an equal probability of being selected as the location leader for any particular channel, only one particular Content Engine is always selected as the location leader.

If you choose not to configure settings using the Location Leader and Forwarder Settings for Content Engine window, the channel routing algorithm works the same way as in the ACNS software releases earlier than 5.2.

Note If there are two Content Engines in the same location (CE1, CE2) and both of them are assigned to two channels (CH1, CH2), and if you want CE1 to be the location leader for CH1 and CE2 to be the location leader for CH2, you cannot configure them using *location leader-preference* and *forwarderLoad-weight*.

In a location, you can specify the probability for each Content Engine acting as the location leader. The specification is channel independent.

The following limitations are associated with specifying a location leader preference:

- When multiple Content Engines are configured with the same *location leader-preference* value and assigned to a channel, they have an equal probability of being selected as the location leader. In such a case, you do not have a control mechanism to decide which Content Engine should be selected as the location leader.
- Only the order among the Content Engines with the highest preference is randomized to determine the location leader preference and not other Content Engines with the same preference value. It is possible that if the Content Engine(s) with the highest preference failed and there are several other Content Engines with the second highest preference value, they will not be balanced across the different channels. The same Content Engine will be the location leader for all channels.
- If a Content Engine with a high *location leader-preference* is assigned to many channels, it is possible that the Content Engine can be selected as the location leader for all these channels.

Live Splitting Probability

Similar to the channel routing application used for content replication that assigns weight and priority to Content Engines to tune location leader and forwarder selections, you can specify a configuration parameter *liveSplit-load-weight* for each Content Engine. This parameter represents the relative probability that a Content Engine is likely to receive live stream splitting traffic as compared to other Content Engines. The specification is channel independent. If you do not configure *liveSplit-load-weight*, the algorithm should work as it currently does.

A weighted load balancing scheme splits the live stream load on the Content Engines according to the *liveSplit-load-weight* value because Content Engines of different types (such as CE-7325 and CE-565) have an equal probability of being selected as the splitting Content Engine on the location path.

The *liveSplit-load-weight* is used when the channel routing generates the ordered list of Content Engines for each location. The Content Engines are ordered in a way that the higher the weight, the greater probability that the Content Engines are being ordered at the beginning of the list.

The weight represents a relative value. The *liveSplit-load-weight* assigned to one Content Engine is compared against the weights of other Content Engines in the same location.

The weight is a probabilistic value. For one particular URL, all Content Engines generate the same ordered list for a location.

It is possible that a Content Engine with a lower weight is being ordered at the beginning while a Content Engine with a higher weight is being ordered at the end of the list. The weighted load balancing is useful only when there are multiple live stream URLs.

This parameter applies both within the home location and upstream locations. This setting is different from the distribution settings where locationLeader-priority controls how you choose Content Engines from the home location, while forwarderLoad-weight controls how you choose Content Engines from upstream locations.

Examples

The **show distribution mcast-data-sender** command displays the state of the multicast sender, and the **show distribution mcast-data-receiver** command displays the state of the multicast receiver. Both commands display information about the multicast cloud.

The following example shows the response if the Content Engine is not a multicast sender:

```
ContentEngine1# show distribution mcast-data-sender
Multicast enabled
Currently not a Multicast Sender for any Cloud
```

The following example shows the response if the Content Engine is a primary multicast sender:

```
ContentEngine# show distribution mcast-data-sender
Multicast enabled
Currently a Primary Multicast Sender for Cloud : MCloudSend [201]

Primary Sender Details
-----------------------
  Multicast Sender state           : PRIMARY_ACTIVE(1)
  Current Send State               : OK to Send
  Current MOUT BW (kbps)           : 100
  Fixed Carousel                   : Disabled
  Multicast Checkpoint Transfer is : Enabled

Backup Sender Details
----------------------
  The backup sender CE name        : kravisha-507
  The backup sender CE id          : 132
  The backup sender CE ip          : 10.43.27.4
```

The following example shows the response if the Content Engine is a backup sender:

```
ContentEngine# show distribution mcast-data-sender
Multicast enabled
Currently a Backup Multicast Sender for Cloud : MCastReceiver [202]

Primary Sender Details
-----------------------
  The primary sender CE name       : jmaypall-507
  The primary sender CE id         : 207
  The primary sender CE ip         : 10.43.27.3
```

```
Backup Sender Details
----------------------
  Multicast Sender state           : BACKUP_STANDBY(2)
  Multicast Checkpoint Transfer is : Enabled
  Current failover grace period    : 30m
  Time elapsed                     : 53s
```

When channels are distributed over multicast, most of the multicast cloud values displayed are from the multicast cloud database.

Table 2-66 describes the fields shown in the **show distribution mcast-data-sender cloud** display.

Table 2-66 show distribution mcast-data-sender cloud Field Descriptions

Field	Description
Multicast Cloud id	Identifier for the multicast cloud.
Multicast Cloud Name	Name of the multicast cloud. The name is unique across the system.
Description	Comments about the multicast cloud.
Advertisement IP address	Multicast address collectively known and used by a multicast cloud to implement a communication channel.
Multicast Port	Port used for file addresses. The default port is 7777.
Start Multicast IP Address	Start of the IP address range, which is within the range 224.0.0.0–239.255.255.255.
End Multicast IP Address	End of the IP address range, which is higher than the start IP address. The end IP address is within the range 224.0.0.0–239.255.255.255.
Default Multicast-out Bandwidth (kbps)	Maximum multicast rate in kilobits per second (kbps). The minimum rate is 10 kbps. This default multicast-out bandwidth is applied 24 hours a day, 7 days a week. The multicast-out bandwidth is associated with the sender Content Engine only.
Backup Sender Default Mout Bandwidth (kbps)	Maximum multicast rate in kilobits per second (kbps) for the backup sender. The minimum rate is 10 kbps.
Multicast medium	Means of transmitting the multicast, either satellite or terrestrial. The default is satellite.
Number of Carousel passes	Maximum number of times (1–1000000000) that a multicast sender will attempt to send missing content for on-demand carousels. The default is 5. **Note** If the multicast sender finishes the last carousel on an object at time *t* and the multicast sender receives a NACK within *t* + *carousel_delay*, the multicast sender starts the next carousel of this object at time *t* + *carousel_delay*. The multicast carousel is not triggered immediately upon receipt of the NACK if the carousel delay is greater than zero (0).
Delay between Carousel Passes (mins)	Delay between file transmissions, in minutes (0–10080) (7 days * 24 hours * 60 minutes). The default is 0 minutes.
TTL	Time To Live expressed as the number of hops (1–255). The default is 255.

Table 2-66 show distribution mcast-data-sender cloud Field Descriptions (continued)

Field	Description
FEC Transmission Group	Size of an FEC (forward error correction) block in packets. (See RFC 3208 *PGM Reliable Transport Protocol Specification* for more information.) The allowable inputs are 2, 4, 8, 16, 32, 64, and 128. The default is 16.
PGM Router Assist	Specifies whether IP routers are to be used to assist in the distribution of content. Checking this check box enables the IP router alert option for PGM packets. **Note** PGM packet routing is supported in Cisco IOS 12.0(5)T and later 12.0 T releases.
Primary-to-backup failover grace period	Amount of time allotted for the backup sender to detect whether the primary sender is active. If the backup sender does not hear a heartbeat from the primary sender within this grace period, the backup sender assumes the active role. The range is 5–7200 minutes. The default is 30 minutes.
Backup-to-primary fallback grace period	Amount of time allotted for the primary sender to detect whether the backup is active. If the backup sender does not respond within this grace period, the primary sender assumes the active role. The range is 5–7200 minutes. The default is 30 minutes.

The **show distribution mcast-data-sender cloud detail** option shows, in addition to the multicast cloud configuration, the Content Engines that are assigned to the multicast cloud.

Table 2-67 describes the additional fields in the **detail** option.

Table 2-67 show distribution mcast-data-sender cloud detail Field Description

Field	Description
CEs Assigned to the Multicast Cloud	
CE ID	Numerical identifier for the Content Engine.
CE Name	Hostname of the Content Engine.
CE IP	IP address of the Content Engine.
CE Role	Role of the Content Engine. Values may include Primary Sender, Receiver, and Backup Sender.

The **show distribution mcast-data-sender channels command** displays the list of channels assigned to the multicast sender.

Table 2-68 describes the fields in the command display.

Table 2-68 show distribution mcast-data-sender channels Field Description

Field	Description
Multicast Cloud-ID	Numerical identifier for the multicast cloud.
Multicast Cloud-Name	Name of the multicast cloud.

Table 2-68 show distribution mcast-data-sender channels Field Description (continued)

Field	Description
Multicast Channels Associated with Multicast Cloud	
Channel-ID	Numerical identifier for the channel.
Channel-Name	Name of the channel.
Multicast IP-Address	Multicast IP address.

The following example shows the response if the Content Engine is not a multicast receiver. The receiver display is similar to that of the multicast sender, but it does not display the current send state and displays the IP address of the multicast cloud sender.

```
ContentEngine1# show distribution mcast-data-receiver
Currently not a multicast receiver
```

The following example shows the response if the ContentEngine is a multicast receiver:

```
ContentEngine2# show distribution mcast-data-receiver
Multicast enabled
Currently Multicast Receiver for Cloud   : MCastReceiver [202]
```

The **show distribution mcast-data-receiver cloud** command shows cloud configuration information for the multicast receiver.

Table 2-69 describes the fields shown in the command display.

Table 2-69 show distribution mcast-data-receiver cloud Field Descriptions

Field	Description
Multicast Cloud id	Identifier for the multicast cloud.
Multicast Cloud Name	Name of the multicast cloud. The name is unique across the system.
Description	Comments about the multicast cloud.
Advertisement IP address	Multicast address collectively known and used by a multicast cloud to implement a communication channel.
Multicast Port	Port used for file addresses. The default port is 7777.
Start Multicast IP Address	Start of the IP address range, which is within the range 224.0.0.0–239.255.255.255.
End Multicast IP Address	End of the IP address range, which is higher than the start IP address. The end IP address is within the range 224.0.0.0–239.255.255.255.
Default Multicast-out Bandwidth (kbps)	Maximum multicast rate in kilobits per second (kbps). The minimum rate is 10 kbps. This default multicast-out bandwidth is applied 24 hours a day, 7 days a week. The multicast-out bandwidth is associated with the sender Content Engine only.
Backup Sender Default Mout Bandwidth (kbps)	Maximum multicast rate in kilobits per second (kbps) for the backup sender. The minimum rate is 10 kbps.
Multicast medium	Means of transmitting the multicast, either satellite or terrestrial. The default is satellite.

Table 2-69 show distribution mcast-data-receiver cloud Field Descriptions (continued)

Field	Description
Number of Carousel passes	Maximum number of times (1–1000000000) that a multicast sender will attempt to send missing content for on-demand carousels. The default is 5. **Note** If the multicast sender finishes the last carousel on an object at time *t* and the multicast sender receives a NACK within *t* + *carousel_delay*, the multicast sender starts the next carousel of this object at time *t* + *carousel_delay*. The multicast carousel is not triggered immediately upon receipt of the NACK if the carousel delay is greater than zero (0).
Delay between Carousel Passes (mins)	Delay between file transmissions, in minutes (0–10080) (7 days * 24 hours * 60 minutes). The default is 0 minutes.
TTL	Time To Live expressed as the number of hops (1–255). The default is 255.
FEC Transmission Group	Size of an FEC (forward error correction) block in packets. (See RFC 3208 *PGM Reliable Transport Protocol Specification* for more information.) The allowable inputs are 2, 4, 8, 16, 32, 64, and 128. The default is 16.
PGM Router Assist	Specifies whether IP routers are to be used to assist in the distribution of content. Checking this check box enables the IP router alert option for PGM packets. **Note** PGM packet routing is supported in Cisco IOS 12.0(5)T and later 12.0 T releases.
Primary-to-backup failover grace period	Amount of time allotted for the backup sender to detect whether the primary sender is active. If the backup sender does not hear a heartbeat from the primary sender within this grace period, the backup sender assumes the active role. The range is 5–7200 minutes. The default is 30 minutes.
Backup-to-primary fallback grace period	Amount of time allotted for the primary sender to detect whether the backup is active. If the backup sender does not respond within this grace period, the primary sender assumes the active role. The range is 5–7200 minutes. The default is 30 minutes.

The **show distribution mcast-data-receiver cloud detail** option shows, in addition to the multicast cloud configuration, the Content Engines that are assigned to the multicast cloud.

Table 2-70 describes the additional fields in the **detail** option.

Table 2-70 show distribution mcast-data-receiver cloud detail Field Description

Field	Description
CEs Assigned to the Multicast Cloud	
CE ID	Numerical identifier for the Content Engine.
CE Name	Hostname of the Content Engine.

Table 2-70 show distribution mcast-data-receiver cloud detail Field Description

Field	Description
CE IP	IP address of the Content Engine.
CE Role	Role of the Content Engine. Values may include Primary Sender, Receiver, and Backup Sender.

The **show distribution mcast-data-receiver channels command** displays the list of channels assigned to the multicast receiver.

Table 2-71 describes the fields in the command display.

Table 2-71 show distribution mcast-data-receiver channels Field Description

Field	Description
Multicast Cloud-ID	Numerical identifier for the multicast cloud.
Multicast Cloud-Name	Name of the multicast cloud.
Multicast Channels Associated with Multicast Cloud	
Channel-ID	Numerical identifier for the channel.
Channel-Name	Name of the channel.
Multicast IP-Address	Multicast IP address.

The following examples show various ways of using the **show distribution remote** commands.

The following example shows the status of the object at a remote Content Engine with the IP address 172.16.2.160 and channel ID 631. The URL of the content object specified in the command must not be the complete source URL. Instead, it must be the relative ACNS network URL of the object.

```
ContentEngine# show distribution remote 172.16.2.160 unicast-sender channel-id 631
relative-cdn-url 101files/100.txt

     Forwarder-Name :                AD-CE08
       Forwarder-ID :                    140
       Forwarder IP :             2.43.10.70
 Forwarder Location :       default-location

   Relative CDN URL :       101files/100.txt
        Actual Size :                     58
   Size Transferred :                     58
        Resource-ID : roVe2aMzp+YhmbhGUfMPpQ
         Content-ID : 7LC5xOlMp4YvkBJlHaQucQ
Last Modified Time :   10:52:38 Jan 04 2005
```

The following example retrieves the metadata from a remote Content Engine with the IP address 172.16.2.160 assigned to channel ID 4999:

```
ContentEngine# show distribution remote 172.16.2.160 metadata-sender channel-id 4999

Getting meta data for channel(4999) from genid -1 to 2
Connecting to 2.43.10.101
Remote CE replied with the following headers:
        Action : Processing metadata records
        Latest Gen id is : 2
        Have more records to process : No
        Is metadata still in full reload: No
add-size: 2, del-size: 0
```

```
Add Logs: 1 to 2
        add #1: UBsSUMwbTdJzzpqDvxSdYg.., basic_auth/public.html
        add #2: NJyVL9CZwpnyCfw+Is26yw.., index.txt
```

The following example probes the remote Content Engine with the IP address 172.16.2.160 for the liveness of its assigned channel ID 153:

```
ContentEngine# show distribution remote 172.16.2.160 unicast-sender channel-id 153 probe
Probe Successful
```

The following example shows the beginning database value of the current version of the multicast cloud at a remote Content Engine with the IP address 10.43.10.101 and channel ID 4999:

```
AD-CE06# show distribution remote 10.43.10.101 metadata-sender channel-id 4999
start-generation-id 0 end-generation-id 5
Getting meta data for channel(4999) from genid -1 to 5
Connecting to 10.43.10.101
Remote CE replied with the following headers:
        Action : Processing metadata records
        Latest Gen id is : 2
        Have more records to process : No
        Is metadata still in full reload: No
add-size: 2, del-size: 0

Add Logs: 1 to 2
        add #1: UBsSUMwbTdJzzpqDvxSdYg.., basic_auth/public.html
        add #2: NJyVL9CZwpnyCfw+Is26yw.., index.txt
```

Note When start and end generation IDs are not specified in the **show distribution remote** command, the current maximum generation ID of –100 will be the start generation ID. The end generation ID is equal to the sum of the start generation ID and 100.

The following example shows the list of forwarders in the path toward the root Content Engine:

```
ContentEngine# show distribution remote traceroute forwarder-next-hop channel-id 4999
trace-till-root

Hop NextHop_CEId  NextHop_CEName NextHop_CEIp      GenID Status/Reason
--- ------------  -------------- ------------      ----- -------------
  1    5884           AD-CE07    192.168.1.69     1        REGULAR
  2    6035           AD-CE13     2.43.10.101     1       LOC-LEAD
  3    5683           AD-CE12     2.43.10.100     1       LOC-LEAD
  4    6026      gnadaraj-507       2.43.27.2     1       LOC-LEAD
  5    5638          devi-507      2.43.27.36     1       LOC-LEAD (Reached RootCE)
```

The following example shows output from the **show distribution object-status** command:

```
ContentEngine# show distribution object-status http://www.cisco.com/index.txt

        ========== Website Information ==========

Name                      :          cisco-crawl
Origin Server FQDN        :        www.cisco.com
Request Routed FQDN       :                  N/A
Content UNS Reference #   :                    1


        ========== Channels Information ==========

        *** Channel 4999 (name = headercheck) ***
```

```
Object Replication
------------------
Replication               :                      Done
File State                :    Ready for distribution
Multicast for Channel     :               Not Enabled
Replication Lock          : Received by Unicast-Receiver/Acquirer
Reference Count           :                         1
Total Size                :                   2208640
Transfered Size           :                   2208640
MD5 of MD5                :  zwhJagyCmRAE4UmTwc0EtA..
Source Url                : http://liqq-linux.cisco.com/index.txt
Source Last Modified Time :  Sun Jul 11 03:23:33 2004

Object Properties
-----------------
Redirect To Origin        :                       Yes
Requires Authentication   :                        No
Alternative URL           :
Serve Start Time          :                       N/A
Serve End Time            :                       N/A
Play servers              :                HTTP HTTPS
Content Metadata          :                      None
Content uns_id            :  yhzR3VZ96MDz5FVHwmGD+A..
Content gen-id            :         5638:1108022220:1


        ========== CDNFS Information ==========

Internal File Name        :
/disk00-04/d/http-liqq-linux.cisco.com-azk2lrqzsytweswexham5w/32/326cf0278da48aac82d796cb1
19b1caa.0.data.txt
Actual File Size          : 2208640 bytes
MD5 of MD5 (Re-calculated): zwhJagyCmRAE4UmTwc0EtA..
Content metadata          : None
Metadata match with       : Channel 4999
Number of Source-urls     : 1

        Source-url to CDN-object mapping:

        Source-url                : http://liqq-linux.cisco.com/index.txt
        Used by CDN object        : ---- Yes ----
        Internal File Name        :
/disk00-04/d/http-liqq-linux.cisco.com-azk2lrqzsytweswexham5w/32/326cf0278da48aac82d796cb1
19b1caa.0.data.txt
        Actual File Size          : 2208640 bytes


        ========== CDNFS lookup output ==========


CDNFS File Attributes:
  Status                 3  (Ready)
  File Size              2208640 Bytes
  Start Time             null
  End Time               null
  Allowed Playback via   HTTP HTTPS
  Last-modified Time     Sun Jul 11 03:23:33 2004
  cdn_uns_id             yhzR3VZ96MDz5FVHwmGD+A..
  last-modified          Sun, 11 Jul 2004 03:23:33 GMT
Internal path to data file:
/disk00-04/d/http-liqq-linux.cisco.com-azk2lrqzsytweswexham5w/32/326cf0278da48aac82d796cb1
19b1caa.0.data.txt
```

The **show distribution channels** output and the **show distribution forwarder-list** output have been enhanced in the ACNS 5.2 software and later releases to display additional channel routing information. The newly added Status/Reason field displays whether the Content Engine is a location leader and the reason for not having a forwarder Content Engine. The mcast-role field in the **show distribution channels** output displays the role of the Content Engine in the multicast distribution.

The following example shows the channel distribution information:

```
ContentEngine# show distribution channels
Channel Name   ID   Priority   Root Forwarder   Status/Reason   Mcast-Role
------------   --   --------   ---- ---------   -------------   ----------
00-AD         527      500    No    Rack89-CE-11   REGULAR         N/A
01-AD         586      500    Yes   N/A            N/A             N/A
00-Live       588      500    Yes   N/A            LIVE            N/A

  LOC-LEAD:   This CE is the location leader for this channel
   REGULAR:   This CE is not the location leader for this channel
      LLMT:   This CE cannot find forwarder because this CE has limit on
              the forwarder lookup level
      FAIL:   This CE cannot find forwarder because there is failed CE along
              the path within specified forwarder lookup level
      NGWT:   This CE cannot find forwarder because there is CE with negative
              forwarder-load-weight along the path within the specified
              forwarder lookup level
      LIVE:   The specified channel is live channel, forwarder not applicable
         *:   MetaData forwarder and Unicast forwarder are different
```

The following example provides channel distribution information for channel ID 527:

```
ContentEngine# show distribution channels channel-id 527
Channel Configuration
---------------------
Channel ID                         :                  527
Channel Name                       :                00-AD
Website Name                       :                Test1
Website Origin FQDN                :         www.test.com
Channel Priority                   :                  500
Configured Distribution Type       :       Multicast only

Root CE Information
-------------------
ID of Configured Root CE           :                  462
Name of Configured Root CE         :         Rack89-CE-11
IP of Configured Root CE           :           2.43.27.38
ID of Effective Root CE            :                  462
Current root-ce-uid                :           1110247321
This CE's Role                     :        Not a Root CE
This CE in Full Reload             :                   No

Root CE Failover/Fallback Information
---------------------------------------
Root CE Failover/Fallback Interval :             120 Mins

Metadata Information
--------------------
Metadata-Forwarder ID              :                  462
Metadata-Forwarder Name            :         Rack89-CE-11
Metadata-Forwarder Primary IP      :           2.43.27.38
Metadata-Forwarder NAT IP/Port     :                  N/A
Address to Poll Metadata-Forwarder :           Primary IP
Metadata-Forwarder Status          :              REGULAR
Last gen-id Switch                 :                Never
Current low-water-marker           :                    1
Current max-gen-id                 :                    0
Current max-del-gen-id             :                    0
Last poll                          :          13 Secs ago
Next poll                          :    107 Secs from now
Idle poll interval                 :             120 Secs
Poll interval multiplier           :                    1

Unicast Information
-------------------
Ucast-Forwarder ID                 :                  462
Ucast-Forwarder Name               :         Rack89-CE-11
Ucast-Forwarder Primary IP         :           2.43.27.38
Ucast-Forwarder NAT IP/Port        :                  N/A
Address to Poll Ucast-Forwarder    :           Primary IP
Ucast-Forwarder Status             :              REGULAR

Multicast Information
---------------------
Mcast Receiving                    :                   No
Mcast Sending                      :                   No

QoS Configuration
-----------------
MetaData QoS (system config)       :                   16 (Effective)

Progress Information
--------------------
Number of jobs completed           :                    0
Has incomplete jobs                :                   No


  LOC-LEAD:   This CE is the location leader for this channel
   REGULAR:   This CE is not the location leader for this channel
      LLMT:   This CE cannot find forwarder because this CE has limit on
              the forwarder lookup level
      FAIL:   This CE cannot find forwarder because there is failed CE along
              the path within specified forwarder lookup level
      NGWT:   This CE cannot find forwarder because there is CE with negative
              forwarder-load-weight along the path within the specified
              forwarder lookup level
      LIVE:   The specified channel is live channel, forwarder not applicable
         *:   MetaData forwarder and Unicast forwarder are different
```

Note The Has Unfinished Job line is only available if the Content Engine is *not* a root Content Engine. It is only available on a receiver Content Engine.

Related Commands

multicast

show multicast

show statistics distribution

show dns

To view the DNS cache status and the memory allocated to the cache, use the **show dns** EXEC command.

show dns

Syntax Description

This command has no arguments or keywords.

Defaults

No default behavior or values

Command Modes

EXEC

Usage Guidelines

Use this command to check the DNS cache status and the memory size allocated to the cache. The retry timeout and period are also displayed. The DNS cache goes online when the **ip name-server** global configuration command is used and goes offline when the last IP name server configuration is deleted with the **no ip name-server** *ip-address* command.

Examples

The following example shows the DNS cache details:

```
ContentEngine# show dns
DNS cache status is online
Memory allocated to the cache is 5 MB
Retry timeout (time between attempts) is 3 sec
Retry period (the maximum time to retry) is 2 sec
Serial lookup is disabled
ContentEngine#
```

show error-handling

To display the error-handling configuration, use the **show error-handling** EXEC command.

show error-handling

Syntax Description

This command has no arguments or keywords.

Defaults

No default behavior or values

Command Modes

EXEC

Examples

The following example shows how to configure the Content Engine to send a cache error:

```
ContentEngine(config)# error-handling send-cache-error
```

The following example displays the error-handling configuration:

```
ContentEngine# show error-handling
error-handling is set to send-cache-error
```

Related Commands

error-handling

show flash

To display the flash memory version and usage information, use the **show flash** EXEC command.

show flash

Syntax Description

This command has no arguments or keywords.

Defaults

No default behavior or values

Command Modes

EXEC

Usage Guidelines

If a new software image has been installed and is waiting to be run after a reboot, the **show flash** command displays this information and the version of ACNS software that will run on the device after reload.

Examples

Table 2-72 describes the fields shown in the **show flash** display.

Table 2-72 show flash Field Descriptions

Field	Description
ACNS software version (disk-based code)	ACNS software version and build number that is running on the device.
System image on flash:	
Version	Version and build number of the software that is stored in flash memory.
System flash directory:	
System image	Number of sectors used by the system image.
Bootloader, rescue image, and other reserved areas	Number of sectors used by the bootloader, rescue image, and other reserved areas.
XX sectors total, XX sectors free	Total number of sectors. Number of free sectors.

show ftp-native

To display the FTP native proxy configuration, use the **show ftp-native** EXEC command.

show ftp-native

Syntax Description

This command has no arguments or keywords.

Defaults

No default behavior or values

Command Modes

EXEC

Examples

The following example shows the current FTP native proxy configuration:

```
CONTENTENGINE# show ftp-native
WCCP FTP service status:                 ENABLED
Maximum size of a FTP cacheable object:  2096128 KBytes
FTP data connection mode with Server:    Always passive mode
Incoming Proxy-Mode:
Configured Proxy mode Native FTP connections on ports: 8088 556
```

Related Commands

clear statistics ftp-native
ftp-native
show statistics ftp-native

show ftp-over-http

To display the caching configuration of the FTP-over-HTTP proxy, use the **show ftp-over-http** EXEC command.

show ftp-over-http

Syntax Description

This command has no arguments or keywords.

Defaults

No default behavior or values

Command Modes

EXEC

Examples

The following example shows the caching configuration of the FTP-over-HTTP proxy:

```
CONTENTENGINE# show ftp-over-http
FTP heuristic age-multipliers: directory-listing 30% file 60%
Maximum time to live in days: directory-listing 3 file 7
Minimum time to live for all objects in minutes: 30
Incoming Proxy-Mode:
  Not servicing incoming proxy mode connections.
Outgoing Proxy-Mode:
  Not using outgoing proxy mode.

 Monitor Interval for Outgoing Proxy Servers is 60 seconds

 Timeout period for probing Outgoing Proxy Servers is 300000 microseconds

 Use of Origin Server upon Proxy Failures is disabled.
Passive mode of FTP transfer is enabled
Maximum size of a FTP cacheable object is 2096128 KBytes
Only ftp directory-listing objects are revalidated on each request
```

Related Commands

clear statistics ftp-over-http
ftp-over-http
show statistics ftp-over-http

show gui-server

To display the current port assignment and operational status of the caching application graphical user interface (GUI) server, use the **show gui-server** EXEC command.

show gui-server

Syntax Description

This command has no arguments or keywords.

Defaults

No default behavior or values

Command Modes

EXEC

Examples

You can configure many parameters by making entries in the Content Engine management GUI. The **show gui-server** command in the following example shows whether or not the GUI is enabled and also its listener port. The following example also displays whether or not the secure access port is enabled. This configuration allows access to the GUI when you enter https://*ContentEngine*:*secure_port* in your browser (where *ContentEngine_IP_Address* is the IP address or hostname of the Content Engine that you want to manage, and *secure_port* is the port number used to access the GUI server).

```
ContentEngine# show gui-server
GUI Server is enabled
Listen on port 8001
Secured GUI Server is enabled
Secured GUI Listen on port 8003
ContentEngine#
```

Related Commands

gui-server

show hardware

To display system the hardware status, use the **show hardware** EXEC command.

show hardware

Syntax Description

This command has no arguments or keywords.

Defaults

No default behavior or values

Command Modes

EXEC

Usage Guidelines

In the ACNS 5.2.3 software and later releases, the output of the **show hardware** EXEC command displays the version of the TV-out hardware that the Content Engine is equipped with. In the following excerpt of the sample output from the **show hardware** command, this particular information is highlighted in bold. Rev 3 in the command output indicates that the TV-out hardware uses the newer Revision 3 MPEG decoder PCI part. The Vela II Revision D and Revision E cards use the Revision 3 part.

```
Content Engine# show hardware
.
.
.
Total 1 CPU.
1024 Mbytes of Physical memory.
1 CD ROM drive (CD-224E)
1 AV card (Vela II)
2 GigabitEthernet interfaces
1 Console interface
2 USB interfaces [Not supported in this version of software]

The following PCI cards were found:
PCI-Slot-1 MPEG-Decoder-AV [1105:8476 (Sigma Designs, Inc.) (rev 3)]
PCI-Slot-2 SCSI
Manufactured As: Pre-FCS 565  [867383Z]
.
.
.
```

Note To support the TV-out service with a Revision D or Revision E card, the Content Engine must be running the newer driver software, which is included in the ACNS 5.2.3 software and later releases, instead of an earlier version of the driver.

In the ACNS 5.2 software and later releases, the output of the **show hardware** EXEC command notifies you if the Content Engine is running a version of the ACNS software that does not support the TV-out hardware contained in the Content Engine. In the following example, you are notified that the Content Engine has a Vela II audio-video (AV) card that is not supported by the version of the ACNS software that is running on the Content Engine. In the following excerpt of the sample output from the **show hardware** command, this particular information is highlighted in bold:

```
Content Engine # show hardware
.
CPU 0 is GenuineIntel Intel(R) Celeron(R) CPU 1.70GHz (rev 1) running at 1699MHz

.
Total 1 CPU.
1024 Mbytes of Physical memory.
1 CD ROM drive (CD-224E)
1 AV card (Vela II) [***Revision not supported in this version of software***]
2 GigabitEthernet interfaces
1 Console interface
2 USB interfaces [Not supported in this version of software]

The following PCI cards were found:
.
.
.
```

Examples

Table 2-73 describes the fields shown in the **show hardware** display.

Table 2-73 show hardware Field Descriptions

Field	Description
Cisco Application and Content Networking System Software (ACNS) Copyright (c) *year* by Cisco Systems, Inc. Application and Content Networking System Software Release XXX (build bXXX month day year)	Software application, copyright, release, and build information.
Version	Version number of the software that is running on the device.
Compiled hour:minute:second month day year by cnbuild	Compile information for the software build.
System was restarted on day of week month day hour:minute:second year	Date and time that the system was last restarted.
The system has been up for X hours, X minutes, X seconds	Length of time the system has been running since the last reboot.
CPU 0 is	CPU manufacturer information.
Total X CPU	Number of CPUs on the device.
XXXX Mbytes of Physical memory	Number of megabytes of physical memory on the device.
X CD ROM drive	Number of CD-ROM drives on the device.
X FastEthernet interfaces	Number of Fast Ethernet interfaces on the device.

Table 2-73 show hardware Field Descriptions (continued)

Field	Description
X Console interface	Number of console interfaces on the device.
Cookie info	
SerialNumber	Serial number of the WAE.
SerialNumber (raw)	Serial number of the WAE as an ASCII value.
TestDate	Date that the WAE was tested.
ModelNum (text)	Hardware model of the device, for example WAE612.
ModelNum (raw)	Internal model number (ASCII value) that corresponds to the ExtModel number.
HWVersion	Number of the current hardware version.
PartNumber	Not implemented.
BoardRevision	Number of revisions for the current system board.
ChipRev	Number of revisions for the current chipset.
VendID	Vendor ID of the cookie.
CookieVer	Version number of the cookie.
Chksum	Checksum of the cookie. showing whether the cookie is valid.
List of all disk drives	
Physical disk information	Lists the disks by number. WAE 7300 series appliances show information for 6 disk drives and WAE 500 and 600 series appliances show information for 2 disk drives.
disk00	Availability of the disk: Present, Not present or Not responding, or Not used (*). Disk identification number and type. Disk size in megabytes and gigabytes.
disk01	Same type of information is shown for each disk.
Mounted filesystems	Table containing the following information:
Device	Path to the partition on the disk.
Type	Type of the file system. Values include PHYS-FS, SYSFS, CDNFS, CFS, MEDIAFS.
Size	Total size of the file system in megabytes and gigabytes.
Mount point	Mount point for the file system. For example, the mount point for SYSFS is /local/local1.
System use	Amount of disk space being used for system use.
Free	Amount of unused disk space available.

Related Commands

show version