wccp rtsp

To configure WCCP Version 2 Real-Time Streaming Protocol (RTSP) protocol transparent interception, use the **wccp rtsp** global configuration command. To disable this function, use the **no** form of this command.

wccp rtsp {**mask** {[**dst-ip-mask** *hex_num*] [**dst-port-mask** *port_hex_num*] [**src-ip-mask** *hex_num*] [**src-port-mask** *port_hex_num*]} | **router-list-num** *num* [**hash-destination-ip**] [**hash-destination-port**] [**hash-source-ip**] [**hash-source-port**] [**l2-redirect**] [**mask-assign**] [**password** *key*] [**weight** *percentage*]}

no wccp rtsp

Syntax Description

mask	Sets the mask used for the Content Engine assignment. Configure at least one mask; the maximum is four masks.
dst-ip-mask	(Optional) Sets the mask used to match the packet destination IP address.
hex_num	IP address mask defined by a hexadecimal number (for example, 0xFC000000). The range is 0x00000000–FE000000.
dst-port-mask	(Optional) Sets the mask used to match the packet destination port number.
port_hex_num	Port mask defined by a hexadecimal number (for example, 0xFC00). The port range is 0–65535.
src-ip-mask	(Optional) Sets the mask used to match the packet source IP address.
src-port-mask	(Optional) Sets the mask used to match the packet source port number.
router-list-num	Sets the router list number.
num	Router list number (1–8).
hash-destination-ip	(Optional) Defines the load-balancing hash of the destination IP address (the default).
hash-destination-port	(Optional) Defines the load-balancing hash of the destination port.
hash-source-ip	(Optional) Defines the load-balancing hash of the source IP address.
hash-source-port	(Optional) Defines the load-balancing hash of the source port.
l2-redirect	(Optional) Specifies the packet forwarding by Layer 2 redirect.
mask-assign	(Optional) Uses the mask method for the Content Engine assignment.
password	(Optional) Sets the authentication password to be used for secure traffic among the Content Engines within a cluster and the router for a specified service. Be sure to enable all other Content Engines and routers within the cluster with the same password.
key	WCCP service password key. Passwords must not exceed eight characters.
weight	(Optional) Sets the weight percentage for load balancing. For more information, see the "Load Balancing" section.
percentage	Percentage value (0–100).

Defaults

wccp reverse-proxy: disabled

dst-ip-mask: 0x00000000

src-ip-mask: 0x00001741

dst-port-mask: 0x0

src-port-mask: 0x0

Command Modes

global configuration

Usage Guidelines

This command applies only to WCCP Version 2.

Examples

The following example associates the router list 8 with the RTSP service, sets the authentication password to be used for secure traffic among the Content Engines within a cluster and the router for the reverse-proxy service, and sets the weight percentage for load balancing:

```
ContentEngine(config)# wccp rtsp router-list-num 8 password mysecret weight 100
```

The following example disables RTSP transparent interception:

```
ContentEngine(config)# no wccp rtsp
```

Related Commands

show rtsp all
show wccp content-engines
show wccp flows rtsp
show wccp masks rtsp
show wccp services
show wccp slowstart rtsp
show wccp status
wccp router-list
wccp version 2

wccp service-number

To enable up to eight dynamic WCCP redirection services on the Content Engine, use the **wccp service-number** global configuration command. The services must also be configured on the router running WCCP Version 2. To disable this function, use the **no** form of this command.

wccp service-number *servnumber* {**mask** {[**dst-ip-mask** *hex_num*] [**dst-port-mask** *port_hex_num*] [**src-ip-mask** *hex_num*] [**src-port-mask** *port_hex_num*]} | **router-list-num** *num* **port-list-num** *port-list* **application** {**cache** | **https-cache** | **streaming**} [**assign-method-strict**] [**hash-destination-ip**] [**hash-destination-port**] [**hash-source-ip**] [**hash-source-port**] [**l2-redirect**] [**mask-assign**] [**match-source-port**] [**password** *key*] [**weight** *percentage*]}

no wccp service-number *servnumber*

Syntax Description

servnumber	WCCP Version 2 service number (90–97).
mask	Sets the mask used for the Content Engine assignment. Configure at least one mask; the maximum is four masks.
dst-ip-mask	(Optional) Sets the mask used to match the packet destination IP address.
hex_num	IP address mask defined by a hexadecimal number (for example, 0xFC000000). The range is 0x00000000–FE000000.
dst-port-mask	(Optional) Sets the mask used to match the packet destination port number.
port_hex_num	Port mask defined by a hexadecimal number (for example, 0xFC00). The port range is 0–65535.
src-ip-mask	(Optional) Sets the mask used to match the packet source IP address.
src-port-mask	(Optional) Sets the mask used to match the packet source port number.
router-list-num	Sets the router list number.
num	Router list number (1–8).
port-list-num	Sets the port list number.
port-list	Port list number (1–8).
application	Specifies the application (caching or streaming media).
cache	Redirects traffic to the caching application running on the Content Engine.
https-cache	Redirects traffic to the HTTPS caching application running on the Content Engine.
streaming	Redirects traffic to the streaming media application running on the Content Engine.
assign-method-strict	(Optional) Forces WCCP to strictly use only the configured assignment method. For more information, see the "Assignment Method" section.
hash-destination-ip	(Optional) Defines the load-balancing hash of the destination IP address (the default).
hash-destination-port	(Optional) Defines the load-balancing hash of the destination port.
hash-source-ip	(Optional) Defines the load-balancing hash of the source IP address.
hash-source-port	(Optional) Defines the load-balancing hash of the source port.
l2-redirect	(Optional) Sets the packet forwarding by Layer 2 redirect. For more information, see the "WCCP Layer 2 Support" section on page 2-839.

mask-assign	(Optional) Uses the mask method for the Content Engine assignment.
match-source-port	(Optional) Matches the source port for redirection.
password	(Optional) Sets the authentication password to be used for secure traffic among the Content Engines within a cluster and the router for a specified service. Be sure to enable all other Content Engines and routers within the cluster with the same password.
key	WCCP service password key. Passwords must not exceed eight characters.
weight	(Optional) Sets the weight percentage for load balancing. For more information, see the "Load Balancing" section.
percentage	Percentage value (0–100).

Defaults

wccp service-number: disabled

dst-ip-mask: 0x00001741

src-ip-mask: 0x00000000

dst-port-mask: 0x0

src-port-mask: 0x0

If a load-balancing hash is defined, the destination IP address is the default.

Command Modes

global configuration

Usage Guidelines

The **application cache** option redirects traffic to the Content Engine cache processes. The **application https-cache** option redirects traffic to the Content Engine HTTPS cache processes. The **application streaming** option redirects traffic to the Content Engine streaming media processes.

Proxy Mode

The Content Engine supports up to eight incoming ports each for File Transfer Protocol (FTP), HTTPS, and HTTP proxy modes. The RTSP proxy currently permits only one proxy port. The incoming proxy ports can be the same ports that are used by transparent-mode services. The incoming proxy ports can be changed without stopping any WCCP services running on the Content Engine or on other Content Engines in the farm.

The Content Engine parses requests received on a port to determine the protocol to be serviced. If the Content Engine is not configured to support a received protocol, the proxy server returns an error. For example, if port 8080 is configured to run an HTTP and HTTPS proxy service, an FTP request coming to this port is rejected.

Some TCP ports are reserved for system or network services and should not be used for proxying services in transparent mode or in proxy mode. If more than eight ports are required, you can configure multiple custom WCCP services. Intercepted FTP, HTTP, and HTTPS requests addressed to other proxy servers (received on transparent-mode ports) are serviced according to the **proxy-protocols transparent** command parameters.

Transparent Mode

Transparent and proxy mode requests can be distinguished by comparing the destination IP address of the request and the IP address of the Content Engine. A nonmatching IP address indicates that the request has been redirected and is transparent. The style of the URL within the request can be proxy-style or server-style (a server-style URL does not include the protocol and hostname). Transparent requests have a server-style URL, but proxy-style ones may also be received, for example, when the Content Engine is intercepting a request destined for a proxy. If a server-style URL is received, only HTTP is supported. If a proxy-style URL is received, all of the protocols enabled on the Content Engine are supported.

The **wccp service-number** command can enable up to eight WCCP redirection services on a Content Engine, provided that the services are also configured on the router. Eight dynamic WCCP services (90 to 97) are available.

Each **wccp service-number** command specifies a router list, single port list (containing up to eight ports), application type, hash parameters, password, and weight. With eight custom services using a maximum number of eight ports each, the maximum number of ports that can be specified for transparent redirection is 64.

The legacy custom web cache and reverse proxy services (service numbers 98 and 99) can be configured with only one port each. If only one legacy service is configured, the total maximum number of transparent redirection ports is 57. If both legacy services are configured, the maximum port total is 50.

Note In the ACNS 5.1 software and earlier releases, a maximum of eight active WCCP services were supported by a WCCP Version 2 router and a Content Engine. In the ACNS 5.2 software and later releases, up to 25 active WCCP Version 2 services are supported. In the ACNS 5.2 software, only 17 WCCP services were defined. In the ACNS 5.3 software, 18 WCCP services are defined.

All ports receiving HTTP that are configured as members of the same WCCP service share the following characteristics:

- They have the same hash parameters as configured with the **wccp service-number** command.
- The service on individual ports cannot be stopped or started individually (WCCP Version 2 restriction).

With Content Engines in a farm, the following restrictions apply:

- All Content Engines that use the same WCCP service are required to configure the same list of ports and the same hash parameters.
- A Content Engine that tries to join the farm with the same WCCP service using a different list of ports or different hash parameters is rejected by the router.
- To change the port list for a particular WCCP service, WCCP service must be stopped on all involved Content Engines and then all must be restarted with the new parameters.

The Content Engine WCCP implementation currently allows global settings that apply to all WCCP services, such as healing parameters, slow start, and others. The multiple service model does not change that, and the settings remain global for the whole WCCP system.

Modifying Configurations

For proxy-mode and transparent-mode commands, entering a new command replaces the old one. In proxy mode, a **no** command that specifies the protocol and no ports disables the service for that protocol. To add or remove ports in proxy mode, enter a new command that specifies all the ports to be used. Ports can also be removed by a **no** command with a list of ports to remove. A **no** command that specifies only some of the configured ports removes these ports from the list, and the service continues to run on the

remaining ports. For example, if HTTPS is received on 8080, 8081, and 8082, the **no https proxy incoming 8081** command disables port 8081 but permits the HTTPS proxy service to continue on ports 8080 and 8082.

In transparent mode, to add or remove ports for a WCCP service, modify the port list or create a new port list for the WCCP service. In transparent mode, a **no** command that specifies the WCCP service number disables the service.

To use the **l2-redirect** forwarding method, the Content Engine must be directly connected at Layer 2 to a switch or router that supports accelerated hardware switching.

See the *Cisco ACNS Software Configuration Guide for Locally Managed Deployments* for a description on how to configure a Content Engine and router to use WCCP Version 2 to support a user-defined WCCP service (services 90 to 97).

Examples

The following example shows that the WCCP dynamic service 90 is configured with router list 1 and port list 1. Port 8080 is the only element in port list 1.

```
ContentEngine(config)# wccp 90 router-list-num 1 port-list-number 1 hash-source-ip
hash-destination-port
ContentEngine(config)# wccp port-list 1 8080
```

IThe following example shows that the Content Engine is configured to accept HTTP and HTTPS proxy requests on ports 81, 8080, and 8081:

```
ContentEngine(config)# http proxy incoming 81 8080 8081
ContentEngine(config)# https proxy incoming 81 8080 8081
```

The following example shows how to configure a WCCP-enabled router to support dynamic WCCP services by turning on WCCP Version 2 on the router:

```
Router# configure terminal
Router(config)# ip wccp version 2
```

The following example shows how to turn on the WCCP feature for the user-defined service (for example, service 90):

```
Router(config)# ip wccp 90
```

The following example shows how to specify the interface on which service 90 will run on the router:

```
Router(config)# interface type number
```

The following example shows how to configure the Content Engine to use the outbound interface for service 90:

```
Router(config-if)# ip wccp 90 redirect out
```

The following example shows how to configure user-defined services (services 90 to 97) on the Content Engine by creating a router list that specifies the routers that support the user-defined service or a set of services:

```
ContentEngine(config)# wccp router-list 1 10.1.202.1
```

In this example, router list 1 has only a single router (the WCCP Version 2-enabled router with an IP address of 10.1.202.1). Only one router list is created to enable the same router to be used for different WCCP services.

These port lists specify the port numbers on which the Content Engine will listen for incoming traffic from specific WCCP Version 2-enabled routers. These ports lists allow you to configure the Content Engine to listen for incoming WCCP requests on more than one port. By default, the Content Engine listens for incoming traffic on port 80. Create one port list for each of the eight user-defined WCCP Version 2 services that you will be creating (services 90 to 97). You can define up to eight ports per port list. In this case, each port list has a single port (for example, port list 1 contains only port 32).

```
ContentEngine(config)# wccp port-list 1 32
ContentEngine(config)# wccp port-list 2 33
ContentEngine(config)# wccp port-list 3 34
ContentEngine(config)# wccp port-list 4 35
ContentEngine(config)# wccp port-list 5 36
ContentEngine(config)# wccp port-list 6 37
ContentEngine(config)# wccp port-list 7 38
ContentEngine(config)# wccp port-list 8 39
```

The port lists in the example specify the port lists number 1 through 8.

The following example shows how to enable WCCP Version 2 on the Content Engine:

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

The following example shows how to enable the first user-defined WCCP service (service 90) on the Content Engine and associate router list 1 and port list 1 with this service. This example specifies that the traffic is to be redirected to the caching application on the Content Engine by entering the **application cache** option:

```
ContentEngine(config)# wccp service-number 90 router-list-num 1 port-list-num 1
application cache
```

This example configures the Content Engine to listen on the ports listed in port list 1 (port 32) for redirected requests from the routers on router list 1.

Tip You must specify the **application cache** option for each user-defined WCCP services (services 90 to 97) that are created in this sample scenario because you want the WCCP routers to redirect the traffic to the caching application (instead of the streaming application) on the Content Engine.

The following example shows how to enable the second user-defined WCCP service (service 91) on the Content Engine and associate router list 1 and port list 2 with this service:

```
ContentEngine(config)# wccp service-number 91 router-list-num 1 port-list-num 2
application cache
```

This example configures the Content Engine to listen on the ports listed in port list 2 (port 33) for redirected requests from the WCCP Version 2-enabled routers in router list 1.

The following example shows how to enable the third user-defined WCCP service (service 92) on the Content Engine and associate router list 1 and port list 3 with this service:

```
ContentEngine(config)# wccp service-number 92 router-list-num 1 port-list-num 3
application cache
```

This example configures the Content Engine to listen on the ports listed in port list 3 (port 34) for redirected requests from the routers on router list 1.

The following example shows how to enable the fourth user-defined WCCP service (service 93) on the Content Engine and associate router list 1 and port list 4 with this service:

```
ContentEngine(config)# wccp service-number 93 router-list-num 1 port-list-num 4
application cache
```

This example configures the Content Engine to listen on the ports listed in port list 4 (port 35) for redirected requests from the routers in router list 1.

The following example shows how to enable the fifth user-defined WCCP service (service 94) on the Content Engine and associate router list 1 and port list 5 with this service:

```
ContentEngine(config)# wccp service-number 94 router-list-num 1 port-list-num 5
application cache
```

This example configures the Content Engine to listen on the ports listed in port list 5 (port 36) for redirected requests from the routers on router list 1.

The following example shows how to enable the sixth user-defined WCCP service (service 95) on the Content Engine and associate router list 1 and port list 6 with this service:

```
ContentEngine(config)# wccp service-number 95 router-list-num 1 port-list-num 6
application cache
```

This example configures the Content Engine to listen on the ports listed in port list 6 (port 37) for redirected requests from the routers on router list 1.

The following example shows how to enable the seventh user-defined WCCP service (service 96) on the Content Engine and associate router list 1 and port list 7 with this service:

```
ContentEngine(config)# wccp service-number 96 router-list-num 1 port-list-num 7
application cache
```

This example configures the Content Engine to listen on the ports listed in port list 7 (port 38) for redirected requests from the routers on router list 1.

The following example shows how to enable the eighth user-defined WCCP service (service 97) on the Content Engine and associate router list 1 and port list 8 with this service:

```
ContentEngine(config)# wccp service-number 97 router-list-num 1 port-list-num 8
application cache
```

This example configures the Content Engine to listen on the ports listed in port list 8 (port 39) for redirected requests from the routers on router list 1.

Related Commands

ftp proxy incoming
http proxy incoming
https proxy incoming
proxy-protocols
rtsp proxy incoming
show http proxy
show https proxy
show wccp flows service-number
show wccp masks service-number
show wccp services
show wccp slowstart service-number
show wccp status
wccp version 2

wccp shutdown

To set the maximum time interval after which the Content Engine will perform a clean shutdown of WCCP, use the **wccp shutdown** global configuration command. To disable the clean shutdown, use the **no** form of the command.

wccp shutdown max-wait *seconds*

no wccp shutdown max-wait

Syntax Description

max-wait	Sets the clean shutdown time interval.
seconds	Time in seconds (0–86400). The default is 120 seconds.

Defaults

The maximum time interval before a clean shutdown is 120 seconds by default.

Command Modes

global configuration

Usage Guidelines

To prevent broken TCP connections, the Content Engine performs a clean shutdown of WCCP after a **reload** or **wccp version** command is entered. The Content Engine does not reboot until either all connections have been serviced or the configured **max-wait** interval has elapsed.

During a clean shutdown, the Content Engine continues to service the flows it is handling but starts to bypass new flows. When the number of flows goes down to zero, the Content Engine takes itself out of the cluster by having its buckets reassigned to other Content Engines by the lead Content Engine. TCP connections can still be broken if the Content Engine crashes or is rebooted without WCCP being cleanly shut down. The clean shutdown can be aborted while in progress.

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

Examples

The following example shows how to configure the Content Engine to wait for 1000 seconds before shutdown:

```
ContentEngine(config)# wccp shutdown max-wait 1000
```

The following example shows how to shut down WCCP Version 2 on the Content Engine by entering the **no wccp version 2** command. After you enter the **no wccp version 2** command, the Content Engine waits for a maximum of 1000 seconds before it shuts down WCCP Version 2.

```
Content Engine(config)# no wccp version 2
```

The following countdown message appears, indicating how many seconds remain before WCCP shuts down on the Content Engine:

```
Waiting (999 seconds) for WCCP shutdown. Press ^C to skip shutdownn
```

The clean shutdown can be aborted while in progress by simultaneously pressing **Ctrl-C** after the countdown message appears.

Related Commands

wccp flow-redirect
wccp slow-start
wccp version

wccp slow-start enable

To enable the slow-start capability of the caching service on the Content Engine, use the **wccp slow-start enable** global configuration command. To disable the slow-start capability, use the **no** form of this command.

wccp slow-start enable

no wccp slow-start enable

Syntax Description

enable	Enables WCCP slow start.

Defaults

Enabled

Command Modes

global configuration

Usage Guidelines

Within a cluster of Content Engines, TCP connections are redirected to other Content Engines as units are added or removed. A Content Engine can be overloaded if it is reassigned new traffic too quickly or introduced abruptly into a fat pipe.

WCCP slow start performs the following tasks to prevent a Content Engine from being overwhelmed when it comes online or is reassigned new traffic:

- TCP flow protection when WCCP 2 is enabled and a Content Engine is introduced into the cluster
- TCP flow protection when WCCP 2 is disabled and a Content Engine is leaving the cluster
- Load assignment to the Content Engine in slow increments rather than a full load at bootup

Slow start is applicable only in the following cases:

- Initial bootup when there is no Content Engine present in the server farm
- When a new Content Engine is added to a cluster that is not handling the full load; for example, when there are some buckets that are being shed by the cluster
- When a Content Engine in a cluster has failed

In all other cases, slow start is not necessary and all the Content Engines can be assigned their share of traffic.

Examples

The following example shows how to enable the slow-start capability of the caching service on a Content Engine:

```
ContentEngine# wccp slow-start enable
```

The following example shows how to disable the slow-start capability of the caching service on a Content Engine:

```
ContentEngine# no wccp slow-start enable
```

Related Commands wccp flow-redirect

wccp spoof-client-ip enable

To enable client IP spoofing, use the **wccp spoof-client-ip** global configuration command. To disable client IP spoofing, use the **no** form of this command.

wccp spoof-client-ip enable

no wccp spoof-client-ip enable

Syntax Description

enable	Enables client IP spoofing.

Defaults

Disabled

Command Modes

global configuration

Usage Guidelines

In transparent caching, when the Content Engine contacts the origin web server, it uses the Content Engine's own IP address instead of the client's IP address, on behalf of which the Content Engine is making the request. With IP spoofing, the transparent redirection process enables the Content Engine to send out the client's IP address for authentication purposes on origin web servers. Enable client IP spoofing with the **wccp spoof-client-ip enable** command. Disable this feature with the **no wccp spoof-client-ip enable** command.

Note Enabling IP spoofing on a Content Engine applies only to HTTP requests.

When the Content Engine is performing normal WCCP caching without IP spoofing, the Content Engine connects the origin servers using its own IP address. Routers identify the requests as coming from the Content Engine using its own source IP address in the request and forwards the requests out onto the WAN. The router forwards requests coming from the clients on the LAN to the Content Engine.

When configured for IP spoofing, the Content Engine connects to the origin server using the client IP address instead of its own IP address. At this point, the router cannot identify requests coming from the Content Engine because the source IP address of the request is not that of the Content Engine. To prevent these Content Engine requests from being returned as unresolved, enter the **ip wccp redirect exclude in** command to prevent the WCCP-enabled router from intercepting any requests coming in on the WCCP-enabled interface to which the Content Engine is connecting. To allow the interception of client requests, you must configure the **ip wccp redirect out** command on those interfaces connected to the client and the WAN.

Note Because the **ip wccp redirect exclude in** and **ip wccp redirect out** commands are contradictory and cannot coexist, different router interfaces must be configured, effectively placing the Content Engines on different subnets.

The ACNS software prior to Release 5.0.7 did not perform IP address spoofing for transparently intercepted proxy-style requests. However, with the ACNS software, Release 5.0.7 and later releases, IP address spoofing is performed for transparently intercepted proxy-style requests when the

Content Engine is configured to use the proxy server from the original request to fetch the content. To configure a Content Engine to use the proxy server from the original request, use the **proxy-protocols transparent original-proxy** global configuration command.

With a proxy-style request, the client sends a proxy-style HTTP request if the client is configured to send HTTP requests directly to a specific Content Engine. The client sends the request to the IP address of the proxy server, with the complete destination URL, including the name of the origin server in the HTTP method (for example, GET).

In the case of a server-style HTTP request, the client sends the request directly to the destination server with the HTTP Host header containing the domain name of the origin server and the HTTP method containing the path to the file or script that the client is requesting.

The proxy-style request for myfile.html located in mydirectory in the domain myserver.com, when transparently intercepted, will have the following initial HTTP line:

GET http://myserver.com/mydirectory/myfile.html HTTP/1.1

The server-style request for myfile.html located in mydirectory in the domain myserver.com, when transparently intercepted, will have the following initial HTTP line:

GET /mydirectory/myfile.html HTTP/1.1

To enable IP spoofing on a Content Engine, enter the **wccp spoof-client-ip enable** global configuration command. To disable IP spoofing, enter the **no wccp spoof-client-ip enable** global configuration command.

Tip The Content Engine can also use authentication traffic bypass to automatically generate a dynamic access list and to connect to a server using the client's IP address for selected client-server pairs. For more information, see the "Authentication Traffic Bypass" section. You can also forward the client's IP address without turning IP spoofing on by using the **http append x-forwarded-for-header** global configuration command on the Content Engine that is serving the request.

IP spoofing is recommended in the following scenarios:

- Logging of user IP addresses
- Filtering based on user IP addresses
- Policy-based routing to provide some users better service than others

These restrictions block IP spoofing of some or all client traffic even though globally, IP spoofing is enabled. Observe the following restrictions on IP spoofing:

- The client request is not transparently redirected (proxy-style requests)
- The request is transparently redirected but the request is proxy style, and the proxy-protocols transparent original-proxy configuration is not present
- An HTTP outgoing proxy has been configured

The client request matches any of the following rule patterns:

  - rule use-proxy
  - rule use-server
  - rule rewrite

For more information on IP spoofing configuration, see the *Cisco ACNS Software Configuration Guide for Locally Managed Deployments*.

Examples

The following example enables client IP spoofing on the Content Engine:

```
ContentEngine(config)# wccp spoof-client-ip enable
```

The following example enables client IP spoofing on the Content Engine:

```
ContentEngine(config)# no wccp spoof-client-ip enable
```

Related Commands

http (global configuration mode)
proxy-protocols

wccp version

To specify the version of WCCP that the Content Engine should use, enter the **wccp version** global configuration command. To disable the currently running version, use the **no** form of this command.

wccp version {**1** | **2**}

no wccp version {**1** | **2**}

Syntax Description

1	WCCP Version 1.
2	WCCP Version 2.

Defaults

No default behavior or values

Command Modes

global configuration

Usage Guidelines

Both WCCP versions allow transparent caching of web content. Only one version of WCCP can be enabled on a Content Engine at the same time. We recommend that you run WCCP Version 2 because it supports a broader set of WCCP options and services and provides multiple router support (router lists). For a detailed description of both versions, see the *Cisco ACNS Software Configuration Guide for Locally Managed Deployments*. It is not necessary to disable WCCP Version 1 before enabling WCCP Version 2, and vice versa. Be sure the routers used in the WCCP environment are running a software version that supports the WCCP version configured on the Content Engine.

When operating with WCCP Version 2, the Content Engine performs a clean shutdown after a **reload**, **wccp version 1**, or **no wccp version 2** command is entered. A clean shutdown prevents broken TCP connections.

Examples

The following example shows how to disable WCCP version 1 on a Content Engine:

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

The following example shows how to enable WCCP version 2 on a Content Engine:

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

Related Commands

wccp home-router

wccp web-cache

To instruct the router to run the web cache service with WCCP Version 2, use the **wccp web-cache** global configuration command. To disable this function, use the **no** form of this command.

wccp web-cache {**mask** {[**dst-ip-mask** *hex_num*] [**dst-port-mask** *port_hex_num*] [**src-ip-mask** *hex_num*] [**src-port-mask** *port_hex_num*]} | **router-list-num** *num* [**assign-method-strict**] [**l2-redirect**] [**mask-assign**] [**password** *key*] [**weight** *percentage*]}

no wccp web-cache

Syntax Description

mask	Sets the mask used for the Content Engine assignment. Configure at least one mask; the maximum is four masks.
dst-ip-mask	(Optional) Sets the mask used to match the packet destination IP address.
hex_num	IP address mask defined by a hexadecimal number (for example, 0xFC000000). The range is 0x00000000–FE000000.
dst-port-mask	(Optional) Sets the mask used to match the packet destination port number.
port_hex_num	Destination port mask defined by a hexadecimal number (for example, 0xFC00). The port range is 0–65535.
src-ip-mask	(Optional) Sets the mask used to match the packet source IP address.
src-port-mask	(Optional) Sets the mask used to match the packet source port number.
router-list-num	Sets the router list number.
num	Router list number (1–8).
assign-method-strict	(Optional) Forces WCCP to strictly use only the configured assignment method. For more information, see the "Assignment Method" section.
l2-redirect	(Optional) Sets the packet forwarding by Layer 2 redirect. For more information, see the "WCCP Layer 2 Support" section on page 2-839.
mask-assign	(Optional) Uses the mask method for the Content Engine assignment.
password	(Optional) Sets the authentication password to be used for secure traffic among the Content Engines within a cluster and the router for a specified service. Be sure to enable all other Content Engines and routers within the cluster with the same password.
key	WCCP service password key. Passwords must not exceed eight characters.
weight	(Optional) Sets the weight percentage for load balancing. For more information, see the "Load Balancing" section.
percentage	Percentage value (0–100).

Defaults

wccp web-cache: disabled

dst-ip-mask: 0x00001741

src-ip-mask: 0x00000000

dst-port-mask: 0x0

src-port-mask: 0x0

Command Modes global configuration

Usage Guidelines The standard web-cache service (service 0) permits a single WCCP Version 1-enabled router or one or more WCCP Version 2-enabled routers to redirect HTTP traffic to Content Engines on port 80 only. In order for a Content Engine to accept redirected HTTP requests on port 80, you must configure the standard web-cache service on the Content Engine (transparent HTTP forward proxy caching).

Use the **wccp web-cache** command to enable web cache service with WCCP Version 2. With web cache service, the router balances the traffic load within a Content Engine cluster based on the destination IP address (for example, the web server IP address).

You must set the **wccp router-list** command before you use this command.

Both **weight** and **password** are optional and can be used together or separately.

To enable the use of a password for a secure web cache cluster, use the **password** *key* option and be sure to enable all other Content Engines and routers within the cluster with the same password.

The **l2-redirect** option permits the Content Engine to receive transparently redirected traffic from a WCCP Version 2-enabled switch or router if the Content Engine has a Layer 2 connection with the device, and the device is configured for Layer 2 redirection.

The **weight** parameter represents a percentage of the total load redirected to the Content Engine (for example, a Content Engine with a weight of 30 receives 30 percent of the total load). If the total of all weight parameters in a Content Engine cluster exceeds 100, the percentage load for each Content Engine is recalculated as the percentage that its weight parameter represents of the combined total.

The web-cache service (service 0) needs to be configured both on a Content Engine and a WCCP-enabled router. By configuring this WCCP service on the router, the WCCP router redirects HTTP requests transparently to the Content Engine on port 80. By configuring this service on the Content Engine, the Content Engine listens on port 80 for redirected HTTP requests. If the Content Engine determines that it should accept and process the redirected HTTP request, it retrieves the requested information from the origin server if it is not already stored in its cache, caches a copy of the content in its local storage if the content is cacheable, and then sends the requested content to the client browser.

Examples The following example shows how to associate the router list 1 with the standard web cache service:

```
ContentEngine(config)# wccp web-cache router-list-num 1
```

The following example shows how to disable the web cache service:

```
ContentEngine(config)# no wccp web-cache
```

The following example shows how to configure the standard web-cache service (service 0) on a Content Engine by enabling WCCP Version 1 or Version 2 on the Content Engine:

```
ContentEngine(config)# wccp version 1
```

or

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

The Content Engine must be running WCCP Version 2 to support any of the WCCP services other than the web-cache service (service 0). If you enable WCCP Version 1 instead of Version 2 on this Content Engine, only a single WCCP router can be configured to support the only supported service (the standard web-cache service). If you select Version 2, up to 32 WCCP routers can be specified to support a particular WCCP service, and all WCCP services are supported.

The following example shows how to create a router list that specifies the routers that will support the web cache service:

```
ContentEngine(config)# wccp router-list 1 10.0.1.1
```

The IP address or multicast address specifies which router will support the web-cache service for this Content Engine. If different routers will be used for different WCCP services, you must create more than one router list. In this example, there is only one router on router list 1 (the router that you just configured for the standard web-cache service, which has an IP address of 10.0.1.1).

The following example shows how to inform the WCCP-enabled router in the specified router list that this Content Engine is accepting redirected web cache requests on port 80:

```
ContentEngine(config)# wccp web-cache router-list-num 1 l2-redirect mask-assign
```

The **l2-redirect** option specifies Layer 2 redirection as the packet-forwarding method (instead of GRE). The **mask-assign** option specifies the mask assignment as the load-balancing method (instead of the default hash assignment method) for this WCCP service.

The following example shows how to exit global configuration mode:

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

The following example shows how to write the running configurations to the nonvolatile memory:

```
ContentEngine# write memory
```

The following example shows how to configure the standard web-cache service (service 0) on a WCCP router by turning on WCCP Version 2 on the router:

```
Router# configure terminal
Router(config)# ip wccp version 2
```

The following example shows how to configure the standard web-cache service on the WCCP-enabled router and set a password for this router:

```
Router(config)# ip wccp web-cache password 0 password
```

where

- **password** directs the WCCP-enabled router to apply MD5 authentication to messages received from the specified service group. Messages that are not accepted by the authentication are discarded.
- 0 is an optional value that indicates that the HMAC MD5 algorithm is used to encrypt the password. This value is generated when an encrypted password is created for the Content Engine.
- *password* is the optional password name that is combined with the HMAC MD5 value to create a secure connection between the WCCP-enabled router and the Content Engine.

The following example shows how to turn on the standard web-cache service on the router:

```
Router(config)# ip wccp web-cache
```

The following example shows how to specify the Ethernet interface 0/1 on which the standard web-cache service will run:

```
Router(config)# interface ethernet 0/1
```

The following example shows how to configure the router to check the HTTP traffic that arrives on the interface on which the standard web-cache service is configured (for example, Ethernet interface 0/1):

```
Router(config-if)# ip wccp web-cache redirect in
```

The router determines whether it should redirect these packets to the Content Engine. This Content Engine functions as a transparent forward proxy server that accepts redirected HTTP requests on port 80 from this WCCP Version 2 router.

Related Commands

show statistics wmt all
show wccp content-engines
show wccp flows web-cache
show wccp masks web-cache
show wccp routers
show wccp services
show wccp slowstart web-cache
show wccp status
show wmt
wccp version 2

wccp wmt

To instruct the router to run the web cache service with WCCP and Windows Media Technologies (WMT), use the **wccp wmt** global configuration command. To disable this function, use the **no** form of this command.

wccp wmt {**mask** {[**dst-ip-mask** *hex_num*] [**dst-port-mask** *port_hex_num*] [**src-ip-mask** *hex_num*] [**src-port-mask** *port_hex_num*]} | **router-list-num** *num* [**assign-method-strict**] [**hash-destination-ip**] [**hash-destination-port**] [**hash-source-ip**] [**hash-source-port**] [**l2-redirect**] [**mask-assign**] [**password** *key*] [**weight** *percentage*]}

no wccp wmt

Syntax Description

mask	Sets the mask used for the Content Engine assignment. Configure at least one mask; the maximum is four masks.
dst-ip-mask	(Optional) Sets the mask used to match the packet destination IP address.
hex_num	IP address mask defined by a hexadecimal number (for example, 0xFC000000). The range is 0x00000000–FE000000.
dst-port-mask	(Optional) Sets the mask used to match the packet destination port number.
port_hex_num	Port mask defined by a hexadecimal number (for example, 0xFC00). The port range is 0–65535.
src-ip-mask	(Optional) Sets the mask used to match the packet source IP address.
src-port-mask	(Optional) Sets the mask used to match the packet source port number.
router-list-num	Specifies the router list number.
num	Router list number (1–8).
assign-method-strict	(Optional) Forces WCCP to strictly use only the configured assignment method. For more information, see the “Assignment Method” section.
hash-destination-ip	(Optional) Defines the load-balancing hash of the destination IP address (the default).
hash-destination-port	(Optional) Defines the load-balancing hash of the destination port.
hash-source-ip	(Optional) Defines the load-balancing hash of the source IP address.
hash-source-port	(Optional) Defines the load-balancing hash of the source port.
l2-redirect	(Optional) Sets the packet forwarding by Layer 2 redirect. For more information, see the “WCCP Layer 2 Support” section on page 2-839.
mask-assign	(Optional) Uses the mask method for the Content Engine assignment.
password	(Optional) Sets the authentication password to be used for secure traffic among the Content Engines within a cluster and the router for a specified service. Be sure to enable all other Content Engines and routers within the cluster with the same password.
key	WCCP service password key. Passwords must not exceed eight characters.
weight	(Optional) Sets the weight percentage for load balancing. For more information, see the “Load Balancing” section.
percentage	Percentage value (0–100).

Defaults

wccp wmt: disabled

dst-ip-mask: 0x00000000

src-ip-mask: 0x00001741

dst-port-mask: 0x0

src-port-mask: 0x0

Command Modes

global configuration

Usage Guidelines

Use the **router-list-num** *num* option to specify the router list.

The **l2-redirect** option permits the Content Engine to receive transparently redirected traffic from a WCCP Version 2-enabled switch or router if the Content Engine has a Layer 2 connection with the device, and the device is configured for Layer 2 redirection.

Both **weight** and **password** are optional and may be used in combination or separately.

To enable the use of a password for a secure web cache cluster, use the **password** *key* option and enable all other Content Engines and routers within the cluster with the same key.

The **weight** parameter is the percentage of the total load redirected to the Content Engine. If the total percentage of every weight parameter in a Content Engine cluster exceeds 100, the percentage for each load is recalculated as a percentage that its weight parameter represents of the combined total.

Note Although typical router configuration in a branch office scenario involves configuring the outgoing interface, you can also configure the incoming interface on the router for traffic redirection (using the **ip wccp** *service number* **redirect in** interface configuration command).

Examples

The following examples show that the **wccp wmt router-list-num** command sets the router to run the WMT service with assigned router list 2 and sets the Layer 2 redirect, the authentication password key, and the cluster load balancing weight percentages:

```
ContentEngine(config)# wccp wmt router-list-num 2

ContentEngine(config)# wccp wmt router-list-num 2 password cisco

ContentEngine(config)# wccp wmt router-list-num 2 l2-redirect

ContentEngine(config)# wccp wmt router-list-num 2 weight 45
```

The following example shows how to configure transparent redirection of WMT requests through WCCP Version 2 on the Content Engine that will be functioning as the transparent proxy server for redirected WMT requests from Router A by enabling WCCP Version 2 on the Content Engine:

Note This example assumes that you have enabled the licensed WMT feature on the Content Engine.

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

The following example shows how to create the numbered router list that you want to associate with this WCCP Version 2 service:

```
ContentEngine(config)# wccp router-list 1 172.16.25.25
```

In the example, there is one WCCP Version 2-enabled router (Router A) associated with router list 1. Router A has an IP addresses of 172.16.25.25.

The following example shows how to enable the router list (router list 1) that you just created in the previous example:

```
ContentEngine(config)# wccp wmt router-list-num 1
```

The following example shows how to save the new configuration on the Content Engine:

```
ContentEngine# copy running-config startup-config
```

Related Commands

show statistics wmt all
show wccp content-engines
show wccp flows wmt
show wccp masks wmt
show wccp routers
show wccp services
show wccp slowstart wmt
show wccp status
show wmt
wccp version 2

wccp wmt-rtspu

To configure WCCP Version 2 WMT RTSP transparent interception, use the **wccp wmt-rtspu** global configuration command. To disable this function, use the **no** form of this command.

wccp wmt-rtspu {**mask** {[**dst-ip-mask** *hex_num*] [**dst-port-mask** *port_hex_num*] [**src-ip-mask** *hex_num*] [**src-port-mask** *port_hex_num*]} | **router-list-num** *num* [**assign-method-strict**] [**hash-destination-ip**] [**hash-destination-port**] [**hash-source-ip**] [**hash-source-port**] [**l2-redirect**] [**mask-assign**] [**password** *key*] [**weight** *percentage*]}

no wccp wmt-rtspu

Syntax Description

mask	Sets the mask used for the Content Engine assignment. Configure at least one mask; the maximum is four masks.
dst-ip-mask	(Optional) Sets the mask used to match the packet destination IP address.
hex_num	IP address mask defined by a hexadecimal number (for example, 0xFC000000). The range is 0x00000000–FE000000.
dst-port-mask	(Optional) Sets the mask used to match the packet destination port number.
port_hex_num	Port mask defined by a hexadecimal number (for example, 0xFC00). The port range is 0–65535.
src-ip-mask	(Optional) Sets the mask used to match the packet source IP address.
src-port-mask	(Optional) Sets the mask used to match the packet source port number.
router-list-num	Specifies the router list number.
num	Router list number (1–8).
assign-method-strict	(Optional) Forces WCCP to strictly use only the configured assignment method. For more information, see the "Assignment Method" section.
hash-destination-ip	(Optional) Defines the load-balancing hash of the destination IP address (the default).
hash-destination-port	(Optional) Defines the load-balancing hash of the destination port.
hash-source-ip	(Optional) Defines the load-balancing hash of the source IP address.
hash-source-port	(Optional) Defines the load-balancing hash of the source port.
l2-redirect	(Optional) Sets the packet forwarding by Layer 2 redirect. For more information, see the "WCCP Layer 2 Support" section on page 2-839.
mask-assign	(Optional) Uses the mask method for the Content Engine assignment.
password	(Optional) Sets the authentication password to be used for secure traffic among the Content Engines within a cluster and the router for a specified service. Be sure to enable all other Content Engines and routers within the cluster with the same password.
key	WCCP service password key. Passwords must not exceed eight characters.
weight	(Optional) Sets the weight percentage for load balancing. For more information, see the "Load Balancing" section.
percentage	Percentage value (0–100).

Defaults

wccp wmt-rtspu: disabled

dst-ip-mask: 0x00000000

src-ip-mask: 0x00001741

dst-port-mask: 0x0

src-port-mask: 0x0

Command Modes

global configuration

Usage Guidelines

Use the **router-list-num** *num* option to specify the router list.

The **l2-redirect** option permits the Content Engine to receive transparently redirected traffic from a WCCP Version 2-enabled switch or router if the Content Engine has a Layer 2 connection with the device, and the device is configured for Layer 2 redirection.

Both **weight** and **password** are optional and may be used in combination or separately.

To enable the use of a password for a secure web cache cluster, use the **password** *key* option and enable all other Content Engines and routers within the cluster with the same key.

The **weight** parameter is the percentage of the total load redirected to the Content Engine. If the total percentage of every weight parameter in a Content Engine cluster exceeds 100, the percentage for each load is recalculated as a percentage that its weight parameter represents of the combined total.

With transparent redirection of WMT requests, a WCCP Version 2-enabled router or a Layer 4 switch transparently redirects WMT RTSP requests to the Content Engine (acting as a transparent proxy server). WMT RTSP transparent redirection is used to support WMT transparent caching on a Content Engine that is running the ACNS 5.3 software and later releases. With this type of transparent redirection, you must configure WMT RTSP redirection on the WCCP Version 2-enabled routers or the Layer 4 switch and on the Content Engine that will receive these redirected WMT requests.

Note To perform WCCP transparent redirection of WMT RTSP traffic, you must enable service 80 and service 83 on the WCCP Version 2-enabled router.

To configure WMT transparent redirection of WMT requests (WMT RTSP redirection) through WCCP Version 2, you must perform both of these tasks:

- Configure WMT RTSP transparent redirection (WCCP Version 2 services 80 and 83) on the WCCP Version 2 routers that will support this Windows Media service
- Configure WMT RTSP transparent redirection on the Content Engine

Note Although typical router configuration in a branch office scenario involves configuring the outgoing interface, you can also configure the incoming interface on the router for traffic redirection (using the **ip wccp** *service number* **redirect in** interface configuration command).

Examples

The following example shows how to configure WMT RTSP transparent interception through WCCP Version 2 on a router by turning on WCCP Version 2 on the router (Router A):

```
RouterA# configure terminal
RouterA(config)# ip wccp version 2
```

The following example shows how to turn on service 80 (the rtsp redirection service) on Router A:

```
RouterA(config)# ip wccp 80
```

The following example shows how to turn on service 83 (the wmt-rtspu redirection service) on Router A:

```
RouterA(config)# ip wccp 83
```

The following example shows how to use the **interface** global configuration command to specify an interface on which the RTSP redirection services will run on Router A:

```
RouterA(config)# interface Ethernet 0
```

The example shows how to configure the outgoing interface to the Internet as Ethernet 0 on Router A.

The following example shows how to enable WCCP redirection to service 80 and 83 on the specified router interface (in this case, the outgoing interface) from interface configuration mode on Router A:

```
RouterA(config-if)# ip wccp 80 redirect out
RouterA(config-if)# ip wccp 83 redirect out
```

The following example shows how to configure WMT RTSP transparent through WCCP Version 2 on the Content Engine that will be functioning as the transparent proxy server for redirected WMT requests from Router A by enabling WCCP Version 2 on the Content Engine:

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

Note This example assumes that you have enabled the licensed WMT feature on the Content Engine, as described in the "Enabling WMT on the Content Engine" section on page 2-905.

The following example shows how to create the numbered router list that you want to associate with services 80 and 83:

```
ContentEngine(config)# wccp router-list 1 172.16.25.25
```

In the example, there is one WCCP Version 2-enabled router (Router A) associated with router list 1. Router A has an IP addresses of 172.16.25.25.

The following example shows how to enable the router list (router list 1) that you just created in the previous example:

```
ContentEngine(config)# wccp wmt-rtspu router-list-num 1
```

The following example shows how to save the new configuration on the Content Engine:

```
ContentEngine# copy running-config startup-config
```

Related Commands

show wccp content-engines
show wccp flows wmt-rtspu
show wccp masks wmt-rtspu
show wccp routers
show wccp services
show wccp slowstart wmt-rtspu
show wccp status
wccp version 2

websense-server

To enable the use of a Websense HTTP URL filtering plug-in in a Content Engine, use the **websense-server** global configuration command. To disable this function, use the **no** form of this command.

websense-server {**enable** | **service** {**edir-agent** {**activate** | **edir-server** {**administrative-dn** *name* | **administrative-passwd** *passwd* | **host** {*hostname* | *ip-address*} [**port** *port-num*] | **root-context** *root-location*}} **eim activate** | **network-agent activate** | **policy** {**local activate** | **remote host** {*host-name* | *ip-address*} [**port** *port-num*]} | **radius-agent** {**activate** | **incoming** {**accnt-port** *port-num* | **auth-port** *port-num*} | **outgoing** {**accnt-port** *port-num* | **auth-port** *port-num* | **host** {*hostname* | *ip-address*}}} | **user activate**}}

no websense-server {**enable** | **service** {**edir-agent** {**activate** | **edir-server** {**administrative-dn** *name* | **administrative-passwd** *passwd* | **host** {*hostname* | *ip-address*} [**port** *port-num*] | **root-context** *root-location*}} **eim activate** | **network-agent activate** | **policy** {**local activate** | **remote host** {*host-name* | *ip-address*} [**port** *port-num*]} | **radius-agent** {**activate** | **incoming** {**accnt-port** *port-num* | **auth-port** *port-num*} | **outgoing** {**accnt-port** *port-num* | **auth-port** *port-num* | **host** {*hostname* | *ip-address*}}} | **user activate**}}

Syntax Description

enable	Enables the local Websense server on the Content Engine.
service	Configures services of the local Websense server.
edir-agent	Configures the installation of the local eDirectory agent.
activate	Activates the local eDirectory agent on the Content Engine.
edir-server	Configures parameters related to the eDirectory server.
administrative-dn	Sets the administrative distinguished name.
name	Administrative distinguished name.
administrative-passwd	Sets the administrative password.
passwd	Administrative password.
host	Specifies the external eDirectory server that the Content Engine is to contact to request a search of an external LDAP database.
hostname	Hostname of the external eDirectory server.
ip-address	IP address of the external eDirectory server.
port	Specifies the port number of the eDirectory server.
port-num	Port number of the eDirectory server (1–65535). The default is 389.
root-context	Specifies the root location (root context) of the directory service.
root-location	Root location of the directory service.
eim	Configures the installation of the local employee Internet management (EIM) server.
activate	Activates the local EIM server on the Content Engine.
network-agent	Configures the installation of the local network agent.
activate	Activates the local network agent on the Content Engine.
policy	Configures the policy server.
local	Configures the installation of the local policy server.
activate	Activates the local policy server on the Content Engine.

remote	Configures the remote policy server.
host	Specifies the remote policy server.
host-name	Hostname of the remote policy server.
ip-address	IP address of the remote policy server.
port	(Optional) Configures the policy server port.
port-num	Policy server port number. The default port is 55806.
radius-agent	Configures the installation of the local RADIUS agent.
activate	Activates the local RADIUS agent on the Content Engine.
incoming	Configures port numbers for incoming requests.
accnt-port	Specifies the port number on which accounting requests should be received from the RADIUS client.
port-num	Port number for receiving accounting requests (1–65535). The default is 12346.
auth-port	Specifies the port number on which authentication requests should be received from the RADIUS client.
port-num	Port number for receiving authentication requests (1–65535). The default is 12345.
outgoing	Configures port numbers for outgoing requests.
accnt-port	Specifies the port number on which accounting requests should be forwarded to the external RADIUS server.
port-num	Port number for forwarding accounting requests (1–65535). The default is 1646.
auth-port	Specifies the port number on which authentication requests sthould be forwarded to the external RADIUS server.
port-num	Port number for forwarding authentication requests (1–65535). The default is 1645.
host	Specifies the external RADIUS server.
hostname	Hostname of the RADIUS server.
ip-address	IP address of the RADIUS server.
user	Configures the installation of the local user service.
activate	Activates the local user service on the Content Engine.

Defaults

websense-server: disabled

edir-server port: 389

radius-agent incoming accnt-port: 12346

radius-agent incoming auth-port: 12345

radius-agent outgoing accnt-port: 1646

radius-agent outgoing auth-port: 1645

Command Modes

global configuration

Usage Guidelines

The ACNS 5.2 software and later releases use Websense server Version 5.2.0, integrated with the Content Engine, for URL filtering. In the Websense 5.2.0 software, five services (the employee Internet management [EIM] server, the local network agent, the local RADIUS agent, the local eDirectory agent, and the local user service) replace the single Websense service (the local Websense server) that was supported in the Websense 5.0.1 software.

Note The terms *Websense plug-in* or *local Websense server* are used to refer collectively to the five Websense processes that are running internally on the Content Engine. These Websense processes are also called services.

In the ACNS 5.3 software and later releases, these five Websense services are all supported. (In the ACNS 5.2.x software, the local RADIUS agent service and the local eDirectory agent service were not supported.)

The local Websense server runs internally to the Content Engine (instead of running on a separate system and communicating with the Content Engine over the network) and uses approximately 60 MB to 140 MB of RAM in the Content Engine. We recommend that you run the local Websense server on Content Engines with at least 512 MB of RAM.

The default memory size for the Websense URL database that is cached in the RAM of the Content Engine is 50 MB. When you upgrade the Content Engine from the ACNS 5.2 software and earlier releases to the ACNS 5.3 software, an additional 300 MB of disk space is required for the Websense database backed up from the ACNS 5.2 software and earlier releases.

Note You can delete the Websense URL database that has been retained after you upgrade the Content Engine to the ACNS software 5.3 software and later releases from the ACNS 5.2 software and earlier releases by deleting the Websense file from the /local1/WebsenseEnterprise/EIM/OldConfigs/Reinstall_*version* directory on the Content Engine.

Websense provides an image of Websense server that resides in the /local/WebsenseEnterprise/EIM directory on the Content Engine. All the executables and the configuration and logging files are stored in this directory. This package requires about 50 MB of disk space in the /local1/WebsenseEnterprise/EIM directory. The current size of the Websense database is about 300 MB. When you update the Websense database, an additional 300 MB might be required when downloading the Websense URL database, increasing the total disk space requirement to about 650 MB.

If you upgrade the Content Engine from the ACNS 5.2 software and earlier releases to the ACNS 5.3 software release and later releases, we recommend that you delete the Websense file from the /local1/WebsenseEnterprise/EIM/OldConfigs/Reinstall_*version* directory to avoid unnecessary disk usage of about 140–300 MB before you download the URL database the next time. To ensure that you have enough disk space to properly download the Websense software, we recommend that you increase the amount of sysfs disk space to be greater than the default sysfs on your Content Engine.

When you enable Websense server and download the Websense URL database to the Content Engine for the first time, CPU usage will be high. We recommend that you enable Websense server during off-peak times or at times of low network traffic; otherwise, other processes running on the Content Engine may be affected. If one of the Websense processes exits, the local Websense server is automatically restarted on the Content Engine.

With the Websense 5.2 software, you can use a local or remote Websense policy server to activate the local EIM server, the local network agent, and the local user service individually on a Content Engine.

Note For information on the services of the local Webesense 5.2 server, see Chapter 11 of the *Cisco ACNS Software Configuration Guide for Locally Managed Deployments*.

Note In the ACNS 5.2 software and later releases, the **websense-server ip-address** and **websense-server user-server external** global configuration commands are deprecated.

In the ACNS software, Release 5.2 and later releases, you can configure a Content Engine to use up to two Websense servers for URL filtering.

Only one URL filtering scheme per protocol can be active at a time. In order to enable Websense URL filtering for requests over HTTP, you should make sure that no other URL filtering scheme is configured per protocol.

Use the show url-filter http EXEC command to display the URL filtering schemes that are currently enabled on this Content Engine for HTTP requests (HTTP, FTP-over-HTTP, and HTTPS-over-HTTP).

Note In the ACNS 5.2.x and 5.3.x software releases, Websense server Version 5.2 is supported on all Cisco Content Engine platforms. For more detailed information about configuring the Websense software, go to the following website: www.websense.com

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.

Configuring Ports for Websense Server

The Websense process requires that four ports be open for connections either from processes internal to the Content Engine or from external processes such as the Websense Manager.

You can modify the ports by exporting a copy of the eimserver.ini file using FTP from the /local1/WebsenseEnterprise/EIM directory on the Content Engine, modifying the file, deleting the eimserver.ini file on the Content Engine, and then sending the modified file to the Content Engine using FTP.

Note Websense server needs to be disabled and then reenabled to pick up newly configured ports. To disable the local Websense server, use the **no websense-server enable** global configuration command. Also make sure that you use the **url-filter http websense server** global configuration command to point the Websense client to the correct Websense server port.

In the ACNS 5.3 software release, the **websense-server service edir-agent edir-server** global configuration command was added. The command has several options that enable you to configure the settings for the eDirectory server.

For the local eDirectory agent to work properly, the IP address and authentication port of the outgoing host must match the IP address and outgoing authentication port of one of the LDAP servers that the Content Engine has been configured to use. You must ensure that you specify the IP address and outgoing authentication port of the external LDAP servers that the Content Engine is to use for authentication before you activate the eDirectory agent.

After specifying the settings for the local eDirectory agent, enter the **websense-server service edir-agent activate** global configuration command to activate the local eDirectory agent on the Content Engine. If you activate the local eDirectory agent before specifying the settings, the following message appears:

```
CONTENTENGINE(config)# websense-server service edir-agent activate
Warning: eDirectory Server Ip, eDirectory Administrator-dn, eDirectory
Administrator-passwd, eDirectory Root-context, not configured. eDirectory-Agent Service
will not run properly.
```

Note For information on configuring the Content Engine to use the local Websense server for URL filtering, see Chapter 11 of the *Cisco ACNS Software Configuration Guide for Locally Managed Deployments* publication.

Specifying the Settings for the Local RADIUS Agent

RADIUS authentication clients reside on the Content Engine running the ACNS 5.x software. When enabled, these clients send authentication requests to a central (remote) RADIUS server, which contains login authentication and network service access information. The local RADIUS agent, which is running on the Content Engine, acts as a proxy that forwards the RADIUS messages between the RADIUS client and the external RADIUS server.

In the ACNS 5.3 software release, the **websense-server service radius-agent** global configuration command was added. The command has several options that enable you to configure the necessary settings for the local RADIUS agent and to activate the agent on the Content Engine.

Use the **websense-server service radius-agent outgoing auth-port** global configuration command to specify the port number on which authentication requests should be forwarded to the external RADIUS server. The port number can be from 1 to 65535. The default port is 1645.

Use the **websense-server service radius-agent outgoing accnt-port** global configuration command to specify the port number on which accounting requests should be forwarded to the external RADIUS server. The port number can be from 1 to 65535. The default port is 1646.

Use the **websense-server service radius-agent outgoing host** global configuration command to specify the IP address of the external RADIUS server to which authentication requests should be forwarded.

For the local RADIUS agent to work properly, the IP address and authentication port number of the outgoing host must match the IP address and outgoing authentication port of one of the RADIUS servers that the Content Engine has been configured to use. You must ensure that you specify the IP address and outgoing authentication port number of the external RADIUS servers that the Content Engine is to use for authentication before you activate the RADIUS agent.

After specifying the settings for the local RADIUS agent, enter the **websense-server service radius-agent activate** global configuration command to activate the local RADIUS agent on the Content Engine. If you activate the RADIUS agent before specifying the settings, the following message appears:

```
CONTENTENGINE(config)# websense-server service radius-agent activate
Warning: Radius Server IP is not configured. Radius-Agent Service will not run properly.
```

Note For information on how to deactivate the local Websense server services on the Content Engine, see Chapter 11 of the *Cisco ACNS Software Configuration Guide for Locally Managed Deployments*.

Examples

The following example configures the Content Engine to use the local Websense policy server:

```
ContentEngine(config)# websense-server service policy local activate
```

The following example shows how to use the **host** command option to point the Content Engine to an external eDirectory server that has an IP address of 172.18.24.10:

```
ContentEngine(config)# websense-server service edir-agent edir-server host 172.18.24.10
```

The following example shows how to use the **administrative-passwd** command option to specify default244 as the administrative password. This administrative password is the password used by the the Content Engine to contact the external eDirectory server to request a database search.

```
ContentEngine(config)# websense-server service edir-agent edir-server
administrative-passwd default244
```

The following example installs the Websense EIM server on the Content Engine:

```
ContentEngine(config)# websense-server service eim activate
```

The following example installs the Websense network agent service on the Content Engine:

```
ContentEngine(config)# websense-server service network-agent activate
```

The following example uses the **show websense-server** EXEC command to show the status of Websense URL filtering before enabling it:

```
ContentEngine# show websense-server
Local Websense Server Information :
------------------------------

Websense Enterprise Version : 5.2.0

Websense Server not running.
```

The following example enables the use of local Websense URL filtering:

```
ContentEngine(config)# websense-server enable
ContentEngine(config)#
```

The following example uses the **show websense-server** EXEC command to show the status of Websense filtering:

```
ContentEngine# show websense-server
Local Websense Server Information :
----------------------------------

Websense Enterprise Version : 5.2.0
Websense Enterprise Build   : 1108699513

Websense server components installed :
Policy Server
EIM Server
User Service
Network Agent
Radius Agent
eDirectory Agent

Status of the components:
-------------------------


Websense Filtering Service running...
```

```
Websense Policy Server running...


Websense User Service running...


Websense Network Agent running...


Websense eDirectory Agent running...


Websense RADIUS Agent running...



Websense Server running...

----------------------------------------
 Sending SERVER_STATUS_REQUEST...
----------------------------------------

  Status Code = 0
  License Count = 0
  Elapsed Time = 1 ms

AVG TIME PER REQUEST = 1 ms

Websense Server Port : 15868
Websense Block Message Port : 15871
Websense Config Server Port : 55806
Websense Diagnostics Server Port : 15869

------------------------------
  Radius Agent Configuration...
------------------------------
Outgoing Requests :
   Radius Server : 10.10.10.10
   Authentication Port : 1645
   Accounting Port : 17000
Incoming Requests :
   Authentication Port : 18000
   Accounting Port : 25000

----------------------------------
  eDirectory Agent Configuration...
----------------------------------
   Administrative DN : xyz
   Administrative Password : ******
   Root Context : <none>

   Servers IP           Port Number
   ----------           -----------
   10.1.1.1              380
```

The following example uninstalls the Websense EIM server on the Content Engine:

```
ContentEngine(config)# no websense-server service eim activate
```

For example, use the **websense-server service radius-agent incoming** global configuration commands to configure the incoming port numbers for the local RADIUS agent requests as follows:

```
ContentEngine(config)# websense-server service radius-agent incoming auth-port
[port-number]
ContentEngine(config)# websense-server service radius-agent incoming accnt-port
[port-number]
```

Related Commands

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

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

The following example displays the username of the user who has logged in to the Content Engine:

```
ContentEngine# whoami
admin
```

Related Commands

pwd

wmt (EXEC)

To start or stop Windows Media Technologies (WMT) multicast stations, use the **wmt** EXEC command.

wmt multicast-station {**start** *name* | **stop** *name*}

Syntax Description

multicast-station	Sets the WMT multicast stations to start or stop.
start	Starts a WMT multicast station.
name	Name of the WMT multicast station to be started.
stop	Stops a WMT multicast station.
name	Name of the WMT multicast station to be stopped.

Defaults

No default behavior or values

Command Modes

EXEC

Usage Guidelines

Use this command to start or stop particular WMT multicast stations.

Examples

The following examples show the **start** and **stop** options on the multicast station named acme:

```
ContentEngine# wmt multicast-station start acme

ContentEngine# wmt multicast-station stop acme
```

Related Commands

multicast
show multicast
show ntp
show statistics wmt
show wmt
wmt multicast

wmt (global configuration)

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 {**live-split** | **proxy-cache** | **vod**} **enable**

wmt accept-license-agreement

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

wmt advanced server log-forwarding enable

wmt bandwidth incoming bypass-list {*ip-address* | *hostname*} [*ip-address* | *hostname*]

wmt broadcast {**alias-name** *name* **source** *url*}

wmt cache {**enable** | **max-obj-size** *size* | **unique-stream-key**}

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

wmt enable

wmt evaluate

wmt extended transaction-log enable

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

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

wmt http allow extension *file_extensions*

wmt incoming *number*

wmt l4-switch enable

wmt license-key *key*

wmt live-url-stripping enable

wmt max-concurrent-sessions *number*

wmt multicast {**station-configuration** *name* [{*dest_addr dest_port media_source*} [**log** {**local** | **webserver** *webserver_url*}] [**play-forever**] [**unicast-url** *url*] | **failover** {**alternate-source** *alt_src_url* | **retry-count** *number* | **retry-interval** *number*} | **schedule-start** {*minute hour day month* | **now**}] | **time-to-live** *ttl*}

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** {**live-split** | **proxy-cache** | **vod**} **enable** | **accept-license-agreement** | **advanced** {**client** {**maximum-packet-size** *number* | **idle-timeout**} | **server log-forwarding enable**} | **bandwidth incoming bypass-list** {*ip-address* | *hostname*} [*ip-address* | *hostname*] | **broadcast** {**alias-name** *name*} | **cache** {**enable** | **max-obj-size** | **unique-stream-key**} |

disallowed-client-protocols {**http** [**rtspt** | **rtspu**] | **rtspt** [**http** | **rtspu**] | **rtspu** [**http** | **rtspt**]} | **enable** | **evaluate** | **extended transaction-log enable** | **fast-cache** {**enable** | **max-delivery-rate** *number*} | **fast-start** {**enable** | **max-bandwidth** *number*} | **http allow extension** *file_extensions* | **incoming** *number* | **l4-switch enable** | **license-key** | **live-url-stripping enable** | **max-concurrent-sessions** | **multicast** {**station-configuration** *name* [{*dest_addr dest_port media_source*} [**log** {**local** | **webserver** *webserver_url*}] [**play-forever**] [**unicast-url** *url*] | **failover** {**alternate-source** *alt_src_url* | **retry-count** [*number*] | **retry-interval** [*number*]} | **schedule-start** {*minute hour day month* | **now**}] | **time-to-live**} | **proxy outgoing** {**http** | **rtsp**} | **transaction-logs format** {**extended** {**wms-41** | **wms-90**} | **wms-41** | **wms-90**}}

Syntax Description

accelerate	Configures the WMT streaming acceleration.
live-split	Configures the performance improvement for live splitting.
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 Content Engine to accelerate the performance of the video on demand.
enable	Enables the performance improvement for the video on demand.
accept-license-agreement	Acknowledges and accepts the end user license agreement (EULA). Although the **no** form of this command is available from the CLI, it simply prints an error message that the EULA acceptance cannot be revoked.
advanced	Configures WMT advanced settings.
client	Configures WMT advanced client features on the Content Engine.
maximum-packet-size	Specifies the client maximum packet size (WMT maximum IP packet size), used in VPN environments.
number	Maximum packet size of WMT stream in bytes (512–2048). The default is 1500 bytes.
idle-timeout	Specifies the maximum amount of time that the Content Engine is to wait for a response from a WMT client before timing out the connection.
number	Timeout value in seconds (30–300). The default is 60 seconds.
server	Configures WMT advanced server features on the Content Engine (added in the ACNS 5.3 software release).
log-forwarding	Specifies whether the Windows Media transaction logs should be sent to the upstream WMT server or upstream Content Engines. This setting applies to all protocols, such as HTTP, RTSPT, and RTSPU.
enable	Enables the current Content Engine to send WMT transaction logs to the upstream WMT server or upstream Content Engines.
bandwidth	Configures WMT bandwidth settings.
incoming	Allows bypassing of incoming bandwidth restrictions for broadcast alias and multicast stations.
bypass-list	Configures a list of up to four Content Engines that are exempted from checking for incoming bandwidth.
ip-address	IP address of an exempt Content Engine.
hostname	Hostname of an exempt Content Engine.

broadcast	Configures a live broadcast.
alias-name	Configures the broadcast alias name.
name	An alias for the source URL or station name to be used for a broadcast.
source	Specifies the broadcast source URL.
url	Broadcast source URL.
cache	Configures the WMT cache.
enable	Enables the WMT media cache.
max-obj-size	Sets the maximum size of the object to be cached.
size	Object size in megabytes (1–1000000). The default is 1024 megabytes.
unique-stream-key	Specifies that the unique stream ID should be used as the cache key The default is to use the stream ID as the cache key. For more information, see the "WMT Unique Stream Key" section on page 2-908.
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.
evaluate	Starts or continues a 60-day evaluation of WMT.
extended	Configures the WMT-extended configuration.
transaction-log	Configures the WMT-extended transaction-log configuration.
enable	Enables the extended transaction-log configuration.
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.
number	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). The default is 5.
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.
number	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.
file_extensions	Filename extensions to be served. A maximum of 20 filename extensions is allowed, with a maximum of 10 characters per extension.

incoming	Configures incoming WMT requests.
number	Port number to listen for requests (1–65535).
l4-switch	Configures Layer 4 switch interoperability for WMT.
enable	Enables Layer 4 switch interoperability for WMT.
license-key	Sets the WMT license key. Although the **no** form of this command is available from the CLI, it simply prints an error message that the license key cannot be removed.
key	License key.
live-url-stripping	Allows the WMT proxy to strip out personalization information from the URL before using it for live-splitting purposes during WMT playback.
enable	Enables live URL stripping.
max-concurrent-sessions	Configures the maximum number of unicast clients that can be served concurrently.
number	Limit for incoming unicast requests; this limit is subject to physical resources on the platform (1–8000).
multicast	Configures multicasting and scheduling.
station-configuration	Configures multicast stations.
name	Multicast station name.
dest_addr	Multicast station destination IP address.
dest_port	Multicast station destination port (1–65535). However, the multicast-enabled network may impose certain restrictions on your choice of port. Normally, port numbers below 1024 should be avoided, but the Content Engine does not enforce any restrictions.
media_source	Multicast station media source. This media source is designated as the primary multicast media source. The source can be any valid WMT URL. If you can play the URL on your Windows Media player, then you can make this URL the source of your multicast.
log	(Optional) Enables logging of multicast URLs.
local	Configures logging of statistics regarding the multicast URLs to local disk using the .nsc file.
webserver	Configures the logging of multicast URLs to a web server. Enter the URL to identify the location of the web server.
webserver_url	Complete URL for the log file on the web server.
play-forever	(Optional) Configures the stream to loop and restart. The default is to play the stream once and stop. The multicast stream stops playing once the end of the source media file has been reached.
unicast-url	(Optional) Configures the unicast rollover URL if station is unable to receive multicast.
url	Complete URL for the unicast rollover log file.
failover	(Optional) Sets the failover configuration for a WMT multicast station.
alternate-source	Specifies an alternate media source for multicast.
alt_src_url	Fully qualified alternate source URL. Up to eight alternative source URLs can be specified. Each alternative URL must be entered on a separate line.

retry-count	Specifies the number of times that the multicast station tries to establish a connection with the multicast source (applies to both primary and alternate sources).
number	Maximum number of retries (0–2000).
retry-interval	Sets the time in minutes between request retries for all multicast media sources (applies to both primary and alternate sources).
number	Time between retries in minutes (1–60).
schedule-start	(Optional) Configures an automatic start schedule.
minute	Start time minute (0–59).
hour	Start time hour (0–23).
day	Start time day (1–31).
month	Start time month (1–12).
now	Starts a WMT multicast station immediately. For more information, see the "Automatically Restarting Multicast Stations" section on page 2-908.
time-to-live	Configures the maximum number of hops needed to reach the Content Engine that functions as the multicast server.
ttl	Number of hops (0–255). The default is 5 hops.
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.
hostname	Hostname of an outgoing proxy.
ip-address	IP address of an outgoing proxy.
port	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

accept-license-agreement: not accepted

wmt: disabled

advanced client maximum-packet-size: 1500 bytes

advanced client idle-timeout: 60 seconds

advanced server log-forwarding: enabled

wmt fast-cache: enabled

wmt fast-start: enabled

max-object-size: 1 GB

multicast station-configuration: play stream once and then stop

multicast time-to-live: 5 hops

wmt incoming *number*: 1755

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.

Note For more information on the Microsoft Media Services, see Chapter 9 of the *Cisco ACNS Software Configuration Guide for Locally Managed Deployments*.

Enabling WMT on the Content Engine

Before enabling licenses for streaming media services on a Content Engine, make sure that your Content Engine 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.

Note You must configure disk space to include mediafs storage with the **disk config** command before you can cache streaming media using WMT.

Enabling Transparent WMT Service Using WCCP-Enabled Routers

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

- The Content Engine is running the ACNS 4.1 and later releases.
- The WMT proxy software is installed with mediafs partitions mounted on the Content Engine.
- You have a WMT license key.
- You have the IP address of the router or routers.

Note For information on enabling a transparent WMT service using WCCP-enabled routers, see Chapter 9 of the *Cisco ACNS Software Configuration Guide for Locally Managed Deployments*.

Enabling Conventional WMT Proxy Service

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

- The Content Engine is running the ACNS 4.1 and later releases.
- The WMT proxy software is installed with mediafs partitions mounted on the Content Engine.
- You have a Microsoft WMT license key.
- You have the IP address of the Content Engine.

Note For information on enabling a conventional WMT proxy service, see Chapter 9 of the *Cisco ACNS Software Configuration Guide for Locally Managed Deployments*.

Removing the WMT License Key

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

Note You must disable the WMT feature using the **no wmt enable** command before uninstalling the WMT license key.

Note Setting the maximum packet size restarts the RTSP gateway. As a result, requests from RealProxy, RealSubscriber, and Cisco Streaming Engine clients will not be served for a brief period of time.

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 either on-demand or broadcast content.

Note By configuring the Content Engine with a broadcast alias, when a client makes a request to the Content Engine as if it were the Windows Media Server, the Content Engine splits the stream to the client. If it is, then the Content Engine joins the stream and splits it to the new client. If the request is the first client request for this stream, the Content Engine sends the request out to the server and then serves it to the new client.

Note For information on enabling the Fast Cache, see Chapter 9 of the *Cisco ACNS Software Configuration Guide for Locally Managed Deployments*.

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 Content Engine 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 and broadcast content are 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.

In the ACNS 5.2.x software, Fast Start is only available for MMS-over-HTTP requests with Windows Media Services Version 9.0. In the ACNS 5.3 software and later releases, support for Fast Start is also available for RTSP requests from a Windows Media 9 player.

When Fast Start is enabled on the Content Engine, 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.

Note For information on enabling Fast Start, see Chapter 9 of the *Cisco ACNS Software Configuration Guide for Locally Managed Deployments*.

Adding or Removing WMT HTTP Allowed Filename Extensions

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

The default list in the Content Engine contains the following filename extensions:

- asf
- none
- nsc
- wma
- wmv

Note The default list of filename extensions includes “none” to enable Content Engines to serve media files without file extensions, such as broadcast aliases or URLs of live encoders. The filename extension nsc is included in the list to enable Content Engines 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.

Automatically Restarting Multicast Stations

If a WMT multicast station has been started before reloading a Content Engine, it will not run after the Content Engine is reloaded because the multicast station schedule configured using the **wmt** EXEC command is not preserved across reboots. In releases of the ACNS software prior to Release 5.3, you had to restart the multicast station by using the **wmt** EXEC command. However, in the ACNS 5.3 software and later releases, you can use the **wmt multicast station-configuration** *name* **schedule-start now** to automatically restart the station after reloading the Content Engine.

Configuring WMT Multicasting

A Content Engine 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 Content Engine 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 Content Engine 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 Content Engine, 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.

Using WMT Multicast

Use the **wmt multicast** {**schedule-start** *name minute hour day month* | **station-configuration** *name dest_addr dest_port media_source* [**log** {*local* | *webserver*}] [**play-forever**] [**unicast-url** *url*] | **time-to-live** *ttl*} global configuration command to enable WMT multicasting for the unicast-in multicast-out and multicast-in multicast-out scenarios on the Content Engine. The **schedule-start** *name minute hour day month* option creates a scheduling option to allow the Content Engine to start a multicast at a specified time. This option only works if you have configured a multicast station first using the **wmt multicast station-configuration** command.

Note A multicast station is a defined location (a multicast IP address and multicast port) from which a player can receive streams. This multicast IP address is not related to the IP address of the Content Engine.

You must enable WMT on the Content Engine before you can use the **wmt multicast** and **wmt broadcast** commands. See the "Enabling WMT on the Content Engine" section on page 2-905.

The **wmt multicast station-configuration** *name dest_addr dest_port media_source* option specifies a multicast station name, an IP multicast address, a port number, and a media source for the multicast station created. Each station needs a multicast IP address. You must enter a valid Class D IP address multicast address in the range 224.0.0.0 to 239.255.255.255, except for the reserved IP ranges based on RFC 1700 and related documents as follows:

- 224.0.0.0–224.0.6.255
- 224.0.13.0–224.0.13.255
- 224.1.0.0–224.2.255.255
- 232.0.0.0–232.255.255.255

Note You must choose a multicast IP address that does not conflict internally within the same multicast-enabled network configuration. This multicast IP address is not related to the IP address of the Content Engine.

The allowed multicast port range defined by the *dest_port* option is 1 through 65535. However, the multicast-enabled network may impose certain restrictions on your choice of port. Normally, port numbers below 1024 should be avoided, but the Content Engine does not enforce any restrictions.

The *media_source* option determines the source of the multicast. The source can be any valid WMT URL. If you can play the URL on your Windows Media player, then you can make this URL the source of your multicast.

In the ACNS software releases prior to Release 5.2, the maximum TTL value for the multicasting of WMT packets was set to five hops and was not user configurable. For clients who were more than five router hops away from a Content Engine functioning as a multicast server, requested content could not be delivered, because the maximum number of hops was exceeded between the server and the clients.

In the ACNS software, Release 5.2 and later releases, a global configuration command is available to configure the TTL for WMT multicast. Use the **wmt multicast time-to-live** *ttl* command to set the value for the TTL between 0 and 255 hops. The default is five hops.

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.

Using WMT Broadcast

Use the **wmt broadcast** {**alias-name** *name* **source** *url*} command to configure the multicast-in unicast-out broadcast scenario on the Content Engine. With this command, you create a broadcasting alias to deliver an incoming stream live to requesting clients using multicast as the source of the streaming media.

Note You can also configure WMT multicasting parameters with the Content Engine GUI. Click the **WMT Config** button to access these parameters.

Configuring Unicast-In Multicast-Out

The Content Engine supports several different sources for a unicast-in multicast-out stream, otherwise known as stream splitting. A unicast input can be from a video-on-demand (VoD) publishing point, a live unicast publishing point, an encoder, or a streaming media source from a local disk. The ASF header obtained from the unicast input and the parameters used to configure the multicast station are used by the Content Engine to automatically create the multicast description .nsc file. The clients use this file to subscribe to the multicast.

Note If a live stream is interrupted on the server side, you must stop the multicast station and then restart the same station to resume live multicasting. Use the **wmt multicast-station stop** *stationname* command in EXEC mode to stop this station. Use the **wmt multicast-station start** *stationname* command in EXEC mode to restart the same station.

Note For information on enabling a WMT multicasting for unicast-in multicast-out, see Chapter 9 of the *Cisco ACNS Software Configuration Guide for Locally Managed Deployments* publication.

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.

Note For information on enabling a WMT multicasting for multicast-in multicast-out, see Chapter 9 of the *Cisco ACNS Software Configuration Guide for Locally Managed Deployments* publication.

Configuring Multicast-In Unicast-Out

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

Note For information on enabling a WMT multicasting for multicast-in unicast-out, see Chapter 9 of the *Cisco ACNS Software Configuration Guide for Locally Managed Deployments* publication.

Configuring Unicast-In Unicast-Out

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

The two ways to configure unicast-in unicast-out are as follows:

- Live splitting without any configuration. The Content Engine acts as a proxy. When clients request the same unicast URL, the Content Engine proxy automatically splits the stream from the source to the clients.
- Configuring the Content Engine with a broadcast alias. A client makes the request to the Content Engine as if it were the Windows Media Server, and the Content Engine checks to see whether the incoming stream is present. If it is present, then the Content Engine joins the stream and splits it to the new client. If the request is the first client request for this stream, the Content Engine sends the request out to the server and then serves it to the new client.

Note For information on configuring a Content Engine to use unicast-in unicast-out to relay live WMT streams to clients, see Chapter 9 of the *Cisco ACNS Software Configuration Guide for Locally Managed Deployments*.

Configuring Outgoing WMT Proxy Servers

The ACNS 5.x software allows you to specify the external WMT server that the Content Engine should use as its upstream WMT server . The Content Engine contacts the specified outgoing proxy server upon a cache miss (if the Content Engine 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 ACNS software, Release 5.2 and later releases use the enhanced logging support provided by Windows Media Services 9 Series in addition to the Windows Media Services Version 4.1 logging format that was supported in the ACNS software, Release 5.1.

The ACNS software, Release 5.2 and later releases support the following transaction log formats 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 In the ACNS software, Release 5.1 and earlier releases, only the standard Windows Media Services 4.1 and the extended Windows Media Services 4.1 logging formats were supported.

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 extended versions of the logging formats are extensions to the standard logging format and contain additional fields that are Content Engine specific (for example, the CE-action field that specifies whether it was a cache hit or miss, and the CE-bytes field that specifies the number of bytes that were sent out from the Content Engine.

The Content Engine'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 2-182.

Table 2-182 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) Content Engine (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 Content Engine (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 Content Engine (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 Content Engine (Windows Media Server) can be parsed and saved to the normal WMT transaction logs that are stored on the Content Engine.

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

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

- CE-action—cache hit, cache miss, VoD, or live create
- CE-bytes—number of bytes served by the Content Engine in the case of a cache hit
- username (username of the person who made the WMT request when Microsoft NT LAN Manager (NTLM) authentication, 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 Content Engine is configured to use the extended format of WMT transaction logging and the extended WMT logging feature is enabled, then the Content Engine logs usernames for any authenticated WMT requests. Usernames are logged not only for NTLM authentication but also 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 Content Engine logs usernames associated with authenticated WMT requests only when the extended logging formats (extended wms-41 and extended wms-90) are used.

Note For information on configuring a WMT transaction logs to be generated in extended Windows Media Services 9.0 format, see Chapter 9 of the *Cisco ACNS Software Configuration Guide for Locally Managed Deployments*.

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 Content Engine:

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

Use the **log** option of the **wmt multicast station-configuration** global configuration command to provide a log of 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.

Forwarding WMT Logs to Upstream Servers

In the ACNS 5.3 software and later releases, you can decide whether you want this Content Engine to forward its WMT logs to the upstream server (a Windows Media server or another Content Engine). By default, Content Engines that are running the ACNS 5.3 software and later releases will forward their WMT logs to the upstream server. This feature applies to all of the supported protocols. To disable this feature and configure the Content Engine to not forward its WMT logs to the upstream server, enter the **no wmt advanced server log-forwarding enable** global configuration command. To reenable 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:

```
ContentEngine# 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%
                     NTLM:          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-BCF5EEEDE3FF} 10.1.0.3055 en-US - - wmplayer.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 CE-action CE-bytes Username
```

Table 2-183 describes the fields shown in this example.

Table 2-183 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.

Table 2-183 wmt multicast logging Field Descriptions (continued)

Field	Description
time	Time (according to Greenwich mean time) when an entry is generated in the log file.
c-dns	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-startime	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 occurs.
c-rate	Mode of Windows Media Player when the last command event was sent: • 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.
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).

Table 2-183 wmt multicast logging Field Descriptions (continued)

Field	Description
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.
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.

Table 2-183 wmt multicast logging Field Descriptions (continued)

Field	Description
CE-action	Action performed by the Content Engine.
CE-bytes	Number of bytes received by the Content Engine.
Username	Username required to access the streaming media retrieved by the WMT player.

The following example shows the display of the **show statistics wmt multicast** command:

```
ContentEngine# show statistics wmt multicast
Multicast statistics
====================

  Total Multicast Outgoing Bytes: 0
Total Multicast Logging Requests: 0

Aggregate Multicast Out Bandwidth (Kbps)
------------------------------------------------
        Current:         0
            Max:         0

Number of Concurrent Active Multicast Sessions
------------------------------------------------
        Current:         0
            Max:         0

List of All Multicast Stations Configured using the CLI
--------------------------------------------------------
Total Number of Configured Multicast Stations: 0
```

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

```
ContentEngine# wmt http allow extension mp3
ContentEngine#
```

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:

```
ContentEngine# show wmt http allow extension
WMT http extensions allowed :
asf mp3 none nsc wma wmv
ContentEngine(config)#
```

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

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

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

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

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

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

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

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

Related Commands

clear cache wmt
clear statistics wmt
clear wmt
show running-config
show tech-support
show statistics wmt
show wccp services
show wccp flows
show wmt
show wmt license-agreement
wmt (EXEC mode)

write

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

write [**erase** | **memory** | **mib-data** | **terminal**]

Syntax Description

erase	(Optional) Erases the startup configuration from NVRAM.
memory	(Optional) Writes the configuration to NVRAM. This setting is the default.
mib-data	(Optional) Saves the MIB persistent configuration data to the disk.
terminal	(Optional) Writes the configuration to a terminal session.

Defaults

The configuration is written to NVRAM by default.

Command Modes

EXEC

Usage Guidelines

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

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

In the ACNS software, Release 5.2 and later releases, 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 ACNS 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 ACNS 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 Content Engine is erased.

Examples

The following command saves the running configuration to NVRAM:

```
ContentEngine# write memory
ContentEngine#
```

Related Commands

copy running-config startup-config
show running-config