

show statistics distribution

To display the statistics of the content distribution components, use the **show statistics distribution** EXEC command.

```
show statistics distribution [all | errors {delivery-service-id delivery-service-id |
delivery-service-name delivery-service-name} | metadata-receiver | metadata-sender |
unicast-data-receiver [delivery-service-id delivery-service-id | delivery-service-name
delivery-service-name | hot-forwarders [forwarder_id | forwarder_name] | idle-forwarders
max_idle_forwarders] | unicast-data-sender]
```

Syntax Description		
all		Displays the content distribution statistics for all distribution components.
errors		Displays the distribution error records for the specified channel.
delivery-service-id		Displays statistics about the specified delivery service ID.
<i>delivery-service-id</i>		Delivery service number.
delivery-service-name		Displays statistics about the specified delivery service name.
<i>delivery-service-name</i>		Delivery service name.
metadata-receiver		Displays the content distribution statistics of the metadata receiver.
metadata-sender		Displays the content distribution statistics of the metadata sender.
unicast-data-receiver		Displays the content distribution statistics of the unicast data receiver.
delivery-service-id	(Optional)	Displays statistics about the specified delivery service ID.
<i>delivery-service-id</i>	(Optional)	Delivery service number.
delivery-service-name	(Optional)	Displays statistics about the specified delivery service name.
<i>delivery-service-name</i>	(Optional)	Delivery service name.
hot-forwarders	(Optional)	Displays the content distribution statistics of hot forwarders.
<i>forwarder_id</i>	(Optional)	Identifier for the hot forwarder SE.
<i>forwarder_name</i>	(Optional)	Name of the hot forwarder SE.
idle-forwarders	(Optional)	Displays the content distribution statistics of idle forwarders.
<i>max_idle_forwarders</i>	(Optional)	Maximum number of idle forwarder SEs to be displayed.
unicast-data-sender		Displays the content distribution statistics of the unicast data sender.

Defaults

The **idle-forwarders** *max_idle_forwarders* default is 3.

Command Modes

EXEC

Usage Guidelines

Cisco Internet Streamer Release 2.4 software supports multicast file transfer features that enhance the reliability and performance of multicast file distribution. Previously, the file transfer session depended on a window of time to resend the missing packets. The sender had to transmit the packets within this window of time for each retransmission request (NACK) from receiver SEs. If a multicast receiver joined the session too late and missed blocks of data that were outside the transmission window, the sender would not resend the missing blocks. The receiver could not receive the entire file, and the transmission

failed. The receiver had to wait until a subsequent carousel pass to recover the missed files. The receiver could only receive the entire file or nothing. A slow receiver often failed to receive a large file if the receiving rate lagged behind the sending rate.

The multicast file transfer enhancements resolve these issues by eliminating the window of time for file transmissions. This feature is called checkpoint. Checkpoint allows the sender to divide the transferring file into blocks and to retransmit any and all blocks until the transfer session ends. At any time during the transfer session, a receiver can request retransmission of any block that it has missed. Also, receiver SEs can receive the blocks of a transfer in any order. Data transmission can occur over a longer period, and receivers can recover missed data blocks to successfully complete the transfer in most situations. File transfers are much more resistant to loss of data.

This feature also solves the problem of a multicast receiver joining a transfer session late. Even if a receiver goes offline and restarts during a transfer, it can recover missing data without requesting retransmission of the blocks that it has already received.

Examples

Table 2-53 describes the fields shown in the **show statistics distribution unicast-data-receiver** display.

Table 2-53 *show statistics distribution unicast-data-receiver Field Descriptions*

Field	Description
Channel ID	Numerical identifier for the channel.
Channel name	Name for the channel.
Current unicast forwarder ID	Numerical identifier for the current unicast forwarder.
Current unicast forwarder name	Name for the current unicast forwarder.
Use hot forwarder	Status of the forwarder SE. Values are Yes or No. Yes means that the forwarder is active, and the job for this channel can be started immediately. No means that the forwarder is currently inactive and may become active some time later depending on the failure reason. For example, any new forwarder must wait at least one minute before starting active jobs.
Current running job	Shows statistics for jobs that are currently running.
relative-cdn-url	Relative URL for the current job.
channel-id	Numerical identifier for the channel for this job.
fwdr ip address	IP address of the current unicast forwarder for this job.
bytes written/total	Total number of bytes written for this job.
last write time	Number of seconds since the last write time for this job.
Cumulative bps	Number of cumulative bits per second.
Last successful job was done at	Time of completion of the last successful job.
#Consecutive failures	Number of consecutive failures.
#Jobs in pending queue(P_Q)	Number of jobs pending.

Table 2-53 *show statistics distribution unicast-data-receiver Field Descriptions (continued)*

Field	Description
#Jobs in suspended queue(S_Q)	Number of jobs suspended.
#Jobs in waiting queue(W_Q)	Number of jobs waiting.
#Bytes of jobs in P_Q and W_Q	Total number of bytes for jobs that are pending and waiting.
#Bytes of jobs in S_Q	Number of bytes for jobs that are suspended.
#Bytes of running jobs	Number of bytes for jobs that are currently running.


Related Commands

clear
show distribution

show statistics flash-media-streaming

To display the statistics for Flash Media Streaming, use the **show statistics flash-media-streaming EXEC** command.

show statistics flash-media-streaming [**connections** | **dvrcast** | **errors** | **livestats** | **performance** | **requests**]

Syntax Description		
connections	(Optional)	Displays Flash Media Streaming connections statistics.
dvrcast	(Optional)	Displays Flash Media Streaming dvrcast application statistics.
		
	Note	The dvrcast option will be available until the 2.4.3 release.
errors	(Optional)	Displays Flash Media Streaming errors statistics.
livestats	(Optional)	Displays Flash Media Streaming live application statistics.
performance	(Optional)	Displays Flash Media Streaming performance statistics.
requests	(Optional)	Displays Flash Media Streaming requests statistics.

Defaults No default behavior or values.

Command Modes EXEC

Examples The following example displays the statistics for Flash Media Streaming:

```
ServiceEngine#show statistics flash-media-streaming
Flash Media Streaming Statistics
Statistics have not been cleared since last Flash Media Streaming starts

Connections
-----
Current           :                0
Current VOD       :                0
Current LIVE      :                0
Current DVRCast   :                0
Max Concurrent    :                0
Total             :                0
Total VOD Req     :                0
Total LIVE Req    :                0
Total DVRCast Req :                0
Total Other Proxy Req:            0

Live Streaming
-----
UpStream BW       :                0 Kbps
DownStream BW     :                0 Kbps
UpStream Bytes    :                0
DownStream Bytes  :                0
Num of Instance Load :            0

DVRCast Streaming
```

```

-----
UpStream BW      :          0 Kbps
DownStream BW    :          0 Kbps
UpStream Bytes   :          0
DownStream Bytes :          0
Num of Instance Load :        0

Flash Video Cache Statistics
-----
Hits             :          0
Misses           :          0
Released         :          0
Bytes in cache   :          0
Bytes in use     :          0

Performance
-----
Server Up Time   :          0 S
Mem Usage        :          0 %
Max Mem Usage    :          0 %

Cache
-----
Cache Hit        :          0
Cache Miss       :          0
Proxy Case       :          0
Cache Hit Percentage :      0.00 %

Preposition
-----
Preposition Hit   :          0

Bytes Served
-----
Total Server Bytes :          0
Local Disk Reads   :          0
HTTP Based Reads   :          0
Bytes From Local Disk:          0
Bytes Through HTTP :          0

Rules
-----
Action Allow      :          0
Action Block      :          0
Validate url Sign :          0
Errors            :          0
Auth Server Allow :          0
Auth Server Deny  :          0

Error
-----
Invalid Error     :          0
Server Error      :          0
Media Not Found   :          0
Media Unauthorize :          0
Invalid Request   :          0

```

Related Commands

flash-media-streaming
show flash-media-streaming

show statistics http

To display SE HTTP statistics, use the **show statistics http** EXEC command.

show statistics http {ims | object | pcmm | performance | requests | rule}

Syntax Description

ims	Displays HTTP if-modified-since statistics.
object	Displays HTTP object statistics.
pcmm	Displays PacketCable Multimedia (PCMM) statistics.
performance	Displays HTTP performance statistics.
requests	Displays HTTP request statistics.
rule	Displays rule statistics.

Defaults

No default behavior or values.

Command Modes

EXEC

Examples

[Table 2-54](#) describes the fields shown in the **show statistics http ims** displays.

Table 2-54 *show statistics http ims Field Descriptions*

Field	Description
Total Issued	
Range Issued	
Responses	
Fresh	
Revalid	
Partial Fresh	

[Table 2-55](#) describes the fields shown in the **show statistics http object** displays.

Table 2-55 *show statistics http object Field Descriptions*

Field	Description
Revalidate Requests	
Stale Content	
No-Cache Requests	
Min TTL Expired	
Max TTL Expired	
Object Expired	

Table 2-55 *show statistics http object Field Descriptions (continued)*

Field	Description
Max Age Header	
Large File Size	
Content Not Modified	
No Content Length	
Range Request	
No Store	
Private	
Auth Required	
Non Cacheable	
Head Request	
Vary Header	
Miscellaneous	

[Table 2-56](#) describes the fields shown in the **show statistics http pcmm** displays.

Table 2-56 *show statistics http pcmm Field Descriptions*

Field	Description
No: of Icap Request	
No: of Signature Generation	
No: of Signature Validation	

[Table 2-57](#) describes the fields shown in the **show statistics http performance** displays.

Table 2-57 *show statistics http performance Field Descriptions*

Field	Description
Total Accesses	
Total kBytes	
Request Per Second	
kBytes Per Second	
kBytes Per Request	

Table 2-58 describes the fields shown in the **show statistics http requests** displays.

Table 2-58 *show statistics http requests Field Descriptions*

Field	Description
Cache Hit	Number of requests that resulted in a cache hit for all SEs in the CDS network.
Cache Miss	Number of requests that resulted in a cache miss (the web object was not available in the cache) for all SEs in the CDS network.
Range Requests	Number of requests in the range.
Partial Hit–Live fill	Number of requests that resulted in a partial hit-live fill.
Partial Hit–Refill	Number of requests that resulted in a partial hit-refill.
Partial Caching–Bypassed	Number of bypassed partial caching requests.
Preposition Hits	Number of preposition hit requests.
Reply Meta	Number of reply meta requests.
Alternate Media	Number of alternate media requests.
Num Lookups	Number of lookup requests.
Lookup Errors	Number of lookup request errors.
Streaming Redirected Requests	Number of client requests for the content redirected by the SR to the closest SE in the CDS network containing that content.
WMT Liveness Requests	Number of WMT liveness requests.
Hierarchical Cache Liveness Requests	Number of hierarchical cache liveness requests.
Client Errors	Number of client error requests or authentication failures handled by the SE.
Server Errors	Number of origin server errors or authentication failures handled by the SE.
HTTP 0.9 Requests	Number of requests made using the HTTP 0.9 version. HTTP/0.9 cannot manage caches because document transfers are not optimized. HTTP/0.9, which is the first version of HTTP, has only the GET method. Everything is performed with this method, including sending data to the server (the requested URI looks like the following: <code>http://www.foo.bar/url?var1=foo</code> ; the string that follows the first question mark means that the variable called <code>var1</code> is set to <code>foo</code>).

Table 2-58 *show statistics http requests Field Descriptions (continued)*

Field	Description
HTTP 1.0 Requests	<p>Number of requests made using the HTTP 1.0 version. HTTP/1.0 provides a simple caching mechanism. An origin server may mark a response, using the Expires header, with a time until which the cache could return the response without violating semantic transparency. A cache may check the current validity of a response using a conditional request. It may include an If-Modified-Since header in a request for the resource, specifying the value in the cached response's Last-Modified header. The server may then either respond with a 304 (Not Modified) status code, implying that the cache entry is valid, or it may send a normal 200 (OK) response to replace the cache entry.</p> <p>HTTP/1.0 also included a mechanism, the Pragma: no-cache header, for the client to indicate that a request should not be satisfied from a cache.</p>
HTTP 1.1 Requests	<p>Number of requests made using the HTTP 1.1 version. HTTP/1.1 includes a number of new conditional request-headers, in addition to If-Modified-Since. The most basic is If-None-Match, which allows a client to present one or more entity tags from its cache entries for a resource. If none of these matches the resource's current entity tag value, the server returns a normal response; otherwise, it may return a 304 (Not Modified) response with an ETag header that indicates which cache entry is currently valid. This mechanism allows the server to cycle through a set of possible responses, while the If-Modified-Since mechanism only generates a cache hit if the most recent response is valid.</p> <p>HTTP/1.1 also adds new conditional headers called If-Unmodified-Since and If-Match, which create other forms of preconditions on requests.</p>
Http Invalid Requests	Number of invalid HTTP requests.
Blocked	Number of blocked requests.
Allowed	Number of allowed requests.

Related Commands

clear
show ftp

show statistics icap

To display Internet Content Adaptation Protocol (ICAP)-related statistics for the SE, use the **show statistics icap** EXEC command.

show statistics icap

Syntax Description This command has no arguments or keywords.

Defaults No default behavior or values.

Command Modes EXEC

Examples [Table 2-59](#) describes the fields shown in the **show statistics icap** display.

Table 2-59 *show statistics icap Field Descriptions*

Field	Description
ICAP-client statistics (http proxy)	Statistics of the client of the ICAP process running on the Service Engine. The client of the ICAP process refers to the caching application that contacts the ICAP process which in turn communicates with the ICAP server. The server of the ICAP process refers to the ICAP server.
Total requests for V1 via RPC	Total number of client requests processed at the vectoring point V1.
Time per ICAP request (last 1k reqs)	Time taken to process each ICAP request. This field displays the time taken per request for the last 1000 requests.
ICAP daemon connection error	Number of errors that occurred when the caching application on the Service Engine attempted to connect to the ICAP daemon on the Service Engine.
Bad packets from ICAP daemon	Number of responses received by the Service Engine from the ICAP daemon that were not correct.
Error parsing HTTP req hdr from ICAP	Number of HTTP response headers received from the ICAP daemon that were incorrect. The ICAP daemon is the outgoing proxy server of the Service Engine.
ICAP daemon internal error	Number of internal errors that occurred on an ICAP daemon.
Total requests via outgoing proxy	Total number of requests sent from the caching application to the ICAP process through the outgoing proxy configured on the Service Engine.
ICAP daemon overloaded	Number of times that the ICAP process running on the Service Engine was overloaded.
Other errors	Number of all other errors that occurred when an ICAP request is processed.

Table 2-59 *show statistics icap Field Descriptions (continued)*

Field	Description
ICAP Daemon statistics	Statistics recorded on the ICAP daemon running on the Service Engine.
Total requests served	Total number of requests that were served by the ICAP daemon running on the Service Engine.
Total bytes to ICAP-Client	Total number of bytes sent to the ICAP client running on the Service Engine.
Average latency in milliseconds	Average delay in serving ICAP requests to ICAP clients in milliseconds.
ICAP Service statistics	Statistics for each ICAP service configured on the Service Engine.
Service	Name of the configured ICAP service.
Service Errors	Number of error messages returned by the ICAP service in response to client requests. These errors are also entered in the transaction log to show the status of the action performed by the ICAP service.
Service Bypasses	Number of requests that bypassed this ICAP service. The value of this field is incremented when you have configured this service to be bypassed when an error occurs with this service.
Server	Hostname or IP address of the ICAP server. Displays the statistics associated with the ICAP server. More than one ICAP service may be associated with an ICAP service.
Total Reqmods (0), Total Respmods (0)	Total number of requests processed at the reqmod-precache , reqmod-postcache , or respmod-precache vector points.
Modifications (Reqmod - 0), (Respmod - 0)	Total number of requests for which the request header or request body was modified after processing at the reqmod-precache , reqmod-postcache , or respmod-precache vector points.
No Modifications (Reqmod - 0), (Respmod - 0)	Total number of requests for which the request header or request body was not modified after processing at the reqmod-precache , reqmod-postcache , or respmod-precache vector points.
Error Responses (Reqmod - 0), (Respmod - 0)	Total number of requests for which error responses were returned to the client after processing at the reqmod-precache , reqmod-postcache , or respmod-precache vector points.
Server Errors	Number of errors that occurred at the ICAP server.
Server Bypasses	Number of times the ICAP server was bypassed during request processing.

Table 2-59 *show statistics icap Field Descriptions (continued)*

Field	Description
Options Req Success	<p>Number of keepalive requests made by the Service Engine to the ICAP server that succeeded. The ICAP OPTIONS method is used by the ICAP client to retrieve configuration information from the ICAP server. In this method, the ICAP client sends a request addressed to a specific ICAP resource and receives a response with options that are specific to the service named by the URL. All OPTIONS requests may also return options that are global to the server (apply to all services). The OPTIONS method consists of a request line, such as the following example:</p> <pre>OPTIONS icap://icap.server.net/sample-service ICAP/1.0 User-Agent: ICAP-client-XYZ/1.001</pre> <p>In the following example, an ICAP Client sends an OPTIONS Request to an ICAP Service named icap.server.net/sample-service to receive configuration information for the service provided.</p> <pre>OPTIONS icap://icap.server.net/sample-service ICAP/1.0 Host: icap.server.net User-Agent: BazookaDotCom-ICAP-Client-Library/2.3</pre>
Options Req Failed	Number of keepalive requests made by the Service Engine to the ICAP server that failed.
Max Conn Available	Maximum number of connections that can be made to the ICAP server.
Used Connections	Number of connections currently established with the ICAP server.
Total Bytes sent	Total number of bytes sent to the ICAP server.
Total Bytes received	Total number of bytes received from the ICAP server.
Total BPS sent	Total number of bytes sent per second to the ICAP server.
Total BPS received	Total number of bytes received per second from the ICAP server.
Server State	Current state of the connections made to the ICAP server. This field displays Not Available, CONNECTED, or DISCONNECTED.

Related Commands icap

show statistics icmp

To display SE Internet Control Message Protocol (ICMP) statistics, use the **show statistics icmp** EXEC command.

show statistics icmp

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

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

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

Usage Guidelines	<p>ICMP messages are sent in several situations, such as when a datagram cannot reach its destination, when the gateway does not have the buffering capacity to forward a datagram, and when the gateway can direct the host to send traffic on a shorter route. The purpose of these control messages is to provide feedback about problems in the communication environment, not to make IP reliable. There is still no guarantee that a datagram will be delivered or a control message will be returned. Some datagrams may still be undelivered without any report of their loss.</p>
-------------------------	--

The ICMP messages typically report errors in the processing of datagrams. To avoid the infinite regress of messages about messages, no ICMP messages are sent about ICMP messages. Also, ICMP messages are only sent about errors in handling fragment zero of fragmented datagrams.

ICMP messages are sent using the basic IP header. The first octet of the data portion of the datagram is on a ICMP type field; the value of this field determines the format of the remaining data.

Many of the type fields contain more specific information about the error condition identified by a code value. ICMP messages have two types of codes:

- Query
- Error

Queries contain no additional information because they ask for information and will show a value of 0 in the code field. ICMP uses the queries as shown in [Table 2-60](#).

Table 2-60 *Queries*

Query	Type Field Value
Echo Reply	0
Echo Request	8
Router Advertisement	9
Router Solicitation	10
Time-stamp Request	13
Time-stamp Reply	14
Information Request (obsolete)	15

Table 2-60 *Queries (continued)*

Query	Type Field Value
Information Reply (obsolete)	16
Address Mask Request	17
Address Mask Reply	18

Error messages give specific information and have varying values that further describe conditions. Error messages always include a copy of the offending IP header and up to 8 bytes of the data that caused the host or gateway to send the error message. The source host uses this information to identify and fix the problem reported by the ICMP error message. ICMP uses the error messages as shown in [Table 2-61](#).

Table 2-61 *Errors*

Error	Type Field Value
Destination Unreachable	3
Source Quench	4
Redirect	5
Time Exceeded	11
Parameter Problems	12

Examples

[Table 2-62](#) describes the fields shown in the **show statistics icmp** display.

Table 2-62 *show statistics icmp Field Descriptions*

Field	Description
ICMP messages received	Total number of ICMP messages received by the SE.
ICMP messages receive failed	Total number of ICMP messages that were not received by the SE.
Destination unreachable	Number of destination-unreachable ICMP packets received by the SE. A destination-unreachable message (Type 1) is generated in response to a packet that cannot be delivered to its destination address for reasons other than congestion. The reason for the nondelivery of a packet is described by the code field value. Destination-unreachable packets use the code field values to further describe the function of the ICMP message being sent.

Table 2-62 *show statistics icmp Field Descriptions (continued)*


Field	Description
Timeout in transit	<p>Number of ICMP time-exceeded packets received by the SE. The time-exceeded message occurs when a router receives a datagram with a TTL of 0 or 1. IP uses the TTL field to prevent infinite routing loops. A router cannot forward a datagram that has a TTL of 0 or 1. Instead, it trashes the datagram and sends a time-exceeded message. Two different time-exceeded error codes can occur, as follows:</p> <ul style="list-style-type: none"> • 0 = Time-To-Live Equals 0 During Transit • 1 = Time-To-Live Equals 0 During Reassembly <p>A router cannot forward a datagram with a TTL of 0 or 1 both during transit or reassembly. The TTL timer is measured in seconds and originally was used before the existence of routers to guarantee that a datagram did not live on the Internet forever. Each gateway processing a datagram reduces this value by at least one if it takes longer to process and forward the datagram. When this value expires, the gateway trashes the datagram and sends a message back to the sender notifying the host of the situation.</p>
Wrong parameters	<p>Number of ICMP packets with parameter problems received by the SE. An IP datagram that has been received with the protocol field of the IP header set to 1 (ICMP) and the type field in the ICMP header set to 12 denote a parameter problem on a datagram. ICMP parameter-problem datagrams are issued when a router has had to drop a malformed datagram. This condition is a normal and necessary type of network traffic; however, large numbers of this datagram type on the network can indicate network difficulties or hostile actions. A host or gateway can send this message when no other ICMP message covering the problem can be used to alert the sending host.</p>
Source quenches	<p>Number of ICMP source-quench packets received by the SE. A receiving host generates a source-quench message when it cannot process datagrams at the speed requested due to a lack of memory or internal resources. This message serves as a simple flow control mechanism that a receiving host can use to alert a sender to slow down its data transmission. When the source host receives this message, it must pass this information on to the upper-layer process, such as TCP, which then must control the flow of the application's data stream. A router generates this message when, in the process of forwarding datagrams, it has run low on buffers and cannot queue the datagram for delivery.</p>

Table 2-62 *show statistics icmp Field Descriptions (continued)*

Field	Description
Redirects	<p>Number of ICMP redirect packets received by the SE. A router sends a redirect error to the sender of an IP datagram when the sender should have sent the datagram to a different router or directly to an end host (if the end host is local). The message assists the sending host to direct a misdirected datagram to a gateway or host. This alert does not guarantee proper delivery; the sending host has to correct the problem if possible.</p> <p>Only gateways generate redirect messages to inform source hosts of misguided datagrams. A gateway receiving a misdirected frame does not trash the offending datagram if it can forward it.</p>
Echo requests	<p>Number of echo ICMP packets received by the SE. An echo request is an IP datagram that has been received with the protocol field of the IP header set to 1 (ICMP) and the type field in the ICMP header set to 8. The ICMP echo request is issued by the source to determine if the destination is alive. When the destination receives the request, it replies with an ICMP echo reply. This request and reply pair is most commonly implemented using the ping utility. Many network management tools use this utility or some derivative of it, and this condition is common as a part of network traffic.</p> <p>Note You should be suspicious when a large number of these packets are found on the network.</p>
Echo replies	<p>Number of echo-reply ICMP packets received by the SE. An echo reply is the message that is generated in response to an echo request message. An echo reply is an IP datagram that has been received with the protocol field of the IP header set to 1 (ICMP) and the type field in the ICMP header set to 0. This condition is common as a part of network traffic.</p> <p>Note You should be suspicious when a large number of these packets are found on the network.</p>
Timestamp requests	<p>Number of ICMP time-stamp request packets received by the SE. An ICMP time-stamp request is an IP datagram that has been received with the protocol field of the IP header set to 1 (ICMP) and the type field in the ICMP header set to 13. The ICMP time-stamp request and reply pair can be used to synchronize system clocks on the network. The requesting system issues the time-stamp request bound for a destination, and the destination system responds with a time-stamp reply message. This condition is normal as a part of network traffic but is uncommon on most networks.</p> <p>Note You should be suspicious when a large number of these packets are found on the network.</p>

Table 2-62 *show statistics icmp Field Descriptions (continued)*

Field	Description
Timestamp replies	Number of ICMP time-stamp reply packets received by the SE. Time-stamp request and reply messages work in tandem. You have the option of using time stamps. When used, a time-stamp request permits a system to query another for the current time. It expects a recommended value returned to be the number of milliseconds since midnight, UTC. This message provides millisecond resolution. The two systems compare the three time stamps and use a round-trip time to adjust the sender's or receiver's time if necessary. Most systems set the transmit and receive time as the same value.
Address mask requests	<p>Number of ICMP address mask request packets received by the SE. An ICMP address mask request is an IP datagram that has been received with the protocol field of the IP header set to 1 (ICMP) and the type field in the ICMP header set to 17. ICMP address mask requests could be used to perform reconnaissance sweeps of networks. The ICMP address mask request and reply pair can be used to determine the subnet mask used on the network. When the requesting system issues the address mask request bound for a destination, the destination system responds with an address mask reply message. This condition can be a part of normal network traffic but is uncommon on most networks.</p> <p>Note You should be suspicious when a large number of these packets are found on the network.</p>
Address mask replies	<p>Number of ICMP address mask reply packets received by the SE. An address mask ICMP reply is an IP datagram that has been received with the protocol field of the IP header set to 1 (ICMP) and the type field in the ICMP header set to 18. No known exploits incorporate this option. The ICMP address mask request and reply pair can be used to determine the subnet mask used on the network. When the requesting system issues the address mask request bound for a destination, the destination system responds with an address mask reply message. This condition can be a part of normal network traffic but is uncommon on most networks.</p> <p>Note You should be suspicious when a large number of these packets are found on the network.</p>
ICMP messages sent	Total number of ICMP messages sent by the SE.
ICMP messages send failed	Total number of ICMP messages that failed to be sent by the SE.
Destination unreachable	Number of destination-unreachable ICMP packets sent by the SE.
Timeout in transit	Number of ICMP time-exceeded packets sent by the SE.
Wrong parameters	Number of ICMP packets with parameter problems sent by the SE.
Source quenches	Number of ICMP source-quench packets sent by the SE.
Redirects	Number of ICMP redirect packets sent by the SE.
Echo requests	Number of echo ICMP packets sent by the SE.

 show statistics icmp**Table 2-62** *show statistics icmp Field Descriptions (continued)*

Field	Description
Echo replies	Number of echo-reply ICMP packets sent by the SE.
Timestamp requests	Number of ICMP time-stamp request packets sent by the SE.
Timestamp replies	Number of ICMP time-stamp reply packets sent by the SE.
Address mask requests	Number of ICMP address mask requests sent by the SE.
Address mask replies	Number of ICMP address mask replies sent by the SE.

Related Commands

clear

show statistics ip

To display SE IP statistics, use the **show statistics ip** EXEC command.

show statistics ip

Syntax Description This command has no arguments or keywords.

Defaults No default behavior or values.

Command Modes EXEC

Examples [Table 2-63](#) describes the fields shown in the **show statistics ip** display.

Table 2-63 *show statistics ip Field Descriptions*

Field	Description
Total packets in	Total number of input datagrams received from interfaces, including those received in error.
with invalid header	Number of input datagrams discarded due to errors in their IP headers, including bad checksums, version number mismatch, other format errors, Time To Live exceeded, errors discovered in processing their IP options, and so on.
with invalid address	Number of input datagrams discarded because the IP address in the IP header's destination field was not a valid address to be received at this entity. This count includes invalid addresses (for example, 0.0.0.0) and addresses of unsupported classes (for example, Class E). For entities that are not IP routers and do not forward datagrams, this counter includes datagrams discarded because the destination address was not a local address.
forwarded	Number of input datagrams for which this entity was not the final IP destination, but the SE attempted to find a route to forward them to that final destination. In entities that do not act as IP routers, this counter will include only those packets that were source-routed through this entity, and the source-route option processing was successful.
unknown protocol	Number of locally addressed datagrams received successfully but discarded because of an unknown or unsupported protocol.
discarded	Number of input IP datagrams that were discarded even though the datagrams encountered no problems to prevent their continued processing. This counter does not include any datagrams discarded while awaiting reassembly.
delivered	Total number of input datagrams successfully delivered to IP user protocols (including ICMP).

Table 2-63 *show statistics ip Field Descriptions (continued)*

Field	Description
Total packets out	Total number of IP datagrams that local IP user protocols (including ICMP) supplied to IP in requests for transmission. This counter does not include any datagrams counted in the forwarded field.
dropped	Number of output IP datagrams that were discarded even though the datagrams encountered no problems that would prevent their transmission to their destination. This counter would include datagrams counted in the forwarded field if any such packets met this (discretionary) discard criterion.
dropped (no route)	Number of IP datagrams that were discarded because the SE found no route to transmit them to their destination. This counter includes any packets counted in the forwarded field that meet this no-route criterion including any datagrams that a host cannot route because all of its default routers are down.
Fragments dropped after timeout	Number of received fragments at this entity that are dropped after being held for the maximum number of seconds while awaiting reassembly at this entity.
Reassemblies required	Number of IP fragments received that needed to be reassembled at this entity.
Packets reassembled	Number of IP datagrams successfully reassembled.
Packets reassemble failed	Number of failures detected by the IP reassembly algorithm (because of reasons such as timed out and errors.) This counter is not necessarily a count of discarded IP fragments because some algorithms (notably the algorithm in RFC 815) can lose track of the number of fragments by combining them as they are received.
Fragments received	Number of IP datagrams that have been successfully fragmented at this entity.
Fragments failed	Number of IP datagrams that have been discarded because they needed to be fragmented at this entity but could not be fragmented for reasons such as the Don't Fragment flag was set.
Fragments created	Number of IP datagram fragments that have been generated as a result of fragmentation at this entity.

Related Commands

clear statistics ip
ip
show ip routes

show statistics movie-streamer

To display statistics for the Movie Streamer, use the **show statistics movie-streamer EXEC** command.

show statistics movie-streamer {all | bw-usage | error | performance | requests | rule}

Syntax Description	all	Displays all statistics.
	bw-usage	Displays bandwidth usage statistics.
	error	Displays error statistics.
	performance	Displays server performance.
	requests	Displays request statistics.
	rule	Displays rule statistics.

Defaults No default behavior or values.

Command Modes EXEC

Examples The following example shows all the Movie Streamer statistics:

```
ServiceEngine#show statistics movie-streamer all
```

```
Movie Streamer Request Statistics
Total
```

```
-----
```

```
Current RTSP Sessions: 3400
Total RTSP Sessions: 283299
Current RTP Connections: 2739
Total RTP Connections: 282885
```

```
CDN Related Statistics
```

```
-----
```

```
Preposition Hits: 0
Cache Hits: 0
Cache Miss: 0
Live Requests: 283299
```

```
Cache Revalidation Statistics
```

```
-----
```

```
Fresh Content Requests: 0
Revalidated Requests: 0
```

```
Movie Streamer Bandwidth Usage Statistics
```

```
Total
```

```
-----
```

```
Current Incoming Bandwidth: 0 bps
Current Outgoing Bandwidth: 3921755 bps
Current Total Bandwidth: 3921755 bps
```

```
Average Incoming Bandwidth: 475217 bps
Average Outgoing Bandwidth: 13038460 bps
Average Total Bandwidth: 13513677 bps
```

show statistics movie-streamer

```

By Type of Connection
-----
Unicast Incoming Bandwidth: 0 bps
Multicast Incoming Bandwidth: 0 bps
Unicast Outgoing Bandwidth: 3816953 bps
Multicast Outgoing Bandwidth: 0 bps

By Type of Content
-----
Live Incoming Bandwidth: 0 bps
VOD Incoming Bandwidth: 0 bps
Live Outgoing Bandwidth: 3816953 bps
VOD Outgoing Bandwidth: 0 bps

Overall Traffic
-----
Incoming Bytes: 709316834819 Bytes
Outgoing Bytes: 62627648126402 Bytes
Total Bytes: 63336964961221 Bytes

Incoming Packets: 652577871
Outgoing Packets: 191008363529
Total Packets: 191660941400

Movie Streamer Error Statistics
Total
Server Error
-----
Internal Error: 0
Not Implemented: 0
Server Unavailable: 0
Gateway Timeout: 0
Others: 0

Client Error
-----
Bad Request: 0
File Not Found: 6
Session Not Found: 0
Method Not Allowed: 0
Not Enough Bandwidth: 0
Client Forbidden: 0
Others: 0

Movie Streamer Performance Statistics
Total
-----
CPU Usage: 0.166702 %
Uptime: 254328 sec
Statistics was last cleared on Monday, 18-May-2009 20:04:42 UTC.

```

The following example shows the Movie Streamer rule statistics:

```

ServiceEngine#show statistics movie-streamer rule
RTSP Rule Template Statistics
=====
URL Rewrite: 0
URL Block: 0
Allow: 0
Redirect: 0

```

Related Commands

movie-streamer
show movie-streamer

show statistics netstat

To display SE Internet socket connection statistics, use the **show statistics netstat** EXEC command.

show statistics netstat

Syntax Description This command has no arguments or keywords.

Defaults No default behavior or values.

Command Modes EXEC

Examples [Table 2-64](#) describes the fields shown in the **show statistics netstat** display.

Table 2-64 *show statistics netstat Field Descriptions*

Field	Description
Proto	Layer 4 protocol used on the Internet connection, such as TCP, UDP, and so forth.
Recv-Q	Amount of data buffered by the Layer 4 protocol stack in the receive direction on a connection.
Send-Q	Amount of data buffered by the Layer 4 protocol stack in the send direction on a connection.
Local Address	IP address and Layer 4 port used at the device end point of a connection.
Foreign Address	IP address and Layer 4 port used at the remote end point of a connection.
State	Layer 4 state of a connection. TCP states include the following: ESTABLISHED, TIME-WAIT, LAST-ACK, CLOSED, CLOSED-WAIT, SYN-SENT, SYN-RCVD, SYN-SENT, SYN-ACK-SENT, and LISTEN.

show statistics qos

To display statistics for the QoS policy service, use the **show statistics qos** EXEC command.

show statistics qos policy-service

Syntax Description	policy-service	Displays statistics of Camiant cdn-am policy service
--------------------	----------------	--

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

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

Examples The following example displays the statistics for the QoS policy service:

```
ServiceEngine#show statistics qos policy-service
Camiant CDN-AM Policy Service Statistics
-----

Application : WMT
  Protocol  : RTSP
    PLAY   : 0
    PAUSE  : 0
    STOP   : 0

  Protocol  : HTTP
    PLAY   : 0
    PAUSE  : 0
    STOP   : 0

Application : WEB-ENGINE
  Protocol  : HTTP
    PLAY   : 0
    PAUSE  : 0
    STOP   : 0

Errors : 0
```

Related Commands	qos show qos
------------------	-------------------------------

show statistics radius

To display SE RADIUS authentication statistics, use the **show statistics radius** EXEC command.

show statistics radius

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

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

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

Examples	<p>The fields in the show statistics radius display are as follows:</p> <ul style="list-style-type: none">• Number of access requests• Number of access deny responses• Number of access allow responses• Number of authorization requests• Number of authorization failure responses• Number of authorization success responses
-----------------	--

Related Commands	<p>clear radius-server show radius-server</p>
-------------------------	--

show statistics replication

To display delivery service replication status and related statistical data, use the following **show statistics replication** EXEC command.

On the CDSM:

```
show statistics replication { content-items { selected-delivery-service delivery-service-name } |
delivery-service [ selected-delivery-service delivery-service-name ] | item url | service engines
{ selected-delivery-service delivery-service-name } }
```

On the SE:

```
show statistics replication { content-items content-name | selected-delivery-service
delivery-service-name }
```

Syntax Description		
content-items		Displays the replication status of the specified content items.
<i>content-name</i>		Content item name or pattern including an asterisk (*) and question mark (?). Use an asterisk to select all content items.
selected-delivery-service		Selects a delivery service.
<i>delivery-service-name</i>		Delivery service name.
delivery-service		Displays replication status of the delivery service.
item		Displays the detailed replication status of a content item across all SEs in a delivery service.
<i>url</i>		URL of the content item.
service-engines		Displays the replication status of the specified SEs.

Defaults No default behavior or values.

Command Modes EXEC

Usage Guidelines The **show statistics replication** command displays the delivery service replication status on the CDSM and the SE and shows the progressive file count status during acquisition and replication.

Examples [Table 2-65](#) describes the fields shown in the **show statistics replication** displays.

Table 2-65 show statistics replication Field Descriptions

Field	Description
Delivery service	Delivery service name.
State	Overall state of the delivery service. Values are Complete or Failed.
User Selected Root SE	Name of the root SE that has been selected for delivery service.

Table 2-65 *show statistics replication Field Descriptions (continued)*

Field	Description
Current Root SE	Name of the currently acting root SE for the delivery service.
Receiver SEs Completed	Total number of SEs that have completed content replication for the delivery service.
Receiver SEs In Progress	Total number of SEs for which content replication is in progress for the delivery service.
Receiver SEs Failed	Total number of SEs that have some error condition and are treated as failed.
Receiver SEs Not Responding	Total number of SEs not responding to the replication status queries from the CDSM.
Device	Name and ID of the device.
Website	Name of the website used for the delivery service.
Type	Role of the device, such as Root or Receiver.
State	State of the SE replication. For receiver SEs, states are Failed, Replicating, or Completed. For the root SE, states are Acquiring Content, Rechecking Content, or Completed.
Status	Replication status. Values are Red for failure and Green for success.
Completed	Number of content items completed.
To Do	Number of content items pending for the delivery service.
Failed	Number of failed content items.
Total	Total number of content items.
Last Report Time	Time that this status was obtained.
Disk Quota Used	Total disk quota used for the delivery service.
Manifest Last Modified	Time at which the manifest file was last modified.
Manifest Last Check	Time at which the manifest file was last checked for freshness.
Manifest State	State of the manifest. Values are Complete or Error with details of the error displayed.

show statistics service-router

To display service router statistics, use the **show statistics service-router** EXEC command.

```
show statistics service-router {all | content-origin name | dns | history | keepalive | se name |  
summary}
```

Syntax Description		
	all	Displays all statistics.
	content-origin	Displays content origin specific statistics.
	<i>name</i>	Content origin name to show.
	dns	Displays DNS statistics.
	history	Displays statistics history.
	keepalive	Displays keepalive statistics.
	se	Displays Service Engine specific statistics.
	<i>name</i>	Service Engine name to show.
	summary	Displays summary statistics.

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

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

Related Commands	service-router show service-router
-------------------------	---

show statistics services

To display SE services statistics, use the **show statistics services** EXEC command.

show statistics services

Syntax Description This command has no arguments or keywords.

Defaults No default behavior or values.

Command Modes EXEC

Examples [Table 2-66](#) describes the fields shown in the **show statistics services** display.

Table 2-66 *show statistics services Field Descriptions*

Field	Description
Port Statistics	Service-related statistics for each port on the Wide Area Application Services (WAAS) device.
Port	Port number.
Total Connections	Number of total connections.

Related Commands **show services**

show statistics snmp

To display SE Simple Network Management Protocol (SNMP) statistics, use the **show statistics snmp** EXEC command.

show statistics snmp

Syntax Description This command has no arguments or keywords.

Defaults No default behavior or values.

Command Modes EXEC

Examples [Table 2-67](#) describes the fields shown in the **show statistics snmp** display.

Table 2-67 *show statistics snmp Field Descriptions*

Field	Description
SNMP packets input	Total number of SNMP packets input.
Bad SNMP version errors	Number of packets with an invalid SNMP version.
Unknown community name	Number of SNMP packets with an unknown community name.
Illegal operation for community name supplied	Number of packets requesting an operation not allowed for that community.
Encoding errors	Number of SNMP packets that were improperly encoded.
Number of requested variables	Number of variables requested by SNMP managers.
Number of altered variables	Number of variables altered by SNMP managers.
Get-request PDUs	Number of GET requests received.
Get-next PDUs	Number of GET-NEXT requests received.
Set-request PDUs	Number of SET requests received.
SNMP packets output	Total number of SNMP packets sent by the router.
Too big errors	Number of SNMP packets that were larger than the maximum packet size.
Maximum packet size	Maximum size of SNMP packets.
No such name errors	Number of SNMP requests that specified a MIB object that does not exist.
Bad values errors	Number of SNMP SET requests that specified an invalid value for a MIB object.

Table 2-67 *show statistics snmp Field Descriptions (continued)*

Field	Description
General errors	Number of SNMP SET requests that failed because of some other error. (It was not a No such name error, Bad values error, or any of the other specific errors.)
Response PDUs	Number of responses sent in reply to requests.
Trap PDUs	Number of SNMP traps sent.

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 notify
snmp-server user

show statistics tacacs

To display Service Engine TACACS+ authentication and authorization statistics, use the **show statistics tacacs** EXEC command.

show statistics tacacs

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

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

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

Examples	The fields shown in the show statistics tacacs display for the service engine are as follows:
-----------------	--

- Number of access requests
- Number of access deny responses
- Number of access allow responses
- Number of authorization requests
- Number of authorization failure responses
- Number of authorization success responses
- Number of accounting requests
- Number of accounting failure responses
- Number of accounting success responses

Related Commands	clear show tacacs tacacs
-------------------------	---

show statistics tcp

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

show statistics tcp

Syntax Description This command has no arguments or keywords.

Defaults No default behavior or values.

Command Modes EXEC

Examples [Table 2-68](#) describes the fields shown in the **show statistics tcp** display.

Table 2-68 *show statistics tcp Field Descriptions*

Field	Description
Server connection openings	Number of connections opened from the SE to the server.
Client connection openings	Number of connections opened from the client to the SE.
Failed connection attempts	Number of incoming SYN connections rejected due to rate limiting or resource shortage.
Connections established	Number of incoming connections that have been set up.
Connections resets received	Number of resets (RSTs) received by the SE.
Connection resets sent	Number of resets (RSTs) sent by the SE.
Segments received	Number of TCP segments received from the client and the server. The value of this field is almost equal to the sum of the values of the Server segments received and the Client segments received fields.
Segments sent	Number of TCP segments sent by the client and the server. The value of this field is almost equal to the sum of the values of the Server segments sent and the Client segments sent fields.
Bad segments received	Number of incoming segments dropped due to checksum or being outside the TCP window.
Segments retransmitted	Number of TCP segments retransmitted by the client and the server. The value of this field is almost equal to the sum of the values of the Server segments retransmitted and the Client segments retransmitted fields.

Table 2-68 *show statistics tcp Field Descriptions (continued)*

Field	Description
Retransmit timer expirations	Number of times that the TCP retransmit timer expires. The TCP sender uses a timer to measure the time that has elapsed between sending a data segment and receiving the corresponding ACK from the receiving side of the TCP transmission. When this retransmit timer expires, the sender (according to the RFC standards for TCP congestion control) must reduce its sending rate.
Server segments received	Number of TCP segments received by the SE from the server.
Server segments sent	Number of TCP segments sent by the SE to the server.
Server segments retransmitted	Number of TCP segments retransmitted by the SE from the server.
Client segments received	Number of TCP segments received by the SE from the client.
Client segments sent	Number of TCP segments sent by the SE to the server.
Client segments retransmitted	Number of TCP segments retransmitted by the SE to the client.
Sync cookies sent	Number of synchronized (SYN) cookies sent by the SE. TCP requires unacknowledged data to be retransmitted. The server is supposed to retransmit the SYN.ACK packet before giving up and dropping the connection. When SYN.ACK arrives at the client but the ACK gets lost, there is a disparity about the establishment state between the client and server. Typically, this problem can be solved by the server's retransmission. But in the case of a SYN cookie, there is no state kept on the server and retransmission is impossible.
Sync cookies received	Number of synchronized (SYN) cookies received by the SE. The entire process of establishing the connection is performed by the ACK packet sent by the client, making the connection process independent of the preceding SYN and SYN.ACK packets. This type of connection establishment opens the possibility of ACK flooding, in the hope that the client will have the correct value to establish a connection. This method also allows you to bypass firewalls that normally only filter packets with SYN bit set.
Sync cookies failed	Number of synchronized (SYN) cookies rejected by the SE. The SYN cookies feature attempts to protect a socket from a SYN flood attack. This feature is a violation of TCP and conflicts with other areas of TCP such as TCP extensions. It can cause problems for clients and relays. We do not recommend that you use this feature as a tuning mechanism for heavily loaded servers to help with overloaded or misconfigured conditions.
Embryonic connection resets	Number of TCP connections that have been reset before the SE accepted the connection.
Prune message called	Number of calls that the SE makes to the function that tries to reduce the number of received but not acknowledged packets.

Table 2-68 *show statistics tcp Field Descriptions (continued)*

Field	Description
Packets pruned from receive queue	Number of packets that the TCP drops from the receive queue (usually due to low memory).
Out-of-order-queue pruned	Number of times that the packet was dropped from the out-of-order queue.
Out-of-window Icmp messages	Number of ICMP packets that were outside the TCP window and dropped.
Lock dropped Icmp messages	Number of ICMP packets that hit a locked (busy) socket and were dropped.
Arp filter	Number of Address Resolution Protocol (ARPs) not sent because they were meant for the SE.
Time-wait sockets	Number of current sockets in the TIME-WAIT state. The TIME-WAIT state removes old duplicates for fast or long connections. The clock-driven ISN selection is unable to prevent the overlap of the old and new sequence spaces. The TIME-WAIT delay allows enough time for all old duplicate segments to die in the Internet before the connection is reopened.
Time-wait sockets recycled	Number of TIME-WAIT sockets that were recycled (the address or port was reused before the waiting period was over). In TCP, the TIME-WAIT state is used as protection against old duplicate segments
Time-wait sockets killed	Number of TIME-WAIT sockets that were terminated to reclaim memory.
PAWS passive	Number of passive connections that were made with Protection Against Wrapped Sequence (PAWS) numbers enabled. PAWS operates within a single TCP connection using a state that is saved in the connection control block.
PAWS active	Number of active connections that were made with PAWS enabled. PAWS uses the same TCP time stamps as the round-trip time measurement mechanism and assumes that every received TCP segment (including the data and ACK segments) contains a time-stamp SEG.TSval that has values that are monotone and nondecreasing in time. A segment can be discarded as an old duplicate if it is received with a time-stamp SEG.TSval less than some time stamp recently received on this connection.
PAWS established	Number of current connections that were made with PAWS enabled.
Delayed acks sent	Number of delayed ACK counters sent by the SE.
Delayed acks blocked by socket lock	Number of delayed ACK counters that were blocked because the socket was in use.
Delayed acks lost	Number of delayed ACK counters lost during transmission.

Table 2-68 *show statistics tcp Field Descriptions (continued)*

Field	Description
Listen queue overflows	Number of times that the three-way TCP handshake was completed, but enough space was not available in the listen queue.
Connections dropped by listen queue	Number of TCP connections dropped due to a resource shortage.
TCP packets queued to prequeue	Number of TCP packets queued to the prequeue.
TCP packets directly copied from backlog	Number of TCP packets delivered to the client from the backlog queue. Packets are queued in the backlog when the TCP receive routine runs and notices that the socket was locked.
TCP packets directly copied from prequeue	Number of TCP packets delivered to the client from the prequeue.
TCP prequeue dropped packets	Number of TCP packets dropped from the prequeue. The prequeue is where the TCP receives routine runs. It notes that the current running process as the TCP target process and queues it directly for copy after the TCP software interrupt is completed.
TCP header predicted packets	Number of incoming packets that successfully matched the TCP header prediction.
Packets header predicted and queued to user	Number of TCP packets copied directly to the user space.
TCP pure ack packets	Number of acknowledgment (ACK) packets that contain no data.
TCP header predicted acks	Number of incoming ACKs that successfully matched the TCP header prediction.
TCP Reno recoveries	Number of times that the TCP fast recovery algorithm recovered a packet loss. TCP Reno induces packet losses to estimate the available bandwidth in the network. When there are no packet losses, TCP Reno continues to increase its window size by one during each round trip. When it experiences a packet loss, it reduces its window size to one half of the current window size. This feature is called additive increase and multiplicative decrease. TCP Reno, however, does not fairly allocate bandwidth because TCP is not a synchronized rate-based control scheme, which is necessary for the convergence.
TCP SACK recoveries	Number of times that the SE recovered from a selective acknowledgment (SACK) packet loss. If the data receiver has received a SACK-Permitted option on the SYN for this connection, the data receiver may choose to generate SACK options. If the data receiver generates SACK options under any circumstance, it should generate them under all permitted circumstances. If the data receiver has not received a SACK-Permitted option for a given connection, it must not send SACK options on that connection.

Table 2-68 *show statistics tcp Field Descriptions (continued)*

Field	Description
TCP SACK renegeing	<p>Number of times that the SE refused to accept packets that have not been acknowledged to the data sender, even if the data has already been reported in a SACK option. Such discarding of SACK packets is discouraged but may be used if the receiver runs out of buffer space. The data receiver may choose not to keep data that it has reported in a SACK option.</p> <p>Because the data receiver may later discard data reported in a SACK option, the sender must not discard data before it is acknowledged by the Acknowledgment Number field in the TCP header.</p>
TCP FACK reorders	Number of forward acknowledgment (FACK) packets that were out of sequence order. The FACK algorithm makes it possible to treat congestion control during recovery in the same manner as during other parts of the TCP state space. The FACK algorithm is based on first principles of congestion control and is designed to be used with the proposed TCP SACK option. By decoupling congestion control from other algorithms, such as data recovery, it attains more precise control over the data flow in the network. FACK takes advantage of the SACK option; it takes into account which segments have been SACKed. It also uses the receipt of a SACK that leaves at least 3*MSS bytes unacknowledged as a trigger for Fast Retransmit.
TCP SACK reorders	Number of selective acknowledgment (SACK) packets that were out of sequence order.
TCP Reno reorders	Number of TCP Renos that were out of sequence order.
TCP TimeStamp reorders	Number of segments received with out-of-order time stamps.
TCP full undos	Number of times that the congestion window (cwnd) was fully recovered.
TCP partial undos	Number of times that the congestion window (cwnd) was partially recovered.
TCP DSACK undos	Number of times that the duplicate selective acknowledgment (DSACK) packets were recovered.
TCP loss undos	Number of times that the congestion window (cwnd) recovered from a packet loss.
TCP losses	Number of times that data was lost and the size of the congestion window (cwnd) decreased.
TCP lost retransmit	Number of times that a retransmitted packet was lost.

Table 2-68 *show statistics tcp Field Descriptions (continued)*

Field	Description
TCP Reno failures	Number of times that the congestion window (cwnd) failed because the TCP fast recovery algorithm failed to recover from a packet loss. The congestion avoidance mechanism, which is adopted by TCP Reno, causes the window size to vary. This situation causes a change in the round-trip delay of the packets, larger delay jitter, and an inefficient use of the available bandwidth due to many retransmissions of the same packets after the packet drops occur. The rate at which each connection updates its window size depends on the round-trip delay of the connection. The connections with shorter delays can update their window sizes faster than other connections with longer delays.
TCP SACK failures	Number of times that the congestion window (cwnd) shrunk because the SE failed to recover from a selective acknowledgment (SACK) packet loss. The selective acknowledgment extension uses two TCP options. The first is an enabling option, SACK-permitted, which may be sent in a SYN segment to indicate that the SACK option can be used once the connection is established. The other is the SACK option, which may be sent over an established connection once permission has been given by the SACK-permitted option.
TCP loss failures	Number of times that the TCP timeout occurred and data recovery failed.
TCP fast retransmissions	Number of TCP fast retransmission counters. TCP may generate an immediate acknowledgment (a duplicate ACK) when an out-of-order segment is received. The duplicate ACK lets the other end know that a segment was received out of order and tells it what sequence number is expected. Because TCP does not know whether a duplicate ACK is caused by a lost segment or just a reordering of segments, it waits for a small number of duplicate ACKs to be received. If there is just a reordering of the segments, there will be only one or two duplicate ACKs before the reordered segment is processed, which will then generate a new ACK. If three or more duplicate ACKs are received in a row, it is a strong indication that a segment has been lost. TCP then retransmits what appears to be the missing segment without waiting for a retransmission timer to expire.

Table 2-68 *show statistics tcp Field Descriptions (continued)*

Field	Description
TCP forward retransmissions	<p>Number of TCP forward retransmission counters. This field applies only to SACK-negotiated connections; this field is the counter for FACK segments. The value of this field is for segments that were retransmitted even though there is no indication that they were actually lost. Retransmission is stopped when either one of the following occurs:</p> <ul style="list-style-type: none"> • The maximum time to wait for a remote response is reached. This timeout occurs when the total time of all retransmission intervals exceeds the maximum time to wait for a remote response. • The number of retransmissions configured in maximum retransmissions per packet is reached.
TCP slowstart retransmissions	<p>Number of TCP slow-start retransmission counters. The slow-start algorithm begins by sending packets at a rate that is determined by the congestion window. The algorithm continues to increase the sending rate until it reaches the limit set by the slow-start threshold (ssthresh) variable. (Initially, the value of the ssthresh variable is adjusted to the receiver's maximum window size [RMSS]. However, when congestion occurs, the ssthresh variable is set to half the current value of the cwnd variable, marking the point of the onset of network congestion for future reference.)</p>
TCP Timeouts	Number of times that a TCP timeout occurred.
TCP Reno recovery fail	<p>Number of times that the TCP fast recovery algorithm failed to recover from a packet loss. In TCP Reno, the maximum number of recoverable packet losses in a congestion window without timeout is limited to one or two packets. No more than six losses can be recovered with a maximum window size of 128 packets. This failure of recovery is because TCP Reno cuts the congestion window by half for each recovered loss.</p>
TCP Sack recovery fail	<p>Number of times that the SE failed to recover from a SACK packet loss. When receiving an ACK containing a SACK option, the data sender should record the selective acknowledgment for future reference. The data sender is assumed to have a retransmission queue that contains the segments that have been transmitted but not yet acknowledged in sequence number order. If the data sender performs repacketization before retransmission, the block boundaries in a SACK option that it receives may not fall within the boundaries of segments in the retransmission queue.</p>
TCP scheduler failed	Number of times that the TCP scheduler failed.
TCP receiver collapsed	Number of times that the data in an out-of-order queue collapsed.

Table 2-68 *show statistics tcp Field Descriptions (continued)*

Field	Description
TCP DSACK old packets sent	Number of duplicate selective acknowledgments (D-SACKs) sent by the SE. The use of D-SACK does not require a separate negotiation between a TCP sender and receiver that have already negotiated SACK. The absence of a separate negotiation for D-SACK means that the TCP receiver could send D-SACK blocks when the TCP sender does not understand this extension to SACK. In this case, the TCP sender discards any D-SACK blocks and processes the other SACK blocks in the SACK option field as it normally would.
TCP DSACK out-of-order packets sent	Number of out-of-order D-SACK packets sent by the SE. A D-SACK block is used only to report a duplicate contiguous sequence of data received by the receiver in the most recent packet. Each duplicate contiguous sequence of data received is reported in at most one D-SACK block. (The receiver sends two identical D-SACK blocks in subsequent packets only if the receiver receives two duplicate segments.) If the D-SACK block reports a duplicate contiguous sequence from a (possibly larger) block of data in the receiver's data queue above the cumulative acknowledgement, then the second SACK block in that SACK option should specify that (possibly larger) block of data.
TCP DSACK packets received	Number of D-SACK packets received by the SE. TCP senders receiving D-SACK blocks should be aware that a segment reported as a duplicate segment could possibly have been from a prior cycle through the sequence number space. This awareness of the TCP senders is independent of the use of PAWS by the TCP data receiver.
TCP DSACK out-of-order packets received	Number of out-of-order D-SACK packets received by the SE. Following a lost data packet, the receiver receives an out-of-order data segment, which triggers the SACK option as specified in RFC 2018. Because of several lost ACK packets, the sender then retransmits a data packet. The receiver receives the duplicate packet and reports it in the first D-SACK block.
TCP connections abort on sync	Number of times that a valid SYN segment was sent in the TCP window and the connection was reset.
TCP connections abort on data	Number of times that the connection closed after reading the data.
TCP connections abort on close	Number of times that the connection aborted with pending data.
TCP connections abort on memory	Number of times that memory was not available for graceful closing of the connection resulting in the connection being aborted immediately.
TCP connections abort on timeout	Number of times that the connection timed out.
TCP connections abort on linger	Number of times that the linger timeout expired resulting in the data being discarded and closing of the connection.

Table 2-68 *show statistics tcp Field Descriptions (continued)*

Field	Description
TCP connections abort failed	Number of times that the TCP connection ran out of memory, transmits failed, or peer TCP Reset (RST) could not be sent.
TCP memory pressures	Number of times that the TCP subsystem encounters memory constraints.

Related Commands

clear
show tcp
tcp

show statistics transaction-logs

To display SE transaction log export statistics, use the **show statistics transaction-logs EXEC** command.

show statistics transaction-logs

Syntax Description This command has no arguments or keywords.

Defaults No default behavior or values.


Command Modes EXEC

Usage Guidelines To display the transaction log export statistics, you must first configure the FTP server.

Examples [Table 2-69](#) describes the fields shown in the **show statistics transaction-logs** display.

Table 2-69 *show statistics transaction-logs Field Descriptions*

Field	Description
Initial Attempts	Initial attempts made to contact the external server at the configured export intervals.
Initial Successes	Number of times that an initial attempt made to contact the external server succeeded.
Initial Open Failures	Number of times that the SE failed to open a connection to the FTP export server.
Initial Put Failures	Number of times that the SE failed to transfer a file to the FTP export server.
Retry Attempts	Number of retries made to contact the external server at the configured export intervals.
Retry Successes	Number of times that a retry made to contact the external server succeeded.
Retry Open Failures	Number of times that the SE failed to open a connection to the FTP export server on a retry.
Retry Put Failures	Number of times that the SE failed to transfer a file to the FTP export server on a retry.

 show statistics transaction-logs**Table 2-69** *show statistics transaction-logs Field Descriptions (continued)*

Field	Description
Authentication Failures	Number of times that the SE failed to authenticate with the FTP export server. This situation might occur if the SE is misconfigured with the wrong password for the FTP server or the password on the FTP server has been changed since the SE was configured.
Invalid Server Directory Failures	Number of times the SE failed to direct traffic to the correct server directory.

Related Commands

clear
show transaction-logging
transaction-log force

show statistics udp

To display SE 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 2-70](#) describes the fields shown in the **show statistics udp** display.

Table 2-70 *show statistics udp Field Descriptions*

Field	Description
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 wmt

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

show statistics wmt { **all** | **bytes** [**incoming** | **outgoing**] | **cache** | **errors** | **multicast** | **requests** | **rule** | **savings** | **streamstat** [**incoming** | **live** | **outgoing** | **stream-id** *1-999999*] | **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.
cache		Displays cache validation 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 SE.
live		(Optional) Displays aggregated live stream statistics.
outgoing		(Optional) Displays statistics of all outgoing WMT streams from the SE.
stream-id		(Optional) Displays statistics of the WMT streams that have the specified stream ID.
<i>1-999999</i>		WMT stream ID to display.
usage		Displays current usage statistics.

Defaults No default behavior or values.

Command Modes EXEC

Usage Guidelines The output of the **show statistics wmt** EXEC command includes information about WMT RTSP requests. For example, the output from the **show statistics wmt** EXEC command 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.

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.

Configuring the HTTP Allow/Block Rule

For the MMS over HTTP request rule, even though the request is served by WMT, it doesn't increment the statistics. The user needs the statistics for all WMT requests. Now the user can execute the **show statistics http rule** command as the rules daemon check is done from the HTTP side, and the request is redirected to WMT.

Examples

Table 2-71 describes the fields shown in the **show statistics wmt all** display.

Table 2-71 *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.

Table 2-71 *show statistics wmt all Field Descriptions (continued)*

Field	Description
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.
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.
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.

Table 2-71 *show statistics wmt all Field Descriptions (continued)*

Field	Description
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	
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.

```
show statistics wmt
```

Table 2-71 *show statistics wmt all Field Descriptions (continued)*

Field	Description
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.
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.
URL Redirect	Number of URL redirects.
URL Block	Number of blocked URLs.
No-Cache	Number of no-cache matches.
Allow	Number of allow matches.
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.

Related Commands

```
clear
show wmt
wmt
```

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 Service Engine. [Table 2-72](#) describes the fields shown in the **show tacacs** display.

Table 2-72 *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 Service 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 Service Engine.
Key	Secret key that the Service 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 Service Engine waits for a response from the specified TACACS+ authentication server before declaring a timeout.
Retransmit	Number of times that the Service Engine is to retransmit its connection to the TACACS+ server 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 2-72 *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
show statistics tacacs
tacacs

show tech-support

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

show tech-support list-files *directory name*

show tech-support [**page**]

show tech-support service authentication [**acquisition-distribution** | **cms** | **flash-media-streaming** | **http** | **icap** | **kernel** | **movie-streamer** | **rules** | **wmt**]

show tech-support service cms [**acquisition-distribution** | **authentication** | **flash-media-streaming** | **http** | **icap** | **kernel** | **movie-streamer** | **rules** | **wmt**]

show tech-support service flash-media-streaming [**acquisition-distribution** | **authentication** | **cms** | **http** | **icap** | **kernel** | **movie-streamer** | **rules** | **wmt**]

show tech-support service http [**acquisition-distribution** | **authentication** | **cms** | **flash-media-streaming** | **icap** | **kernel** | **movie-streamer** | **rules** | **wmt**]

show tech-support service icap [**acquisition-distribution** | **authentication** | **cms** | **flash-media-streaming** | **http** | **kernel** | **movie-streamer** | **rules** | **wmt**]

show tech-support service kernel [**acquisition-distribution** | **authentication** | **cms** | **flash-media-streaming** | **http** | **icap** | **movie-streamer** | **rules** | **wmt**]

show tech-support service movie-streamer [**acquisition-distribution** | **authentication** | **cms** | **flash-media-streaming** | **http** | **icap** | **kernel** | **rules** | **wmt**]

show tech-support service rules [**acquisition-distribution** | **authentication** | **cms** | **flash-media-streaming** | **http** | **icap** | **kernel** | **movie-streamer** | **wmt**]

show tech-support service wmt [**acquisition-distribution** | **authentication** | **cms** | **flash-media-streaming** | **http** | **icap** | **kernel** | **movie-streamer** | **rules**]

Syntax Description

list-files	(Optional) Displays the list of files under a directory.
<i>directory name</i>	Directory name (use absolute path, such as /local1/logs).
page	(Optional) Specifies the pages through the output.
service	(Optional) Displays technical support information specific to a service.
authentication	Displays technical support information related to HTTP authentication.
acquisition-distribution	Displays technical support information related to acquisition and distribution.
cms	Displays technical support information related to CMS.
flash-media-streaming	Displays technical support information related to Flash media streaming.
http	Displays technical support information related to HTTP.
icap	Displays technical support information related to ICAP.
kernel	Displays technical support information related to the kernel.
movie-streamer	Displays technical support information related to the Movie Streamer.

rules	Displays technical support information related to rules.
wmt	Displays technical support information related to WMT.

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 SE. We recommend that you log the output to a disk file. Use the streaming option to view information specific to the streaming feature.

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](#))
- Running configuration ([show running-config](#))
- Processes ([show processes](#))
- Process memory ([show processes memory](#))
- System memory
- File system information
- Interface information
- 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 is as follows:

- CPU or memory processes ([show programs](#))
- WMT streaming connections ([show statistics wmt streamstat](#))
- Bandwidth allocation ([show bandwidth](#))
- Bit rate allocation ([show bitrate](#))
- Acquirer information ([show acquirer](#))
- Rules ([show rule all](#))
- Distribution channel details

Information Specific to WMT

Information that is specific to WMT is as follows:

- WMT bandwidth and proxy mode configuration ([show wmt](#))

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

Information Specific to RTSP

Information that is specific to RTSP is as follows:

- RTSP configuration ([show rtsp](#))

Examples

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

```
ServiceEngine#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_ah_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)
1207 S      0      0      0 (parser_server)
1208 S      0      0      0 (eval_timer_mana)
1211 S      0     46      1 (parser_server)
1443 S      0      0      0 (overload)
1444 S      0      0      0 (standby)
1445 S      0     13     29 (cache)
1446 S      0      0      0 (proxy_poll)
1447 S      0      0      0 (snmpcd)
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     61 (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)
```

show tech-support

```

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	(bdflush)
6	S	0	0.0	0	0	(kupdated)
7	S	0	0.0	0	0	(scsi_eh_0)
45	S	0	0.0	1208320	143	(nodemgr)
46	S	0	0.0	1630208	194	(syslogd)
47	R	0	0.0	1974272	238	(dataserver)
920	S	1088	0.0	1728512	236	(login)
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)
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 (remotereq)
2185      S      0 0.3    8269824      873 (videosvr)
2188      S      0 0.4    7651328    1196 (contentsvr)
2189      S      0 0.3    6103040      953 (routeraux)
2190      S      0 0.4    10272768    1075 (dfcontrolsrvr)
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: GigabitEthernet Slot: 0 Port: 0
Type:Ethernet
Ethernet address:00:05:32:02:DD:74
Internet address:172.16.5.234

```

show tech-support

```
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
MULTICASTMode:autoselect, 100baseTX
```

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:
-----------------	--

```
ServiceEngine#show telnet
telnet service is enabled
```

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

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 an SE, use the show transaction-logging EXEC command. Transaction log file information is displayed for HTTP and WMT caching proxy transactions and TFTP and ICAP transactions.
Examples	<p>The following example displays information about the current configuration of transaction logging on an SE:</p> <pre> ServiceEngine#show transaction-logging Transaction log configuration: ----- Logging is enabled. Archive interval: 1800 seconds Maximum size of archive file: 2000000 KB Maximum number of archive files: 50 files Log File format is apache. Windows domain is not logged with the authenticated username Exporting files to ftp servers is enabled. File compression is disabled. Export interval: 30 minutes server type username directory 10.77.153.110 ftp root /var/ftp/test WMT MMS Caching Proxy/Server Transaction Log File Info Working Log file - size : 556 age: 483497 Archive Log file - mms_export_3.1.18.8_20090522_074807 size: 556 WMT MMS Caching Proxy/Server Transaction Log File Info (WMS-90 format) Working Log file - size : 665 age: 483497 Archive Log file - mms_export_wms_90_3.1.18.8_20090522_074807 size: 665 WMT MMS Caching Proxy/Server Transaction Log File Info (Ext. WMS-90 format) Working Log file - size : 702 </pre>

```

                                age: 483497
Archive Log file - mms_export_e_wms_90_3.1.18.8_20090522_074807      size: 70
2

WMT MMS Caching Proxy/Server Transaction Log File Info (Ext. WMS-41 format)
Working Log file - size : 584
                                age: 483497
Archive Log file - mms_export_e_wms_41_3.1.18.8_20090522_074807      size: 58
4

A&D Transaction Log File Info
Working Log file - size : 138
                                age: 483497
Archive Log file - acqdist_3.1.18.8_20090522_074807      size: 138
Movie Streamer Transaction Log File Info
Working Log file - size : 488
                                age: 482196
Archive Log file - movie-streamer_3.1.18.8_20090522_062602      size: 648
Archive Log file - movie-streamer_3.1.18.8_20090522_064309      size: 805
Archive Log file - movie-streamer_3.1.18.8_20090522_065857      size: 645
Archive Log file - movie-streamer_3.1.18.8_20090522_070038      size: 648
Archive Log file - movie-streamer_3.1.18.8_20090522_074807      size: 645
Archive Log file - movie-streamer_3.1.18.8_20090522_080016      size: 648
Archive Log file - movie-streamer_3.1.18.8_20090523_030829      size: 645
ICAP Transaction Log File Info
Working Log file - size : 61
                                age: 483496
Archive Log file - icap_3.1.18.8_20090522_074807      size: 61

Web Engine Transaction Log File Info - Apache format
Working Log file - size : 86
                                age: 483497
Archive Log file - we_accesslog_apache_3.1.18.8_20090522_074807      size: 82

Web Engine Transaction Log File Info - CLF format
Working Log file - size : 3
                                age: 483497
Archive Log file - we_accesslog_clf_3.1.18.8_20090522_074807      size: 3

Web Engine Transaction Log File Info - Extended Squid format
Working Log file - size : 102
                                age: 483497
Archive Log file - we_accesslog_extsqu_3.1.18.8_20090522_074807      size: 10
2

Cached Content Log File Info
Working Log file - size : 41
                                age: 483496
Archive Log file - cache_content_3.1.18.8_20090522_074807      size: 41

Flash Media Streaming Access Transaction Log File Info
Working Log file - size : 36
                                age: 482196
Archive Log file - fms_access_3.1.18.8_20090522_062602      size: 650
Archive Log file - fms_access_3.1.18.8_20090522_064309      size: 509
Archive Log file - fms_access_3.1.18.8_20090522_065857      size: 650
Archive Log file - fms_access_3.1.18.8_20090522_074807      size: 509
Archive Log file - fms_access_3.1.18.8_20090522_080016      size: 509
Archive Log file - fms_access_3.1.18.8_20090523_030830      size: 650

Flash Media Streaming Authorization Transaction Log File Info
Working Log file - size : 43
                                age: 482196
Archive Log file - fms_auth_3.1.18.8_20090522_062602      size: 4826

```

show transaction-logging

```

Archive Log file - fms_auth_3.1.18.8_20090522_063036 size: 281
Archive Log file - fms_auth_3.1.18.8_20090522_064309 size: 596
Archive Log file - fms_auth_3.1.18.8_20090522_065857 size: 4789
Archive Log file - fms_auth_3.1.18.8_20090522_070038 size: 277
Archive Log file - fms_auth_3.1.18.8_20090522_074807 size: 596
Archive Log file - fms_auth_3.1.18.8_20090523_030830 size: 4790

```

Authserver Transaction Log File Info

```

Working Log file - size : 108
                  age: 483496
Archive Log file - authsvr_3.1.18.8_20090522_065857 size: 108

```

ServiceEngine#

The following example displays information about the current configuration of transaction logging on an SR:

ServiceRouter#show transaction-logging

Transaction log configuration:

```

-----
Logging is enabled.
Archive interval: 120 seconds
Maximum size of archive file: 2000000 KB
Maximum number of archive files: 50 files

```

```

Exporting files to ftp servers is enabled.
File compression is disabled.
Export interval: 1 minute

```

server	type	username	directory
10.74.115.12	sftp	xinwwang	/workspace/xinwwang/test
10.74.124.156	sftp	root	/root/test
10.74.124.157	sftp	root	/root/test
171.71.50.162	sftp	root	/test

Service Router Log File Info

```

Working Log file - size : 96
                  age: 169813
Archive Log file - service_router_3.1.14.70_20090421_222006 size: 256
Archive Log file - service_router_3.1.14.70_20090422_020038 size: 223
Archive Log file - service_router_3.1.14.70_20090422_210022 size: 351
Archive Log file - service_router_3.1.14.70_20090423_020006 size: 1248
Archive Log file - service_router_3.1.14.70_20090423_210021 size: 456
Archive Log file - service_router_3.1.14.70_20090521_000218 size: 402
Archive Log file - service_router_3.1.14.70_20090521_014815 size: 243
Archive Log file - service_router_3.1.14.70_20090521_015020 size: 225
Archive Log file - service_router_3.1.14.70_20090521_015227 size: 243
Archive Log file - service_router_3.1.14.70_20090521_015417 size: 272
Archive Log file - service_router_3.1.14.70_20090521_015601 size: 390
Archive Log file - service_router_3.1.14.70_20090521_015816 size: 243
Archive Log file - service_router_3.1.14.70_20090521_020033 size: 243
Archive Log file - service_router_3.1.14.70_20090521_020249 size: 143
Archive Log file - service_router_3.1.14.70_20090521_032633 size: 168
Archive Log file - service_router_3.1.14.70_20090526_025027 size: 143
Archive Log file - service_router_3.1.14.70_20090526_030002 size: 176
Archive Log file - service_router_3.1.14.70_20090526_030226 size: 250
Archive Log file - service_router_3.1.14.70_20090526_052206 size: 250
Archive Log file - service_router_3.1.14.70_20090526_052413 size: 143
Archive Log file - service_router_3.1.14.70_20090526_200213 size: 168
Archive Log file - service_router_3.1.14.70_20090526_200413 size: 481
Archive Log file - service_router_3.1.14.70_20090526_200645 size: 173
Archive Log file - service_router_3.1.14.70_20090526_201010 size: 250

```

Related Commands

clear
clear
show statistics transaction-logs
transaction-log force

show url-signature

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

show url-signature

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

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

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

Examples	The following example displays the URL signature information:
-----------------	---

```
ServiceEngine#show url-signature
key-id-owner key-id-number key
-----
```


show user

To display the 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 2-73](#) describes the fields shown in the **show user** display.

Table 2-73 *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
show users
username

show users

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

show users administrative

Syntax Description

administrative	Lists users with administrative privileges.
-----------------------	---

Defaults

No default behavior or values.

Command Modes

EXEC

Examples

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

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

Related Commands

clear
show user
username

show version

To display version information about the SE 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 2-74](#) describes the fields shown in the **show version** display.

Table 2-74 *show version Field Descriptions*

Field	Description
Compiled hour:minute:second month day year by cnbuild	Compile information for the software build.
System was restarted on day of week month day hour:minute:second year	Date and time that the system was last restarted.
The system has been up for X hours, X minutes, X seconds	Length of time the system has been running since the last reboot.

show wmt

To display Windows Media Technologies (WMT) bandwidth and proxy mode configuration, use the **show wmt** EXEC command.

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

Syntax Description	
bandwidth	(Optional) Displays WMT bandwidth settings.
incoming	(Optional) Displays WMT incoming bandwidth settings.
bypass-list	Displays the WMT incoming bandwidth bypass list.
detail	(Optional) Displays the detailed WMT configuration.
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 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.

Defaults No default behavior or values.

Command Modes EXEC

Usage Guidelines You can access the following three WMT diagnostic tools through the SE 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 Multimedia Messaging Service (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 RTSP, RTP, and RTCP.

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.

```
ServiceEngine# show wmt diagnostics header-info stream-file 256.wmv
Start dumping ASF header objects...
```

```
Obj: ASF_Header_Object (size 30)
    Header Len: 5342
    Header Num Of Objs: 8
Obj: ASF_File_Properties_Object (size 104)
    file_size: 429275084
    creation_time: 128208475755620000
    packet_count: 53656
    play_duration: 36050290000
    send_duration: 35992950000
    preroll: 5000
    flags: 2
    min_pktsize: 8000
    max_pktsize: 8000
    min_bitrate: 1003200
Obj: ASF_Stream_Properties_Object (size 114)
    time_offset: 0
    stream_type: ASF_Audio_Media
    ecc_type: ASF_Audio_Spread
    type_data_len: 28
    ecc_data_len: 8
    flags: 0x0001 (stream #: 1)
    ASF type specific data: -----
    id_tag: 161                num_channels: 2
    sample_per_sec: 48000      bytes_per_sec: 15875
    block_align: 2032         bits_per_sample: 16
    codec_data(size: 10):
    0x00 0x88 0x00 0x00 0x0f 0x00 0xf0 0x07
    0x00 0x00
    ASF Ecc data: -----
    span: 1
    packet_len: 2032          chunk_len: 2032
    silence_data (1 bytes): 0x00
Obj: ASF_Stream_Properties_Object (size 133)
    time_offset: 0
    stream_type: ASF_Video_Media
    ecc_type: ASF_No_Error_Correction
    type_data_len: 55
    ecc_data_len: 0
    flags: 0x0002 (stream #: 2)
    ASF type specific data: -----
    image_width: 320          image_height: 240
    flags: 2                  data_size: 44
    width: 320                height: 240
    bits_per_pixel: 24         compression_id: 861293911
    data_size: 44              image_size: 0
    h_pixels_per_meter: 0      v_pixels_per_meter: 0
    color_count: 0             important_color_count: 0
    codec_data (4 bytes): 0x4e 0xd9 0x1a 0x01
Obj: ASF_Extended_Content_Description_Object (size 208)
Obj: ASF_Content_Description_Object (size 42)
    title:
    author:
    copyright:
    description:
    rating:
Obj: ASF_Stream_Bitrate_Properties_Object (size 38)
```

```

        bitrate record count: 2
        # 0: flags = 0x0001, bitrate = 129550
        # 1: flags = 0x0002, bitrate = 873650
Obj: ASF_Codec_List_Object (size 252)
    codec_list_entry count: 2
    entry # 0:
        name = Windows Media Audio 9.1
        description = 127 kbps, 48 kHz, stereo Low Delay 1-pass CBR
        0x61 0x01
    entry # 1:
        name = Windows Media Video 9
        description =
        0x57 0x4d 0x56 0x33
Obj: ASF_Header_Extension_Object (size 4421)
Obj: ASF_Language_List_Object (size 39)
Obj: ASF_Extended_Stream_Properties_Object (size 88)
Obj: ASF_Extended_Stream_Properties_Object (size 110)
Obj: ASF_Compatibility_Object (size 26)
Obj: ASF_Metadata_Object (size 224)
Obj: ASF_Padding_Object (size 3850)
Obj: ASF_GUID_Invalid/Unknown_Object (size 38)
    0x20 0xde 0xaa 0xd9 0x17 0x7c 0x9c 0x4f
    0xbc 0x28 0x85 0x55 0xdd 0x98 0xe2 0xa2
Obj: ASF_Data_Object (size 50)
    data_size: 429248050
    packet_count: 53656

```

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 live1.nsc.

```

ServiceEngine# show wmt diagnostics header-info nsc-file live1.nsc
Press Ctrl-C to abort, if no information is shown within 30 secs.

```

```

=====Dumping NSC file - live1.nsc=====

```

```

[Address]
Name=(null)
NSC Format Version=3.0
Multicast Adapter=(null)
IP Address=224.2.2.3
IP Port=96
Time To Live=15
Default Ecc=10
Log URL=http://kinslive.spcdn.net/live1.nsclog
Unicast URL=rtsp://kinslive.spcdn.net/live1
Allow Splitting=1
Allow Caching=1
Cache Expiration Time=86400
[Formats]
Format1=[Binary data skipped], len = 5316, key = 1111

```

```

-----Now trying to dump ASF header(0)-----

```

```

Obj: ASF_Header_Object (size 30)
    Header Len: 5266
    Header Num Of Objs: 8
Obj: ASF_File_Properties_Object (size 104)
    file_size: 5268
    creation_time: 128880472543590000
    packet_count: 4294967295
    play_duration: 0
    send_duration: 0
    preroll: 5000

```

```

    flags: 9
    min_pktsize: 8000
    max_pktsize: 8000
    min_bitrate: 1003200
Obj: ASF_Stream_Properties_Object (size 114)
    time_offset: 0
    stream_type: ASF_Audio_Media
    ecc_type: ASF_Audio_Spread
    type_data_len: 28
    ecc_data_len: 8
    flags: 0x0001 (stream #: 1)
    ASF type specific data: -----
    id_tag: 161                num_channels: 2
    sample_per_sec: 48000      bytes_per_sec: 15875
    block_align: 2032         bits_per_sample: 16
    codec_data(size: 10):
    0x00 0x88 0x00 0x00 0x0f 0x00 0xf0 0x07
    0x00 0x00
    ASF Ecc data: -----
    span: 1
    packet_len: 2032          chunk_len: 2032
    silence_data (1 bytes): 0x00
Obj: ASF_Stream_Properties_Object (size 133)
    time_offset: 0
    stream_type: ASF_Video_Media
    ecc_type: ASF_No_Error_Correction
    type_data_len: 55
    ecc_data_len: 0
    flags: 0x0002 (stream #: 2)
    ASF type specific data: -----
    image_width: 320          image_height: 240
    flags: 2                  data_size: 44
    width: 320                height: 240
    bits_per_pixel: 24         compression_id: 861293911
    data_size: 44             image_size: 0
    h_pixels_per_meter: 0     v_pixels_per_meter: 0
    color_count: 0            important_color_count: 0
    codec_data (4 bytes):     0x4e 0xd9 0x1a 0x01
Obj: ASF_Stream_Bitrate_Properties_Object (size 38)
    bitrate record count: 2
    # 0: flags = 0x0001, bitrate = 129550
    # 1: flags = 0x0002, bitrate = 873650
Obj: ASF_Extended_Content_Description_Object (size 164)
Obj: ASF_Codec_List_Object (size 252)
    codec_list_entry count: 2
    entry # 0:
    name = Windows Media Audio 9.1
    description = 127 kbps, 48 kHz, stereo Low Delay 1-pass CBR
    0x61 0x01
    entry # 1:
    name = Windows Media Video 9
    description =
    0x57 0x4d 0x56 0x33
Obj: ASF_Error_Correction_Object (size 48)
    ecc type: ASF_Error_Correction_Default
    data_len: 4
    ecc span: 10
Obj: ASF_Header_Extension_Object (size 4383)
Obj: ASF_Language_List_Object (size 39)
Obj: ASF_Extended_Stream_Properties_Object (size 88)
Obj: ASF_Extended_Stream_Properties_Object (size 110)
Obj: ASF_Compatibility_Object (size 26)
Obj: ASF_Metadata_Object (size 224)
Obj: ASF_Padding_Object (size 3850)

```

```
Obj: ASF_Data_Object (size 50)
    data_size: 50
    packet_count: 0
```

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 WMT server configurations, the WMT HTTP configurations, and the WMT proxy configurations for the SE. The output of the **show wmt** and **show wmt detail** commands is identical.

```
ServiceEngine#show wmt
----- WMT Server Configurations -----
WMT is enabled
WMT disallowed client protocols: http
WMT bandwidth platform limit: 2000000 Kbits/sec
WMT outgoing bandwidth configured is 2000000 Kbits/sec
WMT incoming bandwidth configured is 2000000 Kbits/sec
WMT max sessions configured: 400
WMT max sessions platform limit: 14000
WMT max sessions enforced: 400 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: 10000 MB
WMT cache revalidate for each request is enabled
WMT cache age-multiplier: 100%
WMT cache min-ttl: 75 minutes
WMT cache max-ttl: 7 days
WMT debug client ip not set
WMT debug server ip not set
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 65535 (Kbps)
WMT fast-cache is enabled
WMT fast-cache acceleration factor is 65535
WMT maximum data packet MTU (TCP) enforced is 1472 bytes
WMT maximum data packet MTU (UDP) is 16000 bytes
WMT client idle timeout is 300 seconds
WMT forward logs is enabled
WMT server inactivity-timeout is 65535
WMT Transaction Log format is Windows Media Services 9.0 logging and SE specific
information
RTSP Gateway incoming 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 configured: 2.2.23.19:86
ServiceEngine#
```

The following example displays the WMT bandwidth settings configured on an SE:

```
ServiceEngine# show wmt bandwidth
Outgoing bandwidth configured 2000000 kbps
```



```
Incoming bandwidth configured 2000000 kbps  
Incoming bandwidth configured 50000 kbps
```

Related Commands

clear
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 on page 2-123 section for alternative mechanism.
-------------------------	--

Examples	The following example shows how to shut down an interface configured on an SE:
-----------------	--



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

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

shutdown (EXEC)

To shut down the SE, SR, or CDSM, 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 an SE 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 an SE but the power is still on. Controlled shutdowns of an SE can help you minimize the downtime when the SE is being serviced.</p> <p>The shutdown EXEC command enables you to shut down and optionally power off an SE:</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 SE is powered down by the Internet Streamer CDS 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.
 Caution	If you do not perform a controlled shutdown, the SE file system can be corrupted. It also takes longer to reboot the SE if the SE is not properly shut down.
 Note	You cannot power on SEs again through software after a software poweroff operation. You must press the power button once on these SEs to bring these SEs back online.

The **shutdown** EXEC command facilitates a proper shutdown for SEs, SRs, or CDSMs. 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 **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. The device console displays a menu after the shutdown process is completed. You need to log in to the SE using a console to display the following menu:

```
ServiceEngine#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
Shutting down all services, will timeout in 15 minutes.
shutdown in progress ..Halt requested by CLI@ttyS0.
.....
Shutdown success

Cisco Service 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 2-75](#) describes the shutdown and shutdown power-off operations for SEs.

Table 2-75 Shutting Down Content Engines Through CLI Commands

Activity	All Content Engine Models	Content Engines with Power Management Capability
User performs a shutdown operation on the SE	ServiceEngine# shutdown	ServiceEngine# shutdown poweroff
User intervention to bring SE back online	<p>To bring an SE that has an on/off switch on the back online after a shutdown operation, flip the on/off switch twice.</p> <p>To bring an SE 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.</p>	After a shutdown poweroff, you must press the power button once to bring the SE back online.
File system check	Will not be performed after you turn the power on again and reboot the SE.	Will not be performed after you turn the power on again and reboot the SE.

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 an SE.

To perform a shutdown on an SE, enter the **shutdown** EXEC command as follows:

```
ServiceEngine# 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 SE:

```
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 Internet Streamer CDS 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 SE, press and hold the power button on the SE, 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 SE CLI, enter the **shutdown poweroff** EXEC command as follows:

```
ServiceEngine# 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:

```
ServiceEngine1# 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 SE:

```
ServiceEngine2# 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 community

To configure the community access string to permit access to the Simple Network Management Protocol (SNMP), use the **snmp-server community** command in global configuration mode. To remove the specified community string, use the **no** form of this command.

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

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

Syntax Description

<i>community-string</i>	Community string that acts like a password and permits access to SNMP.
group	(Optional) Specifies the group to which this community name belongs.
<i>group name</i>	Name of the group.
rw	(Optional) Specifies read-write access with this community string.

Defaults

An SNMP community string permits read-only access to all MIB objects.
A community string is assigned to the Secure Domain Router (SDR) owner.

Command Modes

Global configuration

Usage Guidelines

To use this command, you must be in a user group associated with a task group that includes the proper task IDs. Use the **snmp-server community** command to configure the community access string to permit access to SNMP. To remove the specified community string, use the **no** form of this command.



Note

In a non-owner SDR, a community name provides access only to the object instances that belong to that SDR, regardless of the access privilege assigned to the community name. Access to the owner SDR and system-wide access privileges are available only from the owner SDR.

Examples

The following example shows how to add the community comaccess:

```
ServiceEngine(config)#snmp-server community comaccess rw
```

The following example shows how to remove the community comaccess:

```
ServiceEngine(config)#no snmp-server community comaccess
```

Related Commands

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

snmp-server enable traps

To enable the SE 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** | **entity** | **event** | **service-engine** [**disk-fail** | **disk-read** | **disk-write** | **transaction-log**] | **snmp** [**authentication** | **cold-start**]]

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

Syntax Description	
alarm	(Optional) Enables SE 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.
entity	(Optional) Enables SNMP entity traps.
event	(Optional) Enables Event MIB traps.
service-engine	(Optional) Enables SNMP SE 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.
transaction-log	(Optional) Enables the transaction log write error trap.
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 an SE to generate an SNMP trap for a specific alarm condition. You can configure the generation of SNMP alarm traps on SEs based on the following:

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

Cisco Internet Streamer Release 2.4 software supports six generic alarm traps. These six generic alarm traps provide SNMP and Node Health Manager integration. Each trap can be enabled or disabled through the SE CLI.

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 SE 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 SE to send all traps to the host 172.31.2.160 using the community string public:

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

The following example disables all traps:

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

Related Commands

```
show snmp
snmp-server community
snmp-server contact
snmp-server group
snmp-server host
snmp-server location
snmp-server notify inform
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 Internet Streamer CDS Release 2.4 software supports the following versions of SNMP:

- Version 1 (SNMPv1)—This version is the initial implementation of SNMP. See 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 SEs by authenticating and encrypting packets over the network. The SNMP agent in the Internet Streamer CDS Release 2.4 software supports SNMPv3, SNMPv1, and SNMPv2c.

Using SNMPv3, users can securely collect management information from their SNMP agents. Also, confidential information, such as SNMP set packets that change an SE'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 SE:

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

Related Commands

show snmp
snmp-server community
snmp-server contact
snmp-server enable traps
snmp-server host
snmp-server location
snmp-server notify inform
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 SE.
<i>ip-address</i>		IP address of the SNMP trap host that will be sent in the SNMP trap messages from the SE.
<i>communitystring</i>		Password-like community string sent in the SNMP trap messages from the SE. 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. The default is 15 seconds.
<i>seconds</i>		Timeout value in seconds (1–1000).
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 *seconds*: 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 protocol data unit (PDU). If the sender never receives the response, the inform request can be sent again. Inform requests 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 SE 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.

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 traps** command. Use the **snmp-server enable traps** command to specify which SNMP notifications are sent globally. For a host to receive most notifications, at least one **snmp-server enable traps** 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.

```
ServiceEngine(config)#snmp-server enable traps
ServiceEngine(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:

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

Related Commands

```
show snmp
snmp-server community
snmp-server contact
snmp-server enable traps
snmp-server group
snmp-server location
snmp-server notify inform
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	<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	The following example shows how to configure a system location string: ServiceEngine(config)# snmp-server location Building 3/Room 214
-----------------	--

Related Commands	show snmp snmp-server community snmp-server contact snmp-server enable traps snmp-server group snmp-server host snmp-server notify inform 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 SEs to send inform requests to SNMP managers. SEs can send notifications to SNMP managers when particular events occur. For example, an agent SE might send a message to a manager when the agent SE 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 SE 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 SE 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 SE:

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

Related Commands

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

snmp-server user

To define a user who can access the SNMP 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 SE. 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 SEs:

- If SNMPv3 is going to be used for SNMP requests, you must define at least one SNMPv3 user account on the SE in order for the SE 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.



To send an SNMPv3 inform message, you must configure at least one SNMPv3 user with a remote SNMP ID option on the SE. 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 SE. 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 SE does not perform authentication on SNMP requests from this user.

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

Related Commands

show snmp
snmp-server community
snmp-server contact
snmp-server enable traps
snmp-server group
snmp-server host
snmp-server location
snmp-server notify inform
snmp-server view

snmp-server view

To define a SNMP Version 2 (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:

```
ServiceEngine(config)#snmp-server view contentview ciscoServiceEngineMIB 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:

```
ServiceEngine(config)#snmp-server view phred system included
ServiceEngine(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:

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

Related Commands

- show snmp**
- snmp-server community**
- snmp-server contact**
- snmp-server enable traps**
- snmp-server group**
- snmp-server host**
- snmp-server location**
- snmp-server notify inform**
- snmp-server user**

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** | **timeout** *seconds* | **version** { **1** | **2** } }

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

Syntax Description

enable	Enables the SSH feature.
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 SE but have not entered the username when prompted at the login prompt, the connection will be terminated by the SE 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 SE.
1	Specifies that SSH version 1 is supported on the SE.
2	Specifies that SSH version 2 is supported on the SE.

Defaults

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 SE 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.

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 CDS 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 SE, SFTP support is not available.

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 Internet Streamer CDS software enables support for both SSH version 1 and SSH version 2 on the SE. If you want the SE 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 SE, 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 an SE. Use the **no sshd enable** global configuration command to disable SSH on the SE.

Examples

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

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

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

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

Related Commands

show ssh

sysreport

To save the sysreport to a user-specified file, use the **sysreport** privilege EXEC command.

sysreport { **acquisition-distribution** [**date-range** *start-date end-date* | *filename*] | **authentication** [**date-range** *start-date end-date* | *filename*] | **cms** [**date-range** *start-date end-date* | *filename*] | **dns** | **flash-media-streaming** | **ftp** | **http** | **icap** | **movie-streamer** | **rules** | **wmt** }

Syntax Description	
acquisition-distribution	Generates sysreport information related to acquisition/distribution.
date-range	Specifies the date range of system report.
<i>start-date</i>	Specifies start date of system report following the format yyyy/mm/dd assuming local time zone.
<i>end-date</i>	Specifies the end date of system report following the format yyyy/mm/dd assuming local time zone.
<i>filename</i>	Filename (xxx.tar.gz) for system report.
authentication	Generates sysreport information related to http authentication.
cms	Generates sysreport information related to Centralized Management System (CMS).
dns	Generates sysreport information related to Domain Name Server (DNS).
flash-media-streaming	Generates sysreport information related to Flash Media Streaming.
ftp	Generates sysreport information related to FTP.
http	Generates sysreport information related to HTTP.
icap	Generates sysreport information related to ICAP
movie-streamer	Generates sysreport information related to Movie Streamer.
rules	Generates sysreport information related to rules.
wmt	Generates sysreport information related to Windows Media Technologies (WMT).

Defaults No default behavior or values.

Command Modes Privilege EXEC

Examples The following example saves the sysreport for WMT to a user-specified file:

```
ServiceEngine#sysreport wmt date-range 2009/05/07 2009/05/11 xxx.tar.gz
The sysreport has been saved onto file xxx.tar.gz in local1
```


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

retransmit *retries*: 2

password **ascii**: 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:

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

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

```
ServiceEngine(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:

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

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

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

The Users GUI page or the **username** global configuration command 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+ Service 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 Service 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 Service 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 earlier releases, the password type was not configurable. When users needed to log in to a Service 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 Service 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:

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

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

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

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

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

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

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

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

```
ServiceEngine#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
ServiceEngine#
```

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:

```
ServiceEngine(config)#tacacs password ascii
ServiceEngine(config)#exit
ServiceEngine#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

tcpdump

To dump the network traffic, use the **tcpdump** EXEC command.

tcpdump [*LINE*]

Syntax Description	<i>LINE</i> (Optional) Specifies the dump options.
Defaults	No default behavior or values.
Command Modes	EXEC

Usage Guidelines Use the **tcpdump** command to gather a sniffer trace on the SE, SR, or CDSM for troubleshooting when asked to gather the data by the Cisco TAC. This utility is very similar to the Linux or Unix **tcpdump** command.

The **tcpdump** command allows an administrator (must be an admin user) to capture packets from the Ethernet. On the SE 500 series, the interface names are eth0 and eth1. On all CDS platforms, we recommend that you specify a path/filename in the local1 directory.

You can do a straight packet header dump to the screen by entering the **tcpdump** command. Press **Ctrl-C** to stop the dump.

The **tcpdump** command has the following options:

- **-w <filename>**—writes the raw packet capture output to a file.
- **-s <count>**—captures the first <count> bytes of each packet.
- **-i <interface>**—allows you to specify a specific interface to use for capturing the packets.
- **-c <count>**—limits the capture to <count> packets.

The following example captures the first 1500 bytes of the next 10,000 packets from interface Ethernet 0 and puts the output in a file named dump.pcap in the local1 directory on the SE:

```
ServiceEngine#tcpdump -w /local1/dump.pcap -i eth0 -s 1500 -c 10000
```

When you specify the **-s** option, it sets the packet snap length. The default value captures only 64 bytes, and this default setting saves only packet headers into the capture file. For troubleshooting of redirected packets or higher level traffic (HTTP, authentication, and so on), you must copy the complete packets.

After the TCP dump has been collected, you need to move the file from the SE to a PC so that the file can be viewed by a sniffer decoder.

```
ftp <ip address of the SE>
```

```
!--- Log in using the admin username and password.
```

```
cd local1
bin
hash
```

```
get <name of the file>
```

```
!--- Using the above example, it would be dump.pcap.
```

```
bye
```

We recommend that you use Ethereal as the software application for reading the TCP dump. With Ethereal, you can decode packets that are encapsulated into a GRE tunnel. See the Ethereal website for further information.



Note

In most cases, redirected packets captured by the tcpdump facility with the CDS CLI differ from the data received on the interface. The destination IP address and TCP port number are modified to reflect the device IP address and the port number 8999.

Examples

The following example shows how to dump the TCP network traffic:

```
ServiceEngine#tcpdump
```

```
tcpdump: verbose output suppressed, use -v or -vv for full protocol decode
listening on eth0, link-type EN10MB (Ethernet), capture size 68 bytes
12:45:42.617677 IP ServiceEngine.cisco.com.ssh > 10.77.140.97.4314: P
3342832089:3342832201(112) ack 1248615673 win 15232
12:45:42.618950 IP 172.19.226.63 > ServiceEngine.cisco.com: icmp 36: 172.19.226.63 udp
port 2048 unreachable
12:45:42.619327 IP ServiceEngine.cisco.com.10015 > dns-sj2.cisco.com.domain:
49828+[]domain]
12:45:42.621158 IP dns-sj2.cisco.com.domain > ServiceEngine.cisco.com.10015: 49828
NXDomain*[]domain]
12:45:42.621942 IP ServiceEngine.cisco.com.10015 > dns-sj2.cisco.com.domain:
49829+[]domain]
12:45:42.623799 IP dns-sj2.cisco.com.domain > ServiceEngine.cisco.com.10015: 49829
NXDomain*[]domain]
12:45:42.624240 IP ServiceEngine.cisco.com.10015 > dns-sj2.cisco.com.domain:
49830+[]domain]
12:45:42.626164 IP dns-sj2.cisco.com.domain > ServiceEngine.cisco.com.10015:
49830*[]domain]
12:45:42.702891 802.1d config TOP_CHANGE 8000.00:03:9f:f1:10:63.8042 root
8000.00:01:43:9a:c8:63 pathcost 26 age 3 max 20 hello 2 fdelay 15
12:45:42.831404 IP 10.77.140.97.4314 > ServiceEngine.cisco.com.ssh: . ack 112 win 64351
12:45:42.831490 IP ServiceEngine.cisco.com.ssh > 10.77.140.97.4314: . 112:1444(1332) ack 1
win 15232
12:45:42.831504 IP ServiceEngine.cisco.com.ssh > 10.77.140.97.4314: P 1444:1568(124) ack 1
win 15232
12:45:42.831741 IP ServiceEngine.cisco.com.ssh > 10.77.140.97.4314: P 1568:1696(128) ack 1
win 15232
12:45:43.046176 IP 10.77.140.97.4314 > ServiceEngine.cisco.com.ssh: . ack 1568 win 65535
12:45:43.046248 IP ServiceEngine.cisco.com.ssh > 10.77.140.97.4314: P 1696:2128(432) ack 1
win 15232
12:45:43.046469 IP ServiceEngine.cisco.com.ssh > 10.77.140.97.4314: P 2128:2256(128) ack 1
win 15232
12:45:43.046616 IP ServiceEngine.cisco.com.ssh > 10.77.140.97.4314: P 2256:2400(144) ack 1
win 15232
12:45:43.107700 802.1d config TOP_CHANGE 8000.00:03:9f:f1:10:63.8042 root
8000.00:01:43:9a:c8:63 pathcost 26 age 3 max 20 hello 2 fdelay 15
12:45:43.199710 IP 10.77.140.97.4314 > ServiceEngine.cisco.com.ssh: . ack 1696 win 65407
12:45:43.199784 IP ServiceEngine.cisco.com.ssh > 10.77.140.97.4314: P 2400:2864(464) ack 1
win 15232
12:45:43.199998 IP ServiceEngine.cisco.com.ssh > 10.77.140.97.4314: P 2864:2992(128) ack 1
win 15232
```

```
12:45:43.259968 IP 10.77.140.97.4314 > ServiceEngine.cisco.com.ssh: . ack 2400 win 64703
12:45:43.260064 IP ServiceEngine.cisco.com.ssh > 10.77.140.97.4314: P 2992:3280(288) ack 1
win 15232
12:45:43.260335 IP ServiceEngine.cisco.com.ssh > 10.77.140.97.4314: P 3280:3408(128) ack 1
win 15232
12:45:43.260482 IP ServiceEngine.cisco.com.ssh > 10.77.140.97.4314: P 3408:3552(144) ack 1
win 15232
12:45:43.260621 IP ServiceEngine.cisco.com.ssh > 10.77.140.97.4314: P 3552:3696(144) ack 1
win 15232
12:45:43.413320 IP 10.77.140.97.4314 > ServiceEngine.cisco.com.ssh: . ack 2992 win 65535
12:45:43.413389 IP ServiceEngine.cisco.com.ssh > 10.77.140.97.4314: P 3696:3984(288) ack 1
win 15232
12:45:43.413597 IP ServiceEngine.cisco.com.ssh > 10.77.140.97.4314: P 3984:4112(128) ack 1
win 15232
12:45:43.413741 IP ServiceEngine.cisco.com.ssh > 10.77.140.97.4314: P 4112:4256(144) ack 1
win 15232
12:45:43.473601 IP 10.77.140.97.4314 > ServiceEngine.cisco.com.ssh: . ack 3552 win 64975
12:45:43.473659 IP ServiceEngine.cisco.com.ssh > 10.77.140.97.4314: P 4256:4544(288) ack 1
win 15232
12:45:43.473853 IP ServiceEngine.cisco.com.ssh > 10.77.140.97.4314: P 4544:4672(128) ack 1
win 15232
12:45:43.473994 IP ServiceEngine.cisco.com.ssh > 10.77.140.97.4314: P 4672:4816(144) ack 1
win 15232
12:45:43.474132 IP ServiceEngine.cisco.com.ssh > 10.77.140.97.4314: P 4816:4960(144) ack 1
win 15232
12:45:43.484117 IP 10.77.140.97.4314 > ServiceEngine.cisco.com.ssh: P 1:81(80) ack 3696
win 64831
12:45:43.484167 IP ServiceEngine.cisco.com.ssh > 10.77.140.97.4314: P 4960:5248(288) ack
81 win 15232
12:45:43.484424 IP ServiceEngine.cisco.com.ssh > 10.77.140.97.4314: P 5248:5392(144) ack
81 win 15232
12:45:43.627125 IP 10.77.140.97.4314 > ServiceEngine.cisco.com.ssh: . ack 4112 win 64415
12:45:43.627204 IP ServiceEngine.cisco.com.ssh > 10.77.140.97.4314: P 5392:5680(288) ack
81 win 15232
12:45:43.627439 IP ServiceEngine.cisco.com.ssh > 10.77.140.97.4314: P 5680:5808(128) ack
81 win 15232
12:45:43.627586 IP ServiceEngine.cisco.com.ssh > 10.77.140.97.4314: P 5808:5952(144) ack
81 win 15232
12:45:43.688261 IP 10.77.140.97.4314 > ServiceEngine.cisco.com.ssh: . ack 4544 win 65535
12:45:43.688316 IP ServiceEngine.cisco.com.ssh > 10.77.140.97.4314: P 5952:6240(288) ack
81 win 15232
12:45:43.688495 IP ServiceEngine.cisco.com.ssh > 10.77.140.97.4314: P 6240:6368(128) ack
81 win 15232
12:45:43.688638 IP ServiceEngine.cisco.com.ssh > 10.77.140.97.4314: P 6368:6512(144) ack
81 win 15232
12:45:43.689012 IP 10.77.140.97.4314 > ServiceEngine.cisco.com.ssh: . ack 4960 win 65119
12:45:43.689046 IP ServiceEngine.cisco.com.ssh > 10.77.140.97.4314: P 6512:6800(288) ack
81 win 15232
12:45:43.689170 IP ServiceEngine.cisco.com.ssh > 10.77.140.97.4314: P 6800:6928(128) ack
81 win 15232
12:45:43.689309 IP ServiceEngine.cisco.com.ssh > 10.77.140.97.4314: P 6928:7072(144) ack
81 win 15232
12:45:43.689447 IP ServiceEngine.cisco.com.ssh > 10.77.140.97.4314: P 7072:7216(144) ack
81 win 15232
12:45:43.698391 IP 10.77.140.97.4314 > ServiceEngine.cisco.com.ssh: . ack 5392 win 64687
12:45:43.698437 IP ServiceEngine.cisco.com.ssh > 10.77.140.97.4314: P 7216:7504(288) ack
81 win 15232
12:45:43.698599 IP ServiceEngine.cisco.com.ssh > 10.77.140.97.4314: P 7504:7632(128) ack
81 win 15232
12:45:43.698740 IP ServiceEngine.cisco.com.ssh > 10.77.140.97.4314: P 7632:7776(144) ack
81 win 15232
12:45:43.840558 IP 10.77.140.97.4314 > ServiceEngine.cisco.com.ssh: . ack 5808 win 64271
12:45:43.840622 IP ServiceEngine.cisco.com.ssh > 10.77.140.97.4314: P 7776:8064(288) ack
81 win 15232
```

```

12:45:43.840819 IP ServiceEngine.cisco.com.ssh > 10.77.140.97.4314: P 8064:8192(128) ack
81 win 15232
12:45:43.840962 IP ServiceEngine.cisco.com.ssh > 10.77.140.97.4314: P 8192:8336(144) ack
81 win 15232
12:45:43.901868 IP 10.77.140.97.4314 > ServiceEngine.cisco.com.ssh: . ack 6368 win 65535
12:45:43.901938 IP ServiceEngine.cisco.com.ssh > 10.77.140.97.4314: P 8336:8624(288) ack
81 win 15232
12:45:43.901887 IP 10.77.140.97.4314 > ServiceEngine.cisco.com.ssh: . ack 6928 win 64975
12:45:43.901910 IP 10.77.140.97.4314 > ServiceEngine.cisco.com.ssh: . ack 7216 win 64687
12:45:43.902137 IP ServiceEngine.cisco.com.ssh > 10.77.140.97.4314: P 8624:8752(128) ack
81 win 15232
12:45:43.902281 IP ServiceEngine.cisco.com.ssh > 10.77.140.97.4314: P 8752:8896(144) ack
81 win 15232
12:45:43.902414 IP ServiceEngine.cisco.com.ssh > 10.77.140.97.4314: P 8896:9024(128) ack
81 win 15232
12:45:43.902547 IP ServiceEngine.cisco.com.ssh > 10.77.140.97.4314: P 9024:9152(128) ack
81 win 15232
12:45:43.902687 IP ServiceEngine.cisco.com.ssh > 10.77.140.97.4314: P 9152:9296(144) ack
81 win 15232
12:45:43.902826 IP ServiceEngine.cisco.com.ssh > 10.77.140.97.4314: P 9296:9440(144) ack
81 win 15232
12:45:43.902965 IP ServiceEngine.cisco.com.ssh > 10.77.140.97.4314: P 9440:9584(144) ack
81 win 15232
12:45:43.903104 IP ServiceEngine.cisco.com.ssh > 10.77.140.97.4314: P 9584:9728(144) ack
81 win 15232
12:45:43.922413 IP 10.77.140.97.4314 > ServiceEngine.cisco.com.ssh: . ack 7632 win 64271
12:45:43.922459 IP ServiceEngine.cisco.com.ssh > 10.77.140.97.4314: P 9728:10304(576) ack
81 win 15232
12:45:43.922622 IP ServiceEngine.cisco.com.ssh > 10.77.140.97.4314: P 10304:10432(128) ack
81 win 15232
12:45:43.922764 IP ServiceEngine.cisco.com.ssh > 10.77.140.97.4314: P 10432:10576(144) ack
81 win 15232
12:45:44.053872 IP 10.77.140.97.4314 > ServiceEngine.cisco.com.ssh: . ack 8192 win 65535
12:45:44.053972 IP ServiceEngine.cisco.com.ssh > 10.77.140.97.4314: P 10576:10864(288) ack
81 win 15232
12:45:44.054308 IP ServiceEngine.cisco.com.ssh > 10.77.140.97.4314: P 10864:11104(240) ack
81 win 15232
12:45:44.054453 IP ServiceEngine.cisco.com.ssh > 10.77.140.97.4314: P 11104:11248(144) ack
81 win 15232
12:45:44.054596 IP ServiceEngine.cisco.com.ssh > 10.77.140.97.4314: P 11248:11392(144) ack
81 win 15232
12:45:44.111702 802.1d config TOP_CHANGE 8000.00:03:9f:f1:10:63.8042 root
8000.00:01:43:9a:c8:63 pathcost 26 age 3 max 20 hello 2 fdelay 15
12:45:44.114626 IP 10.77.140.97.4314 > ServiceEngine.cisco.com.ssh: . ack 8752 win 64975
12:45:44.114712 IP ServiceEngine.cisco.com.ssh > 10.77.140.97.4314: P 11392:11712(320) ack
81 win 15232
12:45:44.115219 IP ServiceEngine.cisco.com.ssh > 10.77.140.97.4314: P 11712:11952(240) ack
81 win 15232
12:45:44.115381 IP ServiceEngine.cisco.com.ssh > 10.77.140.97.4314: P 11952:12096(144) ack
81 win 15232
12:45:44.115426 IP 10.77.140.97.4314 > ServiceEngine.cisco.com.ssh: . ack 9152 win 64575
12:45:44.115617 IP ServiceEngine.cisco.com.ssh > 10.77.140.97.4314: P 12096:12336(240) ack
81 win 15232
12:45:44.115760 IP ServiceEngine.cisco.com.ssh > 10.77.140.97.4314: P 12336:12480(144) ack
81 win 15232
12:45:44.115904 IP ServiceEngine.cisco.com.ssh > 10.77.140.97.4314: P 12480:12624(144) ack
81 win 15232
12:45:44.116045 IP ServiceEngine.cisco.com.ssh > 10.77.140.97.4314: P 12624:12768(144) ack
81 win 15232
12:45:44.116094 IP 10.77.140.97.4314 > ServiceEngine.cisco.com.ssh: . ack 9440 win 64287
12:45:44.116114 IP 10.77.140.97.4314 > ServiceEngine.cisco.com.ssh: . ack 9728 win 65535
12:45:44.116332 IP ServiceEngine.cisco.com.ssh > 10.77.140.97.4314: P 12768:13088(320) ack
81 win 15232

```



```
12:45:44.116473 IP ServiceEngine.cisco.com.ssh > 10.77.140.97.4314: P 13088:13232(144) ack
81 win 15232
12:45:44.116614 IP ServiceEngine.cisco.com.ssh > 10.77.140.97.4314: P 13232:13376(144) ack
81 win 15232
12:45:44.116755 IP ServiceEngine.cisco.com.ssh > 10.77.140.97.4314: P 13376:13520(144) ack
81 win 15232
12:45:44.116895 IP ServiceEngine.cisco.com.ssh > 10.77.140.97.4314: P 13520:13664(144) ack
81 win 15232
12:45:44.135947 IP 10.77.140.97.4314 > ServiceEngine.cisco.com.ssh: . ack 10432 win 64831
12:45:44.135996 IP ServiceEngine.cisco.com.ssh > 10.77.140.97.4314: P 13664:13808(144) ack
81 win 15232
12:45:44.136223 IP ServiceEngine.cisco.com.ssh > 10.77.140.97.4314: P 13808:14048(240) ack
81 win 15232
12:45:44.136366 IP ServiceEngine.cisco.com.ssh > 10.77.140.97.4314: P 14048:14192(144) ack
81 win 15232
12:45:44.144104 IP 10.77.140.97.4314 > ServiceEngine.cisco.com.ssh: P 81:161(80) ack 10576
win 64687

102 packets captured
105 packets received by filter
0 packets dropped by kernel
```

The following example shows how to dump the TCP network traffic and redirect it to a file named test:

```
ServiceEngine#tcpdump port 8080 -w test
tcpdump: listening on eth0, link-type EN10MB (Ethernet), capture size 68 bytes
216 packets captured
216 packets received by filter
0 packets dropped by kernel
```

telnet

To log in to a network device using the Telnet client, use the **telnet** EXEC command.

telnet {*hostname* | *ip-address*} [*portnum*]

Syntax Description	<i>hostname</i>	Hostname of the network device.
	<i>ip-address</i>	IP address of the network device.
	<i>portnum</i>	(Optional) Port number (1–65535). Default port number is 23.

Defaults The default port number is 23.

Command Modes EXEC

Usage Guidelines Some UNIX shell functions, such as escape and the **suspend** command, are not available in the Telnet client. In addition, multiple Telnet sessions are also not supported.

The Telnet client allows you to specify a destination port. By entering the *telnet* command, you can test websites by attempting to open a Telnet session to the website from the SE CLI.

Examples The following example shows how to open a Telnet session to a network device using the hostname:

```
ServiceEngine#telnet cisco-ce
```

The following example shows how to open a Telnet session to a network device using the IP address:

```
ServiceEngine#telnet 172.16.155.224
```

The following example shows how to open a Telnet session to a network device on port 8443 using the hostname:

```
ServiceEngine#telnet cisco-ce 8443
```

The following example shows how to open a Telnet session to a network device on port 80 using the hostname:

```
ServiceEngine#telnet www.yahoo.com 80
```

telnet enable

To enable Telnet, use the **telnet enable** global configuration command. To disable Telnet, use the **no** form of this command.

telnet enable

no telnet enable

Syntax Description

This command has no arguments or keywords.

Defaults

Enabled

Command Modes

Global configuration

Usage Guidelines

Use this terminal emulation protocol for a remote terminal connection. The **telnet enable** command allows users to log in to other devices using a Telnet session.

Examples

The following example shows how to enable Telnet on the SE:

```
ServiceEngine(config)#telnet enable
```

Related Commands

show telnet

terminal

To set the number of lines displayed in the console window, or to display the current console **debug** command output, use the **terminal EXEC** command.

terminal {**length** *length* | **monitor** [**disable**]}

Syntax Description

length	Sets the length of the display on the terminal.
<i>length</i>	Length of the display on the terminal (0–512). Setting the length to 0 means that there is no pausing.
monitor	Copies the debug output to the current terminal.
disable	(Optional) Disables monitoring at this specified terminal.

Defaults

The default length is 24 lines.

Command Modes

EXEC

Usage Guidelines

When 0 is entered as the *length* parameter, the output to the screen does not pause. For all nonzero values of *length*, the -More- prompt is displayed when the number of output lines matches the specified *length* number. The -More- prompt is considered a line of output. To view the next screen, press the **Spacebar**. To view one line at a time, press the **Enter** key.

The **terminal monitor** command allows a Telnet session to display the output of the **debug** commands that appear on the console. Monitoring continues until the Telnet session is terminated.

Examples

The following example sets the number of lines to display to 20:

```
ServiceEngine#terminal length 20
```

The following example configures the terminal for no pausing:

```
ServiceEngine#terminal length 0
```

Related Commands

All **show** commands

test-url

To test the accessibility of a URL using FTP, HTTP, or HTTPS, use the **test-url** EXEC command.

```
test-url {ftp url [use-ftp-proxy proxy-url] | http url [custom-header header [head-only]
[use-http-proxy proxy-url] | head-only [custom-header header] [use-http-proxy proxy-url] |
use-http-proxy proxy-url [custom-header header] [head-only]}}
```

Syntax Description		
ftp		Specifies the FTP URL to be tested.
<i>url</i>		FTP URL to be tested. Use one of the following formats to specify the FTP URL: <ul style="list-style-type: none"> <i>ftp://domainname/path</i> <i>ftp://user:password@domainname/path</i>
use-ftp-proxy		(Optional) Specifies the FTP proxy that is used to test the URL.
<i>proxy-url</i>		FTP proxy URL. Use one of the following formats to specify the proxy URL: <ul style="list-style-type: none"> <i>proxy IP Address:proxy Port</i> <i>proxy Username:proxy Password@proxy IP Address:proxy Port</i>
http		Specifies the HTTP URL to be tested.
<i>url</i>		HTTP URL to be tested. Use one of the following formats to specify the HTTP URL: <ul style="list-style-type: none"> <i>http://domainname/path</i> <i>http://user:password@domainname/path</i>
custom-header		(Optional) Specifies the custom header information to be sent to the server.
<i>header</i>		Custom header information to be sent to the server. Use the format <i>header:line</i> to specify the custom header.
head-only		(Optional) Specifies that only the HTTP header information must be retrieved.
use-http-proxy		(Optional) Specifies the HTTP proxy that is used to test the URL.
<i>proxy-url</i>		HTTP proxy URL. Use one of the following formats to specify the HTTP proxy URL: <pre>http://proxyIp:proxyPort http://proxyUser:proxypasswd@proxyIp:proxyPort</pre>
head-only		(Optional) Specifies that only the HTTPS header information must be retrieved.

Defaults No default behavior or values.

Command Modes EXEC

Usage Guidelines

In the Internet Streamer CDS Release 2.4 software and later releases, an HTTP CLI client is supported. This capability allows you to test connectivity and debug caching issues. The **test-url** EXEC command in the Internet Streamer CDS Release 2.4 software and later releases allows the users to test whether a URL is accessible over the FTP, HTTP, and HTTPS protocols. When you test the connectivity using the **test-url** command, the SE sends a request using the protocol that you have specified to the server and fetches the requested contents. The actual content is dumped into the path /dev/null, and the server response with the header information is displayed to the user.

You can use the **test-url ftp** command to test the following for the specified URL:

- Connectivity to the URL
- Connectivity to the URL through the FTP proxy (using the **use-ftp-proxy** option)
- Authentication
- FTP proxy authentication

You can use the **test-url http** command to test the following for the specified URL:

- Test the connectivity to the URL
- Test the connectivity to the URL through the HTTP proxy (using the **use-http-proxy** option)
- Authentication
- HTTP proxy authentication
- Header information only for the specified page (using the **head-only** option) or additional header information (using the **custom-header** option)

Examples

The following example tests the accessibility to the URL `http://192.168.171.22` using HTTP:

```
ServiceEngine#test-url http http://cel.server.com
--02:27:20-- http://cel.server.com/
=> `/dev/null'
Len - 22 , Restval - 0 , contlen - 0 , Res - 134728056Resolving cel.server.com..

done.
Connecting to cel.server.com[192.168.171.22]:80... connected.
HTTP request sent, awaiting response...
 1 HTTP/1.1 200 OK
 2 Date: Mon, 26 Jul 2004 08:41:34 GMT
 3 Server: Apache/1.2b8
 4 Last-Modified: Fri, 25 Apr 2003 12:23:04 GMT
 5 ETag: "1aee29-663-3ea928a8"
 6 Content-Length: 1635
 7 Content-Type: text/html
 8 Via: 1.1 Content Delivery System Software 5.2
 9 Connection: Keep-Alive
(1635 to go)
0% [                                     ] 0           --.-K/s   ETA --:--L
en - 0   ELen - 1635   Keepalive - 1
100%[=====] 1,635           1.56M/s   ETA 00:00

02:27:20 (1.56 MB/s) - `/dev/null' saved [1635/1635]
```

The following example tests the accessibility to the URL `http://192.168.171.22` through the HTTP proxy `10.107.192.148`:

```
ServiceEngine#test-url http http://192.168.171.22 use-http-proxy 10.107.192.148:8090
--15:22:51-- http://10.77.155.246/
=> `/dev/null'
```

```

Len - 1393 , Restval - 0 , contlen - 0 , Res - 134728344Connecting to
10.107.192.148:8090... connected.
Proxy request sent, awaiting response...
 1 HTTP/1.1 401 Authorization Required
 2 Date: Mon, 27 Sep 2004 15:29:18 GMT
 3 Server: Apache/1.3.27 (Unix) tomcat/1.0
 4 WWW-Authenticate: Basic realm="IP/TV Restricted Zone"
 5 Content-Type: text/html; charset=iso-8859-1
 6 Via: 1.1 Content Delivery System Software 5.2.1
 7 Connection: Close
Len - 0 , Restval - 0 , contlen - -1 , Res - -1Connecting to 10.107.192.148:8090...
connected.
Proxy request sent, awaiting response...
 1 HTTP/1.1 401 Authorization Required
 2 Date: Mon, 27 Sep 2004 15:29:19 GMT
 3 Server: Apache/1.3.27 (Unix) tomcat/1.0
 4 WWW-Authenticate: Basic realm="IP/TV Restricted Zone"
 5 Content-Type: text/html; charset=iso-8859-1
 6 Via: 1.1 Content Delivery System Software 5.2.1
 7 Connection: Keep-Alive
(1635 to go)
0% [                               ] 0           --.--K/s      ETA --:--L
en - 0      ELen - 1635      Keepalive - 1
100%[=====] 1,635           1.56M/s      ETA 00:00

02:27:20 (1.56 MB/s) - `/dev/null' saved [1635/1635]

```

The following example tests the accessibility to the URL ftp://ssivakum:ssivakum@10.77.157.148 using FTP:

```

ServiceEngine#test-url ftp ftp://ssivakum:ssivakum@10.77.157.148/antinat-0.90.tar
Mar 30 14:33:44 nramaraj-ce admin-shell: %SE-PARSER-6-350232: CLI_LOG shell_parser_log:
test-url ftp ftp://ssivakum:ssivakum@10.77.157.148/antinat-0.90.tar
--14:33:44-- ftp://ssivakum:*password*@10.77.157.148/antinat-0.90.tar
=> `/dev/null'
Connecting to 10.77.157.148:21... connected.
Logging in as ssivakum ...
220 (vsFTPd 1.1.3)
--> USER ssivakum

331 Please specify the password.
--> PASS Turtle Power!
230 Login successful. Have fun.
--> SYST

215 UNIX Type: L8
--> PWD

257 "/home/ssivakum"
--> TYPE I

200 Switching to Binary mode.
==> CWD not needed.
--> PORT 10,1,1,52,82,16

200 PORT command successful. Consider using PASV.
--> RETR antinat-0.90.tar

150 Opening BINARY mode data connection for antinat-0.90.tar (1771520 bytes).
Length: 1,771,520 (unauthoritative)

0% [                               ] 0           --.--K/s      ETA --:--L
] 0           --.--K/s      ETA --:--L      Len - 0      ELen - 1771520      Keepalive - 0

```

 test-url

```
100%[=====
>] 1,771,520    241.22K/s    ETA 00:00
```

```
226 File send OK.
14:33:53 (241.22 KB/s) - `/dev/null' saved [1771520]
```

```
ServiceEngine#
```

Related Commands **acquirer test-url**

traceroute

To trace the route to a remote host, use the **traceroute** EXEC command.

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

Syntax Description	<i>hostname</i>	Name of the remote host.
	<i>ip-address</i>	IP address of the remote host.

Defaults No default behavior values

Command Modes EXEC

Usage Guidelines Traceroute is a widely available utility on most operating systems. Similar to ping, traceroute is a valuable tool for determining connectivity in a network. Ping allows the user to find out if there is a connection between the two end systems. Traceroute does this as well, but additionally lists the intermediate routers between the two systems. Users can see the routes that packets can take from one system to another. Use the **traceroute** command to find the route to a remote host when either the hostname or the IP address is known.

The **traceroute** command uses the TTL field in the IP header to cause routers and servers to generate specific return messages. Traceroute starts by sending a UDP datagram to the destination host with the TTL field set to 1. If a router finds a TTL value of 1 or 0, it drops the datagram and sends back an ICMP time-exceeded message to the sender. The traceroute facility determines the address of the first hop by examining the source address field of the ICMP time-exceeded message.

To identify the next hop, traceroute sends a UDP packet with a TTL value of 2. The first router decrements the TTL field by 1 and sends the datagram to the next router. The second router sees a TTL value of 1, discards the datagram, and returns the time-exceeded message to the source. This process continues until the TTL is incremented to a value large enough for the datagram to reach the destination host (or until the maximum TTL is reached).

To determine when a datagram has reached its destination, traceroute sets the UDP destination port in the datagram to a very large value that the destination host is unlikely to be using. When a host receives a datagram with an unrecognized port number, it sends an ICMP “port unreachable” error to the source. This message indicates to the traceroute facility that it has reached the destination.

Examples The following example shows how to trace the route to a remote host from the SE:

```
ServiceEngine#traceroute 10.77.157.43
traceroute to 10.77.157.43 (10.77.157.43), 30 hops max, 38 byte packets
 1  10.1.1.50 (10.1.1.50)  2.024 ms  2.086 ms  2.219 ms
 2  sblab2-rtr.cisco.com (192.168.10.1)  3.718 ms 172.19.231.249 (172.19.231.249)  0.653 ms 0.606 ms
 3  sjc22-00lab-gw1.cisco.com (172.24.115.65)  0.666 ms  0.624 ms  0.597 ms
 4  sjc20-lab-gw2.cisco.com (172.24.115.109)  0.709 ms  0.695 ms  0.616 ms
 5  sjc20-sbb5-gw2.cisco.com (128.107.180.97)  0.910 ms  0.702 ms  0.674 ms
 6  sjc20-rbb-gw5.cisco.com (128.107.180.9)  0.762 ms  0.702 ms  0.664 ms
 7  sjc12-rbb-gw4.cisco.com (128.107.180.2)  0.731 ms  0.731 ms  0.686 ms
```

```

 8  sjc5-gb3-fl-0.cisco.com (10.112.2.158)  1.229 ms  1.186 ms  0.753 ms
 9  capnet-hkidc-sjc5-oc3.cisco.com (10.112.2.238)  146.784 ms  147.016 ms  147.051 ms
10  hkidc-capnet-gw1-g3-1.cisco.com (10.112.1.250)  147.163 ms  147.319 ms  148.050 ms
11  hkidc-gb3-g0-1.cisco.com (10.112.1.233)  148.137 ms  148.332 ms  148.361 ms
12  capnet-singapore-hkidc-oc3.cisco.com (10.112.2.233)  178.137 ms  178.273 ms  178.005
ms
13  singapore-capnet2-fa4-0.cisco.com (10.112.2.217)  179.236 ms  179.606 ms  178.714 ms
14  singapore-gb1-fa2-0.cisco.com (10.112.2.226)  179.499 ms  179.914 ms  179.873 ms
15  capnet-chennai-singapore-ds3.cisco.com (10.112.2.246)  211.858 ms  212.167 ms  212.854
ms
16  hclodc1-rbb-gw2-g3-8.cisco.com (10.112.1.213)  213.639 ms  212.580 ms  211.211 ms
17  10.77.130.18 (10.77.130.18)  212.248 ms  212.478 ms  212.545 ms
18  codc-tbd.cisco.com (10.77.130.34)  212.315 ms  212.688 ms  213.063 ms
19  10.77.130.38 (10.77.130.38)  212.955 ms  214.353 ms  218.169 ms
20  10.77.157.9 (10.77.157.9)  217.217 ms  213.424 ms  222.023 ms
21  10.77.157.43 (10.77.157.43)  212.750 ms  217.260 ms  214.610 ms

```

The following example shows how the **tracert** command fails to trace the route to a remote host from the SE:

```

ServiceEngine#tracert 10.0.0.1
tracert to 10.0.0.1 (10.0.0.1), 30 hops max, 38 byte packets
 1  10.1.1.50 (10.1.1.50)  2.022 ms  1.970 ms  2.156 ms
 2  sblab2-rtr.cisco.com (192.168.10.1)  3.955 ms  172.19.231.249 (172.19.231.249)  0.654
ms  0.607 ms
 3  sjc22-00lab-gw1.cisco.com (172.24.115.65)  0.704 ms  0.625 ms  0.596 ms
 4  sjc20-lab-gw1.cisco.com (172.24.115.105)  0.736 ms  0.686 ms  0.615 ms
 5  sjc20-sbb5-gw1.cisco.com (128.107.180.85)  0.703 ms  0.696 ms  0.646 ms
 6  sjc20-rbb-gw5.cisco.com (128.107.180.22)  0.736 ms  0.782 ms  0.750 ms
 7  sjce-rbb-gw1.cisco.com (171.69.7.249)  1.291 ms  1.314 ms  1.218 ms
 8  sjce-corp-gw1.cisco.com (171.69.7.170)  1.477 ms  1.257 ms  1.221 ms
 9  * * *
10  * * *
.
.
.
29  * * *
30  * * *

```

Table 2-76 describes the fields in the **traceroute** command output.

Table 2-76 *traceroute Command Output Fields*

Field	Description
30 hops max, 38 byte packets	Maximum TTL value and the size of the ICMP datagrams being sent.
2.022 ms 1.970 ms 2.156 ms	Total time (in milliseconds) for each ICMP datagram to reach the router or host plus the time it took for the ICMP time-exceeded message to return to the host. An exclamation point following any of these values (for example, 20 ms !) indicates that the port-unreachable message returned by the destination had a TTL of 0 or 1. Typically, this situation occurs when the destination uses the TTL value from the arriving datagram as the TTL in its ICMP reply. The reply does not arrive at the source until the destination receives a traceroute datagram with a TTL equal to the number of hops between the source and destination.
*	An asterisk (*) indicates that the timeout period (default of 5 seconds) expired before an ICMP time-exceeded message was received for the datagram.

Related Commands

ping

traceroute6

To trace the route to a remote IPv6-enabled host, use the **traceroute6** EXEC command.

traceroute6 *ip-address*

Syntax Description	<i>ip-address</i> Remote IPv6-enabled host or IP address.
Defaults	No default behavior values
Command Modes	EXEC
Examples	The following example shows how to trace the route to a remote IPv6-enabled host from the SE: ServiceEngine# traceroute6 <IP address>
Related Commands	ipv6

transaction-log force

To force the archive or export of the transaction log, use the **transaction-log force** EXEC command.

transaction-log force {archive | export}

Syntax Description	archive	Forces the archive of the <i>working.log</i> file.
	export	Forces the archived files to be exported to the server.

Defaults No default behavior or values.

Command Modes EXEC

Usage Guidelines The **transaction-log force archive** command causes the transaction log *working.log* file to be archived to the SE hard disk following the next transaction. This command has the same effect as the **clear transaction-log** command.

The **transaction-log force export** command causes the transaction log to be exported to an FTP server designated by the **transaction-logs export ftp-server** command.

The **transaction-log force** commands do not change the configured or default schedule for archive or export of transaction log files. If the archive interval is configured in seconds or the export interval is configured in minutes, the forced archive or export interval period is restarted after the forced operation.

If a scheduled archive or export job is in progress when a corresponding **transaction-log force** command is entered, the command has no effect. If a **transaction-log force** command is in progress when an archive or export job is scheduled to run, the forced operation is completed and the archive or export is rescheduled for the next configured interval.

Examples The following example shows how to archive the transaction log file to the SE hard disk:

```
ServiceEngine#transaction-log force archive
```

The following example shows that the SE is configured to export its transaction logs to two FTP servers:

```
ServiceEngine(config)# transaction-logs export ftp-server 10.1.1.1 mylogin mypasswd
/ftpdirectory
ServiceEngine(config)# transaction-logs export ftp-server myhostname mylogin mypasswd
/ftpdirectory
```

The following example shows how to export the transaction log file from the SE hard disk to an FTP server designated by the **transaction-logs export ftp-server** command:

```
ServiceEngine#transaction-log force export
```

■ transaction-log force

Related Commands

clear
show statistics transaction-logs
show transaction-logging
transaction-logs

transaction-logs

To configure and enable transaction logs, use the **transaction-logs** command in global configuration mode. To disable transaction logs, use the **no** form of this command.

transaction-logs archive interval *seconds*

transaction-logs archive interval every-day {**at** *hour:minute* | **every** *hours*}

transaction-logs archive interval every-hour {**at** *minute* | **every** *minutes*}

transaction-logs archive interval every-week [**on** *weekdays* **at** *hour:minute*]

transaction-logs archive max-file-number *filenumber*

transaction-logs archive max-file-size *filesize*

transaction-logs enable

transaction-logs export compress

transaction-logs export enable

transaction-logs export ftp-server {*hostname* | *servipaddrs*} *login* *passw* *directory*

transaction-logs export interval *minutes*

transaction-logs export interval every-day {**at** *hour:minute* | **every** *hours*}

transaction-logs export interval every-hour {**at** *minute* | **every** *minutes*}

transaction-logs export interval every-week [**on** *weekdays* **at** *hour:minute*]

transaction-logs export sftp-server {*hostname* | *servipaddrs*} *login* *passw* *directory*

transaction-logs format {**apache** | **custom** *string* | **extended-squid**}

transaction-logs log-windows-domain

no transaction-logs {**archive** {**interval** | **max-file-number** | **max-file-size**} | **enable** | **export** {**compress** | **enable** | **ftp-server** {*hostname* | *servipaddrs*} | **interval** | **sftp-server** {*hostname* | *servipaddrs*}} | **format** | **log-windows-domain**}

Syntax Description

archive	Configures archive parameters.
interval	Determines how frequently the archive file is to be saved.
<i>seconds</i>	Frequency of archiving in seconds (120–604800).
every-day	Archives using intervals of 1 day or less.
at	Specifies the local time at which to archive each day.
<i>hour:minute</i>	Time of day at which to archive in local time (hh:mm).
every	Specifies the interval in hours. Interval aligns with midnight.

<i>hours</i>	Number of hours for daily file archive. 1—Hourly 12—Every 12 hours 2—Every 2 hours 24—Every 24 hours 3—Every 3 hours 4—Every 4 hours 6—Every 6 hours 8—Every 8 hours
every-hour	Specifies the archives using intervals of 1 hour or less.
at	Sets the time to archive at each hour.
<i>minute</i>	Minute alignment for the hourly archive (0–59).
every	Specifies the interval in minutes for hourly archive that aligns with the top of the hour.
<i>minutes</i>	Number of minutes for hourly archive. 10—Every 10 minutes 15—Every 15 minutes 2—Every 2 minutes 20—Every 20 minutes 30—Every 30 minutes 5—Every 5 minutes
every-week	Archives using intervals of 1 or more times a week.
on	(Optional) Sets the day of the week on which to archive.
<i>weekdays</i>	Weekdays on which to archive. One or more weekdays can be specified. Fri—Every Friday Mon—Every Monday Sat—Every Saturday Sun—Every Sunday Thu—Every Thursday Tue—Every Tuesday Wed—Every Wednesday
at	(Optional) Sets the local time at which to archive each day.
<i>hour:minute</i>	Time of day at which to archive in local time (hh:mm).
max-file-number	Sets the maximum number of the archived log file.
<i>filenumber</i>	Maximum number of the archived log file (1–10000).
max-file-size	Sets the maximum archive file size.
<i>filesize</i>	Maximum archive file size in kilobytes (1000–2000000).
enable	Enables the transaction log.
export	Configures file export parameters.
compress	Compresses the archived files in the gzip format before exporting.
enable	Enables the exporting of log files at the specified interval.
ftp-server	Sets the FTP server to receive exported archived files.
<i>hostname</i>	Hostname of the target FTP server.
<i>servipaddrs</i>	IP address of the target FTP server.
<i>login</i>	User login to target FTP server.

<i>passw</i>	User password to target FTP server.
<i>directory</i>	Target directory path for exported files on FTP server.
interval	Determines how frequently the file is to be exported.
<i>minutes</i>	Number of minutes in the interval at which to export a file (1–10080).
every-day	Specifies the exports using intervals of 1 day or less.
at	Specifies the local time at which to export each day.
<i>hour:minute</i>	Time of day at which to export in local time (hh:mm).
every	Specifies the interval in hours for the daily export.
<i>hours</i>	Number of hours for the daily export. 1—Hourly 12—Every 12 hours 2—Every 2 hours 24—Every 24 hours 3—Every 3 hours 4—Every 4 hours 6—Every 6 hours 8—Every 8 hours
every-hour	Specifies the exports using intervals of 1 hour or less.
at	Specifies the time at which to export each hour.
<i>minute</i>	Minute (0–59) alignment for the hourly export.
every	Specifies the interval in minutes that align with the top of the hour.
<i>minutes</i>	Number of minutes for the hourly export. 10—Every 10 minutes 15—Every 15 minutes 2—Every 2 minutes 20—Every 20 minutes 30—Every 30 minutes 5—Every 5 minutes
every-week	Specifies the exports using intervals of 1 or more times a week.
on	(Optional) Specifies the days of the week for the export.
<i>weekdays</i>	Weekdays on which to export. One or more weekdays can be specified. Fri—Every Friday Mon—Every Monday Sat—Every Saturday Sun—Every Sunday Thu—Every Thursday Tue—Every Tuesday Wed—Every Wednesday
at	(Optional) Specifies the time of day at which to perform the weekly export.
<i>hour:minute</i>	Time of day at which to export in the local time (hh:mm).
sftp-server	Sets the Secure File Transfer Protocol (SFTP) server to receive exported archived files.
<i>hostname</i>	Hostname of the target SFTP server.
<i>servipaddr</i>	IP address of the target SFTP server.
<i>login</i>	User login to the target SFTP server (less than 40 characters).

<i>passwd</i>	User password to the target SFTP server (less than 40 characters).																																
<i>directory</i>	Target directory path for exported files on the SFTP server.																																
format	Sets the format to use for the HTTP transaction log entries in the working.log file.																																
apache	Configures the HTTP transaction logs output to the Apache Common Log format (CLF).																																
custom	Configures the HTTP transaction logs output to the custom log format.																																
<i>string</i>	Quoted log format string containing the custom log format.																																
extended-squid	Configures the HTTP transaction logs output to the Extended Squid log format.																																
log-windows-domain	Logs the Windows domain with an authenticated username if available in HTTP transaction log entries.																																
enable	Enables the remote transaction logging.																																
entry-type	Specifies the type of transaction log entry.																																
all	Sets the SE to send all transaction log messages to the remote syslog server.																																
request-auth-failures	<p>Sets the SE to log to the remote syslog server only those transactions that the SE failed to authenticate with the authentication server.</p> <p>Note Only those authentication failures that are associated with an end user who is attempting to contact the authentication server are logged. The transactions in pending state (that have contacted the authentication server, but waiting for a response from the authentication server) are not logged.</p>																																
facility	Configures a unique facility to create a separate log on the remote syslog host for real-time transaction log entries.																																
<i>parameter</i>	<p>Specifies one of the following facilities:</p> <table> <tr><td>auth</td><td>Authorization system</td></tr> <tr><td>daemon</td><td>System daemons</td></tr> <tr><td>kern</td><td>Kernel</td></tr> <tr><td>local0</td><td>Local use</td></tr> <tr><td>local1</td><td>Local use</td></tr> <tr><td>local2</td><td>Local use</td></tr> <tr><td>local3</td><td>Local use</td></tr> <tr><td>local4</td><td>Local use</td></tr> <tr><td>local5</td><td>Local use</td></tr> <tr><td>local6</td><td>Local use</td></tr> <tr><td>local7</td><td>Local use</td></tr> <tr><td>mail</td><td>Mail system</td></tr> <tr><td>news</td><td>USENET news</td></tr> <tr><td>syslog</td><td>Syslog itself</td></tr> <tr><td>user</td><td>User process</td></tr> <tr><td>uucp</td><td>UUCP system</td></tr> </table>	auth	Authorization system	daemon	System daemons	kern	Kernel	local0	Local use	local1	Local use	local2	Local use	local3	Local use	local4	Local use	local5	Local use	local6	Local use	local7	Local use	mail	Mail system	news	USENET news	syslog	Syslog itself	user	User process	uucp	UUCP system
auth	Authorization system																																
daemon	System daemons																																
kern	Kernel																																
local0	Local use																																
local1	Local use																																
local2	Local use																																
local3	Local use																																
local4	Local use																																
local5	Local use																																
local6	Local use																																
local7	Local use																																
mail	Mail system																																
news	USENET news																																
syslog	Syslog itself																																
user	User process																																
uucp	UUCP system																																
host	Configures the remote syslog server.																																
<i>hostname</i>	Hostname of the remote syslog server.																																
<i>ip-address</i>	IP address of the remote syslog server.																																
port	Configures the port to use when sending transaction log messages to the syslog server.																																

<i>port-num</i>	Port number to use when sending transaction log messages to the syslog server (the default is 514).
rate-limit	Configures the rate at which the transaction logger is allowed to send messages to the remote syslog server.
<i>rate</i>	Rate (number of messages per second) at which the transaction logger is allowed to send messages to the remote syslog server.

Defaults

archive: disabled
enable: disabled
export compress: disabled
export: disabled
file-marker: disabled
archive interval: every day, every one hour
archive max-file-size: 2,000,000 KB
export interval: every day, every one hour
format: apache
logging port *port-num*: 514

Command Modes

Global configuration

Usage Guidelines

SEs that are running Cisco Internet Streamer Release 2.4 software can record all errors and access activities. Each content service module on the SE provides logs of the requests that were serviced. These logs are referred to as transaction logs.

Typical fields in the transaction log are the date and time when a request was made, the URL that was requested, whether it was a cache hit or a cache miss, the type of request, the number of bytes transferred, and the source IP address. Transaction logs are used for problem identification and solving, load monitoring, billing, statistical analysis, security problems, and cost analysis and provisioning.

The translog module on the SE handles transaction logging and supports the Apache Common Log Format (CLF), Extended Squid format, and the World Wide Web Consortium (W3C) customizable logging format.



Note

For RTSP, when you choose the **Repeat** option from the Play menu in the Windows Media player to play media files continuously in a loop, an extra entry is logged in the transaction logs for each playback of the file. This situation occurs mostly with the WMT RTSPU protocol due to the behavior of the player.

Enable transaction log recording with the **transaction-logs enable** command. The transactions that are logged include HTTP and FTP. In addition, Extensible Markup Language (XML) logging for MMS-over-HTTP and MMS-over-RTSP (RTSP over Windows Media Services 9) is also supported.

When enabled, daemons create a *working.log* file in */local1/logs/* on the sysfs volume for HTTP and FTP transactions and a separate *working.log* file in */local1/logs/export* for Windows Media transactions. The posted XML log file from the Windows Media Player to the SE (Windows Media server) can be parsed and saved to the normal WMT transaction logs that are stored on the SE.

The *working.log* file is a link to the actual log file with the timestamp embedded in its filename. When you configure the **transaction-logs archive interval** command, the first transaction that arrives after the interval elapses is logged to the *working.log* file as usual, and then actual log file is archived and a new log file is created. Only transactions subsequent to the archiving event are recorded in the new log file. The *working.log* file is then updated to point to the newly created log file. The transaction log archive file naming conventions are shown in [Table 2-83](#). The SE default archive interval is once an hour every day.

Use the **transaction-logs archive max-file-size** command to specify the maximum size of an archive file. The *working.log* file is archived when it attains the maximum file size if this size is reached before the configured archive interval time.

Use the **transaction-logs file-marker** option to mark the beginning and end of the HTTP, HTTPS, and FTP proxy logs. By examining the file markers of an exported archive file, you can determine whether the FTP process transferred the entire file. The file markers are in the form of dummy transaction entries that are written in the configured log format.

The following example shows the start and end dummy transactions in the default native Squid log format.

- 970599034.130 0 0.0.0.0 TCP_MISS/000 0 NONE TRANSLOG_FILE_START - NONE/- -
- 970599440.130 0 0.0.0.0 TCP_MISS/000 0 NONE TRANSLOG_FILE_END - NONE/- -

Use the **format** option to format the HTTP, HTTPS, and FTP proxy log files for custom format, native Squid or Extended Squid formats, or Apache Common Log Format (CLF).

The **transaction-logs format custom** command allows you to use a *log format string* to log additional fields that are not included in the predefined native Squid or Extended Squid formats or the Apache CLF format. The log format string is a string that contains the tokens listed in [Table 2-77](#) and mimics the Apache log format string. The log format string can contain literal characters that are copied into the log file. Two backslashes (\) can be used to represent a literal backslash, and a backslash followed by a single quotation mark (\') can be used to represent a literal single quotation mark. A literal double quotation mark cannot be represented as part of the log format string. The control characters \t and \n can be used to represent a tab and a new line character, respectively.

[Table 2-77](#) lists the acceptable format tokens for the log format string. The ellipsis (...) portion of the format tokens shown in this table represent an optional condition. This portion of the format token can be left blank, as in %a. If an optional condition is included in the format token and the condition is met, then what is shown in the Value column of [Table 2-77](#) is included in the transaction log output. If an optional condition is included in the format token but the condition is not met, the resulting transaction log output is replaced with a hyphen (-). The form of the condition is a list of HTTP status codes, which may or may not be preceded by an exclamation point (!). The exclamation point is used to negate all of the status codes that follow it, which means that the value associated with the format token is logged if none of the status codes listed after the exclamation point (!) match the HTTP status code of the request. If any of the status codes listed after the exclamation point (!) match the HTTP status code of the request, then a hyphen (-) is logged.

For example, %400,501{User-Agent}i logs the User-Agent header value on 400 errors and 501 errors (Bad Request, Not Implemented) only, and %!200,304,302{Referer}i logs the Referer header value on all requests that did not return a normal status.

The custom format currently supports the following request headers:

- User-Agent
- Referer
- Host
- Cookie

The output of each of the following Request, Referer, and User-Agent format tokens specified in the custom *log format string* is always enclosed in double quotation marks in the transaction log entry:

`%r`

`%{Referer}i`

`%{User-Agent}i`

The `%{Cookie}i` format token is generated without the surrounding double quotation marks, because the Cookie value can contain double quotes. The Cookie value can contain multiple attribute-value pairs that are separated by spaces. We recommend that when you use the Cookie format token in a custom format string, you should position it as the last field in the format string so that it can be easily parsed by the transaction log reporting tools. By using the format token string `\'%{Cookie}i\'` the Cookie header can be surrounded by single quotes (`'`).

The following command can generate the well-known Apache Combined Log Format:

```
transaction-log format custom "[ % { %d } t / % { %b } t / % { %Y } t : % { %H } t : % { %M } t : % { %S } t  
% { %z } t] %r %s %b % { Referer } i % { User-Agent } i"
```

The following transaction log entry example in the Apache Combined Format is configured using the preceding custom format string:

```
[11/Jan/2003:02:12:44 -0800] "GET http://www.cisco.com/swa/i/site_tour_link.gif HTTP/1.1"  
200 3436 "http://www.cisco.com/" "Mozilla/4.0 (compatible; MSIE 5.5; Windows NT 5.0)"
```

Table 2-77 Custom Format “Log Format String” Values

Format Token	Value
<code>%...a</code>	IP address of the requesting client.
<code>%...A</code>	IP address of the SE.
<code>%...B</code> <code>%...b</code>	Bytes sent excluding HTTP headers.
<code>%...c</code>	Connection status when response is completed where X = Connection was aborted before the response was completed. + = Connection can be kept alive after the response is sent. – = Connection is closed after the response is sent.
<code>%...f</code>	Filename.
<code>%...h</code>	Remote host (IP address of the requesting client is logged).
<code>%...H</code>	Request protocol.
<code>%...{Foobar}i</code>	Contents of Foobar: header lines in the request that is sent to the server. The value of Foobar can be one of the following headers: User-Agent, Referer, Host, or Cookie.
<code>%...l</code>	Remote log name. Not implemented on the SE, so a hyphen (-) is logged.
<code>%...m</code>	Request method.

Table 2-77 Custom Format “Log Format String” Values (continued)

Format Token	Value
%...p	Canonical port of the server servicing the request. Not applicable on the SE, so a hyphen (-) is logged.
%...P	Process ID of the child that serviced the request.
%...q	Query string (that is preceded by a question mark (?) if a query string exists; otherwise, it is an empty string).
%...r	First line of the request.
%...s	Status. The translog code always returns the HTTP response code for the request.
%...t	Time in common log time format (or standard English format).
%...{format}t	Time in the form given by the format token specified in Table 2-78 .
%...T	Time consumed to serve the request in seconds (a floating point number with three decimal places).
%...u	Remote user.
%...U	URL path requested not including query strings.
%...v %...V	Value of the host request header field reported if the host appeared in the request. If the host did not appear in the host request header, the IP address of the server specified in the URL is reported.

[Table 2-78](#) specifies the format token for the date and time of the format token %...{format}t that is listed in [Table 2-79](#).

Table 2-78 Format Token for Date and Time

Format Token	Value
%a	Abbreviated weekday name.
%A	Full weekday name.
%b	Abbreviated month name.
%B	Full month name.
%c	Date and time representation.
%C	Century number (year/100) as a 2-digit integer.
%d	Day of the month as a decimal number (01–31).
%D	Equivalent to %m/%d/%y. (In countries other than the USA, %d/%m/%y is common. In an international context, this format is ambiguous and should not be used.)
%e	Similar to %d, the day of the month as a decimal number, but a leading zero is replaced by a space.
%G	ISO 8601 year with the century as a decimal number. The 4-digit year corresponding to the ISO week number (see %V). This format token has the same format and value as %y, except that if the ISO week number belongs to the previous or next year, that year is used instead.
%g	Similar to %G, but without a century; that is, with a 2-digit year (00–99).
%h	Equivalent to %b.

Table 2-78 *Format Token for Date and Time (continued)*

Format Token	Value
%H	Hour as a decimal number using a 24-hour clock (00–23).
%I	Hour as a decimal number using a 12-hour clock (01–12).
%j	Day of the year as a decimal number (001–366).
%k	Hour (24-hour clock) as a decimal number (0–23); single digits are preceded by a blank. (See also %H.)
%l	Hour (12-hour clock) as a decimal number (1–12); single digits are preceded by a blank. (See also %I.)
%m	Month as a decimal number (01–12).
%M	Minute as a decimal number (00–59).
%n	New line character.
%p	Either AM or PM according to the given time value, or the corresponding strings for the current locale. Noon is treated as PM and midnight as AM.
%P	Similar to %p but in lowercase: am or pm or a corresponding string for the current locale.
%r	Time in a.m. or p.m. notation. This format token is equivalent to “%I:%M:%S %p.”
%R	Time in 24-hour notation (%H:%M). For a version including the seconds, see %T below.
%s	Number of seconds since the epoch; that is, since 1970-01-01 00:00:00 UTC.
%S	Second as a decimal number (00–61).
%t	Tab character.
%T	Time in 24-hour notation (%H:%M:%S).
%u	Day of the week as a decimal, 1–7, Monday being 1. See also %w.
%U	Week number of the current year as a decimal number (00–53), starting with the first Sunday as the first day of week 01. See also %V and %W.
%V	ISO 8601:1988 week number of the current year as a decimal number (01–53), where week 1 is the first week that has at least 4 days in the current year, and with Monday as the first day of the week. See also %U and %W.
%w	Day of the week as a decimal (0–6) with Sunday as 0. See also %u.
%W	Week number of the current year as a decimal number (00–53), starting with the first Monday as the first day of week 01.
%x	Date representation without the time.
%X	Time representation without the date.
%y	Year as a decimal number without a century (00–99).
%Y	Year as a decimal number, including the century.
%z	Time zone as an hour offset from GMT. Required to emit RFC822-conformant dates (using %a, %d %b %Y %H:%M:%S %z).
%Z	Time zone or name or abbreviation.
%%	Literal % character.

The Extended Squid log format uses the RFC 981 field of the Squid log format for the username. The Extended Squid format logs the associated username for authentication for each record in the log file, if available. The username is also used for billing purposes.

The W3C Customizable Logging Format is limited in that it was defined from the HTTP web server perspective and does not offer certain web cache-specific custom options such as those supplied by the fixed Squid format. Format tokens that are extensions to the W3C Customized Logging Format support additional Cisco and Squid customized logging fields. These format tokens provide support for a Squid-like logging format from within the W3C customizable token set.

The W3C Customizable Logging Format was extended to include support for the following special token sequence:

```
%...{<translog-token>}C
```

The ellipsis (...) is optional. If specified, it can be a sequence of conditional HTTP response codes separated by commas. The uppercase C defines the extended customizable behavior token set, for which tokens are defined by the <translog-token> directive, which is a two-character token directive.

Table 2-79 lists the existing and new <translog-token> directives from the Extended Squid format, which are not immediately supported by the W3C definitions but are supported in the CDS Release 2.4 software.

Table 2-79 Translog Token Directives

Format Token	Value
%...{es}C	Current time presented as the number of seconds that have elapsed since the Epoch (Jan. 1st. 1970).
%...{em}C	Current number of milliseconds that have elapsed since the Epoch (Jan. 1st. 1970).
%...{te}C	Number of milliseconds that have elapsed until the request was completed.
%...{rd}C	Squid-like cache-status code string (for example, TCP_HIT and TCP_CLIENT_REFRESH_MISS).
%...{cs}C	Number of bytes sent to the client (including the protocol headers).
%...{rh}C	Strict Squid-style hierarchy as it applies to the SE.
%...{rh}SE	Extended Squid-style hierarchy. Same as %...{rh}C except when an outgoing-proxy is explicitly defined and is used to satisfy a request, then the DEFAULT_PARENT/proxy_ip_address is logged instead of the DIRECT/origin_server_ip_address.
%...{rt}C	<p>MIME type of the object in the response, as specified by any protocol headers that define such MIME types. Currently, Cisco Internet Streamer Release 2.4 software does not support logging the MIME type of the object that is being requested, and a hyphen (-) is logged instead.</p> <p>Note This restriction also applies to the Squid and Extended Squid logging formats.</p> <p>Tip A MIME-type association enables the browser to invoke a particular application when it encounters an object with a particular MIME-type suffix. A set of default association rules covers the common object types on the Internet. You can edit, add, or delete these MIME-type association rules in the browsers. For example, through a MIME-type association, the client browser launches the Adobe Acrobat reader when it encounters a *.pdf file, and it launches the Windows Media Player when it encounters an *.asf or *.asx file.</p>

Table 2-79 Translog Token Directives (continued)

Format Token	Value
%...{ru}C	URL being requested including any additional query strings.
%...{as}C	Application-specific information. Certain request handling applications might want to log a certain string here, which is supported as part of the Squid format specification. For example, SmartFilter URL filtering logs information where this token sequence is used.

In addition to the tokens listed in [Table 2-79](#), you can condense multiple %...{xx}C style tokens into a single embedded token sequence within the %...{xx}C style. A limited customized logging string validation mechanism has been implemented for all the %...{xy}C style format tokens. This mechanism ensures that the tokens are valid and rejects invalid tokens. To condense multiple style tokens into a single embedded token sequence, you must specify multiple tokens within the {} braces and prefix each token with the percent (%) symbol as follows:

```
%{rh}C %{rt}C %{as}C
```

can be reexpressed in a condensed embedded token format as the following:

```
%{%rh %rt %as}C
```

The command line syntax accepts single tokens represented as the following:

```
%{%rh}C
```

and

```
%{rh}C
```

as equivalents.

Any character that is not part of an embedded token sequence (for example, the space character) is repeated verbatim in the output file.

The above set of tokens allow you to configure an extended Squid-like format line within the W3C Customizable Logging format specification as follows:

```
%{es}C.%{em}C %{te}C %a %{rd}C/%s %{cs}C %m %{ru}C %u %{rh}C %{rt}C %{as}C
```

The following is an example of a Extended Squid-like format that specifies that user-readable time-stamps are used instead of Squid's "seconds-since-epoch" time-stamp format, and that a configured out-going proxy (as specified by "%...{rH}C") is logged:

```
[%{%d/%b/%Y:%H:%M:%S %z}t] %{te}C %a %{rd}C/%s %{cs}C %m %{ru}C %u %{rH}C %{rt}C  
%{as}C
```

Unknown or unsupported translog tokens are logged within the log file as the characters that made up the token. For example, %{xy}C is logged into the log file as xy. All characters outside of a token specification sequence are repeated verbatim within the log file.

Sanitizing Transaction Logs

Use the **sanitized** option to disguise the IP address of clients in the transaction log file. The default is that transaction logs are not sanitized. A sanitized transaction log disguises the network identity of a client by changing the IP address in the transaction logs to 0.0.0.0.

The **no** form of this command disables the sanitize feature. The **transaction-logs sanitize** command does not affect the client IP (%a) value associated with a custom log format string that is configured with the CLI (configured with the **transaction-logs format custom string** global configuration command in

which the string is the quoted log format string that contains the custom log format). To hide the identity of the client IP in the custom log format, either hard code 0.0.0.0 in the custom log format string or exclude the %a token, which represents the client IP, from the format string.

Exporting Transaction Log Files

To facilitate the postprocessing of cache log files, you could export transaction logs to an external host. This feature allows log files to be exported automatically by FTP to an external host at configurable intervals. The username and password used for FTP are configurable. The directory to which the log files are uploaded is also configurable.

The log files automatically have the following filename:

type_ipaddr_yyyymmdd_hhmmss.txt

where

- *type* represents the type of log file with *celog* for cache logs such as HTTP, HTTPS, and FTP, and *mms_export* for the WMT logs.
- *ipaddr* represents the SE IP address.
- *yyymmdd_hhmmss* represents the date and time when the log was archived for export.



Note

WMT logs have no .txt extension in the filename.

Exporting and Archiving Intervals

The transaction log archive and export functions are configured with the following commands:

- The **transaction-logs archive interval** global configuration command allows the administrator to specify when the *working.log* file is archived.
- The **transaction-logs export interval** global configuration command allows the administrator to specify when the archived transaction logs are exported.

The following limitations apply:

- When the interval is scheduled in units of hours, the value must divide evenly into 24. For example, the interval can be every 4 hours, but not every 5 hours.
- When the interval is scheduled in units of minutes, the value must divide evenly into 60.
- Only the more common choices of minutes are supported. For example, the interval can be 5 minutes or 10 minutes, but not 6 minutes.
- The selection of interval alignment is limited. If an interval is configured for every 4 hours, it will align with midnight. It cannot align with 12:30 or with 7 a.m.
- The feature does not support different intervals within a 24-hour period. For example, it does not support an interval that is hourly during regular business hours and then every 4 hours during the night.

Transaction Log Archive Filenaming Convention

The archive transaction log file is named as follows for HTTP and WMT caching:

celog_10.1.118.5_20001228_235959.txt

mms_export_10.1.118.5_20001228_235959

If the **export compress** feature is enabled when the file is exported, then the file extension will be .gz after the file is compressed for the export operation, as shown in the following example:

```
celog_10.1.118.5_20001228_235959.txt.gz
```

```
mms_export_10.1.118.5_20001228_235959.gz
```

Table 2-83 describes the name elements.

Table 2-80 *Archive Log Name Element Descriptions*

Sample of Element	Description
acqdist_	Acquisition and distribution archive log file.
celog_	HTTP caching proxy server archive file.
cifs_server_	Windows file sharing server archive file.
cseaccess	Cisco Streaming Engine archive file.
icap_	ICAP server archive file.
mms_export_	Standard Windows Media Services 4.1 caching proxy server archive file.
mms_export_e_wms_41_	Extended Windows Media Services 4.1 caching proxy server archive file.
mms_export_wms_90_	Standard Windows Media Services 9.0 caching proxy server archive file.
mms_export_e_wms_90_	Extended Windows Media Services 9.0 caching proxy server archive file.
rproxyaccess.log.	RealProxy archive file.
rmsvraccess.log.	RealSubscriber archive file.
tftp_server_	TFTP server archive file.
tvout_	TV-out program archive file.
10.1.118.5_	IP address of the SE creating the archive file.
20001228_	Date on which the archive file was created (yyyy/mm/dd).
235959	Time when the archive file was created (hh/mm/ss).

Table 2-81 lists the directory names and the corresponding examples of the archive filenames.

Table 2-81 *Archive Filename Examples and Directories*

Directory	Archive Filename
logs	celog_10.1.94.4_20050310_231500.txt
logs/export	mms_export_10.1.94.4_20050315_001545
logs/export/extended-wms-41	mms_export_e_wms_41_10.1.94.4_20050315_001545
logs/wms-90	mms_export_wms_90_10.1.94.4_20050315_001545
logs/export/extended-wms-90	mms_export_e_wms_90_10.1.94.4_20050315_001545
logs/acqdist	acqdist_10.1.94.4_20050315_001545
logs/cifs_server	cifs_server_10.1.94.4_20050315_001545
logs/cisco-streaming-engine	cseaccess10.1.94.4_050315000.log
logs/icap	icap_10.1.94.4_20050315_001545
logs/real-proxy	rproxyaccess.log.10.1.94.4_.20050315_001545
logs/real-subscriber	rmsvraccess.log.10.1.94.4_.20050315_001545

Table 2-81 *Archive Filename Examples and Directories (continued)*

Directory	Archive Filename
logs/tftp_server	tftp_server_10.1.94.4_20050315_001545
logs/tvout	tvout_10.1.94.4_20050315_001545

Compressing Archive Files

The **transaction-logs export compress** option compresses an archive into a gzip file format before exporting it. Compressing the archive file uses less disk space on both the SE and the FTP export server. The compressed file uses less bandwidth when transferred. The archive filename of the compressed file has the extension .gz.

Exporting Transaction Logs to External FTP Servers

The **transaction-logs export ftp-server** option can support up to four FTP servers. To export transaction logs, you must first enable the feature and configure the FTP server parameters. The following information is required for each target FTP server:

- FTP server IP address or the hostname
The SE translates the hostname with a DNS lookup and then stores the IP address in the configuration.
- FTP user login and user password
- Path of the directory where transferred files are written
Use a fully qualified path or a relative path for the user login. The user must have write permission to the directory.

Use the **no** form of the **transaction-logs export enable** command to disable the entire transaction logs feature while retaining the rest of the configuration.

Exporting Transaction Logs to External SFTP Servers

Use the **transaction-logs export sftp-server** option to export transaction logs. You must first enable the feature and configure the Secure File Transfer Protocol (SFTP) server parameters. The following information is required for each target SFTP server:

- SFTP server IP address or the hostname
The SE translates the hostname with a DNS lookup and then stores the IP address in the configuration.
- SFTP user login and user password
- Path of the directory where transferred files are written
Use a fully qualified path or a relative path for the user login. The user must have write permission to the directory.

Use the **no** form of the **transaction-logs export enable** command to disable the entire transaction logs feature while retaining the rest of the configuration.

Receiving a Permanent Error from the External FTP Server

A permanent error (Permanent Negative Completion Reply, RFC 959) occurs when the FTP command to the server cannot be accepted, and the action does not take place. Permanent errors can be caused by invalid user logins, invalid user passwords, and attempts to access directories with insufficient permissions.

When an FTP server returns a permanent error to the SE, the export is retried at 10-minute intervals or sooner if the configured export interval is sooner. If the error is a result of a misconfiguration of the **transaction-logs export ftp server** command, then you must re-enter the SE parameters to clear the error condition. The **show statistics transaction-logs** command displays the status of logging attempts to export servers.

The **show statistics transaction-logs** command shows that the SE failed to export archive files.

The **transaction-logs format** command has four options: **extended-squid**, **apache**, and **custom**.

Use the **no** form of the **transaction-logs export enable** command to disable the entire transaction logs feature while retaining the rest of the configuration.

Configuring Intervals Between 1 Hour and 1 Day

The archive or export interval can be set for once a day with a specific time stamp. It can also be set for hour frequencies that align with midnight. For example, every 4 hours means archiving occurs at 0000, 0400, 0800, 1200, and 1600. It is not possible to archive at half-hour intervals such as 0030, 0430, or 0830. The following intervals are acceptable: 1, 2, 3, 4, 6, 8, 12, and 24.

Configuring Intervals of 1 Hour or Less

The interval can be set for once an hour with a minute alignment. It can also be set for frequencies of less than an hour; these frequencies will align with the top of the hour. Every 5 minutes means that archiving will occur at 1700, 1705, and 1710.

Configuring Export Interval on Specific Days

The export interval can be set for specific days of the week at a specific time. One or more days can be specified. The default time is midnight.

You must be aware that archived logs are automatically deleted when free disk space is low. It is important to select an export interval that exports files frequently enough so that files are not automatically removed prior to export.

Monitoring HTTP Request Authentication Failures in Real Time

Cisco Internet Streamer Release 2.4 software supports sending HTTP transaction log messages to a remote syslog server so that you can monitor the remote syslog server for HTTP request authentication failures in real time. This real-time transaction log allows you to monitor transaction logs in real time for particular errors such as HTTP request authentication errors. The existing transaction logging to the local file system remains unchanged.



Note

Because system logging (syslog) occurs through UDP, the message transport to the remote syslog host is not reliable.

Examples

The following example shows how to configure an FTP server:

```
ServiceEngine(config)#transaction-logs export ftp-server 10.1.1.1 mylogin mypasswd
/ftpdirectory
```

```
ServiceEngine(config)#transaction-logs export ftp-server myhostname mylogin mypasswd
/ftpdirectory
```

The following example shows how to delete an FTP server:

```
ServiceEngine(config)#no transaction-logs export ftp-server 10.1.1.1
ServiceEngine(config)#no transaction-logs export ftp-server myhostname
```

Use the **no** form of the command to disable the entire transaction log export feature while retaining the rest of the configuration:

```
ServiceEngine(config)#no transaction-logs export enable
```

The following example shows how to change a username, password, or directory:

```
ServiceEngine(config)#transaction-logs export ftp-server 10.1.1.1 mynewname mynewpass
/newftpdirectory
```


Note

For security reasons, passwords are never displayed.

The following example shows how to restart the export of archive transaction logs:

```
ServiceEngine(config)# transaction-logs export ftp-server 172.16.10.5
goodlogin pass /ftpdirectory
```

The following example shows how to delete an SFTP server from the current configuration:

```
ServiceEngine(config)# no transaction-logs export sftp-server sftphostname
```

The following examples show how to configure the archiving intervals:

```
ServiceEngine(config)#transaction-logs archive interval every-day
at          Specify the time at which to archive each day
every       Specify the interval in hours. It will align with midnight
```

```
ServiceEngine(config)#transaction-logs archive interval every-day at
<0-23>:     Time of day at which to archive (hh:mm)
```

```
ServiceEngine(config)#transaction-logs archive interval every-day every
<1-24>     Interval in hours: {1, 2, 3, 4, 6, 8, 12 or 24}
```

The following example shows that the SE has failed to export archive files:

```
ServiceEngine#show statistics transaction-logs
Transaction Log Export Statistics:
```

```
Server:172.16.10.5
  Initial Attempts:1
  Initial Successes:0
  Initial Open Failures:0
  Initial Put Failures:0
  Retry Attempts:0
  Retry Successes:0
  Retry Open Failures:0
  Retry Put Failures:0
  Authentication Failures:1
  Invalid Server Directory Failures:0
```

The following example shows how to correct a misconfiguration:

```
ServiceEngine(config)#transaction-logs export ftp-server 10.1.1.1 goodlogin pass
/ftpdirectory
```

The working.log file and archived log files are listed for HTTP and WMT.

The following example shows how to export transaction logs to an SFTP server:

```
ServiceEngine(config)#transaction-logs export sftp-server 10.1.1.100 mylogin mypasswd
/mydir
```

The following example shows how to archive every 4 hours and align with the midnight local time (0000, 0400, 0800, 1200, 1600, and 2000):

```
ServiceEngine(config)#transaction-logs archive interval every-day every 4
```

The following example shows how to export once a day at midnight local time:

```
ServiceEngine(config)#transaction-logs export interval every-day every 24
```

The following example shows how to configure export intervals:

```
ServiceEngine(config)#transaction-logs archive interval every-hour ?  
  at          Specify the time at which to archive each day  
  every       Specify interval in minutes. It will align with top of the hour
```

```
ServiceEngine(config)#transaction-logs archive interval every-hour at ?  
  <0-59>      Specify the minute alignment for the hourly archive  
ServiceEngine(config)#transaction-logs archive interval every-hour every ?  
  <2-30>      Interval in minutes: {2, 5, 10, 15, 20, 30}
```

Related Commands

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

type

To display the contents of a file, use the **type** EXEC command.

type *filename*

Syntax Description	<i>filename</i> Name of file.
Defaults	No default behavior or values.
Command Modes	EXEC
Usage Guidelines	Use this command to display the contents of a file within any SE file directory. This command may be used to monitor features such as transaction logging or system logging (syslog).

Examples

The following example displays the syslog file on the SE:

```
ServiceEngine#type /local1/syslog.txt
```

```
Jan 10 22:02:46 (none) populate_ds: %SE-CLI-5-170050: Cisco Internet Streamer CDS Software
starts booting
Jan 10 22:02:47 (none) create_etc_hosts.sh: %SE-CLI-5-170051: HOSTPLUSDOMAIN: NO-HOSTNAME
Jan 10 22:02:47 NO-HOSTNAME : %SE-CLI-5-170053: Recreated etc_hosts (1, 0)
Jan 10 22:02:48 NO-HOSTNAME Nodemgr: %SE-NODEMGR-5-330082: [CLI_VER_NTP] requests stop
service ntpd
Jan 10 22:02:49 NO-HOSTNAME Nodemgr: %SE-NODEMGR-5-330082: [ver_tvout] requests stop
service tvoutsvr
Jan 10 22:02:50 NO-HOSTNAME Nodemgr: %SE-NODEMGR-5-330084: [ver_rtspg] requests restart
service rtspg
Jan 10 22:02:50 NO-HOSTNAME Nodemgr: %SE-NODEMGR-5-330082: [ver_iptv] requests stop
service sbss
Jan 10 22:02:51 NO-HOSTNAME Nodemgr: %SE-NODEMGR-5-330080: [ver_telnetd] requests start
service telnetd
Jan 10 22:02:52 NO-HOSTNAME Nodemgr: %SE-NODEMGR-5-330082: [ver_wmt] requests stop service
wmt_mms
Jan 10 22:02:53 NO-HOSTNAME Nodemgr: %SE-NODEMGR-5-330082: [ver_wmt] requests stop service
wmt_logd
Jan 10 22:02:55 NO-HOSTNAME Nodemgr: %SE-NODEMGR-5-330082: [Unknown] requests stop service
mcast_sender
Jan 10 22:02:55 NO-HOSTNAME Nodemgr: %SE-NODEMGR-5-330082: [Unknown] requests stop service
mcast_receiver
Jan 10 22:02:56 NO-HOSTNAME Nodemgr: %SE-NODEMGR-5-330024: Service 'populate_ds' exited
normally with code 0
Jan 10 22:02:56 NO-HOSTNAME Nodemgr: %SE-NODEMGR-5-330040: Start service 'parser_server'
using: '/ruby/bin/parser_server' with pid: 1753
Jan 10 22:02:56 NO-HOSTNAME Nodemgr: %SE-NODEMGR-5-330040: Start service
'syslog_bootup_msgs' using: '/ruby/bin/syslog_bootup_msgs' with pid:
1754
Jan 10 22:02:56 NO-HOSTNAME syslog_bootup_msgs: %SE-SYS-5-900001: <4>Linux version 2.4.16
(cnbuild@builder2.cisco.com) (gcc version 3.0.4) #1
SMP Fri Jan 7 19:26:58 PST 2005
```



```

Jan 10 22:02:56 NO-HOSTNAME syslog_bootup_msgs: %SE-SYS-5-900001: <6>setup.c: handling
flash window at [15MB..16MB)
Jan 10 22:02:56 NO-HOSTNAME syslog_bootup_msgs: %SE-SYS-5-900001: <6>BIOS-provided
physical RAM map:
Jan 10 22:02:56 NO-HOSTNAME syslog_bootup_msgs: %SE-SYS-5-900001: <4> BIOS-e820:
0000000000000000 - 0000000000009ec00 (usable)
Jan 10 22:02:56 NO-HOSTNAME syslog_bootup_msgs: %SE-SYS-5-900001: <4> BIOS-e820:
0000000000009ec00 - 000000000000a0000 (reserved)
Jan 10 22:02:56 NO-HOSTNAME syslog_bootup_msgs: %SE-SYS-5-900001: <4> BIOS-e820:
000000000000e0800 - 00000000000100000 (reserved)
Jan 10 22:02:56 NO-HOSTNAME syslog_bootup_msgs: %SE-SYS-5-900001: <4> BIOS-e820:
00000000000100000 - 00000000000f00000 (usable)
Jan 10 22:02:56 NO-HOSTNAME syslog_bootup_msgs: %SE-SYS-5-900001: <4> BIOS-e820:
00000000000f00000 - 000000000001000000 (reserved)
Jan 10 22:02:56 NO-HOSTNAME syslog_bootup_msgs: %SE-SYS-5-900001: <4> BIOS-e820:
000000000001000000 - 0000000000010000000 (usable)
Jan 10 22:02:56 NO-HOSTNAME syslog_bootup_msgs: %SE-SYS-5-900001: <4> BIOS-e820:
0000000000010000000 - 00000000000100000000 (reserved)
Jan 10 22:02:56 NO-HOSTNAME syslog_bootup_msgs: %SE-SYS-5-900001: <6>setup.c: reserved
bootmem for INITRD_START = 0x6000000, INITRD_SIZE = 117
09348
Jan 10 22:02:56 NO-HOSTNAME syslog_bootup_msgs: %SE-SYS-5-900001: <4>On node 0 totalpages:
65536
Jan 10 22:02:56 NO-HOSTNAME syslog_bootup_msgs: %SE-SYS-5-900001: <4>zone(0): 4096 pages.
Jan 10 22:02:56 NO-HOSTNAME syslog_bootup_msgs: %SE-SYS-5-900001: <4>zone(1): 61440 pages.
Jan 10 22:02:56 NO-HOSTNAME syslog_bootup_msgs: %SE-SYS-5-900001: <4>zone(2): 0 pages.
Jan 10 22:02:56 NO-HOSTNAME syslog_bootup_msgs: %SE-SYS-5-900001: <4>Local APIC disabled
by BIOS -- reenabling.
Jan 10 22:02:56 NO-HOSTNAME syslog_bootup_msgs: %SE-SYS-5-900001: <4>Found and enabled
local APIC!
Jan 10 22:02:56 NO-HOSTNAME syslog_bootup_msgs: %SE-SYS-5-900001: <4>Kernel command line:
root=/dev/ram ramdisk_size=100000 ramdisk_start=0x60
00000 console=ttyS0,9600n8
Jan 10 22:02:56 NO-HOSTNAME syslog_bootup_msgs: %SE-SYS-5-900001: <6>Initializing CPU#0
--More--
.
.

```

Related Commands

cpfile
 dir
 lls
 ls
 mkfile

type-tail

To view a specified number of lines of the end of a log file or to view the end of the file continuously as new lines are added to the file, use the **type-tail** command in EXEC mode.

type-tail *filename* [*line* | **follow**]

Syntax Description

<i>filename</i>	File to be examined.
<i>line</i>	(Optional) The number of lines from the end of the file to be displayed (1–65535).
follow	(Optional) Displays the end of the file continuously as new lines are added to the file.

Defaults

Ten lines shown

Command Modes

EXEC

Usage Guidelines

This command allows you to monitor a log file by letting you view the end of the file. You can specify the number of lines at the end of the file that you want to view, or you can follow the last line of the file as it continues to log new information. To stop the last line from continuously scrolling, press **Ctrl-C**.

Examples

The following example shows the list of log files in the /local1 directory:

```
stream-ServiceEngine#ls /local1
WS441
Websense
WebsenseEnterprise
Websense_config_backup
WsInstallLog
badfile.txt
codecoverage
core.stunnel.5.3.0.b100.cnbuild.5381
core_dir
crash
crka.log
cse_live
cse_vod
dbdowngrade.log
dbupgrade.log
downgrade
errorlog
http_authmod.unstrip
index.html
logs
lost+found
netscape-401-proxy
netscape-401-proxy1
netscape-dump
newwebsense
oldWsInstallLog
preload_dir
proxy-basic1
```

```

proxy1
proxy2
proxy3
proxy4
proxy5
proxy6
proxy7
proxy8
proxyreply
proxyreply-407
real_vod
ruby.bin.cli_fix
ruby.bin.no_ws_fix
ruby.bin.ws_edir_fix
sa
service_logs
smartfilter
smfnaveen
superwebsense
syslog.txt
syslog.txt.1
syslog.txt.2
temp
two.txt
url.txt
urllist.txt
var
vpd.properties
websense.pre-200
webtarball44
webtarball520
wmt_vod
ws_upgrade.log

```

The following example displays the last ten lines of the syslog.txt file. In this example, the number of lines to display is not specified; however, ten lines is the default.

```

stream-ServiceEngine#type-tail /local1/syslog.txt
Oct  8 21:49:15 stream-ce syslog:(26830)TRCE:input_serv.c:83-> select_with
return 0, ready = 0
Oct  8 21:49:15 stream-ce syslog:(26832)TRCE:al_master.c:246-> select_with
return 0, ready = 0
Oct  8 21:49:15 stream-ce syslog:(26832)TRCE:in_mms.c:1747-> tv = NULL
Oct  8 21:49:17 stream-ce syslog:(26830)TRCE:input_serv.c:83-> select_with
return 0, ready = 0
Oct  8 21:49:17 stream-ce syslog:(26832)TRCE:al_master.c:246-> select_with
return 0, ready = 0
Oct  8 21:49:17 stream-ce syslog:(26832)TRCE:in_mms.c:1747-> tv = NULL
Oct  8 21:49:19 stream-ce syslog:(26830)TRCE:input_serv.c:83-> select_with
return 0, ready = 0
Oct  8 21:49:19 stream-ce syslog:(26832)TRCE:al_master.c:246-> select_with
return 0, ready = 0
Oct  8 21:49:19 stream-ce syslog:(26832)TRCE:in_mms.c:1747-> tv = NULL
Oct  8 21:49:21 stream-ce syslog:(26830)TRCE:input_serv.c:83-> select_with
return 0, ready = 0

```

The following example displays the last 20 lines of the syslog.txt file:

```

stream-ServiceEngine#type-tail /local1/syslog.txt 20
Oct  8 21:49:11 stream-ce syslog:(26832)TRCE:al_master.c:246-> select_with
return 0, ready = 0
Oct  8 21:49:11 stream-ce syslog:(26832)TRCE:in_mms.c:1747-> tv = NULL
Oct  8 21:49:13 stream-ce syslog:(26830)TRCE:input_serv.c:83-> select_with
return 0, ready = 0

```

```

Oct  8 21:49:13 stream-ce syslog:(26832)TRCE:al_master.c:246-> select_with
return 0, ready = 0
Oct  8 21:49:13 stream-ce syslog:(26832)TRCE:in_mms.c:1747-> tv = NULL
Oct  8 21:49:15 stream-ce syslog:(26830)TRCE:input_serv.c:83-> select_with
return 0, ready = 0
Oct  8 21:49:15 stream-ce syslog:(26832)TRCE:al_master.c:246-> select_with
return 0, ready = 0
Oct  8 21:49:15 stream-ce syslog:(26832)TRCE:in_mms.c:1747-> tv = NULL
Oct  8 21:49:17 stream-ce syslog:(26830)TRCE:input_serv.c:83-> select_with
return 0, ready = 0
Oct  8 21:49:17 stream-ce syslog:(26832)TRCE:al_master.c:246-> select_with
return 0, ready = 0
Oct  8 21:49:17 stream-ce syslog:(26832)TRCE:in_mms.c:1747-> tv = NULL
Oct  8 21:49:19 stream-ce syslog:(26830)TRCE:input_serv.c:83-> select_with
return 0, ready = 0
Oct  8 21:49:19 stream-ce syslog:(26832)TRCE:al_master.c:246-> select_with
return 0, ready = 0
Oct  8 21:49:19 stream-ce syslog:(26832)TRCE:in_mms.c:1747-> tv = NULL
Oct  8 21:49:21 stream-ce syslog:(26830)TRCE:input_serv.c:83-> select_with
return 0, ready = 0
Oct  8 21:49:21 stream-ce syslog:(26832)TRCE:al_master.c:246-> select_with
return 0, ready = 0
Oct  8 21:49:21 stream-ce syslog:(26832)TRCE:in_mms.c:1747-> tv = NULL
Oct  8 21:49:23 stream-ce syslog:(26830)TRCE:input_serv.c:83-> select_with
return 0, ready = 0
Oct  8 21:49:23 stream-ce syslog:(26832)TRCE:al_master.c:246-> select_with
return 0, ready = 0
Oct  8 21:49:23 stream-ce syslog:(26832)TRCE:in_mms.c:1747-> tv = NULL

```

The following example follows the file as it grows:

```

stream-ServiceEngine#type-tail /local1/syslog.txt ?
  <1-65535> The numbers of lines from end
  follow    Follow the file as it grows
  <cr>
stream-ServiceEngine#type-tail /local1/syslog.txt follow
Oct  8 21:49:39 stream-ce syslog:(26832)TRCE:in_mms.c:1747-> tv = NULL
Oct  8 21:49:41 stream-ce syslog:(26830)TRCE:input_serv.c:83-> select_with
return 0, ready = 0
Oct  8 21:49:41 stream-ce syslog:(26832)TRCE:al_master.c:246-> select_with
return 0, ready = 0
Oct  8 21:49:41 stream-ce syslog:(26832)TRCE:in_mms.c:1747-> tv = NULL
Oct  8 21:49:43 stream-ce syslog:(26830)TRCE:input_serv.c:83-> select_with
return 0, ready = 0
Oct  8 21:49:43 stream-ce syslog:(26832)TRCE:al_master.c:246-> select_with
return 0, ready = 0
Oct  8 21:49:43 stream-ce syslog:(26832)TRCE:in_mms.c:1747-> tv = NULL
Oct  8 21:49:45 stream-ce syslog:(26830)TRCE:input_serv.c:83-> select_with
return 0, ready = 0
Oct  8 21:49:45 stream-ce syslog:(26832)TRCE:al_master.c:246-> select_with
return 0, ready = 0
Oct  8 21:49:45 stream-ce syslog:(26832)TRCE:in_mms.c:1747-> tv = NULL
Oct  8 21:49:47 stream-ce syslog:(26830)TRCE:input_serv.c:83-> select_with
return 0, ready = 0
Oct  8 21:49:47 stream-ce syslog:(26832)TRCE:al_master.c:246-> select_with
return 0, ready = 0
Oct  8 21:49:47 stream-ce syslog:(26832)TRCE:in_mms.c:1747-> tv = NULL
Oct  8 21:49:49 stream-ce syslog:(26830)TRCE:input_serv.c:83-> select_with
return 0, ready = 0
Oct  8 21:49:49 stream-ce syslog:(26832)TRCE:al_master.c:246-> select_with
return 0, ready = 0
Oct  8 21:49:49 stream-ce syslog:(26832)TRCE:in_mms.c:1747-> tv = NULL

```

undebug

To disable debugging functions, use the **undebug** EXEC command.

undebug *option*

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

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

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

Usage Guidelines	We recommend that you use the debug and undebug commands only at the direction of Cisco TAC. See the “debug” section on page 2-73 for more information about debug functions.
-------------------------	---

Related Commands	debug no debug show debugging
-------------------------	--

url-signature

The CDS uses a combination of key owners, key ID numbers, and a word value to generate URL signature keys. To configure the url signature, use the **url-signature** global configuration command.

url-signature key-id-owner *num* **key-id-number** *num* {**key** *word* | **public key** *url* [**symmetric key** *word* | **private key** *url*]}

no url-signature key-id-owner *num* **key-id-number** *num*

Syntax Description

key-id-owner	Configures the owner ID for this key.
<i>num</i>	Specifies the ID for the owner of this key (1-32).
key-id-number	Configures the number ID for this key.
<i>num</i>	Specifies the ID for the number of this key (1-16).
key	Configures the encryption key for signing a URL.
<i>word</i>	Text of encryption key (16 characters maximum, no spaces).
public-key	Configures the Public Key file location (PEM).
<i>url</i>	The URL from where the Public Key file can be downloaded (54 characters maximum).
symmetric-key	(Optional) Configure the Symmetric Key.
<i>word</i>	The Symmetric Key (Must be 16 characters, no spaces).
private-Key	(Optional) Configures the Private Key file location (PEM).
<i>url</i>	The URL from where the Private Key file can be downloaded (54 characters maximum).

Command Modes

Global configuration

Usage Guidelines

Service Rules for Directing Requests to a Policy Server

If your network is configured to work with Camiant PCMM-compliant third-party policy servers for servicing requests that require guaranteed bandwidth, you can use the following rule patterns and rule actions to filter the requests and to direct them to the policy server. The rule patterns and rule actions also enable you to generate URL signatures in the response for a valid request for a Windows Media metafile (.asx file extension), Movie Streamer file, or Flash Media Streaming file, and to validate the URL signature on incoming requests to the SE. URL signature key authentication is implemented by using the generate-url-signature and validate-url-signature rule actions that can be applied to specific rule patterns.



Note

Movie Streamer and Flash Media Streaming support URL signing. Flash Media Streaming only supports the following actions: allow, block, and validate-url-signature.

The following table lists the rule patterns that support the use-icap-service rule action for directing requests that require guaranteed bandwidth to the third-party policy server:

Rule Pattern	Description
url-regex	Filters the request based on any regular expression in the URL.
domain	Filters the request based on the domain name specified.
src-ip	Filters the request based on the IP address of the source.
header-field user-agent	Filters the request based on the user agent specified in the request header.
header-field referer	Filters the request based on the referer in the request header.
header-field request-line	Filters the request based on the request line in the request header.

You can set the use-icap-service rule action for any of the rule patterns above. If the request matches the parameters that you have set for the rule pattern, then the SE redirects the request to the third-party policy server using ICAP services. However, you must make sure that your network is configured to interoperate with the third-party policy server using ICAP services. You can set up the necessary ICAP configurations from the ICAP Services page. You can also use the rule pattern and rule action to generate URL signatures in the response for a valid request for a Windows Media metafile. You can use the following rule patterns to filter out requests for which you want to generate a URL signature key:

Rule Pattern	Description
url-regex	Filters the request based on any regular expression in the URL.
domain	Filters the request based on the domain name specified.

For the rule patterns mentioned above, you can set the following rule actions:

Rule Action	Description
generate-url-signature	Generates the URL signatures in the Windows Media metafile response associated with pre-positioned content, based on the SE configuration for the URL signature and this rule action.
validate-url-signature	Validates the URL signature for a request by using the configuration on your SE for the URL signature and allows the request processing to proceed for this request.



Note

When configuring service rules, you must configure the same service rules on all SEs participating in a delivery service in order for the service rules to be fully implemented. The rule action must be common for all client requests because the SR may redirect a client request to any SE in a delivery service depending on threshold conditions.

URL Signing Components

However, because any of these strings in the URL could potentially be edited manually and circumvented by any knowledgeable user, it is important to generate and attach a signature to the URL. This can be achieved by attaching a keyed hash to the URL, using a secret key shared only between the signer (the portal) and the validating component (CDS).

The URL signing script offers three different versions:

- MD5 hash algorithm
- SHA-1 hash algorithm
- SHA-1 hash algorithm with the protocol removed from the beginning of the URL

When a URL is signed for RTSP and a player does a fallback to HTTP for the same URL, the validation fails because the URL signature includes RTSP. If the URL signature does not include the protocol, the fallback URL is validated correctly even though the protocol is HTTP.

If you do not specify a version for the script, MD5 is used and the SIGV string in the script is not added.

At the portal, URLs can be signed for a particular user (client IP address) and expiry time using a URL signing script. The URL signing script example included in this section requires Python 2.3.4 or higher.

Following is an example of the URL signing script using the MD5 security hash algorithm:

```
python cds-ims-urlsign.py http://www.cisco.com/index.html 8.1.0.4 200000 1 2 cisco
```

An example of the resulting signed URL follows:

```
http://www.cisco.com/index.html?IS=0&ET=1241194518&CIP=8.1.0.4&KO=1&KN=2&US=deebacde45bf716071c8b2fecaa755b9
```

If you specify version 1 for the script, SHA-1 is used and the SIGV=1 string is added.

Following is an example of the URL signing script using the SHA-1 security hash algorithm:

```
python cds-ims-urlsign.py http://www.cisco.com/index.html 8.1.0.4 200000 1 2 cisco 1
```

An example of the resulting signed URL follows:

```
http://www.cisco.com/index.html?SIGV=1&IS=0&ET=1241194679&CIP=8.1.0.4&KO=1&KN=2&US=8349348ffac7987d11203122a98e7e64e410fa18
```

If you specify version 2 for the script, SHA-1 is used. The protocol from the beginning of the URL is also removed before the signature is generated, and the SIGV=2 string is added. The protocol is RTSP, HTTP, or RTMP. The URL is signed without the protocol, but the final signed URL is printed with the protocol.

Following is an example of the URL signing script using the SHA-1 security hash algorithm with version 2 specified:

```
python cds-ims-urlsign.py http://www.cisco.com/index.html 8.1.0.4 200000 1 2 cisco 2
```

An example of the resulting signed URL follows:

```
http://www.cisco.com/index.html?SIGV=2&IS=0&ET=1241194783&CIP=8.1.0.4&KO=1&KN=2&US=68b5f5ed97d1255a0ec42a42a4f779e794df679c
```

For additional information on URL Signing, see the “Configuring URL Signing” section and the “URL Signing and Validation” appendix in the *Cisco Internet Streamer CDS 2.4 Software Configuration Guide*.

Examples

Following is an example of generating and encrypting the public key and private key using the **url-signature** command:

```
ServiceEngine(config)#url-signature key-id-owner 1 key-id-number 10 public-key  
http://1.1.1.1/ec_pub_key private-key http://1.1.1.1/ec_pub_key symmetric-key
```

Following is an example of the URL signing script using the MD5 security hash algorithm:

```
python cds-ims-urlsign.py http://www.cisco.com/index.html 8.1.0.4 200000 1 2 cisco
```


An example of the resulting signed URL follows:

```
http://www.cisco.com/index.html?IS=0&ET=1241194518&CIP=8.1.0.4&KO=1&KN=2&US=deebacde45bf716071c8b2fecaa755b9
```

If you specify version 1 for the script, SHA-1 is used and the SIGV=1 string is added.

Following is an example of the URL signing script using the SHA-1 security hash algorithm:

```
python cds-ims-urlsign.py http://www.cisco.com/index.html 8.1.0.4 200000 1 2 cisco 1
```

An example of the resulting signed URL follows:

```
http://www.cisco.com/index.html?SIGV=1&IS=0&ET=1241194679&CIP=8.1.0.4&KO=1&KN=2&US=8349348ffac7987d11203122a98e7e64e410fa18
```

If you specify version 2 for the script, SHA-1 is used. The protocol from the beginning of the URL is also removed before the signature is generated, and the SIGV=2 string is added. The protocol is RTSP, HTTP, or RTMP. The URL is signed without the protocol, but the final signed URL is printed with the protocol.

Following is an example of the URL signing script using the SHA-1 security hash algorithm with version 2 specified:

```
python cds-ims-urlsign.py http://www.cisco.com/index.html 8.1.0.4 200000 1 2 cisco 2
```

An example of the resulting signed URL follows:

```
http://www.cisco.com/index.html?SIGV=2&IS=0&ET=1241194783&CIP=8.1.0.4&KO=1&KN=2&US=68b5f5ed97d1255a0ec42a42a4f779e794df679c
```

username

To establish username authentication, use the **username** global configuration command.

```
username name { cifs-password | samba-password } { 0 plainword | 1 lancrypto ntcrypto | cleartext } | password { 0 plainword | 1 cryptoword | cleartext } [uid uid] | privilege { 0 | 15 }
```

```
no username name
```

Syntax Description

<i>name</i>	Username.
cifs-password	Sets the Windows user password.
samba-password	Deprecated, same as cifs-password .
0	Specifies a clear-text password. This is the default password setting.
<i>plainword</i>	Clear-text user password.
1	Specifies a type 1 encrypted password.
<i>lancrypto</i>	Encrypted password for LAN Manager networks.
<i>ntcrypto</i>	Encrypted password for Windows NT networks.
<i>cleartext</i>	Unencrypted (clear-text) password for Windows NT networks.
password	Sets the user password.
<i>cryptoword</i>	Encrypted user password.
uid	Sets the user ID for a clear-text password or an encrypted password.
<i>uid</i>	Encrypted password user ID (2001–65535).
privilege	Sets the user privilege level.
0	Sets the user privilege level for a normal user.
15	Sets the user privilege level for a superuser.

Defaults

The **password** value is set to 0 (clear text) by default.

Default administrator account:

- **Uid:** 0
- **Username:** admin
- **Password:** default
- **Privilege:** superuser (15)

Command Modes

Global configuration

Usage Guidelines

The **username** global configuration command changes the password and privilege level for existing user accounts.

User Authentication

User access is controlled at the authentication level. For every HTTP or HTTPS request that applies to the administrative interface, including every CLI and API request that arrives at the CDS network devices, the authentication level has visibility into the supplied username and password. Based on CLI-configured parameters, a decision is then made to either accept or reject the request. This decision is made either by checking local authentication or by performing a query against a remote authentication server. The authentication level is decoupled from the authorization level, and there is no concept of role or domain at the authentication level.

When local CLI authentication is used, all configured users can be displayed by entering the **show running-config** command. Normally, only administrative users need to have username authentication configured.



Note

Every CDS network device should have an administrative password that can override the default password.

User Authorization

Domains and roles are applied by the CDSM at the authorization level. Requests must be accepted by the authentication level before they are considered by the authorization level. The authorization level regulates the access to resources based on the CDSM GUI role and domain configuration.

Regardless of the authentication mechanism, all user authorization configuration is visible in the GUI.

Examples

When you first connect an CDS device to an CDS network, you should immediately change the password for the username *admin*, which has the password *default*, and the privilege level superuser.

The following example shows how to change the password:

```
ServiceEngine(config)#username admin password yoursecret
```

The following example shows how passwords and privilege levels are reconfigured:

```
ServiceEngine#show user username abeddoe
Uid          : 2003
Username     : abeddoe
Password     : ghQ.GyGhP96K6
Privilege    : normal user
ServiceEngine#show user username bwhidney
Uid          : 2002
Username     : bwhidney
Password     : bhlohlbIwAMOk
Privilege    : normal user
ServiceEngine(config)#username bwhidney password 1 victoria
ServiceEngine(config)#username abeddoe privilege 15
User's privilege changed to super user (=15)
ServiceEngine#show user username abeddoe
Uid          : 2003
Username     : abeddoe
Password     : ghQ.GyGhP96K6
Privilege    : super user

ServiceEngine#show user username bwhidney
Uid          : 2002
Username     : bwhidney
Password     : mhYWyw.7PlLd6
Privilege    : normal user
```

■ username

Related Commands

- show user
- show users

whoami

To display the username of the current user, use the **whoami** EXEC command.

whoami

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

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

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

Usage Guidelines	Use this command to display the username of the current user.
-------------------------	---

Examples	<p>The following example displays the username of the user who has logged in to the SE:</p> <pre>ServiceEngine#whoami admin</pre>
-----------------	---

Related Commands	pwd
-------------------------	------------

wmt

To configure Windows Media Technologies (WMT), use the **wmt** global configuration command. To negate these actions, use the **no** form of this command.

wmt accelerate {proxy-cache | vod} enable

wmt advanced client {maximum-packet-size *number* | idle-timeout *number*}

wmt advanced server {log-forwarding enable | inactivity-timeout *number*}

wmt cache {age-multiplier *number* | enable | max-obj-size *size* | max-ttl {days *number* | hours *number* | minutes *number* | seconds *number*} | min-ttl *number* | reval-each-request}

wmt disallowed-client-protocols {http [rtspt | rtspu] | rtspt [http | rtspu] | rtspu [http | rtspt]}

wmt enable

wmt fast-cache {enable | max-delivery-rate *number*}

wmt fast-start {enable | max-bandwidth *number*}

wmt http allow extension *file_extensions*

wmt max-concurrent-sessions *number*

wmt proxy outgoing {http | rtsp} host {hostname | ip-address} port

wmt transaction-logs format {extended {wms-41 | wms-90} | wms-41 | wms-90}

no wmt {accelerate {proxy-cache | vod} enable | advanced {client {maximum-packet-size *number* | idle-timeout} | server {log-forwarding | inactivity-timeout} enable} | cache {age-multiplier *number* | enable | max-obj-size *size* | max-ttl {days *number* | hours *number* | minutes *number* | seconds *number*} | min-ttl *number* | reval-each-request} | disallowed-client-protocols {http [rtspt | rtspu] | rtspt [http | rtspu] | rtspu [http | rtspt]} | enable | fast-cache {enable | max-delivery-rate *number*} | fast-start {enable | max-bandwidth *number*} | http allow extension *file_extensions* | max-concurrent-sessions | proxy outgoing {http | rtsp} | transaction-logs format {extended {wms-41 | wms-90} | wms-41 | wms-90}}

Syntax Description

accelerate	Configures the WMT streaming acceleration.
enable	Enables the performance improvement for live splitting.
proxy-cache	Configures the performance improvement for proxy caching.
enable	Enables the performance improvement for proxy caching.
vod	Sets the SE to accelerate the performance of the video on demand.
enable	Enables the performance improvement for the video on demand.
advanced	Configures WMT advanced settings.
client	Configures WMT advanced client features on the SE.
maximum-packet-size	Specifies the client maximum packet size (WMT maximum IP packet size), used in Virtual Private Network (VPN) environments.

<i>number</i>	Maximum packet size of WMT stream in bytes (512–2048).
idle-timeout	Specifies the maximum amount of time that the SE is to wait for a response from a WMT client before timing out the connection.
<i>number</i>	Timeout value in seconds (30–300).
server	Configures WMT advanced server features on the SE.
log-forwarding	Specifies whether the Windows Media transaction logs should be sent to the upstream WMT server or upstream SEs. This setting applies to all protocols, such as HTTP, RTSPT, and RTSPU.
inactivity-timeout	Specifies the server data channel inactivity timeout.
<i>number</i>	Server data channel inactivity timeout (60–65535).
cache	Configures the WMT cache.
age-multiplier	Specifies the WMT caching heuristic modifiers.
<i>number</i>	Expiration time as a percentage of their age (0–100).
enable	Enables the WMT media cache.
max-obj-size	Sets the maximum size of the object to be cached.
<i>size</i>	Object size in megabytes (1–1000000). The default is 1024 megabytes.
max-ttl	Specifies the maximum time to live for objects in the cache.
days	Specifies the maximum time to live units in days.
<i>number</i>	Maximum time to live (1–1825).
hours	Specifies the maximum time to live units in hours.
<i>number</i>	Maximum time to live (1–43800)
minutes	Specifies the maximum time to live units in minutes.
<i>number</i>	Maximum time to live (1–2628000).
seconds	Specifies the maximum time to live units in seconds.
<i>number</i>	Maximum time to live (1–157680000).
min-ttl	Specifies the minimum time to live for objects in the cache.
<i>number</i>	Minimum time to live (0–86400).
reval-each-request	Revalidates cache on every request.
disallowed-client-protocols	Specifies disallowed WMT client protocols.
http	Disallows streaming over the HTTP protocol (http://).
rtspt	Disallows streaming over the RTSPT protocol (rtspt://).
rtspu	Disallows streaming over the RTSPU protocol (rtspu://).
enable	Enables the WMT server.
fast-cache	Configures WMT Fast Cache. Fast Cache is supported for MMS-over-HTTP only.
enable	Enables WMT Fast Cache.
max-delivery-rate	Configures the maximum delivery rate allowed per media player when Fast Cache is used to serve packets to the media player.
<i>number</i>	Maximum delivery rate per player when Fast Cache is used to serve packets to the media player, expressed as a multiple of the normal delivery rate of a media stream (1–65535).
fast-start	Configures WMT Fast Start.

enable	Enables WMT Fast Start.
max-bandwidth	Configures the maximum burst bandwidth allowed per media player when Fast Start is used to serve packets to the media player.
<i>number</i>	Limit for maximum burst bandwidth allowed per player when Fast Start is used to serve packets to the media player. The default is 3500 kbps.
http	Sets HTTP configurations.
allow	Configures the HTTP filename extensions to be served.
extension	Sets the HTTP filename extensions to be served.
<i>file_extensions</i>	Filename extensions to be served. A maximum of 20 filename extensions is allowed, with a maximum of 10 characters per extension.
max-concurrent-sessions	Configures the maximum number of unicast clients that can be served concurrently.
<i>number</i>	Limit for incoming unicast requests; this limit is subject to physical resources on the platform (1–8000).
proxy	Configures a proxy.
outgoing	Configures an outgoing proxy.
http	Configures an outgoing HTTP proxy server for Windows Media requests.
rtsp	Configures an RTSP outgoing server for WMT RTSP requests from Windows Media 9 players.
host	Configures the host of an outgoing MMS-over-HTTP proxy.
<i>hostname</i>	Hostname of an outgoing proxy.
<i>ip-address</i>	IP address of an outgoing proxy.
<i>port</i>	Port number of an outgoing proxy (1–65535).
transaction-logs	Configures the logging format of the WMT transaction logs.
format	Sets the format for WMT transaction logs.
extended	Specifies the WMT-extended configuration for transaction logs. Enables username logging in the WMT transaction log.
wms-41	Sets the WMT to generate transaction logs in the extended Windows Media Services version 4.1 format.
wms-90	Sets the WMT to generate transaction logs in the extended Windows Media Services Version 9.0 format.
wms-41	Sets the WMT to generate transaction logs in the standard Windows Media Services Version 4.1 format.
wms-90	Sets the WMT to generate transaction logs in the standard Windows Media Services Version 9.0 format.

Defaults

wmt: disabled
advanced client maximum-packet-size: 1500 bytes
advanced client idle-timeout: 60 seconds
advanced server log-forwarding: enabled

wmt cache max-ttl days: 1 day
wmt cache max-ttl hours: 72 hours
wmt cache max-ttl minutes: 4320 minutes
wmt cache max-ttl seconds: 259200 seconds
wmt cache min-ttl: 60 minutes
wmt fast-cache: enabled
wmt fast-start: enabled
max-object-size: 1 GB
wmt http allow extension *file_extensions*: asf, none, nsc, wma, wmv

Command Modes

Global configuration

Usage Guidelines

The Windows Media Services (WMS) is the Microsoft streaming solution for creating, distributing, and playing back digital media files on the Internet. Windows Media Services 9 Series (WMS 9) is the new Windows Media solutions from Microsoft.

Enabling WMT on the Service Engine

Before enabling licenses for streaming media services on an SE, make sure that your SE clock and calendar settings are correct; otherwise, you will see an error message and the services will fail to install. Use the **show clock EXEC** command to display the system clock. To set the system clock, use the **clock set EXEC** command.

Enabling Conventional WMT Proxy Service

During conventional proxy caching, the user media player is pointed to the SE to access the streaming media. Before enabling conventional WMT proxy service, be sure you have fulfilled the following requirements:

- You have a Microsoft WMT license key.
- You have the IP address of the SE.

Enabling Fast Cache

Fast Cache allows streaming of content to the Windows Media Player's cache as fast as the network allows, reducing the likelihood of an interruption in play due to network problems. When used with the Windows Media Player 9 Series, Fast Cache provides a way to stream content to clients faster than the data rate specified by the stream format. For example, with Fast Cache enabled, the server can transmit a 128-kbps stream at 700 kbps. In Windows Media Player, the stream is still rendered at the specified data rate, but the media player can buffer a much larger portion of the content before rendering it. This buffering allows the client to handle variable network conditions without impacting the playback quality of on-demand content.

Enabling Fast Start

Fast Start helps reduce buffering time. Typically, Windows Media Player must buffer a certain amount of data before it can start rendering content. If the clients connecting to the SE are using Windows Media Player for Windows XP or a later version of Windows Media Player, Fast Start can be used to provide

data directly to the buffer at speeds higher than the bit rate of the content requested. This buffering enables users to start receiving content more quickly. After the initial buffer requirement has been fulfilled, on-demand content is streamed at the bit rate defined by the content stream.

**Note**

Fast Start is not available to the first client connecting to a live stream.

When Fast Start is enabled on the SE, the increased bandwidth that Fast Start initially uses to send data to the media players can overburden a network if many media players connect to the stream at the same time. To reduce the risk of network congestion, use the **wmt fast-start max-bandwidth** global configuration command to limit the amount of bandwidth that Fast Start can use to stream content to each media player.

Adding or Removing WMT HTTP Allowed Filename Extensions

SEs use a list of filename extensions to decide whether a type of media file should be served by WMT. Typically, SEs are shipped with a default list of filename extensions to be served by WMT.

The default list in the SE contains the following filename extensions:

- asf
- none
- nsc
- wma
- wmv

**Note**

The default list of filename extensions includes “none” to enable SEs to serve media files without file extensions, such as URLs of live encoders. The filename extension nsc is included in the list to enable SEs to multicast media files.

Use the **wmt http allow extension *file_extensions*** global configuration command to add new filename extensions to the list. Use the **no wmt http allow extension *file_extensions*** command to remove filename extensions from the list.

The following restrictions apply to adding new filename extensions to the list:

- You cannot have more than 20 extensions in the list of allowed filename extensions.
- Filename extensions must be alphanumeric, and the first character of every extension must be a letter.
- You cannot have more than ten characters in a filename extension.

WMT Unique Stream Key

Normally, a caching proxy uses the URL string as the content identifier, so that a cache hit occurs when the request URL matches the content URL. This process is often unreliable, because some websites use dynamically generated URLs, which create different URL strings for the same content. When the URL string is used as the content identifier in this case, the likelihood of a cache hit is reduced. The unique stream key produces an identifier that is based on domain name, file size, bit rate, and other content-specific properties. This identifier is almost always unique for a piece of content. Using the unique stream key feature increases the likelihood of a cache hit.

Configuring WMT Multicasting

An SE can receive and deliver WMT streaming content through IP multicast as described in the next few sections.

Unicast-in multicast-out multicast delivery enables you to distribute streaming media efficiently by allowing different devices on the IP multicast to receive a single stream of media content from the SE simultaneously. This delivery mechanism can save significant network bandwidth consumption, because a single stream is sent to many devices, rather than sending a single stream to a single device every time that this stream is requested. This multicast delivery feature is enabled by setting up a multicast address on the SE to which different devices, configured to receive the content from the same channel, can subscribe. The delivering device sends the content to the multicast address set up at the SE, from which it becomes available to all subscribed receiving devices.

Multicast-in multicast-out multicast receive enables you to receive multicast WMT streams delivered through IP multicasting and then relay them to end users through another delivery channel (unicast or multicast).

The two WMT multicast-out features combined enable you to receive and deliver WMT streaming media content through IP multicasting and to do conversions from multicast to unicast (and vice versa).

The multicast-in unicast-out scenario enables you to create a broadcasting publishing point to deliver an incoming stream live to requesting clients using multicast as the source of the streaming media.

WMT Multicast Logging

Use the **log** option to provide multicast statistics to multicast server administrators. These statistics include a multicast IP address, a port number, a start time, and a number of clients. When configuring this option, you can choose to provide either a local URL where the multicast logging statistics can be sent, or an external fully qualified server URL that can receive these statistics. The multicast logging URL option can point to the multicast server or to any web server that can process the posted information from the users who subscribed to the multicast address.

Configuring Multicast-In Multicast-Out

In this multicasting scenario, a description file *.nsc is created that is accessible through multicast-out to clients. This scenario is similar to the unicast-in multicast-out scenario except that the input source is multicast. The clients use this description file to subscribe to the multicast.

Configuring Multicast to SE and Multicast to Client

In the Internet Streamer CDS 2.4 release, Multicast to SE and Multicast to client options were added. The administrator can configure inter-SE multicast for live programs if the network is multicast enabled. If the network is not multicast enabled, the result is undefined and streaming may not work as expected. Therefore, this requires a special configuration on the Live Programs page to turn this feature on and off.

To enable multicast delivery to the SEs for a program, you must choose multicast as a delivery mechanism. Choose Services > Live Video > Live Programs > Live Streaming. The Live Stream Settings page is displayed. Check the Enable Multicast Delivery to SE check box and click Submit.

Configuring Multicast-In Unicast-Out

In this scenario, a unicast-out publishing point is created to deliver the incoming stream live to requesting clients.

Configuring Unicast-In Unicast-Out

Unicast-in unicast-out provides a point-to-point connection between the client and the SE. The advantage of unicasting when streaming media over a network is that only a single stream needs to be pulled over the network between the origin server and SE, but that stream can be delivered to multiple

clients in a nonmulticast environment. A server running Windows Media Services can provide a unicast video stream to multiple clients through a single stream delivered to the SE. Typically, unicast-in unicast-out is used to broadcast live events.

In this scenario, unicast-in unicast-out provides a point-to-point connection between the client and the SE. The SE makes a single connection to the media server. Multiple requests for the same stream can be split by the SE so that each client receives a distinct data stream directly from the SE, while the SE maintains its single stream connection to the media server.

You can configure unicast-in unicast-out using live splitting without any configuration. The SE acts as a proxy. When clients request the same unicast URL, the SE proxy automatically splits the stream from the source to the clients.

Configuring Outgoing WMT Proxy Servers

You can specify the external WMT server that the SE should use as its upstream WMT server. The SE contacts the specified outgoing proxy server upon a cache miss (if the SE does not have the requested WMT content already stored in its local cache).

Configuring WMT Transaction Logs

WMT transaction logs allow content providers to track what content customers viewed, how long they viewed it, and the quality of transmission. The Internet Streamer CDS software uses the enhanced logging support provided by Windows Media Services 9 Series in addition to the Windows Media Services Version 4.1 logging format.

The following transaction log formats are supported for WMT:

- Standard Windows Media Services 4.1
- Extended Windows Media Services 4.1
- Standard Windows Media Services 9.0
- Extended Windows Media Services 9.0



Note

For RTSP, when you choose the **Repeat** option from the Play menu in the Windows Media player to play media files continuously in a loop, an extra entry is logged in the transaction logs for each playback of the file. This situation occurs with the WMT RTSPU protocol due to the behavior of the Windows Media player.

The SE's transaction logging format for WMT streaming is consistent with that of the Windows Media Services and the World Wide Web Consortium (W3C)-compliant log format. A log line is written for every stream accessed by the client. The location of the log is not configurable. These logs can be exported using FTP. When transaction logging is enabled, daemons create a separate working.log file in /local1/logs/export for WMT transactions.

All client information in the transaction logs is sent to the origin server by default.

Log Formats Accepted by Windows Media Services 9

Windows Media Players connect to a Windows Media Server using the following protocols:

- Windows Media Players earlier than Version 9.0 (Windows Media 6 and 7 Players) use HTTP 1.0 or the MMS protocol.
- Windows Media 9 Players use HTTP 1.0, HTTP 1.1, and RTSP.

Depending on the version of the Windows Media Player, logs are sent in different formats, such as text, binary, or Extensible Markup Language (XML). See [Table 2-82](#).

Table 2-82 Log Formats Accepted by Windows Media Services 9

Protocol	Player and Distributor	Log Type
HTTP/1.0	Windows Media Player earlier than Version 9.0 (for example, Windows Media 6.4 or 7.0 Players) SE (caching and proxy server) is running Windows Media Services Version 9.0 and streaming from a WMT server that is running Windows Media Services 4.1	World Wide Web Consortium (W3C) standard space-delimited text log
MMS	Windows Media Player earlier than Version 9.0 (for example, Windows Media 6.4 or 7.0 Players)	Binary structure log
HTTP/1.1	Windows Media Player Version 9.0 Distribution server is running Windows Media Services 9.0 SE (caching and proxy server) is running Windows Media Services 9.0	XML structure log
RTSP	Windows Media Player Version 9.0 Distribution server is running Windows Media Services 9.0 SE (caching and proxy server) is running Windows Media Services 9.0	XML structure log

The posted XML log file from the Windows Media Player to the SE (Windows Media Server) can be parsed and saved to the normal WMT transaction logs that are stored on the SE.

To specify the format for the WMT transaction logs on SEs, use the **wmt transaction-logs format** global configuration command. By default, the standard Windows Media Services 4.1 logging format is used (no SE-specific details are logged).

When you use the extended format in Windows Media Services 4.1 and 9.0, the SE includes the following three additional fields in the transaction log:

- SE-action—cache hit, cache miss, VoD, or live create
- SE-bytes—number of bytes served by the SE in the case of a cache hit
- username (username of the person who made the WMT request when Microsoft Negotiate authentication, Microsoft Digest authentication, and basic authentication are used)

**Note**

Microsoft Negotiate authentication is an authentication method in which the WMS Negotiate Authentication plug-in is used to authenticate the client. This method of authentication uses the client's logon credentials. It uses the encrypted password and username that the user entered during the login process.

Microsoft Digest authentication is an authentication method in which an initial authentication of

the client is performed when the server receives the first challenge response from the client. After the server verifies that the client has not been authenticated yet, it accesses the services of a domain controller to perform the initial authentication of the client. When the initial authentication of the client is successfully completed, the server receives a Digest session key. The server caches the session key and uses it to authenticate subsequent requests for resources from the authenticated client.

If the SE is configured to use the extended format of WMT transaction logging and the extended WMT logging feature is enabled, then the SE logs usernames for any authenticated WMT requests. Usernames are logged for Negotiate, Digest, and basic authentication.

**Note**

Negotiate and Digest authentication is applicable for the HTTP protocol only.

By default, the extended WMT logging feature is disabled. If the extended logging format is enabled (using the **wmt transaction-logs format extended** global configuration command) but the extended WMT logging feature is disabled, the username field in the WMT transaction log will be empty.

**Note**

The SE logs usernames associated with authenticated WMT requests only when the extended logging formats (extended wms-41 and extended wms-90) are used.

WMT Multicast Logging

WMT logs are logged to a working log on the local disk in one of the following files, depending upon where the sysfs is mounted on the SE:

- The file named /local1/logs/export/working.log
- The file named /local2/logs/export/working.log

Forwarding WMT Logs to Upstream Servers

You can decide whether you want this SE to forward its WMT logs to the upstream server (a Windows Media server or another SE). By default, SEs forward their WMT logs to the upstream server. This feature applies to all of the supported protocols. To disable this feature and configure the SE to not forward its WMT logs to the upstream server, enter the **no wmt advanced server log-forwarding enable** global configuration command. To re-enable this feature, enter the **wmt advanced server log-forwarding enable** global configuration command.

Examples

The following example displays request statistics. In this example, the statistics reported are the total number of requests served, type of content (live or VoD), transport protocol, and source of content:

```
ServiceEngine#show statistics wmt requests
```

```
Unicast Requests Statistics
```

```
=====
```

```
Total unicast requests received: 0
```

```
-----
```

	Total	% of Total Unicast Requests
	-----	-----
Streaming Requests served:	0	0.00%
Mcast nsc file Request:	0	0.00%

```

Authenticate Requests:      0      0.00%
Requests error:            0      0.00%

                                Total      % of Total
                                Streaming Requests
-----

By Type of Content
-----
    Live content:          0      0.00%
    On-Demand Content:    0      0.00%

By Transport Protocol
-----
    HTTP:                 0      0.00%
    RTSPT:                0      0.00%
    RTSPU:                0      0.00%

By Source of Content
-----
    Local:                0      0.00%
    Remote HTTP:          0      0.00%
    Remote RTSP:          0      0.00%
    Multicast:            0      0.00%

CDN-Related WMT Requests
-----
    CDN Content Hits:      0      0.00%
    CDN Content Misses:    0      0.00%
    CDN Content Live:      0      0.00%
    CDN Content Errors:    0      0.00%

Fast Streaming related WMT Requests
-----
    Normal Speed:          0      0.00%
    Fast Start Only:       0      0.00%
    Fast Cache Only:       0      0.00%
    Fast Start and Fast Cache: 0      0.00%

                                Total      % of Total
                                Authenticated Requests
-----

By Type of Authentication
-----
    Negotiate:             0      0.00%
    Digest:                0      0.00%
    Basic:                 0      0.00%

```

The following example displays the multicast logging statistics sent to the multicast server:

```

10.1.101.2 2003-05-11 13:39:21 - asfm://239.1.4.5:4000 0 30 1 200
{5DC90EEB-CEB1-467C-9F7A-BCF5EEDE3FF} 10.1.0.3055 en-US - - wmpplayer.exe 10.1.0.3055
Windows_2000 10.0.0.2195 Pentium 0 152543 65389 asfm UDP WINDOWS_MEDIA_AUDIO_V2
MICROSOFT_MPEG-4_VIDEO_CODEC_V3 http://172.16.192.91/cisco.nsc - 166245 - 176 0 0 0 0 0 01
0 100 239.1.4.5 - - -

```

The format of the example shown is as follows:

```

c-ip date time c-dns cs-uri-stem c-starttime x-duration c-rate c-status c-playerid
c-playerversion c-playerlanguage cs(User-Agent) cs(Referer) c-hostexe c-hostexever c-os
c-osversion c-cpu filelength filesize avgbandwidth protocol transport audiocodec
videocodec channelURL sc-bytes c-bytes s-pkts-sent c-pkts-received c-pkts-lost-client

```

```
c-pkts-lost-net c-pkts-lost-cont-net c-resendreqs c-pkts-recovered-ECC
c-pkts-recovered-resent c-buffercount c-totalbuffertime c-quality s-ip s-dns
s-totalclients s-cpu-util SE-action SE-bytes Username
```

Table 2-83 describes the fields shown in this example.

Table 2-83 wmt multicast logging Field Descriptions

Field	Description
c-ip	IP address of the client computer. A client that is not connected properly provides a client proxy server IP address, not the client IP address.
date	Date (according to Greenwich mean time) when an entry is generated in the log file.
time	Time (according to Greenwich mean time) when an entry is generated in the log file.
c-dns	Domain Name Server (DNS) name of the client computer.
cs-uri-stem	Name of the file that is playing: an .asf file for a unicast and an .asx file for a multicast.
c-starttime	Time stamp (in seconds) of the stream when an entry is generated in the log file.
x-duration	Length of time that a client played content before a client event (FF, REW, pause, stop, or jump to marker). A log entry is generated whenever one of these client events occur.
c-rate	Mode of Windows Media Player when the last command event was sent: <ul style="list-style-type: none"> 1 = Windows Media Player was paused or stopped during a play, fast-forward, rewind, or marker jump operation. -5 = Windows Media Player was rewound from a play, stop, or pause operation. 5 = Windows Media Player was fast-forwarded from a play, stop, or pause operation.
c-status	Codes that describe client status. Mapped to HTTP/1.1 and RTSP client status codes described in RFC 2068 and RFC 2326. Windows Media Services includes the extensible client status codes 480 (simultaneous client connections exceeded the maximum client limit of the server) and 483 (stream exceeded maximum file bit-rate limit of the server).
c-playerid	Globally unique identifier (GUID) of the player.
c-playerversion	Version number of the player.
c-playerlanguage	Language country code of the client computer.
cs(User-Agent)	Browser type used if Windows Media Player was embedded in a browser.
cs(Referer)	URL of the web page in which Windows Media Player was embedded (if it was embedded).
c-hostexe	Host application; for example, a web page in a browser (iexplore.exe), a Microsoft Visual Basic applet (vb.exe), or standalone Microsoft Windows Media Player (mplayer2.exe).
c-hostexever	Version number of the host application.
c-os	Operating system of the client computer.

Table 2-83 *wmt multicast logging Field Descriptions (continued)*

Field	Description
c-osversion	Operating system version number of the client computer.
c-cpu	CPU type of the client computer.
filelength	Length of the file (in seconds). This value is 0 for a live stream.
filesize	Size of the file (in bytes). This value is 0 for a live stream.
avgbandwidth	Average bandwidth (in bits per second) at which the client was connected to the server.
protocol	Protocol used to access the stream: HTTP, or ASFM (multicast protocol).
transport	Transport protocol used to deliver the stream (UDP, TCP, or UDP over IP multicast).
audiocodec	Audio codec used in the stream.
videocodec	Video codec used to encode the stream.
channelURL	URL to the .nsc file. A unicast client information log file records a hyphen (-) for this field.
sc-bytes	Bytes sent by the server to the client.
c-bytes	Number of bytes received by the client from the server. For unicast, the c-bytes value and sc-bytes value must be identical. If not, packet loss has occurred.
s-pkts-sent	Total number of packets sent by the server.
c-pkts-received	Number of packets from the server (s-pkts-send) that are received correctly by the client on the first try.
c-pkts-lost-client	Number of packets lost during transmission from the server to the client and not recovered at the client layer through an error correction or at the network layer through User Datagram Protocol (UDP) resends.
c-pkts-lost-net	Number of packets lost on the network layer.
c-pkts-lost-cont-net	Maximum number of continuously lost packets on the network layer during a transmission from the server to the client.
c-resendreqs	Number of client requests to receive new packets. This field contains a value only if the client is using UDP resend.
c-pkts-recovered-ECC	Number of packets repaired and recovered on the client layer. Packets repaired and recovered at the client layer are equal to the difference between c-pkts-lost-net and c-pkts-lost-client.
c-pkts-recovered-resent	Number of packets recovered because they were resent using UDP.
c-buffercount	Number of times that the client buffered while playing the stream.
c-totalbuffertime	Time (in seconds) that the client used to buffer the stream. If the client buffers the stream more than once before a log entry is generated, c-totalbuffertime is the total amount of time that the client spent buffering the stream.

Table 2-83 *wmt multicast logging Field Descriptions (continued)*

Field	Description
c-quality	The percentage of packets that were received by the client, indicating the quality of the stream. If cPacketsRendered is all packets received by the client, including packets recovered by error correction and UDP resend (c-pkts-received + c-pkts-recovered-ECC + c-pkts-recovered-resent), then c-quality can be calculated as: [cPacketsRendered / (cPacketsRendered + c-pkts-lost-client)] * 100.
s-ip	Server IP address.
s-dns	Server DNS.
s-totalclients	Clients connected to the server (but not necessarily receiving streams).
s-cpu-util	Average load on the server processor as a percentage (0–100%). If multiple processors exist, this value is the average for all processors.
SE-action	Action performed by the SE.
SE-bytes	Number of bytes received by the SE.
Username	Username required to access the streaming media retrieved by the WMT player.

The following example adds the filename extension mp3 to the list of filename extensions to be served by WMT:

```
ServiceEngine#wmt http allow extension mp3
```

The **show wmt http allow extension EXEC** command shows the filename extensions included in the list after you have added or deleted filename extensions.

The following example shows that the filename extension mp3 has been added to the list of file extensions:

```
ServiceEngine#show wmt http allow extension
```

```
WMT http extensions allowed :
asf mp3 none nsc wma wmv
```

The following example shows that an SE at a branch office is configured to send all its WMT cache miss traffic to a central SE at 172.16.30.30 through port 8080:

```
ServiceEngine(config)# wmt proxy outgoing http host 172.16.30.30 8080
```

The following example shows that an SE at a branch office is configured to send all its cache miss traffic to a central SE at 172.16.30.31 through port 1700:

```
ServiceEngine(config)# wmt proxy outgoing http host 172.16.30.31 1700
```

The following example sets the SE to generate WMT transaction logs in the extended Windows Media Services, Version 9.0 format:

```
ServiceEngine#wmt transaction-logs format extended wms-90
```

The following example enables the logging of usernames to the WMT transaction log:

```
ServiceEngine#wmt extended transaction-log enable
```

Related Commands

clear
show running-config
show tech-support
show statistics wmt
show wmt
wmt (EXEC mode)

write

To save startup configurations, use the **write** EXEC command.

write [erase | memory | terminal]

Syntax Description

erase	(Optional) Erases the startup configuration from NVRAM.
memory	(Optional) Writes the configuration to NVRAM. This setting is the default.
terminal	(Optional) Writes the configuration to a terminal session.

Defaults

The configuration is written to NVRAM by default.

Command Modes

EXEC

Usage Guidelines

Use this command to either save running configurations to NVRAM or erase memory configurations. Following a **write erase** command, no configuration is held in memory, and a prompt for configuration specifics occurs after you reboot the SE.

Use the **write terminal** command to display the current running configuration in the terminal session window. The equivalent command is **show running-config**.

The **write memory** command saves modified Websense configuration files (the eimserver.ini, config.xml, and websense.ini files and the Blockpages directory) across disk reconfiguration and Internet Streamer CDS software release upgrades.



Note

Clicking the **Save Changes** button from the Websense Enterprise Manager window does not save the Websense configuration modifications across device reboots. You need to use the **write memory** command to save the Websense configuration changes across reboots.

You must execute the **write memory** command in order to save the most recent configuration modifications, including websense.ini file modifications and Websense URL filtering configuration changes. The **write memory** command enables the changes made from the external Websense Manager GUI to be saved across disk reconfiguration and upgrades (which might erase disk content).

The Websense configurations from the last use of the **write memory** command are retained under the following situations:

- If the **write memory** command is not used before a reboot but after a disk reconfiguration or an Internet Streamer CDS software upgrade that erases disk content.
- If you are using the CLI and did not answer “yes” when asked if you wanted to save the configurations at the reload prompt.

However, if the **write memory** command has never been used before, then default configurations will be applied when the content in the /local1/WebsenseEnterprise/EIM directory on the SE is erased.

Examples

The following command saves the running configuration to NVRAM:

```
ServiceEngine#write memory
```

Related Commands

copy
show running-config

■ write