

show statistics udp

To display Content Engine User Datagram Protocol (UDP) statistics, use the **show statistics udp** EXEC command.

show statistics udp

Syntax Description This command has no arguments or keywords.

Defaults No default behavior or values

Command Modes EXEC

Examples [Table 2-155](#) describes the fields shown in the **show statistics udp** display.

Table 2-155 *show statistics udp Field Descriptions*

Field	Description
UDP Statistics	
Packets received	Total number of UDP packets received.
Packets to unknown port received	Number of packets to unknown ports received.
Packet receive error	Number of packet receive errors.
Packet sent	Number of UDP packets sent.

show statistics url-filter

To display Content Engine URL filtering statistics, use the **show statistics url-filter** EXEC command.

show statistics url-filter {http {local-list | N2H2 | websense} | rtsp local-list | wmt local-list}

Syntax Description	http	Displays URL filtering statistics for HTTP.
	local-list	Displays URL filtering statistics for the HTTP local list.
	N2H2	Displays URL filtering statistics for HTTP N2H2.
	websense	Displays URL filtering statistics for HTTP Websense.
	rtsp	Displays URL filtering statistics for RTSP.
	local-list	Displays URL filtering statistics for the RTSP local list.
	wmt	Displays URL filtering statistics for WMT.
	local-list	Displays URL filtering statistics for the WMT local list.

Defaults No default behavior or values

Command Modes EXEC

Examples [Table 2-156](#) describes the fields shown in the **show statistics url-filter http n2h2** display.

Table 2-156 *show statistics url-filter http n2h2 Field Descriptions*

Field	Description
Lookup requests transmitted	Number of URL filtering requests sent by the Content Engine to the N2H2 server to perform necessary lookups for the URL.
Lookup response received	Number of URL filtering requests sent by the Content Engine to the N2H2 server for which responses were received after the N2H2 server performed necessary lookups for the URL.
Requests timed out	Number of URL filtering requests sent by the Content Engine to the N2H2 server that timed out while waiting for the maximum amount of time for a response from the N2H2 server.
Number of retransmits	Number of URL filtering requests retransmitted by the Content Engine to the N2H2 server to perform necessary lookups for the URL.
Requests BLOCKed by N2H2	Number of request URLs blocked by the N2H2 server. The Content Engine blocks the HTTP request by redirecting the browser to a page where a blocking message is displayed.
Requests OKed by N2H2	Number of request URLs served by the N2H2 server.
No available connection	Number of times that the Content Engine could not establish a connection with the N2H2 server to send an IFP request.

Table 2-156 *show statistics url-filter http n2h2 Field Descriptions (continued)*

Field	Description
Error sending lookup requests	Number of times that an error occurred when the Content Engine sent an IFP request to the N2H2 server to perform lookups for the URL from the server database.
Error recving lookup responses	Number of times that an error occurred when the Content Engine received an IFP response from the N2H2 after necessary lookups for the URL were performed.
Server error in responses	Number of times that an error occurred in the N2H2 server when it sent an IFP response.
Error in Filter Server	Number of errors that occurred in the filter server. The N2H2 server consults one among the many filter servers to make a filtering decision.
Error in IFP server	Number of errors that occurred in the IFP (N2H2) server.
Seq number mismatch	Number of responses received from the N2H2 server in which the data packets are out of sequence.
Multiple responses rcvd	Number of URL filtering requests sent by the Content Engine to the N2H2 server for which multiple responses were received after lookups.
Bad network endpoint	Number of times that the socket is invalid.
Network unreachable	Number of times that the Content Engine is unable to connect to the external N2H2 server. The server might not be functioning or network connectivity to the server is down.
Underlying connection broken	Number of times that the connection to the client or N2H2 server is broken.
Timeout specified is reached	Number of seconds (1–120) that the Content Engine waits for an IFP response from the N2H2 server before timing out the connection.
Register read cancelled	Number of times that the Content Engine canceled the read operation on the socket.
Other errors	Number of all other types of errors.
Number of xacts in Queue	Number of requests queued into a waiting queue after the maximum number of connections that the Content Engine makes with the N2H2 server has been reached. To prevent overloading the server, once the timeout expires, the requests from the waiting queue are blocked or served depending on the allowmode configuration.
Avg total process time	Average amount of time taken by the Content Engine to process the client requests for URL filtering.
Avg response time	Average amount of time taken to receive an IFP response from the N2H2 server.
Socket update count	Number of socket failures.

Table 2-157 describes the fields shown in the **show statistics url-filter http websense** display.

Table 2-157 *show statistics url-filter http websense Field Descriptions*

Field	Description
Websense URL Filtering Statistics	Status of whether URL filtering statistics is displayed for primary or secondary Websense servers.
Transmission statistics	Statistics regarding the requests and responses sent by the Content Engine and Websense server.
Lookup requests transmitted	Number of URL filtering requests sent by the Content Engine to Websense server to perform necessary lookups for the URL.
Lookup requests timed-out	Number of URL filtering requests sent by the Content Engine to Websense server that timed out while waiting for the maximum amount of time for a response from Websense server.
Lookup responses received	Number of URL filtering requests sent by the Content Engine to Websense server for which responses were received after Websense server performed necessary lookups for the URL.
Lookup responses received with error	Number of URL filtering requests sent by the Content Engine to Websense server for which erroneous responses were received.
Multiple response received	Number of URL filtering requests sent by the Content Engine to Websense server for which multiple responses were received after lookups.
Sequence number mismatch	Number of responses received from Websense server in which the data packets are out of sequence.
TCP errors	Statistics for the persistent TCP connection between the Content Engine and Websense server.
Connection reset	Number of times that the TCP connection between the Content Engine and Websense server was reset.
Connection timeout	Number of times that the TCP connection between the Content Engine and Websense server exceeded the maximum wait time for a response from Websense server.
Other errors	Number of times that errors other than reset and timeout occurred on the TCP connection between the Content Engine and Websense server.
Filter results	Websense URL filtering statistics.
Requests BLOCKed by Websense	Number of request URLs that are blocked by Websense server. If the Websense process blocks a URL, it sends a redirect URL to the user. The redirect URL is configured to print out the blocked page and policy for the user. The Websense process listens on this port to receive the pages blocked and serviced by a thread in Websense server. This thread sends the blocked page in response to the redirected request.
Requests OKed by Websense	Number of request URLs that are served by Websense server. By default, Websense server receives requests for the content filtering according to the Websense protocol on port 15868.

Table 2-157 *show statistics url-filter http websense Field Descriptions (continued)*

Field	Description
Sent to Allowmode ok	Number of times that client request URLs are served if the Content Engine has problems communicating with Websense server (the allowmode configuration is enabled).
Sent to Allowmode block	Number of times that client request URLs are blocked if the Content Engine has problems communicating with Websense server (the allowmode configuration is disabled).
desc_filtered_and_passed	Number of request URLs that are filtered and served by Websense server.
desc_category_blocked	Number of request URLs for a site that is listed in a blocked category in the Websense master database. If the site is listed in a blocked category, Websense displays a message alerting the user that the requested site is blocked and identifies the category under which it is blocked.
desc_category_not_blocked	Number of request URLs for a site that is not found in the master database or is listed in a permitted category. If the site is listed in a permitted category, access to the site is allowed.
desc_category_blocked_custom_deny	Number of requests for a custom URL to which access is denied by Websense server. The Websense custom URLs feature enables you to add sites to Websense that are not in the Websense master database and filter sites differently than their master database categories. Websense considers custom URLs before URLs in the master database and filters the site according to the category assigned to the custom URL.
desc_category_not_blocked_custom_permit	Number of requests for a custom URL to which access is allowed by Websense server. Custom URLs/Recategorized are sites you want to filter differently than their master database categories (including sites classified under Miscellaneous/Uncategorized). When editing a custom URL list, you can select any category and add a URL to it. Websense then filters the site according to the filtering option set for that category.
Websense log statistics	Statistics about the logging of requests by Websense server.
Logs sent successfully	<p>Number of times that Websense server sends the log information to Log Server.</p> <p>Log entries contain the following information:</p> <ul style="list-style-type: none"> • Date and time a site was requested • Category/yes list or keyword and protocol under which it was filtered • User or workstation requesting the site • URL and IP address of the requested site • How the requested site was filtered • Number of bytes contained in the requested site (called bytes transferred)

Table 2-157 *show statistics url-filter http websense Field Descriptions (continued)*

Field	Description
Connection error	Number of times that Websense server is unable to connect to Log Server. Log Server is installed with Reporter, a separate program available free with Websense Enterprise. Websense Enterprise Explorer also uses Log Server. With Log Server, Websense can save detailed log entries for all filtered Internet requests. Reporter can analyze the log entries and create reports of Internet activity in graphical or tabular format.
Common Statistics	Statistics common to primary and secondary Websense servers.
Error during log processing	Number of errors that occurred during the transmission of logs from Websense server to Log Server.
Log not complete	Number of entries in the log that are incomplete when transferred to Log Server.
Log not sent because Websense disabled	Number of times logs could not be sent to Log Server because Websense server was not enabled.
No available connection	Number of times when a connection could not be established between Websense server and Log Server.
Congestion statistics	Statistics about network congestion related to Websense URL filtering and logging of requests.
Pending requests	Number of requests that are pending to be processed by Websense server.
Pending log requests	Number of log requests that are pending to be transmitted from Websense server to Log Server.

Related Commands

clear statistics url-filter
show url-filter
url-filter

show statistics wccp

To display Content Engine WCCP statistics, use the **show statistics wccp** EXEC command.

show statistics wccp gre

Syntax Description	gre Displays WCCP generic routing encapsulation packet-related statistics.
---------------------------	---

Defaults	No default behavior or values
-----------------	-------------------------------

Command Modes	EXEC
----------------------	------

Usage Guidelines

GRE is a Layer 3 technique that allows datagrams to be encapsulated into IP packets at the WCCP-enabled router and then redirected to a Content Engine (the transparent proxy server). At this intermediate destination, the datagrams are decapsulated and then routed to an origin server to satisfy the request if a cache miss occurs. In doing so, the trip to the origin server appears to the inner datagrams as one hop. Usually, the redirected traffic using GRE is referred to as GRE tunnel traffic. With GRE, all redirection is handled by the router software.

With WCCP redirection, a Cisco router does not forward the TCP SYN packet to the destination because the router has WCCP enabled on the destination port of the connection. Instead, the WCCP-enabled router encapsulates the packet using GRE tunneling and sends it to the Content Engine that has been configured to accept redirected packets from this WCCP-enabled router.

After receiving the redirected packet, the Content Engine does the following:

1. Strips the GRE layer from the packet.
2. Decides whether it should accept this redirected packet and process the request for the content as follows:
 - a. If the Content Engine decides to accept the request, it sends a TCP SYN ACK packet to the client. In this response packet, the Content Engine uses the IP address of the original destination (origin server) that was specified as the source address so that the Content Engine can be invisible (transparent) to the client; it pretends to be the destination that the client's TCP SYN packet was trying to reach.
 - b. If the Content Engine decides not to accept the request, it reencapsulates the TCP SYN packet in GRE and sends it back to the WCCP-enabled router. The router understands that the Content Engine is not interested in this connection and forwards the packet to its original destination (the origin server).

For example, a Content Engine would decide not to accept the request because it is configured to bypass requests that originate from a certain set of clients or that are destined to a particular set of servers.

Examples Table 2-158 describes the fields shown in the **show statistics wccp gre** display.

Table 2-158 *show statistics wccp gre Field Descriptions*

Field	Description
Transparent GRE packets received	Total number of GRE packets received by the Content Engine, regardless of whether they have been intercepted by WCCP or not. GRE is a Layer 3 technique that allows packets to reach the Content Engine even if there are any number of routers in the path to the Content Engine.
Transparent non-GRE packets received	Number of non-GRE packets received by the Content Engine either using the traffic interception and redirection functions of WCCP in the router hardware at Layer 2 or Layer 4 switching (a Content Services Switch [CSS]) that redirects requests transparently to the Content Engine.
Transparent non-GRE packets passed through	Number of non-GRE packets transparently intercepted by a Layer 4 switch and redirected to the Content Engine.
Total packets accepted	Total number of packets that are transparently intercepted and redirected to the Content Engine to serve client requests for content.
Invalid packets received	Number of packets that are dropped either because the redirected packet is a GRE packet and the WCCP GRE header has invalid data or the IP header of the redirected packet is invalid.
Packets received with invalid service	Number of WCCP version 2 GRE redirected packets that contain an invalid WCCP service number.
Packets received on a disabled service	Number of WCCP version 2 GRE redirected packets that specify the WCCP service number for a service that is not enabled on the Content Engine. For example, a HTTPS request redirected to the Content Engine when the HTTPS-caching service (service 70) is not enabled.
Packets received too small	Number of GRE packets redirected to the Content Engine that do not contain the minimum amount of data required for a WCCP GRE header.
Packets dropped due to zero TTL	Number of GRE packets that are dropped by the Content Engine because the redirected packet's IP header has a zero TTL.
Packets dropped due to bad buckets	<p>Number of packets that are dropped by the Content Engine because the WCCP flow redirection could not be performed due to a bad mask or hash bucket determination.</p> <p>Note A bucket is defined as a certain subsection of the allotted hash assigned to each Content Engine in a Content Engine cluster. If only one Content Engine exists in this environment, it has 256 buckets assigned to it.</p>
Packets dropped due to no redirect address	Number of packets that are dropped because the flow redirection destination IP address could not be determined.
Packets dropped due to loopback redirect	Number of packets that are dropped by the Content Engine when the destination IP address is the same as the loopback address.

Table 2-158 *show statistics wccp gre Field Descriptions (continued)*

Field	Description
Connections bypassed due to load	Number of connection flows that are bypassed when the Content Engine is overloaded. When the overload bypass option is enabled, the Content Engine bypasses a bucket and reroutes the overload traffic. If the load remains too high, another bucket is bypassed, and so on, until the Content Engine can handle the load.
Packets sent back to router	Number of requests that are passed back by the Content Engine to the WCCP-enabled router from which the request was received. The router then sends the flow toward the origin web server directly from the web browser, which bypasses the Content Engine.
Packets sent to another CE	Number of packets that are redirected to another Content Engine in the WCCP service group. Service groups consist of up to 32 Content Engines and 32 WCCP-enabled routers. In both packet-forwarding methods, the hash parameters specify how redirected traffic should be load balanced among the Content Engines in the various WCCP service groups.
GRE fragments redirected	Number of GRE packets received by the Content Engine that are fragmented.
Packets failed GRE encapsulation	Number of GRE packets that are dropped by the Content Engine because they could not be redirected due to problems while encapsulating the packet with a GRE header.
Packets dropped due to invalid fwd method	Number of GRE packets that are dropped by the Content Engine because it was redirected using GRE but the WCCP service was configured for Layer 2 redirection.
Packets dropped due to insufficient memory	Number of GRE packets that are dropped by the Content Engine due to the failure to allocate additional memory resources required to handle the GRE packet.
Packets bypassed, no conn at all	Number of packets that failed to be associated with an existing flow. WCCP can also handle asymmetric packet flows and always maintains a consistent mapping of web servers to caches regardless of the number of switches or routers used in a WCCP service group (up to 32 routers or switches communicating with up to 32 Content Engines in a cluster).
Packets bypassed, no pending connection	Number of packets that failed to be associated with a pending connection.
Packets due to clean wccp shutdown	Number of connection flows that are bypassed due to a clean WCCP shutdown. During a proper shutdown of WCCP, the Content Engine continues to service the flows it is handling but starts to bypass new flows. When the number of flows goes down to zero, the Content Engine takes itself out of the cluster by having its buckets reassigned to other Content Engines by the lead Content Engine.

Table 2-158 *show statistics wccp gre Field Descriptions (continued)*

Field	Description
Packets bypassed due to bypass-list lookup	Number of connection flows that are bypassed due to a bypass list entry. When the Content Engine receives an error response from an origin server, it adds an entry for the server to its bypass list. When it receives subsequent requests for the content residing on the bypassed server, it redirects packets to the bypass gateway. If no bypass gateway is configured, then the packets are returned to the redirecting Layer 4 switch.
Packets received with client IP addresses	Number of packets that are associated to a connection flow that is being spoofed. By spoofing a client's IP address, the Content Engine can receive packets with the client IP (which is different from the Content Engine's own IP address) and send the packet to the correct application that is waiting for the packet.
Conditionally Accepted connections	Number of connection flows that are accepted by the Content Engine due to the conditional accept feature.
Conditionally Bypassed connections	Number of connection flows that are bypassed by the Content Engine due to the conditional accept feature.
L2 Bypass packets destined for loopback	Number of packets that are dropped by the Content Engine due to the destination IP address being the loopback address when the WCCP-enabled router or switch tries to perform Layer 2 redirection.
L2 Packets fragmented for bypass	Number of GRE packets that do not contain enough data to hold an IP header.
Packets dropped due to IP access-list deny	Number of packets that are dropped by the Content Engine when an IP access list that the Content Engine applies to WCCP GRE encapsulated packets denies access to WCCP applications (the wccp access-list command).
Packets w/WCCP GRE received too small	Number of packets transparently intercepted by the WCCP-enabled router at Layer 2 and sent to the Content Engine that need to be fragmented for the packets to be redirected using GRE.

Related Commands

wccp

show statistics wmt

To display Content Engine WMT (Windows Media Technologies) statistics, use the **show statistics wmt** EXEC command.

```
show statistics wmt {all | bytes [incoming | outgoing] | errors | multicast | requests | rule |
savings | streamstat [incoming | live | outgoing | stream-id 1-999999] | urlfilter | usage}
```

Syntax Description		
all		Displays all WMT statistics.
bytes		Displays unicast byte statistics.
incoming		(Optional) Displays unicast incoming byte statistics.
outgoing		(Optional) Displays unicast outgoing byte statistics.
errors		Displays error statistics.
multicast		Displays multicast statistics.
requests		Displays unicast request statistics.
rule		Displays the Rule Template statistics.
savings		Displays savings statistics.
streamstat		Displays Windows Media streaming connections.
incoming		(Optional) Displays statistics of all incoming WMT streams from the Content Engine.
live		(Optional) Displays aggregated live stream statistics.
outgoing		(Optional) Displays statistics of all outgoing WMT streams from the Content Engine.
stream-id		(Optional) Displays statistics of the WMT streams that have the specified stream ID.
<i>1-999999</i>		WMT stream ID to display.
urlfilter		Displays URL filtering statistics.
usage		Displays current usage statistics.

Defaults No default behavior or values

Command Modes EXEC

Usage Guidelines In the ACNS 5.3 software, the output of the **show statistics wmt** EXEC commands was modified to include information about WMT RTSP requests. For example, the output from the **show statistics wmt** EXEC commands was changed as follows:

- RTSP-related information was added to the **show statistics wmt all** command output.
- Information about RTSPT and RTSPU was added in the transport protocol portion of the **show statistics wmt bytes** command output.

- RTSPT and RTSPU errors were added to the **show statistics wmt errors** command output.
- The **show statistics wmt requests** command output includes the RTSPT and RTSPU protocols and Fast Start and Fast Cache data.

In the ACNS 5.3 software, the **live** option was added to the **show statistics wmt streamstat EXEC** command to enable you to display aggregated live statistics. Also, the **incoming**, **outgoing**, and **stream-id** options were added to the **show statistics wmt streamstat EXEC** command to display statistics of all incoming WMT streams, outgoing WMT streams, and streams with the specified ID.

Examples

Table 2-159 describes the fields shown in the **show statistics wmt all** display.

Table 2-159 *show statistics wmt all Field Descriptions*

Field	Description
Unicast Requests Statistics	
Total unicast requests received	Total number of Unicast requests received. Display shows the number of requests in each category and calculates the percentage of the total for each category.
Streaming Requests served	Number of streaming requests received.
Mcast nsc file Request	Number of multicast NSC file requests received.
Authenticate Requests	Number of authenticated requests received.
Requests error	Number of request errors received.
By Type of Content	
Live content	Number of live content requests received.
On-Demand Content	Number of on-demand content requests received.
By Transport Protocol	
HTTP	Number of HTTP requests received.
RTSPT	Number of RTSPT requests received.
RTSPU	Number of RTSPU requests received.
Unicast Savings Statistics	
Total bytes saved	Total number of bytes saved.
By Source of Content	
Local	Number of local bytes saved.
Remote HTTP	Number of remote HTTP bytes saved.
Remote RTSP	Number of remote RTSP bytes saved.
Multicast	Number of multicast bytes saved.
CDN-Related WMT Requests	
CDN Content Hits	Number of CDN content request hits.

Table 2-159 *show statistics wmt all Field Descriptions (continued)*

Field	Description
CDN Content Misses	Number of CDN content request misses.
CDN Content Live	Number of CDN live content requests.
CDN Content Errors	Number of CDN content request errors.
Fast Streaming related WMT Requests	
Normal Speed	Number of normal-speed Fast Streaming-related WMT requests.
Fast Start Only	Number of Fast Start WMT requests.
Fast Cache Only	Number of Fast Cache WMT requests.
Fast Start and Fast Cache	Number of Fast Start and Fast Cache WMT requests.
Authenticated Requests	
By Type of Authentication	
Negotiate	Number of negotiated authentication authenticated requests.
NTLM	Number of NTLM authentication authenticated requests.
Digest	Number of Digest authentication authenticated requests.
Basic	Number of basic authentication authenticated requests.
Unicast Bytes Statistics	
Total unicast incoming bytes	Total number of bytes incoming as Unicast streams.
By Type of Content	
Live content	Number of bytes incoming as Unicast streams for live content.
On-Demand Content	Number of bytes incoming as Unicast streams for on-demand content.
By Transport Protocol	
HTTP	Number of bytes incoming as Unicast streams using the HTTP transport protocol.
RTSPT	Number of bytes incoming as Unicast streams using the RTSPT transport protocol.
Total unicast outgoing bytes	Total number of bytes outgoing as Unicast streams.
Unicast Savings Statistics	
Total bytes saved	Total number of bytes saved.
By pre-positioned content	Number of bytes saved for pre-positioned content.
By live-splitting	Number of bytes saved for live-splitting content.
By cache-hit	Number of bytes saved for cached content.
Live Splitting	

Table 2-159 *show statistics wmt all Field Descriptions (continued)*

Field	Description
Incoming bytes	Number of bytes incoming as live-split streams.
Outgoing bytes	Number of bytes outgoing as live-split streams.
Bytes saved	Number of bytes saved.
Caching	
Bytes cache incoming	Number of bytes incoming for the cache.
Bytes cache outgoing	Number of bytes outgoing from the cache.
Bytes cache total	Total number of bytes cached.
Bytes cache-bypassed	Number of bytes that bypassed the cache.
Cacheable requests	Number of cacheable requests.
Req cache-miss	Number of cacheable requests that were cache misses.
Req cache-hit	Number of cacheable requests that were cache hits.
Req cache-partial-hit	Number of cacheable requests that were partial cache hits.
Req cache-total	Total number of requests that were cached.
Objects not cached	Number of objects that were not cached.
Cache bypassed	Number of objects that were not cached because they bypassed the cache.
Exceed max-size	Number of objects that were not cached because they exceeded the maximum cacheable size limit.
Usage Summary	
Concurrent Unicast Client Sessions	Total number of concurrent unicast client sessions.
Current	Number of concurrent unicast client sessions currently running.
Max	Maximum number of concurrent unicast client sessions recorded.
Concurrent Remote Server Sessions	Total number of concurrent remote server sessions.
Concurrent Active Multicast Sessions	Total number of concurrent active multicast sessions.
Concurrent Unicast Bandwidth (Kbps)	Total amount of bandwidth being used (in kilobits per second) for concurrent unicast sessions.
Concurrent Bandwidth to Remote Servers (Kbps)	Total amount of bandwidth being used (in kilobits per second) for concurrent remote server sessions.
Concurrent Multicast Out Bandwidth (Kbps)	Total amount of bandwidth being used (in kilobits per second) for concurrent multicast out sessions.

Table 2-159 *show statistics wmt all Field Descriptions (continued)*

Field	Description
Error Statistics	
Total request errors	Total number of request errors.
Errors generated by this box	Number of request errors generated by this device.
Errors generated by remote servers	Number of request errors generated by remote servers.
Other Statistics	
Authentication Retries from Clients	Number of authentication retries from clients.
WMT Rule Template Statistics	
URL Rewrite	Number of URL rewrites.
Connection Reset	Number of connection resets.
URL Block	Number of blocked URLs.
No-Auth	Number of No-Auth matches.
No-Cache	Number of No-Cache matches.
Cache-only	Number of Cache-only matches.
Allow	Number of allow matches.
WMT URL Filter Statistics	
URL Allowed	Number of URLs allowed.
URL Filtered	Number of URLs filtered.
Multicast Statistics	
Total Multicast Outgoing Bytes	Total number of bytes outgoing as multicast-out streams.
Total Multicast Logging Requests	Total number of multicast logging requests.
Aggregate Multicast Out Bandwidth (Kbps)	Aggregated amount of bandwidth being used (in kilobits per second) for multicast out sessions.
Current	Number of concurrent multicast out sessions currently running.
Max	Maximum number of multicast out sessions recorded.
Number of Concurrent Active Multicast Sessions	Number of concurrent active multicast sessions.
List of All Multicast Stations Configured using the CLI	
Total Number of Configured Multicast Stations	Total number of configured multicast stations.

Related Commands clear statistics wmt
 show wmt
 wmt

show sysfs

To display system file system (sysfs) information, use the **show sysfs** EXEC command.

show sysfs volumes

Syntax Description	volumes	Displays system file system volumes.
---------------------------	----------------	--------------------------------------

Defaults	No default behavior or values
-----------------	-------------------------------

Command Modes	EXEC
----------------------	------

Examples	The system file system (sysfs) stores log files, including transaction logs, syslogs, and internal debugging logs. It also stores system image files and operating system files. The show sysfs volumes command displays the disk volume number and its size.
-----------------	--

[Table 2-160](#) describes the fields shown in the **show sysfs volumes** display.

Table 2-160 *show sysfs volumes Field Descriptions*

Field	Description
sysfs 00–04	System file system and disk number.
/local/local1–5	Mount point of the volume.
nnnnnnKB	Size of the volume in kilobytes.
nn% free	Percentage of free space in the SYSFS partition.

Related Commands	disk config sysfs show disks details sysfs
-------------------------	---

show tacacs

To display TACACS+ authentication protocol configuration information, use the **show tacacs EXEC** command.

show tacacs

Syntax Description This command has no arguments or keywords.

Defaults No default behavior or values

Command Modes EXEC

Examples The **show tacacs** command displays the TACACS configuration for the Content Engine. [Table 2-161](#) describes the fields shown in the **show tacacs** display.

Table 2-161 show tacacs Field Descriptions

Field	Description
Login Authentication for Console/Telnet Session	Status of whether TACACS+ server is enabled for login authentication.
Configuration Authentication for Console/Telnet Session	Status of whether TACACS+ server is enabled for authorization or configuration authentication.
Authentication scheme fail-over reason	Status of whether Content Engines fail over to the secondary method of administrative login authentication whenever the primary administrative login authentication method is used.
TACACS+ Configuration	TACACS+ server parameters.
TACACS+ Authentication	Status of whether TACACS+ authentication is enabled on the Content Engine.
Key	Secret key that the Content Engine uses to communicate with the TACACS+ server. The maximum number of characters in the TACACS+ key should not exceed 99 printable ASCII characters (except tabs).
Timeout	Number of seconds that the Content Engine waits for a response from the specified TACACS+ authentication server before declaring a timeout.
Retransmit	Number of times that the Content Engine is to retransmit its connection to the TACACS+ if the TACACS+ timeout interval is exceeded.
Password type	Mechanism for password authentication. By default, the Password Authentication Protocol (PAP) is the mechanism for password authentication.

Table 2-161 *show tacacs Field Descriptions (continued)*

Field	Description
Server	Hostname or IP address of the TACACS+ server.
Status	Status of whether server is the primary or secondary host.

Related Commands

clear statistics tacacs
show statistics tacacs
tacacs

show tcp

To display Transmission Control Protocol (TCP) configuration information, use the **show tcp** EXEC command.

show tcp

Syntax Description This command has no arguments or keywords.

Defaults No default behavior or values

Command Modes EXEC

Examples The **show tcp** command displays TCP configuration details for the Content Engine.

[Table 2-162](#) describes the fields shown in the **show tcp** display.

Table 2-162 show tcp Field Descriptions

Field	Description
TCP keepalive timeout	Length of time that the Content Engine keeps a connection open before disconnecting.
TCP keepalive probe count	Number of times that the Content Engine can retry a connection before it is considered unsuccessful.
TCP keepalive probe interval	Length of time that the Content Engine keeps an idle connection open.
TCP server r/w timeout	Period after which the Content Engine times out when attempting to read or write to the network.
TCP client r/w timeout	Period after which the Content Engine times out when attempting to read or write to the network.
TCP server send buffer	TCP sending buffer size in kilobytes (1–512) for outgoing TCP packets.
TCP server receive buffer	TCP receiving buffer size in kilobytes (1–512) for incoming TCP packets.
TCP client send buffer	TCP sending buffers size in kilobytes (1–512) for outgoing TCP packets.
TCP client receive buffer	TCP receiving buffer size in kilobytes (1–512) for incoming TCP packets.
TCP server max segment size	Maximum packet size sent to the server.
TCP satellite (RFC1323)	Status of whether the client and server TCP compliance are set to the RFC 1323 standard.
TCP client max segment size	Maximum packet size sent to clients.
TCP explicit congestion notification	Status of whether reduction of delay and packet loss is enabled.

Table 2-162 *show tcp Field Descriptions (continued)*

Field	Description
TCP type of service	Type of Service.
TCP cwnd base value	Congestion window size in segments.
TCP initial slowstart threshold value	Threshold for slow start in segments.
TCP increase (multiply) retransmit timer	Factor used to modify the length of the retransmit timer.

Related Commands

clear statistics tcp
show statistics tcp
tcp

show tech-support

To view information necessary for Cisco's Technical Assistance Center (TAC) to assist you, use the **show tech-support EXEC** command.

show tech-support [page | streaming]

Syntax Description	page	(Optional) Specifies the pages through the output.
	streaming	(Optional) Displays technical support information specific to the streaming feature.

Defaults No default behavior or values

Command Modes EXEC

Usage Guidelines Use this command to view system information necessary for TAC to assist you with your Content Engine. We recommend that you log the output to a disk file. Use the streaming option to view information specific to the streaming feature.

Examples The following example shows the types of information available about the ACNS software. Because the **show tech-support** command output is comprehensive and can be extensive, only excerpts are shown in the following example.

ContentEngine# **show tech-support**

CPU Usage:

```
cpu: 0.39% User, 0.42% System, 0.33% User(nice), 98.86% Idle
cpu0: 0.39% User, 0.42% System, 0.33% User(nice), 98.86% Idle
```

```
-----
PID  STATE PRI  User T   SYS T      COMMAND
-----
  1   S    0   4386 1706 (init)
  2   S    0     0  0 (keventd)
  3   S   19     0  0 (ksoftirqd_CPU0)
  4   S    0     0  0 (kswapd)
  5   S    0     0  0 (bdflood)
  6   S    0     0  0 (kupdated)
  7   S    0     0  0 (scsi_eh_0)
 45   S    0   4733 4114 (nodemgr)
 46   S    0     0  0 (syslogd)
 47   R    0    83  65 (dataserver)
 920  S    0     0  0 (login)
 921  S    0    123  68 (inetd)
1207  S    0     0  0 (parser_server)
1208  S    0     0  0 (eval_timer_mana)
1211  S    0    46  1 (parser_server)
1442  S    0     0  0 (wccp)
1443  S    0     0  0 (overload)
1444  S    0     0  0 (standby)
1445  S    0    13  29 (cache)
```

show tech-support

```

1446 S 0 0 0 (proxy_poll)
1447 S 0 0 0 (snmpced)
1448 S 0 0 0 (http_authmod)
1458 S 0 0 0 (http_authmod)
1465 S 0 0 0 (http_authmod)
1466 S 0 0 0 (http_authmod)
1467 S 0 0 0 (http_authmod)
1537 S 0 0 0 (cache)
1538 S 0 0 0 (unified_log)
1540 S 0 0 1 (webserver)
1541 S 0 2 2 (mcm)
1542 S 0 0 0 (cache)
1543 S 0 0 0 (cache)
1550 S 0 0 0 (cache)
1551 S 0 0 0 (cache)
1556 S 0 0 0 (cache)
1567 S 0 0 0 (mcm)
1568 S 0 0 0 (mcm)
1629 S 0 18982 4140 (crond)
1936 S 0 1669 611 (bootnet)
1937 S 10 0 0 (tracknet)
1938 S 10 33545 5556 (checkup)
1983 S 0 0 0 (srcpd)
2023 S 0 1 0 (admin-shell)
2024 S 0 0 0 (parser_server)
2150 S 0 0 0 (rsvpd)
2152 S 0 0 0 (rtspd)
2153 S 0 1635 1067 (httpsd)
2164 S 0 0 0 (librarian)
2167 S 0 1667 2105 (libaux)
2170 S 0 0 0 (mapper)
2178 S 0 32 37 (cache)
2179 S 0 0 0 (router)
2180 S 0 0 0 (fill)
2183 S 0 0 0 (remotereq)
2185 S -20 0 0 (videosvr)
2188 S 0 9 4 (contentsvr)
2189 S 0 0 0 (routeraux)
2190 S 0 0 1 (dfcontrolsrvr)
2226 S 0 0 0 (smbd)
2228 S 0 0 0 (nmbd)
2973 Z 0 0 0 (cache)
8446 S 0 0 0 (httpsd)
8447 S 0 0 0 (gcache)
18173 S 0 0 0 (in.telnetd)
18174 S 0 0 0 (login)
18175 S 0 2 2 (admin-shell)
18176 S 0 0 0 (parser_server)
19426 S 0 0 0 (httpsd)
19427 S 0 0 0 (httpsd)
19456 Z 0 0 0 (cache)
19503 Z 0 30 3 (crond)
19515 S 0 0 0 (more)
19516 S 0 6 18 (exec_show_tech-)
19553 R 0 0 0 (exec_show_proce)

```

----- process memory -----

Total	Used	Free	Shared	Buffers	Cached
1050943488	564785152	486158336	0	5222400	475176960

PID	State	TTY	%MEM	VM Size	RSS (pages)	Name
1	S	0	0.0	1146880	119	(init)
2	S	0	0.0	0	0	(keventd)
3	S	0	0.0	0	0	(ksoftirqd_CPU0)
4	S	0	0.0	0	0	(kswapd)
5	S	0	0.0	0	0	(bdfld)
6	S	0	0.0	0	0	(kupdated)
7	S	0	0.0	0	0	(scsi_eh_0)
45	S	0	0.0	1208320	143	(nodemgr)
46	S	0	0.0	1630208	194	(syslogd)
47	R	0	0.0	1974272	238	(dataserver)
920	S	1088	0.0	1728512	236	(login)
921	S	0	0.0	1191936	130	(inetd)
1207	S	0	0.3	4980736	847	(parser_server)
1208	S	0	0.0	1933312	151	(eval_timer_mana)
1211	S	0	0.3	4980736	847	(parser_server)
1442	S	0	0.0	2232320	163	(wccp)
1443	S	0	0.0	1548288	154	(overload)
1444	S	0	0.0	1724416	161	(standby)
1445	S	0	5.9	65646592	15266	(cache)
1446	S	0	0.0	1957888	173	(proxy_poll)
1447	S	0	0.1	2097152	290	(snmpcd)
1448	S	0	0.0	1757184	205	(http_authmod)
1458	S	0	0.0	1757184	205	(http_authmod)
1465	S	0	0.0	1757184	205	(http_authmod)
1466	S	0	0.0	1757184	205	(http_authmod)
1467	S	0	0.0	1757184	205	(http_authmod)
1537	S	0	5.9	65646592	15266	(cache)
1538	S	0	0.0	1789952	169	(unified_log)
1540	S	0	0.4	10817536	1164	(webserver)
1541	S	0	0.0	2150400	251	(mcm)
1542	S	0	5.9	65646592	15266	(cache)
1543	S	0	5.9	65646592	15266	(cache)
1550	S	0	5.9	65646592	15266	(cache)
1551	S	0	5.9	65646592	15266	(cache)
1556	S	0	5.9	65646592	15266	(cache)
1567	S	0	0.0	2150400	251	(mcm)
1568	S	0	0.0	2150400	251	(mcm)
1629	S	0	0.0	1187840	137	(crond)
1936	S	0	0.6	7532544	1605	(bootnet)
1937	S	0	0.2	3215360	545	(tracknet)
1938	S	0	0.2	3637248	654	(checkup)
1983	S	0	0.3	4374528	838	(srcpd)
2023	S	1088	0.0	2146304	182	(admin-shell)
2024	S	0	0.3	4980736	847	(parser_server)
2150	S	0	0.0	1679360	188	(rsvpd)
2152	S	0	0.3	6217728	881	(rtspd)
2153	S	0	0.1	2527232	329	(httpsd)
2164	S	0	0.3	6533120	990	(librarian)
2167	S	0	0.4	7110656	1144	(libaux)
2170	S	0	0.3	5955584	863	(mapper)
2178	S	0	0.3	6135808	927	(cache)
2179	S	0	0.3	6287360	948	(router)
2180	S	0	0.3	5955584	926	(fill)
2183	S	0	0.3	5832704	852	(remotereg)
2185	S	0	0.3	8269824	873	(videosvr)
2188	S	0	0.4	7651328	1196	(contentsvr)
2189	S	0	0.3	6103040	953	(routeraux)
2190	S	0	0.4	10272768	1075	(dfcontrolsvr)
2226	S	0	0.1	3559424	504	(smbd)
2228	S	0	0.0	2084864	247	(nmbd)
2973	Z	0	0.0	0	0	(cache)
8446	S	0	0.1	2506752	327	(httpsd)


```

      8447      S      0 0.0    1421312      116 (gcache)
     18173      S      0 0.0    1220608      132 (in.telnetd)
     18174      S 34816 0.0    1736704      238 (login)
     18175      S 34816 0.0    2162688      184 (admin-shell)
     18176      S      0 0.3    4980736      847 (parser_server)
     19426      S      0 0.1    2551808      350 (httpsd)
     19427      S      0 0.1    2576384      354 (httpsd)
     19456      Z      0 0.0          0          0 (cache)
     19503      Z      0 0.0          0          0 (crond)
     19515      S 34816 0.0    1163264      109 (more)
     19516      S 34816 0.0    1941504      168 (exec_show_tech-)
     19554      R 34816 0.1    2277376      266 (exec_show_proce)

```

```
----- system memory -----
```

```

Total physical memory   :    1026312 KB
Total free memory       :     474692 KB
Total memory shared     :           0 KB
Total buffer memory     :       5100 KB
Total cached memory     :    464040 KB

```

```
----- interfaces -----
```

```

Interface type: FastEthernet Slot: 0 Port: 0
Type:Ethernet
Ethernet address:00:05:32:02:DD:74
Internet address:172.16.5.234
Broadcast address:172.16.5.255
Netmask:255.255.255.0
Maximum Transfer Unit Size:1500
Metric:1
Packets Received: 513241
Input Errors: 0
Input Packets Dropped: 0
Input Packets Overruns: 0
Input Packets Frames: 0
Packet Sent: 153970
Output Errors: 0
Output Packets Dropped: 0
Output Packets Overruns: 0
Output Packets Carrier: 0
Output Queue Length:100
Collisions: 0
Interrupts:9
Flags:UP BROADCAST RUNNING MULTICAST
Mode:autoselect, full-duplex, 100baseTX

```

The following types of information are available when using the streaming option with the **show tech-support** command.

General Information

You can access the following general information when you enter the **show tech-support** command:

- Version and hardware ([show version](#))
- show running-config
- Processes ([show processes](#))
- Process memory ([show processes memory](#))
- System memory ([show memory](#))
- File system information

- Interface information
- WCCP information (HTTP and streaming protocols)
- Media file system statistics
- Application and kernel core dump information
- Netstat

Information Common to WMT and RTSP

Information that is common to both WMT and RTSP are as follows:

- [show programs](#)
- [show statistics wmt streamstat](#)
- [show bandwidth](#)
- [show bitrate](#)
- [show acquirer channels](#)
- Rules ([show rule all](#))
- Distribution channel details

Information Specific to WMT

Information that is specific to WMT is as follows:

- [show wmt](#)
- [show statistics wmt all](#)

Information Specific to RTSP

Information that is specific to RTSP is as follows:

- [show rtsp all](#)
- [show statistics rtsp server cisco-streaming-engine all](#)
- [show statistics rtsp proxy media-real requests](#)
- [show statistics rtsp proxy media-real savings](#)

show telnet

To display the Telnet services configuration, use the **show telnet** EXEC command.

show telnet

Syntax Description	This command has no arguments or keywords.
---------------------------	--

Defaults	Enabled
-----------------	---------

Command Modes	EXEC
----------------------	------

Examples	The following example displays the Telnet service details:
-----------------	--

```
ContentEngine# show telnet  
telnet service is enabled
```

Related Commands	exec-timeout telnet enable
-------------------------	---

show tftp-server

To display the Trivial File Transfer Protocol (TFTP) server directory, use the **show tftp-server** EXEC command.

show tftp-server

Syntax Description This command has no arguments or keywords.

Defaults No default behavior or values

Command Modes EXEC

Examples [Table 2-163](#) describes the fields shown in the **show tftp-server** display.

Table 2-163 *show tftp-server Field Descriptions*

Field	Description
TFTP protocol is disabled	Status of whether TFTP service is enabled on the Content Engine.
TFTP Server Directory List	One or more local directories that the Content Engine should search for requested files when the full pathname is not included in the TFTP request.
TFTP Server Gateway Configuration	TFTP gateway parameters configured on the Content Engine.
No.	Serial number of the listed parameters.
Proto	Protocol used to access the origin server to which the TFTP gateway will forward requests when the file cannot be found in a local directory.
Server-IP-Addr	Hostname or IP address of the origin server.
Pri	Priority (1 or 2) of the origin server.
Dir-Path	Pathname to search for files on the origin server.
Usr-Name	Username for authentication to the origin server.
Password	Password for authentication to the origin server.
TFTP Access-list configured	IP ACL that allows access to the TFTP server and gateway.

Related Commands tftp-server

show transaction-logging

To display the transaction log configuration settings and a list of archived transaction log files, use the **show transaction-logging EXEC** command.

show transaction-logging

Syntax Description	This command has no arguments or keywords.
---------------------------	--

Defaults	No default behavior or values
-----------------	-------------------------------

Command Modes	EXEC
----------------------	------

Usage Guidelines	To display information about the current configuration of transaction logging on a Content Engine, use the show transaction-log or show transaction-logging EXEC commands. Both of these EXEC commands display the same output. Transaction log file information is displayed for HTTP and WMT caching proxy transactions and TFTP and ICAP transactions.
-------------------------	---



Note

For security reasons, passwords are never displayed in the output of the **show transaction-log EXEC** command.

Examples	The following example displays information about the current configuration of transaction logging on a Content Engine:
-----------------	--

```
ContentEngine# show transaction-logging
Transaction log configuration:
-----
Logging is enabled.
End user identity is visible.
File markers are disabled.
Archive interval: every-day every 1 hour
Maximum size of archive file: 2000000 KB
Log File format is squid.
Windows domain is not logged with the authenticated username

Exporting files to ftp servers is disabled.
File compression is disabled.
Export interval: every-day every 1 hour

HTTP Caching Proxy logging to remote syslog host is disabled.
Remote syslog host is not configured.
Facility is the default "*" which is "user".
Log HTTP request authentication failures with auth server to remote syslog host.

HTTP Caching Proxy Transaction Log File Info
Working Log file - None existing
```

WMT MMS Caching Proxy/Server Transaction Log File Info

Working Log file - size : 584

age: 3449567

Archive Log file - mms_export_10.1.1.21_20040805_160226 size: 584

Archive Log file - mms_export_10.1.1.21_20040803_155410 size: 584

Translog directory doesn't exist. Maybe because /local1 has no sysfs mounted.

Translog directory doesn't exist. Maybe because /local1 has no sysfs mounted.

A&D Transaction Log File Info

Working Log file - size : 138

age: 98065

Archive Log file - acqdist_10.1.1.21_20040814_080005 size: 425249

Archive Log file - acqdist_10.1.1.21_20040814_090020 size: 248744

Archive Log file - acqdist_10.1.1.21_20040814_100001 size: 402181

Archive Log file - acqdist_10.1.1.21_20040814_110006 size: 494790

Archive Log file - acqdist_10.1.1.21_20040814_120000 size: 402

Archive Log file - acqdist_10.1.1.21_20040814_120001 size: 232137

Archive Log file - acqdist_10.1.1.21_20040814_130001 size: 32752

Archive Log file - acqdist_10.1.1.21_20040814_140020 size: 243

Archive Log file - acqdist_10.1.1.21_20040815_120022 size: 356

Archive Log file - acqdist_10.1.1.21_20040820_110032 size: 248

Archive Log file - acqdist_10.1.1.21_20040820_120015 size: 1946

Archive Log file - acqdist_10.1.1.21_20040902_150040 size: 1306338

Archive Log file - acqdist_10.1.1.21_20040903_180000 size: 388

Archive Log file - acqdist_10.1.1.21_20040903_180001 size: 1352868

Archive Log file - acqdist_10.1.1.21_20040903_190017 size: 466

Archive Log file - acqdist_10.1.1.21_20040913_110048 size: 138

Translog directory doesn't exist. Maybe because /local1 has no sysfs mounted.

Real Proxy Transaction Log File Info

Working Log file - None existing

Cisco Streaming Engine Transaction Log File Info

Working Log file - size : 923

age: 3622854

Archive Log file - cseaccess.log size: 923

Archive Log file - cseaccess.040517000.log size: 1785

Archive Log file - cseaccess.040524000.log size: 1153

Archive Log file - cseaccess.040531000.log size: 693

Archive Log file - cseaccess.040607000.log size: 923

Archive Log file - cseaccess.040614000.log size: 865

Archive Log file - cseaccess.040621000.log size: 1325

Archive Log file - cseaccess.040628000.log size: 1324

Archive Log file - cseaccess.040705000.log size: 865

Archive Log file - cseaccess.040712000.log size: 923

Archive Log file - cseaccess.040719000.log size: 981

Archive Log file - cseaccess.040727000.log size: 1268

TV-out Transaction Log File Info

Working Log file - size : 48

age: 3449566

Archive Log file - tvout_10.1.1.21_20040805_160227 size: 48

Archive Log file - tvout_10.1.1.21_20040803_155411 size: 48

CIFS (Windows File Sharing) Transaction Log File Info

Working Log file - size : 58

age: 3449566

Archive Log file - cifs_server_10.1.1.21_20040803_155411 size: 58

Archive Log file - cifs_server_10.1.1.21_20040805_160227 size: 58

TFTP Transaction Log File Info

Working Log file - size : 88

age: 3449566

Archive Log file - tftp_server_10.1.1.21_20040803_155411 size: 88

Archive Log file - tftp_server_10.1.1.21_20040805_160227 size: 88

ICAP Transaction Log File Info

Working Log file - size : 61

age: 3449566

show transaction-logging

```
Archive Log file - icap_10.1.1.21_20040803_155411      size: 61
Archive Log file - icap_10.1.1.21_20040805_160227      size: 61
ContentEngine#
```

Related Commands

```
clear statistics transaction-logs
clear transaction-logs
show statistics transaction-logs
transaction-log force
transaction-logs
```

show tvout

To display TV-out information, use the **show tvout** EXEC command.

show tvout

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 software and later releases, the output of the show tvout EXEC command also 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 excerpt of the sample output from the show tvout command, this particular information is highlighted in bold:
-------------------------	--

```
Content Engine # show tvout
.
.
.
TV-out model: ce565-002 (sigma)
  ***Hardware revision level not supported in this version of software***

TV-out service is not enabled
TV-out signal: ntsc

TV-out service is not running
.
.
.
```

Examples	The following example shows the output of the show tvout EXEC command:
-----------------	---

```
ContentEngine# show tvout
TV-out model:ce560-AV-001

TV-out service is enabled
TV-out signal:ntsc
```

Related Commands	show running-config show statistics tvout tvout
-------------------------	--

show url-filter

To display URL filter configuration information, use the **show url-filter** EXEC command.

show url-filter {http | rtsp | wmt}

Syntax Description	http	Displays URL filter configurations for HTTP.
	rtsp	Displays URL filter configurations for RTSP.
	wmt	Displays URL filter configurations for WMT.

Defaults URL filtering is disabled by default.

Command Modes EXEC

Examples [Table 2-164](#) describes the fields shown in the **show url-filter http** display.

Table 2-164 *show url-filter http Field Descriptions*

Field	Description
URL filtering is DISABLED	Status of whether URL filtering is disabled or enabled to use N2H2 external server, Websense server, SmartFilter plug-in, or local list files.
Good-list file name	Status of the path to the file of the HTTP good sites' list. This file contains URLs to which access is allowed.
Bad-list file name	Status of the path to the file of the HTTP good sites' list. This file contains URLs to which access is denied.
Custom message directory	Pathname for the remote file that contains the custom message directory. This file creates a customized URL blocking message to display to the client. This custom message must be an administrator-created HTML file named block.html.
Websense server IP	Hostname or IP address of the primary or secondary Websense servers.
Websense server port	Port number from which the primary or secondary Websense server is accepting requests.
Websense server timeout	Number of seconds that the Content Engine should wait for a response from the primary or secondary Websense server.
Websense server connections	Number of persistent connections to the primary or secondary Websense server to fine tune the performance of Websense.
Websense allow mode is ENABLED	Status of whether the Content Engine can allow the request to be served if there is no response from Websense server.
N2H2 server IP	Hostname or IP address of N2H2 servers.
N2H2 server port	Port number from which N2H2 server is accepting requests.

Table 2-164 *show url-filter http Field Descriptions (continued)*

Field	Description
N2H2 server timeout	Number of seconds that the Content Engine should wait for a response from the N2H2 server.
N2H2 allow mode is ENABLED	Status of whether the Content Engine will time out all HTTP requests (HTTP, FTP-over-HTTP, or HTTPS-over-HTTP requests) and either block or allow all HTTP traffic when it fails to receive a response from the N2H2 server.

Related Commands

clear statistics url-filter
debug url-filter
show statistics url-filter
url-filter
url-filter local-list-reload

show user

To display user identification number and username information for a particular user, use the **show user** EXEC command.

show user { **uid** *number* | **username** *name* }

Syntax Description

uid	Displays the user's identification number.
<i>number</i>	Identification number (0–65535).
username	Displays the name of user.
<i>name</i>	Name of the user.

Defaults

No default behavior or values

Command Modes

EXEC

Examples

[Table 2-165](#) describes the fields shown in the **show user** display.

Table 2-165 *show user Field Descriptions*

Field	Description
Uid	User ID number.
Username	Username.
Password	Login password. This field does not display the actual password.
Privilege	Privilege level of the user.
Configured in	Database in which the login authentication is configured.

Related Commands

clear users
show users
user

show users

To display users, use the **show users** EXEC command.

show users {administrative | request-authenticated}

Syntax Description	administrative	Lists users with administrative privileges.
	request-authenticated	Lists users authenticated by an HTTP request.

Defaults	No default behavior or values
-----------------	-------------------------------

Command Modes	EXEC
----------------------	------

Examples The following example displays the list of users with administrative privileges:

```
ContentEngine# show users administrative
      UID USERNAME
      0 admin
```

The following example displays the list of users authenticated by HTTP requests:

```
ContentEngine# show users request-authenticated
      USERNAME  MODE
```

The following example shows the output if no users are authenticated by HTTP request:

```
ContentEngine# show users request-authenticated
There are no users authenticated by HTTP request
```

Related Commands	clear users
	show user
	user

show version

To display version information about the Content Engine software, use the **show version** EXEC command.

show version

Syntax Description This command has no arguments or keywords.

Defaults No default behavior or values

Command Modes EXEC

Examples [Table 2-166](#) describes the fields shown in the **show version** display.

Table 2-166 *show version Field Descriptions*

Field	Description
Cisco Application and Content Networking 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	Complete 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.

show wccp

To display WCCP information, use the **show wccp EXEC** command.

show wccp content-engines

show wccp flows { **custom-web-cache** | **dns** | **ftp-native** | **https-cache** | **reverse-proxy** | **rtsp** | **service-number** *service_num* | **web-cache** | **wmt** | **wmt-rtspu** } [**summary**]

show wccp gre

show wccp masks { **custom-web-cache** | **dns** | **ftp-native** | **https-cache** | **reverse-proxy** | **rtsp** | **service-number** *service_num* | **web-cache** | **wmt** | **wmt-rtspu** }

show wccp modules

show wccp port-list

show wccp routers

show wccp services [**detail**]

show wccp slowstart { **custom-web-cache** | **dns** | **ftp-native** | **https-cache** | **reverse-proxy** | **rtsp** | **service-number** *service_num* | **web-cache** | **wmt** | **wmt-rtspu** }

show wccp status

Syntax	Description
content-engines	Displays which Content Engines are seen by which routers.
flows	Displays the WCCP packet flow count by bucket.
custom-web-cache	Displays custom web caching service packet flows.
dns	Displays the state of DNS caching services.
ftp-native	Displays the state of native FTP caching services.
https-cache	Displays the state of HTTPS caching services.
reverse-proxy	Displays reverse proxy web caching service packet flows.
rtsp	Displays RTSP caching service packet flows.
service-number	Displays the WCCP service number.
<i>service_num</i>	Service number (90–97).
web-cache	Displays standard web caching service packet flows.
wmt	Displays WMT caching service packet flows.
wmt-rtspu	Displays WMT RTSPU caching service packet flows.
summary	(Optional) Displays summary information.
gre	Displays WCCP generic routing encapsulation packet-related information.
masks	Displays WCCP mask assignments for a given service.
modules	Displays the running status of WCCP registered modules.
port-list	Displays the running status of WCCP port lists.
routers	Displays routers seen and not seen by this Content Engine.
services	Displays WCCP services configured.

detail	(Optional) Displays the detail of services.
slowstart	Displays the WCCP slow-start state for the selected service.
status	Displays the version of WCCP that is enabled and running.

Defaults

No default behavior or values

Command Modes

EXEC

Usage Guidelines

Use the **show wccp services** command to list all the services that are configured on the Content Engine.

Use the **show wccp services detail** command to display the details for a particular WCCP service.

Examples

[Table 2-167](#) describes the fields shown in the **show wccp gre** command.

Table 2-167 *show statistics wccp gre Field Descriptions*

Field	Description
Transparent GRE packets received	Total number of GRE packets received by the Content Engine, regardless of whether they have been intercepted by WCCP or not. GRE is a Layer 3 technique that allows packets to reach the Content Engine even if there are any number of routers in the path to the Content Engine.
Transparent non-GRE packets received	Number of non-GRE packets received by the Content Engine either using the traffic interception and redirection functions of WCCP in the router hardware at Layer 2 or Layer 4 switching (a Content Services Switch [CSS]) that redirects requests transparently to the Content Engine.
Transparent non-GRE packets passed through	Number of non-GRE packets transparently intercepted by a Layer 4 switch and redirected to the Content Engine.
Total packets accepted	Total number of packets that are transparently intercepted and redirected to the Content Engine to serve client requests for content.
Invalid packets received	Number of packets that are dropped either because the redirected packet is a GRE packet and the WCCP GRE header has invalid data or the IP header of the redirected packet is invalid.
Packets received with invalid service	Number of WCCP version 2 GRE redirected packets that contain an invalid WCCP service number.
Packets received on a disabled service	Number of WCCP version 2 GRE redirected packets that specify the WCCP service number for a service that is not enabled on the Content Engine. For example, a HTTPS request redirected to the Content Engine when the HTTPS-caching service (service 70) is not enabled.

Table 2-167 *show statistics wccp gre Field Descriptions (continued)*

Field	Description
Packets received too small	Number of GRE packets redirected to the Content Engine that do not contain the minimum amount of data required for a WCCP GRE header.
Packets dropped due to zero TTL	Number of GRE packets that are dropped by the Content Engine because the redirected packet's IP header has a zero TTL.
Packets dropped due to bad buckets	<p>Number of packets that are dropped by the Content Engine because the WCCP flow redirection could not be performed due to a bad mask or hash bucket determination.</p> <p>Note A bucket is defined as a certain subsection of the allotted hash assigned to each Content Engine in a Content Engine cluster. If only one Content Engine exists in this environment, it has 256 buckets assigned to it.</p>
Packets dropped due to no redirect address	Number of packets that are dropped because the flow redirection destination IP address could not be determined.
Packets dropped due to loopback redirect	Number of packets that are dropped by the Content Engine when the destination IP address is the same as the loopback address.
Connections bypassed due to load	Number of connection flows that are bypassed when the Content Engine is overloaded. When the overload bypass option is enabled, the Content Engine bypasses a bucket and reroutes the overload traffic. If the load remains too high, another bucket is bypassed, and so on, until the Content Engine can handle the load.
Packets sent back to router	Number of requests that are passed back by the Content Engine to the WCCP-enabled router from which the request was received. The router then sends the flow toward the origin web server directly from the web browser, which bypasses the Content Engine.
Packets sent to another CE	Number of packets that are redirected to another Content Engine in the WCCP service group. Service groups consist of up to 32 Content Engines and 32 WCCP-enabled routers. In both packet-forwarding methods, the hash parameters specify how redirected traffic should be load balanced among the Content Engines in the various WCCP service groups.
GRE fragments redirected	Number of GRE packets received by the Content Engine that are fragmented.
Packets failed GRE encapsulation	Number of GRE packets that are dropped by the Content Engine because they could not be redirected due to problems while encapsulating the packet with a GRE header.
Packets dropped due to invalid fwd method	Number of GRE packets that are dropped by the Content Engine because it was redirected using GRE but the WCCP service was configured for Layer 2 redirection.
Packets dropped due to insufficient memory	Number of GRE packets that are dropped by the Content Engine due to the failure to allocate additional memory resources required to handle the GRE packet.

Table 2-167 *show statistics wccp gre Field Descriptions (continued)*

Field	Description
Packets bypassed, no conn at all	Number of packets that failed to be associated with an existing flow. WCCP can also handle asymmetric packet flows and always maintains a consistent mapping of web servers to caches regardless of the number of switches or routers used in a WCCP service group (up to 32 routers or switches communicating with up to 32 Content Engines in a cluster).
Packets bypassed, no pending connection	Number of packets that failed to be associated with a pending connection.
Packets due to clean wccp shutdown	Number of connection flows that are bypassed due to a clean WCCP shutdown. During a proper shutdown of WCCP, the Content Engine continues to service the flows it is handling but starts to bypass new flows. When the number of flows goes down to zero, the Content Engine takes itself out of the cluster by having its buckets reassigned to other Content Engines by the lead Content Engine.
Packets bypassed due to bypass-list lookup	Number of connection flows that are bypassed due to a bypass list entry. When the Content Engine receives an error response from an origin server, it adds an entry for the server to its bypass list. When it receives subsequent requests for the content residing on the bypassed server, it redirects packets to the bypass gateway. If no bypass gateway is configured, then the packets are returned to the redirecting Layer 4 switch.
Packets received with client IP addresses	Number of packets that are associated to a connection flow that is being spoofed. By spoofing a client's IP address, the Content Engine can receive packets with the client IP (which is different from the Content Engine's own IP address) and send the packet to the correct application that is waiting for the packet.
Conditionally Accepted connections	Number of connection flows that are accepted by the Content Engine due to the conditional accept feature.
Conditionally Bypassed connections	Number of connection flows that are bypassed by the Content Engine due to the conditional accept feature.
L2 Bypass packets destined for loopback	Number of packets that are dropped by the Content Engine due to the destination IP address being the loopback address when the WCCP-enabled router or switch tries to perform Layer 2 redirection.
L2 Packets fragmented for bypass	Number of GRE packets that do not contain enough data to hold an IP header.
Packets dropped due to IP access-list deny	Number of packets that are dropped by the Content Engine when an IP access list that the Content Engine applies to WCCP GRE encapsulated packets denies access to WCCP applications (the wccp access-list command).
Packets w/WCCP GRE received too small	Number of packets transparently intercepted by the WCCP-enabled router at Layer 2 and sent to the Content Engine that need to be fragmented for the packets to be redirected using GRE.

Table 2-168 describes the fields shown in the **show wccp modules** display.

Table 2-168 *show wccp modules Field Descriptions*

Field	Description
Modules registered with WCCP on this Content Engine	
Module	Number used by WCCP to identify the module.
Socket	Socket used by the module to communicate with WCCP.
Expire(sec)	Number of seconds after which the module is assumed to be inactive if it does not respond to keepalive messages.
Name	Names of the WCCP registered modules.
Supported Services	Services supported by a module.

Table 2-169 describes the fields in the **show wccp routers** display.

Table 2-169 *show wccp routers Field Descriptions*

Field	Description
Router Information for Service:	Name of WCCP service.
Routers Configured and Seeing this Content Engine	Number of routers configured and that are seeing this CE.
Router Id	Address obtained from the I_SEE_YOU message sent by the router. This address is used to identify the router to which this Content Engine is connected.
Sent To	IP address to which the Content Engine sends the HERE_I_AM message.
Recv ID	Number that is used to synchronize the Content Engine with the router.
Routers not Seeing this Content Engine	IP addresses of all routers that are not seeing this Content Engine.
Routers Notified of but not Configured	Addresses obtained from the I_SEE_YOU message sent by the router when the router is not configured in the router-list.
Multicast Addresses Configured	Multicast addresses, if configured in the router-list. If no multicast addresses are configured, the display will show NONE.

Related Commands

wccp

show websense-server

To display URL filtering statistics for the local Websense server, use the **show websense-server** EXEC command.

show websense-server

Syntax Description This command has no arguments or keywords.

Defaults No default behavior or values

Command Modes EXEC

Examples [Table 2-169](#) describes the fields in the **show websense-server** display.

Table 2-170 *show websense-server Field Descriptions*

Field	Description
Local Websense Server Information	
Websense Enterprise Version	Websense version.
Websense Enterprise Build	Package version.
Websense server components installed	Lists components installed on the Content Engine, such as the following: Policy Server EIM Server User Servicess Network Agent Radius Agent eDirectory Agent
Status of the components	Provides status information about the components. For example, Websense Filtering Service running... Websense Policy Server running... Websense User Service running....
Sending SERVER_STATUS_REQUEST...	
Status Code = 0	
License Count = 0	
Elapsed Time = 1 ms	
AVG TIME PER REQUEST = 1 ms	

Table 2-170 *show websense-server Field Descriptions (continued)*

Field	Description
Websense Server Port	TCP port that receives requests for content filtering according to the Websense protocol.
Websense Block Message Port	Port that the Websense process listens on to receive blocked pages. When Websense process blocks a URL, it sends a redirect URL to the user. The redirect URL is configured to print out the blocked page and the policy for the user.
Websense Config Server Port	Port for the Websense Policy Server to which the Websense GUI Manager connects.
Websense Diagnostics Server Port : 15869	Websense server port to which the diagnostics utilities connect. The Websense server has an exhaustive set of diagnostics that users can run remotely to diagnose problems in the Websense process.
Radius Agent Configuration...	
Outgoing Requests:	
Radius Server	IP address of the RADIUS Server that the Radius-Agent contacts and to which the Radius-Agent forwards authentication requests.
Authentication Port	Port from which authentication requests are sent to the RADIUS server.
Accounting Port	Port from which accounting requests are sent to the RADIUS server.
Incoming Requests:	
Authentication Port	Port from which authentication requests are received from the RADIUS client.
Accounting Port	Port from which accounting requests are received from the RADIUS client.
eDirectory Agent Configuration...	
Administrative DN	Administrative Distinguished name.
Administrative Password	Administrative password.
Root Context	Directory service root location.
Servers IP	eDirectory Server IP addresses that are configured. The maximum number of IP addresses and ports that can be configured is 8.
Port Number	Port number.

Related Commands

```

debug url-filter websense
show statistics url-filter http websense
show url-filter http
url-filter http websense
websense-server

```

show wmt

To display Windows Media Technologies (WMT) bandwidth, broadcast, multicast, and proxy mode configuration and license information, use the **show wmt** EXEC command. You can access the WMT diagnostic tools using the **show wmt diagnostics** EXEC command. If the **license-agreement** option is included in the command string, the full text of the WMT license agreement is displayed.

show wmt [**bandwidth** [**incoming** **bypass-list**] | **broadcast** | **detail** | **diagnostics** {**header-info** {**stream-file** *word* | **nsc-file** *.nsc-filename*} | **network-trace** *word*} **http** **allow** **extension** | **license-agreement** | **multicast** | **proxy**]

Syntax Description

bandwidth	(Optional) Displays WMT bandwidth settings.
incoming	(Optional) Displays WMT incoming bandwidth settings.
bypass-list	Displays the WMT incoming bandwidth bypass list.
broadcast	(Optional) Displays the WMT broadcast configuration.
detail	(Optional) Displays the detailed WMT configuration.
Note In the ACNS 5.3 software, the output of the show wmt and show wmt detail commands is identical.	
diagnostics	(Optional) Displays a set of WMT diagnostics tools.
header-info	Displays the file header information.
stream-file	Displays the headers of a Windows Media file.
<i>word</i>	An .asf, .wma, .wmv URL, or local file.
nsc-file	Displays the .nsc file headers.
<i>.nsc-filename</i>	Name of a local or remote WMT multicast station.
network-trace	Displays WMT diagnostics information.
<i>word</i>	Name of a local tcpdump file.
http	(Optional) Displays HTTP configurations.
allow	Displays the HTTP filename extensions allowed to be served using WMT.
extension	Displays the list of HTTP filename extensions to be served using WMT.
license-agreement	(Optional) Displays the WMT end user license agreement.
multicast	(Optional) Displays the WMT multicast configuration.
proxy	(Optional) Displays the WMT proxy mode configuration.

Defaults

No default behavior or values

Command Modes

EXEC

Usage Guidelines

In the ACNS 5.3 software and later releases, you can access the following three WMT diagnostic tools through the Content Engine CLI:

- **asfhead**—Examine the headers of a Windows Media file (for example, an .asf, .wmv, or .wma file). To access the asfhead tool, enter the **show wmt diagnostics header-info stream-file word** EXEC command.
- **nschead**—Examine the .nsc file headers. To access the nschead tool, enter the **show wmt diagnostics header-info nsc-file .nsc-filename** EXEC command.
- **mmsdig**—Use this text-based tool to decode the MMS protocol (a binary protocol) that is captured in tcpdump traces (or any standard network trace output). To access this tool, enter the **show wmt diagnostics network trace word** EXEC command.

The mmsdig tool does not currently support decoding for the RTSP, RTP, and RTCP protocols. In the ACNS 5.3 software, RTSP support for Windows Media 9 clients and servers was added.

Examples

The following example shows sample output of the **show wmt diagnostics header-info stream-file** EXEC command. In this example, this command is used to display the headers of a .wmv file named 256.wmv. Annotations about this sample command output are highlighted in boldface text and enclosed within parentheses.

```
ContentEngine# show wmt diagnostics header-info stream-file 256.wmv
Description object:
VBR info object  (2 streams)  (VBR: variable bit rate file)
streamid 1 bitrate 34319 (0000860f)  (in this file there are 2 bit rates)
streamid 2 bitrate 195798 (0002fcd6)
Global properties object:
    GUID = f9 dd df 7e 6c 80 85 46 83 3c c1 23 4b 2f 4f 6b  (object ID)
    File Size = 8104309  (object size)
    Total Packets = 2816  (total number of data packets)
    Time_mkin = 0x0000000001c0debc
    Time_mkout = 0x00000000a9e1af60
    Send duration = 2817510000  (Specifies the time needed to send the file
                                in 100-nanosecond units)
    Broadcast Flag = 0
    Seekable Flag = 1
    Minimum Packet Size = 2877  (Specifies the minimum data packet size in bytes)
    Max Packet Size = 2877  (Specifies the maximum data packet size in bytes)
    Max Bitrate = 230117  (Specifies the maximum instantaneous bit rate in
                           bits per second for the entire file)
Unknown object:
uuid: b5 3 bf 5f 2e a9 cf 11 8e e3 0 c0 c 20 53 65 len: 0x16
00000000: 11 d2 d3 ab ba a9 cf 11 8e e6 00 c0 0c 20 53 65
00000010: 06 00 00 00 00 00 00
Extended Content Description Object (two entries):
WMFSDKVersion (unicode) 7.00.00.1956
WMFSDKNeeded (unicode) 0.0.0.0000
Codec description object (2 streams)  (audio and video codec information)
    Type = 0x2
    Description = Windows Media Audio V7
                  32 kbps, 32 kHz, stereo
    ID = 61 01
    Type = 0x1
    Description = Windows Media Video V7
    ID = 57 4d 56 31
Index:
    Index #0
    Interval = 10000000
    Max_pkt# = 9
```

```

Entry = 286
Stream ID = 0

```

The following example shows an excerpt of sample output from the **show wmt diagnostics header-info nsc-file EXEC** command. In this example, this command is used to display the headers of the .nsc file named testmcast.nsc. Annotations about this sample command output are highlighted in boldface text and enclosed within parentheses.

```

ContentEngine# show wmt diagnostics header-info nsc-file testmcast.nsc
Press Ctrl-C to abort, if no information is shown within 30 secs.

```

```

[Address]
Time To Live=0x00000005 (ttl)
IP Address=233.33.33.33 (Multicast address is configured.)
IP Port=0x00000D10 (Multicast port is configured.)
Delivery Mode=0x00000002
NSC Format Version=3.0 (version of NSC format)
[Description]
Description=
Auto Archive=0x00000000
Format1 id 4d2
Description1=http://128.107.192.4:8080 (Source of the media stream,
for example, the WMT server/encoder)

```

header information:

```

output_head() starting
Description object:
clip:          WMT Live testing (title of the stream, configured on the
WMT server/encoder)
author:        John Doe (author of the stream, configured on the WMT server/encoder)
VBR info object (2 streams)
streamid 1 bitrate 34995 (000088b3)
streamid 2 bitrate 253711 (0003df0f)
Global properties object:
    GUID = 33 36 59 e6 6a 7d 4e 49 99 5e cf 71 39 b9 61 54 (guid:
unique identifier of this stream)
    File Size = 2810 (size in bytes)
    Total Packets = 4294967295
    Time_mkin = 0x00000000001c4de45
    Time_mkout = 0x0000000000000000
    Send duration = 0
    Broadcast Flag = 1
    Seekable Flag = 0
    Minimum Packet Size = 1444
    Max Packet Size = 1444
    Max Bitrate = 288706
Unknown object:
uuid: b5 3 bf 5f 2e a9 cf 11 8e e3 0 c0 c 20 53 65 len: 0x707
00000000: 11 d2 d3 ab ba a9 cf 11 8e e6 00 c0 0c 20 53 65
00000010: 06 00 f1 06 00 00 a9 46 43 7c e0 ef fc 4b b2 29
00000020: 39 3e de 41 5c 85 27 00 00 00 00 00 00 00 01 00
00000030: 0c 65 00 6e 00 2d 00 75 00 73 00 00 00 cb a5 e6
00000040: 14 72 c6 32 43 83 99 a9 69 52 06 5b 5a 58 00 00
00000050: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
00000060: 00 00 00 00 00 00 7d 00 00 88 13 00 00 00 00 00
00000070: 00 00 7d 00 00 88 13 00 00 00 00 00 00 00 03 00
00000080: 00 00 00 00 00 01 00 00 00 00 00 00 00 00 00 00
00000090: 00 00 00 00 00 cb a5 e6 14 72 c6 32 43 83 99 a9
000000a0: 69 52 06 5b 5a 6e 00 00 00 00 00 00 00 00 00 00
000000b0: 00 00 00 00 00 00 00 00 00 00 00 00 00 68 ad 03
000000c0: 00 88 13 00 00 00 00 00 00 68 ad 03 00 88 13 00
000000d0: 00 00 00 00 00 00 00 00 00 00 00 00 00 02 00 00

```

```

000000e0: 00 63 17 05 00 00 00 00 00 00 01 00 50 94 bd
000000f0: c6 7f 86 07 49 83 a3 c7 79 21 b7 33 ad 02 00 00
00000100: 00 00 00 5d 8b f1 26 84 45 ec 47 9f 5f 0e 65 1f
00000120: c5 af 5b 77 48 84 67 aa 8c 44 fa 4c ca e0 00 00
00000130: 00 00 00 00 00 04 00 00 00 01 00 0c 00 02 00 02
00000140: 00 00 00 49 00 73 00 56 00 42 00 52 00 00 00 00
00000150: 00 00 00 01 00 34 00 00 00 06 00 00 00 44 00 65
00000160: 00 76 00 69 00 63 00 65 00 43 00 6f 00 6e 00 66
00000170: 00 6f 00 72 00 6d 00 61 00 6e 00 63 00 65 00 54
00000180: 00 65 00 6d 00 70 00 6c 00 61 00 74 00 65 00 00
00000190: 00 4c 00 32 00 00 00 00 00 02 00 0c 00 02 00 02
000001a0: 00 00 00 49 00 73 00 56 00 42 00 52 00 00 00 00
000001b0: 00 00 00 02 00 34 00 00 00 0c 00 00 00 44 00 65
000001c0: 00 76 00 69 00 63 00 65 00 43 00 6f 00 6e 00 66
000001d0: 00 6f 00 72 00 6d 00 61 00 6e 00 63 00 65 00 54
000001e0: 00 65 00 6d 00 70 00 6c 00 61 00 74 00 65 00 00
000001f0: 00 4d 00 50 00 40 00 4c 00 4c 00 00 00 74 d4 06
00000200: 18 df ca 09 45 a4 ba 9a ab cb 96 aa e8 0a 05 00
00000210: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
.
.
.
000006e0: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
000006f0: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
00000700: 00 00 00 00 00 00 00
Extended Content Description Object (3 entries):
WMFSDKVersion (unicode) 9.00.00.2980
WMFSDKNeeded (unicode) 0.0.0.0000
IsVBR (bool) 0
Codec description object (2 streams)
    Type = 0x2
    Description = Windows Media Audio 9
                  32 kbps, 32 kHz, stereo (A/V) 1-pass CBR
    ID = 61 01
    Type = 0x1
    Description = Windows Media Video 9
                  ID = 57 4d 56 33
Multicast object (len 0x18):
00000000: 50 0f 1d 54 4b 5b cf 11 a8 fd 00 80 5f 5c 44 2b
00000010: 04 00 00 00 0a 00 00 00
ContentEngine#

```

Some of the fields are common between the command output from the **show wmt diagnostics header-info stream-file** and **show wmt diagnostics header-info nsc-file** EXEC commands.

The following example shows the the WMT server configurations, the WMT HTTP configurations, and the WMT proxy configurations for the Content Engine. In the ACNS 5.3 software and later releases, the output of the **show wmt** and **show wmt detail** commands is identical.

```

ContentEngine# show wmt

----- WMT Server Configurations -----
WMT golden license key installed
WMT outgoing bandwidth limit enforced: 250000 Kbits/sec
WMT end user license agreement accepted
WMT is enabled
WMT disallowed client protocols: none
WMT outgoing bandwidth configured is 250000 Kbits/sec
WMT incoming bandwidth configured is 250000 Kbits/sec
WMT max sessions configured: 3568
WMT max sessions platform limit: 3568
WMT max sessions enforced: 3568 sessions
WMT max outgoing bit rate allowed per stream has no limit

```



```

WMT max incoming bit rate allowed per stream has no limit
WMT cache is enabled
WMT cache max-obj-size: 1024 MB
WMT debug level: 0
WMT L4 switch is not enabled
WMT debug client ip not set
WMT debug server ip not set
WMT/REAL cache space partition: wmt 70%, real 30%
WMT Stripping ? from Live URL is not enabled
WMT accelerate live-split is enabled
WMT accelerate proxy-cache is enabled
WMT accelerate VOD is enabled
WMT fast-start is enabled
WMT fast-start max. bandwidth per player is 3500 (Kbps)
WMT fast-cache is enabled
WMT fast-cache acceleration factor is 5
WMT maximum data packet MTU (TCP) enforced is 1472 bytes
WMT maximum data packet MTU (UDP) is 1500 bytes
WMT client idle timeout is 60 seconds
WMT forward logs is enabled
WMT server inactivity-timeout is 65535
WMT Transaction Log format is Windows Media Services 9.0 logging and CE specific
information
RTSP Gateway incoming port 554
RTSP Gateway L4-switch not enabled
RTSP Gateway Transparent Interception (WCCP):
    Configured on port: 554

```

----- WMT HTTP Configurations -----

```

WMT http extensions allowed:
asf none nsc wma wmv nsclog

```

----- WMT Proxy Configurations -----

```

Outgoing Proxy-Mode:
-----
MMS-over-HTTP Proxy-Mode:
    is not configured.
RTSP Proxy-Mode:
    is not configured.
ContentEngine#

```

The following example displays the current WMT proxy configuration for a Content Engine. In the ACNS 5.3 software, the command output was modified to include configuration information about the new WMT RTSP proxy server that can now be enabled on a Content Engine that is acting as a Windows Media 9 server.

```
ContentEngine# show wmt proxy
```

```

Outgoing Proxy-Mode:
-----
MMS-over-HTTP Proxy-Mode:
    is not configured.
RTSP Proxy-Mode:
    is not configured.

```

The following example displays the WMT bandwidth settings configured on a Content Engine:

```

CE-590# show wmt bandwidth
Outgoing bandwidth limit enforced 168000 kbps
Outgoing bandwidth configured 168000 kbps
Incoming bandwidth configured 50000 kbps

```

The following example shows an excerpt of sample output from the **show wmt license-agreement** command:

```
CE-590# show wmt license-agreement
```

```
END USER LICENSE
```

```
PLEASE READ THE LICENSE AGREEMENT AND ACCEPT BY ENTERING THE FOLLOWING  
CLI CONFIGURATION COMMAND: wmt accept-license-agreement
```

```
END USER LICENSE AND SOFTWARE WARRANTY  
Software License
```

```
PLEASE READ THIS SOFTWARE LICENSE CAREFULLY BEFORE DOWNLOADING,  
INSTALLING OR USING CISCO OR CISCO-SUPPLIED SOFTWARE.
```

```
BY DOWNLOADING OR INSTALLING THE SOFTWARE, OR USING THE EQUIPMENT THAT  
CONTAINS THIS SOFTWARE, YOU ARE CONSENTING TO BE BOUND BY THIS LICENSE.  
IF YOU DO NOT AGREE TO ALL OF THE TERMS OF THIS LICENSE, THEN (A) DO NOT  
DOWNLOAD, INSTALL OR USE THE SOFTWARE, AND (B) YOU MAY RETURN THE  
SOFTWARE FOR A FULL REFUND, OR, IF THE SOFTWARE IS SUPPLIED AS PART OF  
ANOTHER PRODUCT, YOU MAY RETURN THE ENTIRE PRODUCT FOR A FULL REFUND.  
YOUR RIGHT TO RETURN AND REFUND EXPIRES 30 DAYS AFTER PURCHASE FROM  
CISCO OR AN AUTHORIZED CISCO RESELLER, AND APPLIES ONLY IF YOU ARE THE  
ORIGINAL PURCHASER.
```

```
The following terms govern your use of the Software except to the extent  
a particular program (a) is the subject of a separate written agreement  
with Cisco or (b) includes a separate click-on license agreement as part  
of the installation and/or download process. To the extent of a conflict  
between the provisions of the foregoing documents, the order of  
precedence shall be (1) the written agreement, (2) the click-on  
agreement, and (3) this Software License.
```

```
.  
.
.
```

Related Commands

```
clear statistics wmt  
show statistics wmt  
wmt
```

shutdown (interface configuration)

To shut down a specific hardware interface, use the **shutdown** interface configuration command. To restore an interface to operation, use the **no** form of this command.

shutdown

no shutdown

Syntax Description	This command has no arguments or keywords.
---------------------------	--

Defaults	No default behavior or values
-----------------	-------------------------------

Command Modes	interface configuration
----------------------	-------------------------

Usage Guidelines	See the “interface” section for alternative syntax.
-------------------------	---

Examples	The following example shows how to shut down an interface configured on a Content Engine:
-----------------	---



```
ContentEngine(config-if)# shutdown
```

Related Commands	interface show interface show running-config show startup-config
-------------------------	---

shutdown (EXEC)

To shut down the Content Engine, Content Router, Content Distribution Manager, or Cisco IP/TV Program Manager, use the **shutdown** EXEC command.

shutdown [poweroff]

Syntax Description	poweroff (Optional) Turns off the power after closing all applications and the operating system.
Defaults	No default behavior or values
Command Modes	EXEC
Usage Guidelines	<p>A controlled shutdown refers to the process of properly shutting down a Content Engine without turning off the power on the device. With a controlled shutdown, all of the application activities and the operating system are properly stopped on a Content Engine but the power is still on. Controlled shutdowns of a Content Engine can help you minimize the downtime when the Content Engine is being serviced.</p> <p>The shutdown EXEC command enables you to shut down and optionally power off a Content Engine:</p> <ul style="list-style-type: none">• Shutdown means that all application activities (applications and operating system) are stopped, but the power is still on. This shutdown is similar to the Linux halt command.• Shutdown poweroff means that the Content Engine is powered down by the ACNS software after being shut down. This operation is also referred to as a software poweroff. The implementation of the shutdown poweroff feature uses the Advanced Configuration and Power Interface (ACPI) power management interface. <div><div>Caution</div><div>If you do not perform a controlled shutdown, the Content Engine file system can be corrupted. It also takes longer to reboot the Content Engine if the Content Engine is not properly shut down.</div></div> <div><div>Note</div><div>You cannot power on Content Engines again through software after a software poweroff operation. You must press the power button once on these Content Engines to bring these Content Engines back online.</div></div>

The **shutdown** EXEC command facilitates a proper shutdown for Content Engines, Content Routers, Content Distribution Managers, and Cisco IP/TV Program Managers. Where the **shutdown** command is supported on all content networking hardware models, the **shutdown poweroff** command is supported only on those models that support ACPI. The following content networking hardware models support the **shutdown poweroff** command:

- CE-510
- CE-510A
- CE-511
- CE-511A
- CE-565
- CE-565A
- CE-566
- CE-566A
- CE-7305
- CE-7305A
- CE-7320
- CE-7325
- CE-7326

The **shutdown** command closes all applications and stops all system activities but keeps the power on. The fans continue to run and the power LED is on, indicating that the device is still powered on. When you enter the **shutdown** command, you are prompted to save your configuration changes, if any, and the Content Engine waits for the maximum amount of time before shutdown if WCCP has been enabled on the Content Engine. The device console displays a menu after the shutdown process is completed. You need to log in to the Content Engine using console to display the following menu:

```
CONTENTENGINE# shutdown
System configuration has been modified. Save?[yes]:yes
Device can not be powered on again through software after shutdown.
Proceed with shutdown?[confirm]yes
Proceed with clean WCCP shutdown?[confirm]yes
Waiting (1 seconds) for WCCP shutdown. Press ^C to skip shutdownn
WCCP clean shutdown wait time exceeded
Shutting down all services, will timeout in 15 minutes.
shutdown in progress ..Halt requested by CLI@ttyS0.
.....
Shutdown success
```

Cisco Content Engine Console

```
Username: admin
Password:
```

```
===== SHUTDOWN SHELL =====
System has been shut down.
```

```
You can either
Power down system by pressing and holding power button
or
1. Reload system through software
2. Power down system through software
Please select [1-2]:
```

The **shutdown poweroff** command closes all applications and the operating system, stops all system activities, and turns off the power. The fans stop running and the power LED starts flashing, indicating that the device has been powered off.

**Note**

If you use the **shutdown** or **shutdown poweroff** commands, the device does not perform a file system check when you power on and boot the device the next time.

Table 2-171 describes the shutdown-only operation and the shutdown power-off operation for Content Engines.

Table 2-171 Shutting Down Content Engines Through CLI Commands

Activity	All Content Engine Models Supported in the ACNS 5.2 Software and Later Releases	Content Engines with Power Management Capability in the ACNS 5.2 Software and Later Releases
User performs a shutdown operation on the Content Engine	Shutdown only ContentEngine# shutdown	Shutdown poweroff ContentEngine# shutdown poweroff
User intervention to bring Content Engine back online	To bring a Content Engine that has an on/off switch on the back (for example, the CE-507, CE-507AV, CE-560, CE-560AV, or the CE-590) back online after a shutdown operation, turn the on/off switch twice. To bring a Content Engine that has a power button (instead of an on/off switch on the back) back online after a shutdown operation, first press and hold the power button for several seconds to power off these models, and then press the power button once again.	After a shutdown poweroff, you must press the power button once to bring the Content Engine back online.
File system check	Will not be performed after you turn the power on again and reboot the Content Engine.	Will not be performed after you turn the power on again and reboot the Content Engine.

You can enter the **shutdown** EXEC command from a console session or from a remote session (Telnet or SSH version 1 or SSH version 2) to perform a shutdown on a Content Engine.

To perform a shutdown on a Content Engine, enter the **shutdown** EXEC command as follows:

```
ContentEngine# shutdown
```

When you are asked if you want to save the system configuration, enter **yes** as follows:

```
System configuration has been modified. Save?[yes]:yes
```

When you are asked if you want to proceed with the shutdown, press **Enter** to proceed with the shutdown operation as follows:

```
Device can not be powered on again through software after shutdown.
Proceed with shutdown?[confirm]
```

The following message appears, reporting that all services are being shut down on this Content Engine:

```
Shutting down all services, will timeout in 15 minutes.
shutdown in progress ..System halted.
```

After the system is shut down (the system has halted), an ACNS software shutdown shell displays the current state of the system (for example, “System has been shut down”) on the console. You are asked whether you want to perform a software power off (the Power down system by software option), or if you want to reload the system through the software.

```
===== SHUTDOWN SHELL =====
System has been shut down.
You can either
    Power down system by pressing and holding power button
or
1. Reload system through software
2. Power down system through software
```

To power down the Content Engine, press and hold the power button on the Content Engine, or use one of the following methods to perform a shutdown poweroff:

- From the console command line, enter **2** when prompted as follows:

```
===== SHUTDOWN SHELL =====
System has been shut down.
You can either
    Power down system by pressing and holding power button
or
1. Reload system through software
2. Power down system through software
```

- From the Content Engine CLI, enter the **shutdown poweroff** EXEC command as follows:

```
ContentEngine# shutdown poweroff
```

When you are asked if you want to save the system configuration, enter **yes** as follows:

```
System configuration has been modified. Save?[yes]:yes
```

When you are asked to confirm your decision, press **Enter**.

```
Device can not be powered on again through software after poweroff.
Proceed with poweroff?[confirm]
Shutting down all services, will timeout in 15 minutes.
poweroff in progress ..Power down.
```

Examples

The following example shows that the **shutdown** command is used to close all applications and stop all system activities:

```
ContentEngine1# shutdown
System configuration has been modified. Save?[yes]:yes
Device can not be powered on again through software after shutdown.
Proceed with shutdown?[confirm]
Shutting down all services, will timeout in 15 minutes.
shutdown in progress ..System halted.
```

The following example shows that the **shutdown poweroff** command is used to close all applications, stop all system activities, and then turn off power to the Content Engine:

```
ContentEngine2# shutdown poweroff
System configuration has been modified. Save?[yes]:yes
Device can not be powered on again through software after poweroff.
Proceed with poweroff?[confirm]
Shutting down all services, will timeout in 15 minutes.
poweroff in progress ..Power down.
```


snmp-server access-list

To configure a standard access control list to allow access through an SNMP agent, use the **snmp-server access-list** global configuration command. To remove a standard access control list, use the **no** form of this command.

snmp-server access-list {*std-acl-num* | *std-acl-name*}

no snmp-server access-list {*std-acl-num* | *std-acl-name*}

Syntax Description

<i>std-acl-name</i>	Standard access list name. Alphanumeric identifier of up to 30 characters, beginning with a letter that identifies the ACL to apply to the current interface.
<i>std-acl-num</i>	Standard access list number (1–99). Numeric identifier that identifies the access list to apply to the current interface.

Defaults

No default behavior or values

Command Modes

global configuration

Usage Guidelines

The **snmp-server access-list** *std-acl-num* global configuration command configures an access control list to allow access to an SNMP agent. The *std-acl-num* variable is a number in the range 1 to 99, indicating a standard access control list. SNMP checks against the specified access control list before accepting or dropping incoming packets.

You can enable SNMP applications to be attached to a particular interface (such as management services to the private IP address space) so that the Content Engine can have one interface in the customer's IP address space that serves content, and another interface in a private IP address space that the administrator uses for management purposes. This setting ensures that clients can access the Content Engine only in the public IP address space for serving content and not access it for management purposes. A device attempting to access one of these applications associated with an access control list (ACL) must be on the list of trusted devices to be allowed access.

See the *Cisco ACNS Software Configuration Guide for Locally Managed Deployments*, for a description on how to use standard IP ACLs to control access to the SNMP agent on a Content Engine.

Examples

The following example allows the SNMP agent to check against access control list 12 before accepting or dropping packets:

```
ContentEngine(config)# snmp-server access-list 12
```



Note

You must first create access list 12 using the **ip access-list standard** global configuration command.

Related Commands

ip access-list
show running-configuration

snmp-server community

To enable the SNMP agent and set up the community access string to permit access to the SNMP agent, use the **snmp-server community** global configuration command. Use the **no** form of this command to disable the SNMP agent and to remove the previously configured community string.

snmp-server community *string* [**group** *groupname* | **rw**]

no snmp-server community *string* [**group** *groupname* | **rw**]

Syntax Description

<i>string</i>	Community string that acts like a password and permits access to the SNMP agent. Supports up to a maximum of 64 characters.
group	(Optional) Specifies the group to which the community string belongs.
<i>groupname</i>	Name of the group. Supports up to a maximum of 64 characters.
rw	(Optional) Enables read-write access to this community string.

Defaults

The SNMP agent is disabled and a community string is not configured. When configured, an SNMP community string by default permits read-only access to all objects.

Usage Guidelines

The SNMP community string is used as a password for authentication when accessing the SNMP agent on the Content Engine. To be authenticated, the Community Name field of any SNMP message sent to the Content Engine must match the SNMP community string defined on the Content Engine.

The SNMP agent on the Content Engine is enabled when an SNMP community string is defined on the Content Engine. The maximum number of SNMP communities that can be created is 10.

The **snmp-server community string** global configuration command provides view-based access control for SNMPv1, SNMPv2c, and SNMPv3 but also continues to provide backward compatibility between different versions.



Tip Any SNMP message sent to the Content Engine must have the Community Name field of the message match the community string defined here in order to be authenticated.

In the ACNS 5.x software prior to the ACNS 5.1 software release, the **snmp-server community string** global configuration command did not have an option to create a community string that allows SNMP messages to execute a set operation on a MIB object. A **rw** option has been introduced for this purpose. Also, the previous version of the SNMP agent did not provide selective access control to MIB objects. Access to any MIB object was denied or granted based on authentication of the SNMP community string.

With the introduction of view-based access control, it is now possible to configure a community string that grants access to only part of the MIB subtree. To provide backward compatibility with the previous version of this command, a default read group or default write group (if the **rw** option is specified on the command line) is associated with the community string if no group name is specified. Both of these default groups are hidden from users, are not displayed in the configuration file or in the **show snmp group EXEC** command, and are created during initialization of the SNMP agent.

Command Modes global configuration

Examples The following example enables the SNMP agent and assigns the community string comaccess to SNMP:

```
ContentEngine(config)# snmp-server community comaccess
```

The following example disables the SNMP agent and removes the previously defined community string:

```
ContentEngine(config)# no snmp-server community
```

Related Commands

- show snmp**
- snmp-server contact**
- snmp-server enable traps**
- snmp-server group**
- snmp-server host**
- snmp-server location**
- snmp-server mib**
- snmp-server notify**
- snmp-server user**
- snmp-server view**

snmp-server contact

To set the system server contact (sysContact) string, use the **snmp-server contact** global configuration command. To remove the system contact information, use the **no** form of this command.

snmp-server contact *line*

no snmp-server contact

Syntax Description	contact	Specifies text for MIB-II object sysContact.
	<i>line</i>	Identification of the contact person for this managed node.
Command Modes	global configuration	
Defaults	No system contact string is set.	
Usage Guidelines	The system contact string is the value stored in the MIB-II system group sysContact object.	
Examples	The following example shows how to configure a system contact string:	
	ContentEngine(config)# snmp-server contact Dial System Operator at beeper # 27345	
	The following example resets the system contact string:	
	ContentEngine(config)# no snmp-server contact	
Related Commands	show snmp snmp-server community snmp-server enable traps snmp-server group snmp-server host snmp-server location snmp-server mib snmp-server notify snmp-server user snmp-server view	

snmp-server enable traps

To enable the Content Engine to send SNMP traps, use the **snmp-server enable traps** global configuration command. To disable all SNMP traps or only SNMP authentication traps, use the **no** form of this command.

snmp-server enable traps [**alarm** [**clear-critical** | **clear-major** | **clear-minor** | **raise-critical** | **raise-major** | **raise-minor**] | **config** | **content-engine** [**disk-fail** | **disk-read** | **disk-write** | **overload-bypass** | **transaction-log**] | **entity** | **event** | **snmp** [**authentication** | **cold-start**]]

no snmp-server enable traps [**alarm** [**clear-critical** | **clear-major** | **clear-minor** | **raise-critical** | **raise-major** | **raise-minor**] | **config** | **content-engine** [**disk-fail** | **disk-read** | **disk-write** | **overload-bypass** | **transaction-log**] | **entity** | **event** | **snmp** [**authentication** | **cold-start**]]

Syntax Description

alarm	(Optional) Enables Content Engine alarm traps.
clear-critical	(Optional) Enables the clear-critical alarm trap.
clear-major	(Optional) Enables the clear-major alarm trap.
clear-minor	(Optional) Enables the clear-minor alarm trap.
raise-critical	(Optional) Enables the raise-critical alarm trap.
raise-major	(Optional) Enables the raise-major alarm trap.
raise-minor	(Optional) Enables the raise-minor alarm trap.
config	(Optional) Enables CiscoConfigManEvent traps.
content-engine	(Optional) Enables SNMP Content Engine traps.
disk-fail	(Optional) Enables the disk failure error trap.
disk-read	(Optional) Enables the disk read error trap.
disk-write	(Optional) Enables the disk write error trap.
overload-bypass	(Optional) Enables the WCCP overload bypass error trap.
transaction-log	(Optional) Enables the transaction log write error trap.
entity	(Optional) Enables SNMP entity traps.
event	(Optional) Enables Event MIB traps.
snmp	(Optional) Enables SNMP-specific traps.
authentication	(Optional) Enables the authentication trap.
cold-start	(Optional) Enables the cold-start trap.

Defaults

This command is disabled by default. No traps are enabled.

Command Modes

global configuration

Usage Guidelines

You can configure a Content Engine to generate an SNMP trap for a specific alarm condition. You can configure the generation of SNMP alarm traps on Content Engines based on the following:

- The severity of the alarm (critical, major, or minor)
- The action (the alarm is raised or cleared)

The ACNS 5.3 software and later releases support six generic alarm traps. These six general alarm traps provide SNMP and Node Health Manager integration. Each trap can be enabled or disabled through the Content Engine CLI.



Note

For information on generic alarm traps, see Chapter 21 of the *Cisco ACNS Software Configuration Guide for Locally Managed Deployments*.

SNMP notifications can be sent as traps or inform requests. The **snmp-server enable traps** command enables both traps and inform requests for the specified notification types.

To configure traps, you must enter the **snmp-server enable traps** command. If you do not enter the **snmp-server enable traps** command, no traps are sent.

If you do not enter an **snmp-server enable traps** command, no notifications controlled by this command are sent. In order to configure the Content Engine to send these SNMP notifications, you must enter at least one **snmp-server enable traps** command. If you enter the command with no keywords, all notification types are enabled. If you enter the command with a keyword, only the notification type related to that keyword is enabled. In order to enable multiple types of notifications, you must enter a separate **snmp-server enable traps** command for each notification type and notification option.

The **snmp-server enable traps** command is used with the **snmp-server host** command. Use the **snmp-server host** command to specify which host or hosts receive SNMP traps. To send traps, you must configure at least one host using the **snmp-server host** command.

For a host to receive a trap, you must enable both the **snmp-server enable traps** command and the **snmp-server host** command for that host.

In addition, you must enable SNMP with the **snmp-server community** command.

To disable the sending of the MIB-II SNMP authentication trap, you must enter the **no snmp-server enable traps snmp authentication** command.

Examples

The following example enables the Content Engine to send all traps to the host 172.31.2.160 using the community string public:

```
ContentEngine(config)# snmp-server enable traps
ContentEngine(config)# snmp-server host 172.31.2.160 public
```

The following example disables all traps:

```
ContentEngine(config)# no snmp-server enable traps
```

Related Commands

show snmp
snmp-server access-list
snmp-server community
snmp-server contact
snmp-server group
snmp-server host
snmp-server location

snmp-server mib
snmp-server notify
snmp-server user
snmp-server view

snmp-server group

To define a user security model group, use the **snmp-server group** global configuration command. To remove the specified group, use the **no** form of this command.

```
snmp-server group name { v1 [notify name] [read name] [write name] | v2c [notify name] [read name] [write name] | v3 {auth [notify name] [read name] [write name] | noauth [notify name] [read name] [write name] | priv [notify name] [read name] [write name]}}
```

```
no snmp-server group name { v1 [notify name] [read name] [write name] | v2c [notify name] [read name] [write name] | v3 {auth [notify name] [read name] [write name] | noauth [notify name] [read name] [write name] | priv [notify name] [read name] [write name]}}
```

Syntax Description	
<i>name</i>	Name of the SNMP group. Supports up to a maximum of 64 characters.
v1	Specifies the group using the Version 1 Security Model.
notify	(Optional) Specifies a notify view for the group that enables you to specify a notify, inform, or trap.
<i>name</i>	Notify view name. Supports up to a maximum of 64 characters.
read	(Optional) Specifies a read view for the group that enables you only to view the contents of the agent.
<i>name</i>	Read view name. Supports up to a maximum of 64 characters.
write	(Optional) Specifies a write view for the group that enables you to enter data and configure the contents of the agent.
<i>name</i>	Write view name. Supports up to a maximum of 64 characters.
v2c	Specifies the group using the Version 2c Security Model.
v3	Specifies the group using the User Security Model (SNMPv3).
auth	Specifies the group using the AuthNoPriv Security Level.
noauth	Specifies the group using the noAuthNoPriv Security Level.
priv	Specifies the group using the AuthPriv Security Level.

Defaults The default is that no user security model group is defined.

Command Modes global configuration

Usage Guidelines The maximum number of SNMP groups that can be created is 10.

Select one of three SNMP security model groups: Version 1 (**v1**) Security Model, Version 2c (**v2c**) Security Model, or the User Security Model (**v3** or SNMPv3). Optionally, you then specify a notify, read, or write view for the group for the particular security model chosen. The **v3** option allows you to specify the group using one of three security levels: **auth** (AuthNoPriv Security Level), **noauth** (noAuthNoPriv Security Level), or **priv** (AuthPriv Security Level).

The ACNS 5.x software supports the following versions of SNMP:

- Version 1 (SNMPv1)—This version is the initial implementation of SNMP. See the RFC 1157 for a full description of its functionality.
- Version 2 (SNMPv2c)—This version is the second release of SNMP, described in RFC 1902. It provides additions to data types, counter size, and protocol operations.
- Version 3 (SNMPv3)—This version is the most recent SNMP version, defined in RFC 2271 through RFC 2275.

SNMP Security Models and Security Levels

SNMPv1 and SNMPv2c do not have any security (authentication or privacy) mechanisms to keep SNMP packet traffic on the wire confidential. As a result, packets on the wire can be detected and SNMP community strings can be compromised.

To solve the security shortcomings of SNMPv1 and SNMPv2c, SNMPv3 provides secure access to Content Engines by authenticating and encrypting packets over the network. The SNMP agent in the ACNS 5.x software supports SNMPv3, SNMPv1, and SNMPv2c.



Note

For information on security features provided in SNMPv3, SNMP security models, and security levels, see Chapter 21 of the *Cisco ACNS Software Configuration Guide for Locally Managed Deployments*.

Using SNMPv3, users can securely collect management information from their SNMP agents. Also, confidential information, such as SNMP set packets that change a Content Engine's configuration, can be encrypted to prevent their contents from being exposed on the wire. Also, the group-based administrative model allows different users to access the same SNMP agent with varying access privileges.

Examples

The following example configures the SNMP group name, security model, and notify view on the Content Engine:

```
ContentEngine(config)# snmp-server group acme v1 notify mymib
```

Related Commands

```
show snmp
snmp-server access-list
snmp-server community
snmp-server contact
snmp-server enable traps
snmp-server host
snmp-server location
snmp-server mib
snmp-server notify
snmp-server user
snmp-server view
```

snmp-server host

To specify the recipient of a host SNMP trap operation, use the **snmp-server host** global configuration command. To remove the specified host, use the **no** form of this command.

```
snmp-server host {hostname | ip-address} communitystring [v2c [retry number] [timeout
seconds] | [v3 {auth [retry number] [timeout seconds] | noauth [retry number] [timeout
seconds] | priv [retry number] [timeout seconds]}]]
```

```
no snmp-server host {hostname | ip-address} [v2c [retry number] [timeout seconds] | [v3 {auth
[retry number] [timeout seconds] | noauth [retry number] [timeout seconds] | priv [retry
number] [timeout seconds]}]] communitystring
```

Syntax Description

<i>hostname</i>	Hostname of the SNMP trap host that will be sent in the SNMP trap messages from the Content Engine.
<i>ip-address</i>	IP address of the SNMP trap host that will be sent in the SNMP trap messages from the Content Engine.
<i>communitystring</i>	Password-like community string sent in the SNMP trap messages from the Content Engine. You can enter a maximum of 64 characters.
v2c	(Optional) Specifies the Version 2c Security Model.
retry	(Optional) Sets the count for the number of retries for the inform request. (The default is 2 tries.)
<i>number</i>	Number of retries for the inform request (1–10).
timeout	(Optional) Sets the timeout for the inform request (1–1000). (The default is 15 seconds.)
<i>seconds</i>	Timeout value in seconds.
v3	(Optional) Specifies the User Security Model (SNMPv3).
auth	Sends notification using the AuthNoPriv Security Level.
noauth	Sends notification using the noAuthNoPriv Security Level.
priv	Sends notification using the AuthPriv Security Level.

Defaults

This command is disabled by default. No traps are sent. The version of the SNMP protocol used to send the traps is SNMP Version 1.

retry number: 2 retries.

timeout: 15 seconds.

Command Modes

global configuration

Usage Guidelines

SNMP notifications can be sent as traps or inform requests. Traps are unreliable because the receiver does not send acknowledgments when it receives traps. The sender cannot determine if the traps were received. However, an SNMP entity that receives an inform request acknowledges the message with an SNMP response PDU. If the sender never receives the response, the inform request can be sent again. Informs are more likely to reach their intended destination.

However, informs consume more resources in the agent and in the network. Unlike a trap, which is discarded as soon as it is sent, an inform request must be held in the memory until a response is received or the request times out. Also, traps are sent only once, while an inform may be retried several times. The retries increase traffic and contribute to a higher overhead on the network.

If you do not enter an **snmp-server host** command, no notifications are sent. To configure the Content Engine to send SNMP notifications, you must enter at least one **snmp-server host** command. To enable multiple hosts, you must enter a separate **snmp-server host** command for each host. You can specify multiple notification types in the command for each host.

When multiple **snmp-server host** commands are given for the same host and kind of security model, each succeeding command overwrites the previous command. Only the last **snmp-server host** command will be in effect. For example, if you enter an **snmp-server host v2c** command for a host and then enter another **snmp-server host v3** command for the same host, the second command will replace the first.

The maximum number of SNMP hosts that can be created by entering the **snmp-server host** commands is eight in the ACNS 5.2 software and later releases. However, you can configure only up to four hosts on a Content Engine running the ACNS 5.1.x software and earlier releases.

When multiple **snmp-server host** commands are given for the same host, the community string in the last command is used.

The **snmp-server host** command is used with the **snmp-server enable** command. Use the **snmp-server enable** command to specify which SNMP notifications are sent globally. For a host to receive most notifications, at least one **snmp-server enable** command and the **snmp-server host** command for that host must be enabled.

**Note**

You must enable SNMP with the **snmp-server community** command.

Examples

The following example sends the SNMP traps defined in RFC 1157 to the host specified by the IP address 172.16.2.160. The community string is comaccess.

```
ContentEngine(config)# snmp-server enable traps
ContentEngine(config)# snmp-server host 172.16.2.160 comaccess
```

The following example removes the host 172.16.2.160 from the SNMP trap recipient list:

```
ContentEngine(config)# no snmp-server host 172.16.2.160
```

Related Commands

```
show snmp
snmp-server access-list
snmp-server community
snmp-server contact
snmp-server enable traps
snmp-server group
snmp-server location
snmp-server mib
snmp-server notify
snmp-server user
snmp-server view
```

snmp-server location

To set the SNMP system location string, use the **snmp-server location** global configuration command. To remove the location string, use the **no** form of this command.

snmp-server location *line*

no snmp-server location

Syntax Description	<table> <tr> <td>location</td><td>Specifies the text for MIB-II object sysLocation.</td></tr> <tr> <td><i>line</i></td><td>String that describes the physical location of this node.</td></tr> </table>	location	Specifies the text for MIB-II object sysLocation.	<i>line</i>	String that describes the physical location of this node.
location	Specifies the text for MIB-II object sysLocation.				
<i>line</i>	String that describes the physical location of this node.				
Defaults	No system location string is set.				
Command Modes	global configuration				
Usage Guidelines	The system location string is the value stored in the MIB-II system group system location object. You can see the system location string with the show snmp EXEC command.				
Examples	<p>The following example shows how to configure a system location string:</p> <pre>ContentEngine(config)# snmp-server location Building 3/Room 214</pre>				
Related Commands	<ul style="list-style-type: none"> show snmp snmp-server access-list snmp-server community snmp-server contact snmp-server enable traps snmp-server group snmp-server host snmp-server mib snmp-server notify snmp-server user snmp-server view 				

snmp-server mib

To configure persistence for the SNMP Event MIB, use the **snmp-server mib** global configuration command. To disable the Event MIB, use the **no** form of this command.

snmp-server mib persist event

no snmp-server mib persist event

Syntax Description

persist	Configures MIB persistence.
event	Enables MIB persistence for the Event MIB.

Defaults

No default behavior or values

Command Modes

global configuration

Usage Guidelines

The Event MIB can set the threshold on any MIB variables supported by the ACNS 5.x software and store the threshold permanently on disk. The MIB persistence features allow the SNMP data of a MIB to be persistent across reloads; MIB information retains the same set object values each time that the Content Engine reboots.

You enable MIB persistence by using the **snmp-server mib persist event** command. Enter the **write mib-data** command to write the MIB data of all MIBs that have had persistence enabled using the **snmp-server mib persist event** command to a persistent partition on a disk storage. Any modified MIB data must be written to a persistent partition on a disk using the **write mib-data** command.

The ACNS 5.x software implementation of SNMP supports the following MIBs:

- MIB-II
- ENTITY-MIB
- HOST-RESOURCES-MIB
- CISCO-CONTENT-ENGINE-MIB
- CISCO-ENTITY-ASSET-MIB
- CISCO-CONFIG-MAN-MIB
- CISCO-CDP-MIB
- EVENT-MIB

The EVENT-MIB can set the threshold on any MIB variables supported by the ACNS 5.x software and store the threshold permanently on disk. The HOST-RESOURCES-MIB provides statistics on system resources.

**Note**

In the ACNS 5.2 software and later releases, the CISCO-CONTENT-ENGINE-MIB supports streaming WMT, RealProxy, and Cisco Streaming Engine statistics. Content Engines support WMT and RealProxy. (Cisco Streaming Engine is only supported on Content Engines that are registered with a Content Distribution Manager; Cisco Streaming Engine is not supported on Content Engines.)

In the ACNS 5.3 software release, the CISCO-CONTENT-ENGINE-MIB was modified to add support for WMT RTSP streaming for Windows Media 9 clients and servers (Windows Media 9 players and Windows Media 9 servers).

In the ACNS 5.2 software and later releases, there are six generic alarm traps in the CISCO-CONTENT-ENGINE-MIB for SNMP and Node Health Manager integration.

For each 64-bit counter MIB object, a 32-bit counter MIB object is implemented so that SNMP clients using the SNMPv1 protocol can retrieve data associated with 64-bit counter MIB objects. The MIB objects of each of these groups are read-only:

- The WMT MIB group provides statistics about WMT proxy and server performance. Twenty-eight MIB objects are implemented in this group. Six of these MIB objects are implemented as 64-bit counters.
- The Cisco Streaming Engine MIB group provides statistics about the RTSP streaming engine performance. Seven MIB objects are implemented in this group. Two of these MIB objects are implemented as 64-bit counters.
- The RealProxy MIB group provides statistics about RealProxy performance. Fourteen MIB objects are implemented in this group.

In the ACNS 5.1.1 software and later releases, you can use IP access control lists (ACLs) to control SNMP access on a Content Engine.

You can download the MIB files for all of the MIBs that are supported by a Content Engine that is running the ACNS 5.x software from the following Cisco FTP site:

<ftp://ftp.cisco.com/pub/mibs/v2>

The MIB objects that are defined in each MIB are described in the MIB files at the above FTP site and are self explanatory.

Examples

The following example sets persistence for the Event MIB:

```
ContentEngine(config)# snmp-server mib persist event
```

Related Commands

```
show snmp
snmp-server access-list
snmp-server community
snmp-server contact
snmp-server enable traps
snmp-server group
snmp-server host
snmp-server location
snmp-server notify
snmp-server user
snmp-server view
```

snmp-server notify inform

To configure the SNMP notify inform request, use the **snmp-server notify inform** global configuration command. To return the setting to the default value, use the **no** form of this command.

snmp-server notify inform

no snmp-server notify inform

Syntax Description

This command has no arguments or keywords.

Defaults

If you do not enter the **snmp-server notify inform** command, the default is an SNMP trap request.

Command Modes

global configuration

Usage Guidelines

The **snmp-server host** command specifies which hosts will receive informs. The **snmp-server enable traps** command globally enables the production mechanism for the specified notifications (traps and informs).

In order for a host to receive an inform, you must enable the inform globally by entering the **snmp-server notify inform** command.

The SNMP inform requests feature allows Content Engines to send inform requests to SNMP managers. Content Engines can send notifications to SNMP managers when particular events occur. For example, an agent Content Engine might send a message to a manager when the agent Content Engine experiences an error condition.

SNMP notifications can be sent as traps or inform requests. Traps are unreliable because the receiver does not send any acknowledgment when it receives a trap. The sender cannot determine if the trap was received. However, an SNMP manager that receives an inform request acknowledges the message with an SNMP response protocol data unit (PDU). If the manager does not receive an inform request, it does not send a response. If the sender never receives a response, the inform request can be sent again. Informs are more likely to reach their intended destination.

Because they are more reliable, informs consume more resources in the Content Engine and in the network. Unlike a trap, which is discarded as soon as it is sent, an inform request must be held in the memory until a response is received or the request times out. Also, traps are sent only once, while an inform may be retried several times. The retries increase traffic and contribute to a higher overhead on the network. Traps and inform requests provide a trade-off between reliability and resources.



Tip

If it is important that the SNMP manager receives every notification, then you should use inform requests in your network. If you are concerned about traffic on your network or about the memory in the Content Engine and you do not need to receive every notification, then you should use traps in your network.

Examples

The following example configures the SNMP notify inform request on the Content Engine:

```
ContentEngine(config)# snmp-server notify inform
```

Related Commands

- show snmp**
- snmp-server access-list**
- snmp-server community**
- snmp-server contact**
- snmp-server enable traps**
- snmp-server group**
- snmp-server host**
- snmp-server location**
- snmp-server mib**
- snmp-server user**
- snmp-server view**

snmp-server user

To define a user who can access the SNMP server, use the **snmp-server user** global configuration command. To remove access, use the **no** form of this command.

```
snmp-server user name group [auth {md5 password [priv password] | sha password [priv password]}] | remote octetstring [auth {md5 password [priv password] | sha password [priv password]}]]
```

```
no snmp-server user name group [auth {md5 password | sha password} [priv password] | remote octetstring [auth {md5 password | sha password} [priv password]]]
```

Syntax Description

<i>name</i>	Name of the SNMP user. Use letters, numbers, dashes, and underscores, but no blanks. This is the name of the user on the SNMP host who wants to communicate with the SNMP agent on the Content Engine. You can enter a maximum of 64 characters.
<i>group</i>	Name of the group to which the SNMP user belongs. You can enter a maximum of 64 characters.
auth	(Optional) Configures user authentication parameters.
md5	Configures the Hashed-Based Message Authentication Code Message Digest 5 (HMAC MD5) authentication algorithm.
<i>password</i>	HMAC MD5 user authentication password.
priv	(Optional) Configures authentication parameters for the packet.
<i>password</i>	HMAC MD5 user private password. You can enter a maximum of 256 characters.
sha	Configures the HMAC Secure Hash Algorithm (SHA) authentication algorithm.
<i>password</i>	HMAC SHA authentication password. You can enter a maximum of 256 characters.
remote	(Optional) Specifies the engine identity of the remote SNMP entity to which the user belongs.
<i>octetstring</i>	Globally unique identifier for a remote SNMP entity (for example, the SNMP network management station) for at least one of the SNMP users.

Defaults

No default behavior or values

Command Modes

global configuration

Usage Guidelines

The maximum number of SNMP users that can be created is 10. Follow these guidelines when defining SNMP users for Content Engines:

- If the SNMPv3 protocol is going to be used for SNMP requests, you must define at least one SNMPv3 user account on the Content Engine in order for the Content Engine to be accessed through SNMP.
- A group defined with the SNMPv1 or SNMPv2c security model should not be associated with SNMP users; they should only be associated with the community strings.

**Tip**

To send an SNMPv3 inform message, you must configure at least one SNMPv3 user with a remote SNMP ID option on the Content Engine. The SNMP ID is entered in octet string form. For example, if the IP address of a remote SNMP entity is 192.147.142.129, then the octet string would be 00:00:63:00:00:00:a1:c0:93:8e:81.

Examples

The following example shows that an SNMPv3 user account is created on the Content Engine. The SNMPv3 user is named acme and belongs to the group named admin. Because this SNMP user account has been set up with no authentication password, the SNMP agent on the Content Engine does not perform authentication on SNMP requests from this user.

```
ContentEngine(config)# snmp-server user acme admin
```

Related Commands

show snmp
snmp-server access-list
snmp-server community
snmp-server contact
snmp-server enable
snmp-server group
snmp-server host
snmp-server location
snmp-server mib
snmp-server notify
snmp-server view

snmp-server view

To define a Version 2 SNMP (SNMPv2) MIB view, use the **snmp-server view** global configuration command. To undefine the MIB view, use the **no** form of this command.

snmp-server view *viewname* *MIBfamily* { **excluded** | **included** }

no snmp-server view *viewname* *MIBfamily* [**excluded** | **included**]

Syntax Description

<i>viewname</i>	Name of this family of view subtrees. You can enter a maximum of 64 characters.
<i>MIBfamily</i>	An object identifier that identifies a subtree of the MIB. You can enter a maximum of 64 characters.
excluded	Excludes the MIB family from the view.
included	Includes the MIB family from the view.

Defaults

No default behavior or values

Command Modes

global configuration

Usage Guidelines

An SNMP view is a mapping between SNMP objects and the access rights available for those objects. An object can have different access rights in each view. Access rights indicate whether the object is accessible by either a community string or a user. The **snmp-server view** command is used with the **snmp-server group** to limit the read-write access of MIB trees based on the group. Because the group can be associated with the SNMP community string or users, using the **snmp-server view** command extends the limit to users and community strings. If the view is not configured, read-write access to the community string applies to the MIB tree and all users (SNMPv3).

The maximum number of views that can be created is 10. You can configure the SNMP view settings only if you have previously configured the SNMP server settings.

To remove a view record, use the **no snmp-server view** command.

You can enter the **snmp-server view** command multiple times for the same view record. Later lines take precedence when an object identifier is included in two or more lines.

Examples

The following example shows how to configure the view name, family name, and view type:

```
Content Engine(config)# snmp-server view contentview ciscoContentEngineMIB included
```

The following example creates a view that includes all objects in the MIB-II system group and all objects in the Cisco enterprise MIB:

```
Content Engine(config)# snmp-server phred system included
Content Engine(config)# snmp-server view phred cisco included
```

The following example creates a view that includes all objects in the MIB-II system group except for sysServices (System 7) in the MIB-II interfaces group:

```
ContentEngine(config)# snmp-server view agon system included
ContentEngine(config)# snmp-server view agon system.7 excluded
```

Related Commands

- show snmp
- snmp-server access-list
- snmp-server community
- snmp-server contact
- snmp-server enable
- snmp-server group
- snmp-server host
- snmp-server location
- snmp-server mib
- snmp-server notify
- snmp-server user

speed

To set the Fibre Channel interface speed, use the **speed** interface configuration command. To reset the interface speed, use the **no** form of this command.

```
speed {1 | 2 | autosense}

no speed {1 | 2 | autosense}
```

Syntax Description	1	Sets the Fibre Channel interface speed to 1 gigabit per second (Gbps).
	2	Sets the Fibre Channel interface speed to 2 Gbps.
	autosense	Sets the Fibre Channel to automatically sense the interface speed.

Defaults	No default behavior or values
----------	-------------------------------

Command Modes	interface configuration
---------------	-------------------------

Examples

The following example shows how to configure the speed of the Fibre Channel interface to 1 Gbps:

```
ContentEngine(config)# interface FibreChannel 0/1
ContentEngine(config-if)# speed 1
```

The following example shows how to configure the speed of the Fibre Channel interface to autosense:

```
ContentEngine(config)# interface FibreChannel 0/1
ContentEngine(config-if)# speed autosense
```

Related Commands	interface show interface show running-config show startup-config
------------------	---

sshd

To enable the Secure Shell (SSH) daemon, use the **sshd** global configuration command. To disable SSH, use the **no** form of this command.

sshd {**enable** | **password-guesses** *number* | **timeout** *seconds* | **version** {**1** | **2**}}

no sshd {**enable** | **password-guesses** | **timeout** | **version** {**1** | **2**}}

Syntax Description

enable	Enables the SSH feature.
password-guesses	Specifies the number of allowable password guesses per connection.
<i>number</i>	Maximum number of incorrect password guesses allowed (1–99). (The default is 3.)
timeout	Configures the number of seconds for which an SSH session will be active during the negotiation (authentication) phase between the client and the server before it times out.
	Note If you have established an SSH connection to the Content Engine but have not entered the username when prompted at the login prompt, the connection will be terminated by the Content Engine even after successful login if the grace period expires.
<i>seconds</i>	SSH login grace time value in seconds (1–99999). (The default is 300.)
version	Configures the SSH version to be supported on the Content Engine.
1	Specifies that SSH version 1 is supported on the Content Engine.
2	Specifies that SSH version 2 is supported on the Content Engine.

Defaults

password-guesses *number*: 3 guesses.
timeout *seconds*: 300 seconds.
version: Both SSH version 1 and 2 are enabled.

Command Modes

global configuration

Usage Guidelines

SSH enables login access to the Content Engine through a secure and encrypted channel. SSH consists of a server and a client program. Like Telnet, you can use the client program to remotely log on to a machine that is running the SSH server, but unlike Telnet, messages transported between the client and the server are encrypted. The functionality of SSH includes user authentication, message encryption, and message authentication.

In the ACNS software 5.3 and later releases, when you enable the SSH server, the Secure File Transfer Protocol (SFTP) server is also enabled. The SFTP is a file transfer program that provides a secure and authenticated method for transferring files between ACNS devices and other workstations or clients.

**Note**

SFTP is the standard file transfer protocol introduced in SSH version 2. The SFTP client functionality is provided as part of the SSH component. If you use SSH version 1 on the Content Engine, SFTP support is not available.

Before you enable the **sshd** command, use the **ssh-key-generate** command to generate a private and a public host key, which the client programs use to verify the server's identity.

Although the **sshd password-guesses** command specifies the number of allowable password guesses from the SSH server side, the actual number of password guesses for an SSH login session is determined by the combined number of allowable password guesses of the SSH server and the SSH client. Some SSH clients limit the maximum number of allowable password guesses to three (or to one in some cases), even though the SSH server side allows more than this number of guesses.

When you enter the **sshd password-guesses** command, specifying *n* allowable password guesses causes certain SSH clients to interpret this *number* as *n*+1. For example, when configuring the number of guesses to two by entering the **sshd password-guesses 2** command for a particular device, SSH sessions from some SSH clients will allow three password guesses.

The **sshd version** global configuration command allows you to enable support for either SSH version 1 or SSH version 2. When you enable SSH using the **sshd enable** global configuration command, the ACNS software enables support for both SSH version 1 and SSH version 2 on the Content Engine. If you want the Content Engine to support only one version of SSH (for example SSH version 2), you must disable the other version (in this example, SSH version 1) by using the **no sshd version 1** command.

When support for both SSH version 1 and SSH version 2 are enabled in the Content Engine, the **show running-config EXEC** command output does not display any sshd configuration. If you have disabled the support for one version of SSH, the **show running-config EXEC** command output contains the following line:

```
no sshd version version_number
```

**Note**

You cannot disable both SSH versions in a Content Engine. Use the **no sshd enable** global configuration command to disable SSH on the Content Engine.

Examples

The following example shows how to enable the SSH daemon and configure the number of allowable password guesses and timeout for the Content Engine:

```
ContentEngine(config)# sshd enable
ContentEngine(config)# sshd password-guesses 4
ContentEngine(config)# sshd timeout 20
```

The following example disables the support for SSH version 1 in the Content Engine:

```
ContentEngine(config)# no sshd version 1
```

Related Commands

ssh-key-generate
show ssh

ssh-key-generate

To generate the Secure Shell (SSH) host key, use the **ssh-key-generate** global configuration command. To remove the SSH key, use the **no** form of this command.

ssh-key-generate [**key-length** *length*]

no ssh-key-generate [**key-length** *length*]

Syntax Description	key-length	(Optional) Configures the length of the SSH key.
	<i>length</i>	Specifies the number of bits to create an SSH key (512–2048).

Defaults	key-length <i>length</i> : 1024 bits
----------	---

Command Modes	global configuration
---------------	----------------------

Usage Guidelines	Before you enable the sshd command, use the ssh-key-generate command to generate a private and a public host key, which the client programs use to verify a server's identity.
	When a user runs an SSH client and logs in to the Content Engine, the public key for the SSH daemon running on the Content Engine is recorded in the client machine <code>known_hosts</code> file in the user's home directory. If the Content Engine administrator subsequently regenerates the host key by entering the ssh-key-generate command, the user must delete the old public key entry associated with the Content Engine in the <code>known_hosts</code> file before running the SSH client program to log in to the Content Engine. When the user runs the SSH client program after deleting the old entry, the <code>known_hosts</code> file is updated with the new SSH public key for the Content Engine.

Examples	The following example generates an SSH public key and then enables the SSH daemon:
----------	--

```
ContentEngine(config)# ssh-key-generate
Ssh host key generated successfully
Saving the host key to box ...
Host key saved successfully
ContentEngine(config)# sshd enable
Starting ssh daemon ...
Ssh daemon started successfully
```

Related Commands	sshd
------------------	-------------

standby

To configure an interface to be a backup for another interface, use the **standby** interface configuration command. To restore the default configuration of the interface, use the **no** form of this command.

standby *group_number* { **description** *text* | **errors** *max-errors* | **ip** *ip-address netmask* | **priority** *priority_level* | **shutdown** }

no standby *group_number* { **description** *text* | **errors** *max-errors* | **ip** *ip-address netmask* | **priority** *priority_level* | **shutdown** }

Syntax Description

<i>group_number</i>	Standby group number (1–4).
description	(Optional) Sets the description for the specified interface.
<i>text</i>	Description for the specified interface. The maximum length of the description text is 240 characters.
errors	Sets the maximum number of errors allowed on the active interface before the interface is shut down and the standby interface is brought up. This option is disabled by default.
<i>max-errors</i>	Maximum number of errors (0–4294967295).
ip	Sets the IP address for the specified standby group (Standby Group 1, 2, 3, or 4).
<i>ip-address</i>	IP address of the specified standby group (Standby Group 1, 2, 3, or 4). The group IP address and netmask of a standby group must be configured on all of the member interfaces.
<i>netmask</i>	Netmask of the specified standby group (Standby Group 1, 2, 3, or 4).
priority	Sets the priority of the member interface within a standby group. The priority of a member interface can be changed at run time. The member interface that has the highest priority after this change becomes the new active interface (the default action is to preempt the currently active interface if an interface with higher priority exists).
<i>priority_level</i>	Each member interface is assigned a priority number. The member interface with the highest priority number is the active interface for that standby group. Only the active interface uses the group IP address. If the priority option is specified without a priority number, the default value of 100 is used.
shutdown	(Optional) Shuts down the specified standby group (Standby Group 1, 2, 3, or 4). You can shut down a standby group even if you have not configured a group IP address for the standby group. Note When a standby group is shut down, all of the alarms previously raised by this standby group are cleared.

Defaults

There are no standby interfaces by default. The **errors** option is disabled by default.

Command Modes

interface configuration

Usage Guidelines

In the ACNS 5.2 software and later releases, you can configure one or more interfaces to act as a backup interface (a standby interface) for another interface on a Content Engine. This feature is called standby interface support. Standby groups, which are logical groups of interfaces, are used to implement this feature. When an active network interface fails (because of cable trouble, Layer 2 switch failure, high error count, or other failures) and that interface is part of a standby group, a standby interface can become active and take the load off the failed interface.

A standby group must have at least two interfaces. Interfaces that are part of a standby group are called member interfaces. After you create a standby group, you define which interfaces should be assigned to this logical group. As part of defining the member interfaces, you specify the priority of each member interface in a standby group. The member interface with the highest assigned priority is the active interface for that particular standby group. If the active interface fails, the operational member interface with the next highest priority in the standby group comes up, and so forth. If all member interfaces of a particular standby group are down and then one of the member interfaces comes up, the ACNS software detects this situation and brings up the standby group on the member interface that just came up.

In the ACNS 5.3 software and later releases, the failure or failover of member interfaces within a standby group triggers alarms and traps (if alarms and traps are enabled on the Content Engine). Alarms are sent out when failover occurs between member interfaces in a standby group. Specifically, minor alarms are sent out when member interfaces fail, and these alarms are cleared automatically when the interface failover has been successfully completed. Major alarms are sent out if the standby group goes down (no member interface in a standby group can be brought up.)



Note

A physical interface can belong to more than one standby group, and a single interface can act as a standby interface for more than one standby group.

This standby interface feature can also be used to support a redundant network that uses Layer 4 Cisco Content Services Switch [CSS] switches) to load balance requests to multiple Content Engines. The CSS switch supports active-standby configuration. If the active CSS switch fails, the standby CSS switch takes over all of the load. In such a case, the Content Engine detects this failure and starts serving the same IP address (shared IP address) on the standby network interface card (NIC), which preserves the existing TCP sessions (session-level redundancy). Session-level redundancy is only possible if the CSS switch can preserve sessions in a failure scenario. If the CSS switch loses the sessions, the session will be lost.

To configure standby interfaces, interfaces are logically assigned to standby groups. The following rules define the standby group relationships:

- Each standby group is assigned a unique standby IP address, shared by all member interfaces of the standby group. The IP address of the standby group is shared among the member interfaces; however, only the active interface of the standby group uses this shared IP address at any one time. This shared IP address is configured as an alias on the active interface.
- In the ACNS 5.2.x software, a physical interface needed a dummy or valid IP address assigned to it before you could add the physical interface to a standby group. In the ACNS 5.3 software and later releases, this assignment of an IP address to a physical interface is no longer a requirement.
- The duplex and speed settings of the member interfaces can be configured for better reliability.
- If all the member interfaces of a standby group fail and then one recovers, the ACNS software brings up the standby group on the operational member interface.
- If a physical interface is a member of a port-channel group, it cannot join a standby group. If a physical interface is a member of a standby group, it cannot join a port-channel group.

- A standby group comprises two or more interfaces.
- The maximum number of standby groups on a Content Engine is four.

**Note**

Interface IP addresses and standby group IP addresses must be on different subnets to ensure reliable operation. You can use dummy IP addresses in the private address space to serve as interface primary IP addresses, and use the real Content Engine IP address to serve as the standby group IP address in a different subnet to satisfy this requirement. When dummy IP addresses are used, these interface IP addresses serve only as substitutes to bring up the interface. For example, the Content Engine interface requires an IP address on an interface for initialization. Make sure to configure the interface default gateway using the **ip default-gateway** global configuration command instead of the **ip route** command.

- Each interface in a standby group is assigned a priority. The operational interface with the highest priority in a standby group is the active interface. Only the active interface uses the group IP address.
- If the active interface fails, the operational interface in its standby group that is assigned the next highest priority becomes active. However, when the interface with the higher priority recovers, it does not become active again without manual intervention.
- The priority of an interface in a standby group can be changed at run time. The member interface that has the highest priority after this change becomes the new active interface (the default action is to preempt the currently active interface if an interface with higher priority exists).
- The maximum number of errors allowed on the active interface before the interface is shut down and the standby is brought up is configured with the **errors** option, which is disabled by default.

**Tip**

If an interface belongs to more than one standby group, you can configure the interface with a different priority in each standby group for better load balancing. For example, interfaces FE 0/0 and FE 0/1 are both in standby group 1 and in standby group 2. If you configure FE 0/0 with the highest priority in standby group 1 and configure FE 0/1 with the highest priority in standby group 2, standby group 1 will use FE 0/0 as the active interface, while standby group 2 will use FE 0/1 as the active interface. This configuration allows each interface to back up the other one, if one of them fails.

Use the **interface standby** global configuration command to create standby groups on Content Engines. In the ACNS 5.3 software, the CLI syntax for configuring standby groups was changed to make it more similar to the port channel CLI syntax.

**Note**

Unlike port channels, standby groups do not support IP ACLs at a group level. However, you can configure a member interface of a standby group to support an IP ACL at the interface level. For example, you can individually configure the two member interfaces of Standby Group 1 (the Fast Ethernet slot 0/port 0 interface and the Fast Ethernet slot 0/port 1 interface) to support an IP ACL named ACL1 but you cannot configure the Standby Group 1 to support ACL1.

Examples

The following example configures three Fast Ethernet interfaces to be part of the same standby group, with interface 3/0 as the active interface:

```
Console(config-if)# interface fastEthernet 3/0 standby 1 ip 172.16.10.10 255.255.254.0
Console(config-if)# interface fastEthernet 3/1 standby 1 ip 172.16.10.10 255.255.254.0
Console(config-if)# interface fastEthernet 3/2 standby 1 ip 172.16.10.10 255.255.254.0
Console(config-if)# interface fastEthernet 3/0 standby 1 priority 300
Console(config-if)# interface fastEthernet 3/1 standby 1 priority 200
Console(config-if)# interface fastEthernet 3/2 standby 1 priority 100
```

```

Console(config-if)# interface fastEthernet 3/0 standby 1 errors 10000
Console(config-if)# interface fastEthernet 3/1 standby 1 errors 10000
Console(config-if)# interface fastEthernet 3/2 standby 1 errors 10000

```

The following example displays information about the standby group configuration by entering the **show standby** EXEC command. In the following sample command output, one standby group (Standby Group 1) is configured on this Content Engine. The command output also shows which member interface is the active interface. In this case, the active interface is the Fast Ethernet slot 3/port 0 interface.

```

ContentEngine# show standby
Standby Group:1
IP address: 172.16.10.10, netmask: 255.255.254.0
Maximum errors allowed on the active interface: 10000
  Member interfaces:
    FastEthernet 3/0      priority: 300
    FastEthernet 3/1      priority: 200
    FastEthernet 3/2      priority: 100

  Active interface: FastEthernet 3/0

```



Note To display information about a specific standby group configuration, enter the **show interface standby group_number** EXEC command.

The following example creates a standby group, Standby Group 1:

```

ContentEngine# configure
ContentEngine(config)# interface standby 1
ContentEngine(config-if)#

```

The following example assigns a group IP address of 10.10.10.10 and a netmask of 255.0.0.0 to Standby Group 1. In the ACNS 5.3 software and later releases, you can configure a group IP address regardless of whether the standby group is shut down or not.

```

ContentEngine(config-if)# ip address 10.10.10.10 255.0.0.0
ContentEngine(config-if)# errors 500

```

The following example shows how to add two Fast Ethernet interfaces to Standby Group 1 and then assign each of these member interfaces a priority within the group:

- a. Add a Fast Ethernet interface (slot 0/port 0) to Standby Group 1 and assign a priority of 150:

```

ContentEngine(config)# interface FastEthernet 0/0
ContentEngine(config-if)# standby 1 priority 150

```

- b. Add a second Fast Ethernet interface (slot 0/port 1) to Standby Group 1 and assign a priority of 100 (the default value):

```

ContentEngine(config)# interface FastEthernet 0/1
ContentEngine(config-if)# standby 1
ContentEngine(config-if)# exit
ContentEngine(config)#

```

Because Fast Ethernet 0/0 is assigned the highest priority (a priority number of 150) of all the member interfaces in the group, it will be chosen as the active interface for the group if it can be brought up.

The following example removes the Fast Ethernet slot 0/port 1 interface) from Standby Group 1 using the **no** form of the **standby** command:

```

ContentEngine(config)# interface FastEthernet 0/1
ContentEngine(config-if)# no standby 1

```

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

The following example shows how to shut down Standby Group 1. When a standby group is shut down, all of the alarms previously raised by this standby group are cleared.

```
ContentEngine(config)# interface standby 1
ContentEngine(config-if)# exit
ContentEngine(config)# exit
```

The following example shows how to tear down Standby Group 1:

```
ContentEngine(config)# interface standby 1
ContentEngine(config-if)# no ip address 10.10.10.10 255.0.0.0
Please remove member interface(s) from this standby group first.
ContentEngine(config)# interface GigabitEthernet 2/0
ContentEngine(config-if)# no standby 1
ContentEngine(config-if)# exit
ContentEngine(config)# interface standby 1
ContentEngine(config-if)# no ip address 10.10.10.10 255.0.0.0
ContentEngine(config-if)# exit
ContentEngine(config)# no interface standby 1
ContentEngine(config)# exit
```

Related Commands

```
interface
show interface
show running-config
show standby
show startup-config
```

tacacs

To configure TACACS+ server parameters, use the **tacacs** command in global configuration mode. To disable individual options, use the **no** form of this command.

tacacs {**enable** | **host** {*hostname* | *ip-address*} [**primary**] | **key** *keyword* | **password** **ascii** | **retransmit** *retries* | **timeout** *seconds*}

no tacacs {**enable** | **host** {*hostname* | *ip-address*} [**primary**] | **key** | **password** **ascii** | **retransmit** | **timeout**}

Syntax Description	
enable	Enables the TACACS+ authentication.
host	Sets a server address.
<i>hostname</i>	Hostname of the TACACS+ server.
<i>ip-address</i>	IP address of the TACACS+ server.
primary	(Optional) Sets the server as the primary server.
key	Sets the security word.
<i>keyword</i>	Keyword. An empty string is the default.
password ascii	Specifies ASCII as the TACACS+ password type.
retransmit	Sets the number of times that requests are retransmitted to a server.
<i>retries</i>	Number of retry attempts allowed (1–3). The default is 2 retry attempts.
timeout	Sets the number of seconds to wait before a request to a server is timed out.
<i>seconds</i>	Timeout in seconds (1–20). The default is 5 seconds.

Defaults

keyword: none (empty string).

timeout *seconds*: 5.

retries: 2.

password: The default password type is PAP.

Command Modes

global configuration

Usage Guidelines

Using the **tacacs** command, configure the TACACS+ key, the number of retransmits, the server hostname or IP address, and the timeout.

You must execute the following two commands to enable user authentication with a TACACS+ server:

```
ContentEngine(config)# authentication login tacacs enable
ContentEngine(config)# authentication configuration tacacs enable
```

You must enable TACACS+ for HTTP request authentication as follows:

```
ContentEngine(config)# tacacs enable
```

TACACS+ can be disabled but remain configured for user authentication with a TACACS+ server if you use the **no** option of the command as follows:

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

HTTP request authentication is independent of user authentication options and must be disabled with the following separate commands:

```
ContentEngine(config)# no authentication login tacacs enable
ContentEngine(config)# no authentication configuration tacacs enable
```

The Users GUI page or the **user** global configuration commands provide a way to add, delete, or modify usernames, passwords, and access privileges in the local database. The TACACS+ remote database can also be used to maintain login and configuration privileges for administrative users. The **tacacs host** command or the TACACS+ Content Engine GUI page allows you to configure the network parameters required to access the remote database.

One primary and two backup TACACS+ servers can be configured; authentication is attempted on the primary server first and then on the others in the order in which they were configured. The primary server is the first server configured unless another server is explicitly specified as primary with the **tacacs host hostname primary** command.

Use the **tacacs key** command to specify the TACACS+ key that is used to encrypt the packets transmitted to the server. This key must be the same as the one specified on the server daemon. The maximum number of characters in the key should not exceed 99 printable ASCII characters (except tabs). An empty key string is the default. All leading spaces are ignored; spaces within and at the end of the key string are not ignored. Double quotes are not required even if there are spaces in the key, unless the quotes themselves are part of the key.

The **tacacs timeout** is the number of seconds that the Content Engine waits before declaring a timeout on a request to a particular TACACS+ server. The range is from 1 to 20 seconds with 5 seconds as the default. The number of times that the Content Engine repeats a retry-timeout cycle before trying the next TACACS+ server is specified by the **tacacs retransmit** command. The default is two retry attempts.

Three unsuccessful login attempts are permitted. TACACS+ logins may appear to take more time than local logins depending on the number of TACACS+ servers and the configured timeout and retry values.

Use the **tacacs password ascii** command to specify the TACACS+ password type as ASCII. The default password type is Password Authentication Protocol (PAP). In releases prior to the ACNS software 5.x, the password type was not configurable. When users needed to log in to a Content Engine, a TACACS+ client sent the password information in PAP format to a TACACS+ server. However, TACACS+ servers that were configured for router management required the passwords to be in ASCII clear text format instead of PAP format to authenticate users logging in to the Content Engine. The password type to authenticate user information to ASCII was configurable from the CLI.



Note

When the **no tacacs password ascii** command is used to disable the ASCII password type, the password type is once again reset to PAP.

The TACACS+ client can send different requests to the server for user authentication. The client can send a TACACS+ request with the PAP password type. In this scenario, the authentication packet includes both the username and the user's password. The server must have an appropriately configured user's account.

Alternatively, the client can send a TACACS+ request with the ASCII password type as another option. In this scenario, the authentication packet includes the username only and waits for the server response. Once the server confirms that the user's account exists, the client sends another Continue request with the user's password. The authentication server must have an appropriately configured user's account to support either type of password.

Examples

The following example configures the key used in encrypting packets:

```
ContentEngine(config)# tacacs key human789
```

The following example configures the host named spearhead as the primary TACACS+ server:

```
ContentEngine(config)# tacacs host spearhead primary
```

The following example sets the timeout interval for the TACACS+ server:

```
ContentEngine(config)# tacacs timeout 10
```

The following example sets the number of times that authentication requests are retried (retransmitted) after a timeout:

```
ContentEngine(config)# tacacs retransmit 5
```

The following example shows the password type to be PAP by default:

```
ContentEngine# show tacacs
  Login Authentication for Console/Telnet Session: enabled (secondary)
  Configuration Authentication for Console/Telnet Session: enabled (secondary)

TACACS+ Configuration:
-----
TACACS+ Authentication is off
Key           = *****
Timeout       = 5
Retransmit    = 2
Password type: pap

Server                               Status
-----
10.107.192.148                        primary
10.107.192.168
10.77.140.77
ContentEngine#
```

However, you can configure the password type to be ASCII using the **tacacs password ascii** command. You can then verify the changes using the **show tacacs** command as follows:

```
ContentEngine(config)# tacacs password ascii
ContentEngine(config)# exit
ContentEngine# show tacacs
  Login Authentication for Console/Telnet Session: enabled (secondary)
  Configuration Authentication for Console/Telnet Session: enabled (secondary)

TACACS+ Configuration:
-----
TACACS+ Authentication is off
Key           = *****
Timeout       = 5
Retransmit    = 2
Password type: ascii
```


Server	Status
-----	-----
10.107.192.148	primary
10.107.192.168	
10.77.140.77	

Related Commands

authentication
show authentication
show statistics authentication
show statistics tacacs
show tacacs