



Configuring Transparent Redirection for Standalone Content Engines

This chapter discusses the following methods for transparently redirecting content requests to standalone Content Engines:

- Web Cache Communication Protocol (WCCP)-enabled routers that intercept content requests and redirect them to standalone Content Engines
- A Layer 4 switch that intercepts content requests and redirects them to standalone Content Engines

This chapter includes the following sections:

- [Overview of WCCP Transparent Redirection, page 6-2](#)
- [Configuring Standalone Content Engines for WCCP Transparent Redirection, page 6-9](#)
- [Disabling and Reenabling WCCP Flow Redirection on Standalone Content Engines, page 6-17](#)
- [Shutting Down WCCP on Standalone Content Engines, page 6-17](#)
- [Configuring a Router for WCCP Transparent Redirection, page 6-18](#)
- [Configuring WCCP Services on a Router, page 6-27](#)
- [Clearing WCCP Statistics on a Router, page 6-34](#)
- [Configuring WCCP Layer 2 Support, page 6-34](#)
- [Examples of Configuring WCCP Services for Standalone Content Engines, page 6-38](#)
- [Configuring Layer 4 Switching as a Redirection Method, page 6-50](#)

The Cisco ACNS transparent caching solution uses a WCCP-enabled router and various advanced techniques to ensure that the Content Engine remains transparent, even if web browsers are nonoperational or web servers are not HTTP-compliant. One of these techniques is the bypass feature. For information about how to configure the bypass feature, see the “[Configuring Bypass Settings on Standalone Content Engines](#)” section on page 15-3.



Note

For complete syntax and usage information for the Content Engine CLI commands used in this chapter, refer to the *Cisco ACNS Software Command Reference, Release 5.5* publication.

For further information on WCCP, refer to the *Cisco IOS Configuration Fundamentals Configuration Guide* and the *Cisco IOS Configuration Fundamentals Command Reference*.

Overview of WCCP Transparent Redirection

Cisco developed WCCP within Cisco IOS software to enable routers or switches to intercept packets based on the IP, UDP, and TCP header information and then redirect those packets transparently to Content Engines running the ACNS 5.x software.

There are two versions of WCCP, Version 1 and Version 2. WCCP Version 1 only supports one WCCP service (the standard web-cache service [service 0]) and a single router). The main features of WCCP Version 1 include:

- Support for only one router (home router)
- Support for traffic redirection on port 80 only
- Support for up to 32 Content Engines per WCCP service
- No bypass support (for example, static bypass, error bypass, and authentication bypass are not supported)
- No generic routing encapsulation (GRE) on return

WCCP Version 2 enables more TCP ports to have traffic redirected to the Content Engine. Previously, web-cached information could be redirected only if it was destined for TCP port 80. Many applications require packets intended for other ports to be redirected, for example, proxy web cache handling, FTP proxy caching, web caching for ports other than 80, RealAudio, and video.

We recommend that you use WCCP Version 2 because it supports a wider set of features and services (see [Table 6-3](#)) as well as multiple routers.

**Note**

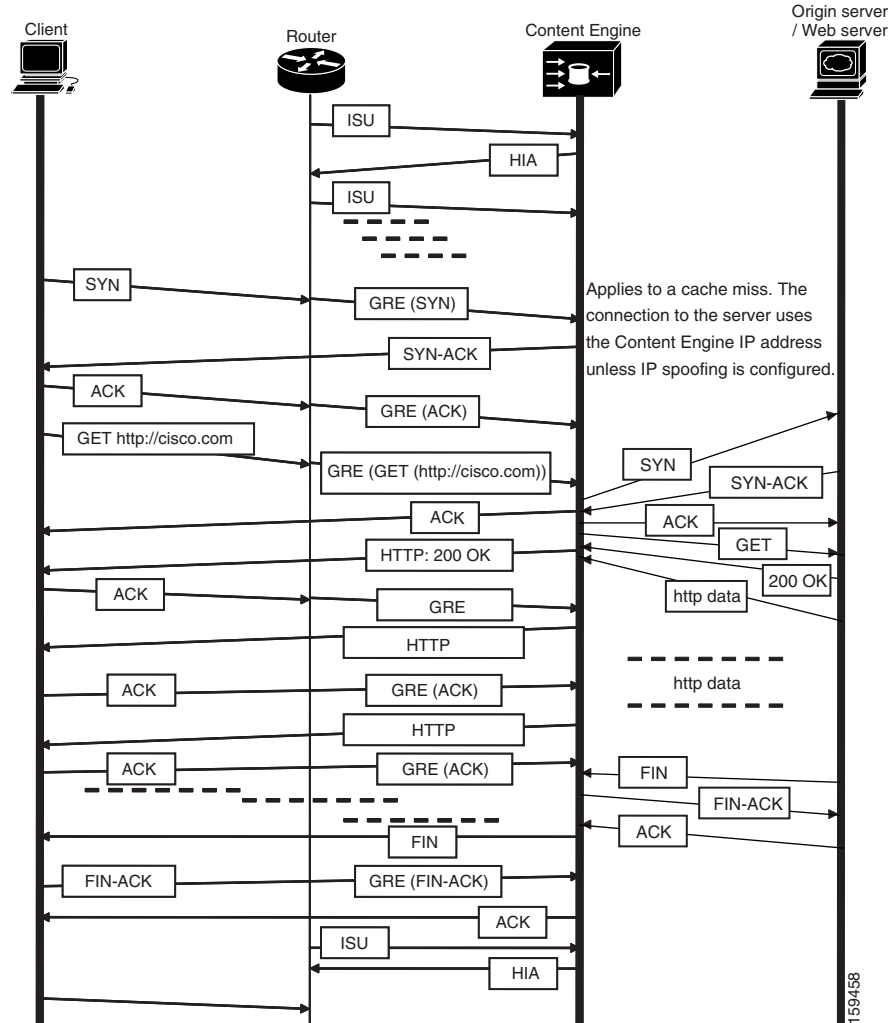
WCCP works with IP networks only.

Redirection of Packets with WCCP

When transparent redirection with a WCCP-enabled router is used to redirect requests to a Content Engine, the web clients send their content requests to the source and are not aware that their requests are being redirected to the Content Engine by a WCCP-enabled router. Because this interception and redirection process is completely invisible, or transparent, to the client who is requesting the content, no desktop changes are required (clients do not have to configure their browsers or media players to point to a specific proxy server). The Content Engine operation is transparent to the network; the WCCP-enabled router operates entirely in its normal role for nonredirected traffic.

[Figure 6-1](#) shows the packet flow between a Content Engine and a router.

Figure 6-1 Packet Flow Diagram



In [Figure 6-1](#), if the Content Engine does not have the data, a cache miss occurs, and the Content Engine sends the request to the origin server. As the Content Engine receives the data from the origin server, it saves the data and forwards the data to the client.

To use WCCP transparent redirection, you must first define a WCCP service on the WCCP-enabled router. The parameters for a given service are its name, service identifier (service number), and the router interface that is to be used to support this WCCP service.

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 in question remain global for the whole WCCP system.

For information about configuring healing mode for Content Engines that are part of a Content Engine cluster, see the [“Configuring Healing Mode for Content Engine Clusters”](#) section on page 7-70. For information about configuring advanced caching features such as WCCP slow start, see [Chapter 15, “Configuring Advanced Transparent Caching Features on Standalone Content Engines.”](#)

Redirection of Packets with Content Engine Clusters and WCCP-Enabled Routers

If there is a cluster (group) of Content Engines, the one seen by all the WCCP Version 2-enabled routers and the one that has the lowest IP address becomes the lead Content Engine. The role of this lead Content Engine is to determine how traffic should be allocated across the Content Engines in the cluster. The assignment information is passed to the entire service group from the designated lead Content Engine so that the WCCP-enabled routers of the group can redirect the packets properly and the Content Engines in the group can better manage their load.

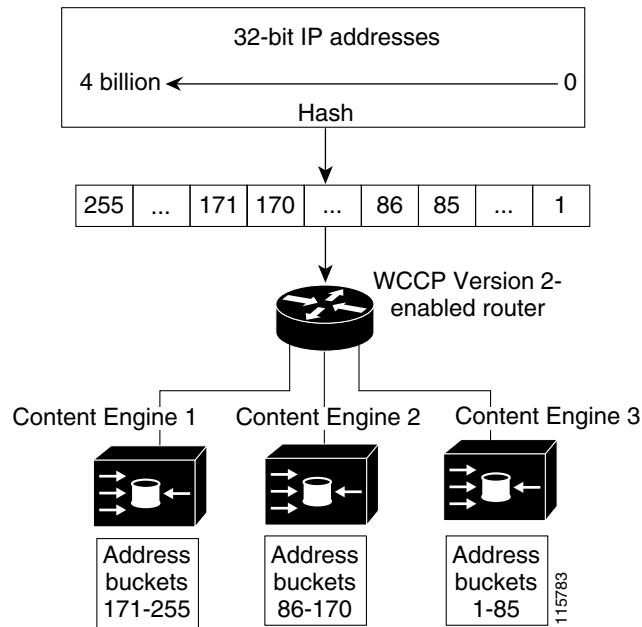
The following describes how a Content Engine in a Content Engine cluster is designated the lead:

1. Each Content Engine is configured with a list of WCCP-enabled routers.
With WCCP Version 1, only a single WCCP-enabled router services a cluster, becoming the default home router for the cluster.
With WCCP Version 2, multiple WCCP-enabled routers (each router list can contain up to eight routers) can service a cluster. This allows any of the available routers in a service group to redirect packets to each of the Content Engines in the cluster.
2. Each Content Engine announces its presence to each router on the router list. The routers reply with their view of Content Engines in the service group.
3. After the view is consistent across all of the Content Engines in the cluster, one Content Engine is designated the lead Content Engine and sets the policy that the WCCP-enabled routers need to deploy in redirecting packets.

About Dynamic Load Distribution with WCCP Version 2

When an IP packet is received by a WCCP Version 2-enabled router, it is examined to determine if it is a request that should be directed to a Content Engine. This is done by matching the request to the defined service criteria. These packets are passed to the router's processing routine to determine which Content Engine, if any, should receive the redirected packets.

The determination of which Content Engine should receive the intercepted packets is made by performing a hash function on the destination IP address to obtain an address bucket to which the packet is assigned. These address buckets are then mapped to a particular Content Engine depending on how many Content Engines are present and how busy they are. See [Figure 6-2](#).

Figure 6-2 Load Balancing Through Hashing of IP Addresses**Note**

Packets that the Content Engines do not service are tunneled back to the same router from which they were received. When a router receives a formerly redirected packet, it knows not to redirect it again.

WCCP Version 2 supports dynamic load distribution that allows the routers to adjust the loads being forwarded to the individual Content Engines in the cluster. WCCP uses two techniques to perform this task:

- Load balancing allows the set of hash address buckets assigned to a Content Engine to be adjusted so that the load can be shifted from an overwhelmed Content Engine to other Content Engines that have available capacity.
- Load shedding enables the WCCP-enabled router to selectively redirect the load to avoid exceeding the capacity of the Content Engines.

About Packet-Forwarding Methods

A WCCP-enabled router redirects intercepted requests to a standalone Content Engine using two packet-forwarding methods:

- Generic routing encapsulation (GRE)—Allows packets to reach the Content Engine even if there are any number of routers in the path to the Content Engine.
- Layer 2 redirection—Allows packets to be switched at Layer 2 (MAC layer) and reach the Content Engine.

Table 6-1 describes these two methods.

Table 6-1 Packet-Forwarding Methods

Packet-Forwarding Method	Load-Balancing Method: Hashing	Load-Balancing Method: Masking
GRE (Layer 3)	Packet redirection is completely handled by the router software.	Packet redirection is handled by the router software. Mask assignment is not recommended when GRE is being used as the packet-forwarding method.
Layer 2 redirection	First redirected packet is handled by the router software; all subsequent redirected packets are handled by the router hardware.	All packets are handled by the router hardware (currently supported on only the Catalyst 6000 series switches because special hardware is required).

**Note**

In both packet-forwarding methods, the hash parameters specify how redirected traffic should be load balanced among the Content Engines in the various WCCP service groups.

The term *assignment method* denotes the method used by WCCP to perform load distribution across Content Engines. There are two possible load-balancing assignment methods: hashing and masking. If the mask load-balancing method is not specified, then the hash load-balancing method (see [Figure 6-2](#)), which is the default method, is used.

The redirection mode is controlled by the Content Engine. The first Content Engine that joins the WCCP service group decides the forwarding method (GRE or Layer 2 redirection) and the assignment method (hashing or masking). The term *mask assignment* is used to refer to WCCP Layer 2 Policy Feature Card 2 (PFC2) input redirection.

Regarding fallback capabilities for the forwarding method, if the router is using the GRE redirection because other Content Engines are using GRE for the web-cache service and the new Content Engine has been configured to use Layer 2 redirection (the **wccp web-cache router-list-num 1 12-redirect** global configuration command has been specified), the Content Engine attempts to negotiate Layer 2 redirection. The Content Engine will advertise its Layer 2 redirection capability but when the router replies with GRE, the Content Engine will fall back to GRE. This is expected behavior because it is currently not possible to configure the Content Engine to only use Layer 2 redirection.

Regarding fallback capabilities for the assignment method, the Content Engine falls back to the assignment method supported in hardware unless the **assign-method-strict** option is used (for example, if the **wccp https-cache assign-method-strict** command is used to specify the **assign-method-strict** option for the https-cache service) rather than remain out of the Content Engine cluster indefinitely. If masking is selected with WCCP output redirection, then the Content Engine falls back to the original hardware acceleration that is used with the Multilayer Switch Feature Card (MSFC) and the Policy Feature Card (PFC).

For example, WCCP Version 2 filters packets to determine which redirected packets have been returned from the Content Engine and which ones have not. It does not redirect the ones that have been returned, because the Content Engine has determined that the packets should not be processed. WCCP Version 2 returns packets that the Content Engine does not service to the same router from which they were transmitted.

The following list includes some typical reasons why a Content Engine would reject packets and initiate packet return:

- The Content Engine is overloaded and has no resources to service the packets.
- The Content Engine activates the automatic bypass feature as a result of a server error or authentication failure. In this situation, the client can reach the server directly. The Content Engine, therefore, is not the reason for the failure.
- The Content Engine is filtering out certain conditions that make processing packets counterproductive, for example, when IP authentication has been turned on.
- You have configured a static bypass list on the Content Engine.

**Note**

The packets are redirected to the source of the connection between the WCCP-enabled router and the Content Engine. Depending on the Cisco IOS software version used, this could be either the address of the outgoing interface or the router IP address. In the latter case, it is important that the Content Engine has the IP address of the WCCP-enabled router stored in the router list. For more information on router lists, see the [“Defining Router Lists on Standalone Content Engines”](#) section on page 6-12.

Cisco Express Forwarding (CEF) has been integrated into WCCP Version 2 to achieve optimal performance during packet redirection. WCCP Version 2 also allows you to configure multiple routers (router lists) to support a particular WCCP service (for example, RTSP redirection), as described in the [“Configuring Standalone Content Engines for WCCP Transparent Redirection”](#) section on page 6-9. For a list of supported WCCP Version 2 features and services, see [Table 6-3](#).

For more information about these packet-forwarding methods, see:

- [Using Layer 3 GRE as a Packet-Forwarding Method, page 6-7](#)
- [Using Layer 2 Redirection as a Packet-Forwarding Method, page 6-8](#)

Using Layer 3 GRE as a Packet-Forwarding Method

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

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

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

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

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

Using Layer 2 Redirection as a Packet-Forwarding Method

Layer 2 redirection is a term for a situation in which WCCP on a router or switch can take advantage of switching hardware that either partially or fully implements the traffic interception and redirection functions of WCCP in router hardware at Layer 2. This type of redirection is currently supported only with the Cisco Catalyst 6000 and 6500 series switches. With Layer 2 redirection, the first redirected traffic packet is handled by the router software. The rest of the traffic is handled by the router hardware. With Layer 2 redirection, the Content Engine instructs the router or switch to apply a bitmask to certain packet fields, which in turn provides a mask result or index mapped to the Content Engines in the cluster in the form of a mask index address table. The redirection process is accelerated in the switching hardware, which makes Layer 2 redirection more efficient than Layer 3 GRE.

**Note**

WCCP is only licensed on the Content Engine and not on the redirecting router. WCCP does not interfere with normal router or switch operations.

For information about configuring Layer 2 redirection, see the [“Configuring WCCP Layer 2 Support” section on page 6-34](#).

Configuring Standalone Content Engines for WCCP Transparent Redirection

The type of WCCP services supported by a standalone Content Engine and a WCCP-enabled router varies based on whether WCCP Version 1 or Version 2 is used, as indicated in [Table B-3](#). All of the services except for the standard web-cache service (service 0) requires that WCCP Version 2 (as opposed to WCCP Version 1) is running on the router and the standalone Content Engine for a particular WCCP service to be supported. [Table 6-2](#) lists the supported WCCP transparent redirection services for standalone Content Engines

Table 6-2 Supported WCCP Transparent Redirection Services

WCCP Transparent Redirection Service	WCCP Service Name	Description	More Information
Service 80	rtsp	RTSP redirection service used to redirect RTSP requests from RealMedia clients to the Content Engine. For transparent redirection of RealMedia requests, you only need to configure service 80. For WMT RTSP transparent redirection support (transparent redirection of RTSP requests from Windows Media 9 players), you must configure service 80 as well as service 83.	When the clients are RealMedia players, the backend RTSP server is the RealProxy server. When the clients are Windows Media 9 players, the backend RTSP server is the Windows Media 9 server that is running on the Content Engine.
Service 81	mmstu	MMSTU redirection service used to redirect MMS-over-HTTP requests from WMT clients to the Content Engine.	Clients are WMT clients (for example, Windows Media players Version 6, 7, or 9).
Service 82	mmsu	MMSU redirection service used to redirect MMS-over-HTTP requests from WMT clients to the Content Engine. For MMS redirection of WMT requests, you must configure service 81 and 82 on the router.	Clients are WMT clients (for example, Windows Media players Version 6, 7, or 9).
Service 83	wmt-rtspu	WMT RTSP redirection service used to redirect RTSP requests from WMT clients (Windows Media 9 Players) to the Content Engine (that is functioning as a Windows Media 9 server).	Clients and server are WMS 9 (for example, Windows Media 9 players and the Windows Media Series 9 server that is running on the Content Engine and functioning as the backend RTSP server for WMT RTPS requests).

To view a list of WCCP options and services that you can configure on a standalone Content Engine, enter the **wccp EXEC** command followed by a question mark (“?”). The following sample output is from a Content Engine that has WCCP Version 2 enabled. If you are using WCCP Version 1, only one router is supported (home router) and a single WCCP service (the standard web-cache service).

```
ContentEngine(config)# wccp ?
access-list      Configure an IP access-list for inbound WCCP encapsulated traffic
custom-web-cache Custom web caching service
dns              Caching Domain Name Service
flow-redirect    Redirect moved flows
ftp-native       Transparent FTP proxy caching service
home-router      WCCP Version 1 Home Router Ip address
https-cache      HTTPS caching service
port-list        Port list for use in WCCP service
reverse-proxy    Reverse Proxy web caching service
router-list      Router List for use in WCCP services
rtsp             RTSP protocol transparent interception
service-number   WCCPv2 service number
shutdown         Wccp Shutdown parameters
slow-start       accept load in slow-start mode
spooof-client-ip Use client IP while connecting to the origin server
version          WCCP Version Number
web-cache        Standard web caching service
wmt             Windows media caching service
wmt-rtspu        Windows media RTSPU (port 5005) transparent interception
```

For information about configuring the standard web-cache service (service 0) using WCCP Version 1, see the “[Example 1—Configuring the Web-Cache Service with WCCP Version 1](#)” section on page 6-39.

See [Table 6-3](#) for a list of these WCCP Version 2 options and services. An asterisk (“*”) in [Table 6-3](#) indicates whether a particular capability is a WCCP option or a WCCP service.

Table 6-3 WCCP Version 2 Options and Services

WCCP Option or Service Name	WCCP Option	WCCP Service	More Information
access-list	*		Configuring WCCP Access Lists for Standalone Content Engines
custom-web-cache		*	Configuring the Custom-Web-Cache Service (Service 98) on a Router
dns		*	Configuring the DNS Caching Service (Service 53) on a Router
flow-redirect	*		Disabling and Reenabling WCCP Flow Redirection on Standalone Content Engines
ftp-native		*	Configuring the FTP-Native Caching Service (Service 60) on a Router
https-cache		*	Configuring the HTTPS-Cache Service (Service 70) on a Router
port-list	*		Defining Port Lists on Standalone Content Engines
reverse-proxy		*	Configuring the Reverse-Proxy Service (Service 99) on a Router
rtsp		*	Configuring the RTSP Service (Service 80) on a Router
service-number	*		Configuring User-Defined WCCP Services (Services 90–97) on a Router
shutdown	*		Defining Port Lists on Standalone Content Engines
slow-start	*		Configuring WCCP Slow Start
spooof-client-ip	*		Configuring WCCP IP Spoofing
version	*		Enabling WCCP on Standalone Content Engines

Table 6-3 WCCP Version 2 Options and Services (continued)

WCCP Option or Service Name	WCCP Option	WCCP Service	More Information
web-cache		*	Example 2—Configuring the Web-Cache Service with WCCP Version 2
wmt-rtspu		*	Configuring the WMT-RTSPU Service (Service 83) on a Router

**Note**

In the ACNS software releases earlier than Release 5.2, a maximum of eight active WCCP services were supported by a WCCP Version 2-enabled router and a Content Engine. The ACNS 5.2.1 software and later releases up to 25 active WCCP Version 2 services are supported. In the ACNS 5.3.1 software release, the wmt-rtspu (Service 83) was added. In the ACNS 5.3.1 software release, there are currently 18 services that can be configured.

In the ACNS 5.2 software, another interception mode (the accept-all mode) was added for the https-cache service. This mode was added to support the filtering of HTTPS traffic. This mode works the same way as traditional WCCP services (for example, the standard web-cache service [service 0] that intercepts all web traffic by default). If the **wccp https-cache accept-all** global configuration command is used, the HTTPS cache (the Content Engine that has the https-cache service configured and enabled) will work in accept all mode (all HTTPS traffic is intercepted by the Content Engine); otherwise, the HTTPS cache works in accept only mode, as in the ACNS 5.1.x software.

To use WCCP for transparent redirection, the Content Engine must be properly configured. Keep these important points in mind:

- The Content Engine must be configured to accept redirected packets from one or more WCCP-enabled routers. With WCCP Version 1, only a single router (home router) is supported. With WCCP Version 2, multiple routers (router lists) are supported.
- Versions of software on the Content Engines must be compatible with those installed on the WCCP-enabled router.
- The Content Engines must not have their packets encrypted or compressed and should be part of the inside Network Address Translation (NAT) firewall if one is present.
- Placing a Content Engine beyond a web cache redirect-enabled interface and along the route to the server will not cause the IP route cache to be populated with an entry.
- After enabling WCCP on the router, you must configure the router and the Content Engine for transparent caching services.

For information about how to configure standalone Content Engines for WCCP transparent redirection, see:

- [Enabling WCCP on Standalone Content Engines, page 6-12](#)
- [Defining Port Lists on Standalone Content Engines, page 6-12](#)
- [Defining Router Lists on Standalone Content Engines, page 6-12](#)
- [Configuring WCCP Services on Standalone Content Engines, page 6-14](#)
- [Displaying WCCP Configuration Information for Standalone Content Engines, page 6-16](#)

For sample scenarios of how to configure WCCP services for standalone Content Engines, see the “[Examples of Configuring WCCP Services for Standalone Content Engines](#)” section on page 6-38. For information about how to configure WCCP transparent interception on a router, see the “[Configuring a Router for WCCP Transparent Redirection](#)” section on page 6-18.

Enabling WCCP on Standalone Content Engines

To enable WCCP on a standalone Content Engine, enter the **wccp version** global configuration command. Specify the WCCP version that you want the standalone Content Engine to run. Ensure that the routers used in the WCCP environment are running a software version that supports the WCCP version configured on the standalone Content Engine.

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

```
Content Engine (config)# wccp version 2
```

The following example shows how to enable WCCP Version 1 on a standalone Content Engine:

```
Content Engine (config)# wccp version 1
```

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). See [Table 6-3](#) for a list of the features and services supported by WCCP Version 2.

It is not necessary to disable WCCP Version 1 before enabling WCCP Version 2, and vice versa. However, to properly shut down WCCP on a Content Engine, you must disable the currently running version as described in the [“Disabling and Reenabling WCCP Flow Redirection on Standalone Content Engines”](#) section on page 6-17.

Defining Port Lists on Standalone Content Engines

With WCCP Version 1, web-cached information can only be redirected to a Content Engine if it was destined for TCP port 80. Many applications require packets intended for other ports to be redirected, for example, proxy web cache handling, web caching for ports other than port 80, RealAudio, and video. If a router is configured for WCCP Version 2 instead of WCCP Version 1, then additional TCP ports other than port 80 can be configured on the WCCP-enabled router to redirect traffic to a Content Engine.

You can configure up to eight port lists (port lists number 1 through 8). These port lists specify the port numbers on which the Content Engine will listen for incoming WCCP redirected traffic. 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 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 consists of 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
```

Defining Router Lists on Standalone Content Engines

As part of configuring a WCCP Version 2 service on a Content Engine, you must create a list of WCCP Version 2-enabled routers that will support a specific WCCP Version 2 service (for example, the rtsp service) for the Content Engine.

To create a router list on a standalone Content Engine, use the **wccp router-list** global configuration command. Enter the IP address of every WCCP Version 2-enabled router that will support a particular WCCP service for the Content Engine. If different routers will be used for different WCCP services, you must create more than one router list. Each router list can contain up to eight routers.

In the following example, router list number 1 is created, and it contains a single router (the WCCP Version 2-enabled router with IP address 10.10.10.1):

```
ContentEngine(config)# wccp router-list 1 10.10.10.1
```

The following example shows how to create a router list (router list 1) and then configure the Content Engine to accept redirected WMT traffic (the WCCP service named “wmt”) from the WCCP Version 2-enabled router on router list 1:

```
ContentEngine(config)# wccp router-list 1 10.10.10.2
ContentEngine(config)# wccp wmt router-list 1
ContentEngine(config)# wccp version 2
```

With WCCP Version 1, you can only configure a single WCCP-enabled router to support the standard web-cache service (service 0). Even if there is a cluster of Content Engines, only a single WCCP Version 1-enabled router services a cluster of Content Engines, becoming the default home router for the cluster. For information about how to configure a router as a home router for the standard web-cache service using WCCP Version 1, see the [“Example 1—Configuring the Web-Cache Service with WCCP Version 1”](#) section on page 6-39.

**Note**

The **ip wccp** global configuration command must be used to enable WCCP on each router that is included on the router list.

When configuring a Content Engine for WCCP Version 2, you can configure an IP multicast address instead of a list of routers on the Content Engine. The use of a list of routers on the Content Engine precludes the need to use IP multicast but requires more configuration on each Content Engine. The use of an IP multicast address reduces the configuration on the Content Engine as well as the protocol overhead.

With IP multicasting, an IP multicast address is configured on the Content Engine. The WCCP Version 2-enabled routers are configured to receive the IP multicast address on one or more interfaces. These routers then send their redirected requests to the specified IP multicast address on the Content Engine. Multicast addresses must be between 224.0.0.0 and 239.255.255.255. The Internet Assigned Numbers Authority (IANA) controls the assignment of IP multicast addresses. The IANA has assigned the IPv4 Class D address space to be used for IP multicast. Therefore, all IP multicast group addresses fall in the range from 224.0.0.0 through 239.255.255.255. However, some combinations of source and group address should not be routed for multicasting purposes. For a list of the unusable multicast address ranges and the reasons they should not be used, see the [“Unusable Multicast Address Assignments”](#) section on page B-11.

The additional configuration required on the WCCP Version 2-enabled routers that are intended to become members of the service group when IP multicast is used is as follows:

- The IP multicast address for use by the service group must be configured.
- The interface or interfaces that the WCCP Version 2-enabled router wants to receive the IP multicast address need to be configured with the **ip wccp {web-cache | service-number} group-listen** command.

For network configurations in which another router must be traversed to get to the target router, the router being traversed must be configured to perform IP multicast routing:

- IP multicast routing needs to be enabled by configuring it for the router with the **ip multicast-routing** command.
- The router interfaces that connect to the Content Engines must be configured to receive multicast with an **ip pim** command.

Configuring WCCP Services on Standalone Content Engines

Some of the WCCP services that WCCP-enabled routers and Content Engines can support have a well-known set of criteria and a predefined service identifier (for example, the standard web-cache service [service 0]). These services are called *predefined* WCCP services. Other examples include the reverse-proxy caching service (service 99), the https-caching service (service 70), and the rtsp service (service 80).

Other WCCP services that are not well known may be defined by specifying a set of criteria and assigning these user-defined WCCP Version 2 services a service identifier. WCCP Version 2 allows you to define up to eight user-defined WCCP services (services 90 to 97). Each of these user-defined services supports up to eight ports.

For more information about configuring a standalone Content Engine to support a user-defined WCCP service, see the [“Configuring Standalone Content Engines to Support User-Defined WCCP Services” section on page 6-15](#).



Note

See [Table B-3](#) for a list of supported WCCP services.

Keep these important points in mind when configuring a WCCP services for standalone Content Engines:

- WCCP Version 1 only supports one WCCP service (the standard web-cache service [service 0]) and a single router. Consequently, we recommend that you use WCCP Version 2 because it supports a wider set of features and services as well as multiple routers.
- You must create a list of WCCP Version 2-enabled routers that will support a specific WCCP service used on the standalone Content Engine. If different routers will be used for different WCCP services, it is necessary to create more than one router list.
- WCCP Version 2 must also be enabled on each of the routers listed in the router lists. You must also enable the specified WCCP service on each of the routers listed in a particular router list that has been associated with a specific WCCP service.
- After enabling WCCP Version 2 on the standalone Content Engine and defining a list of WCCP Version 2-enabled routers that will support this service, you must enable the specific WCCP services on the Content Engine and the WCCP-enabled routers that will be supporting that particular WCCP service.

The following example shows how to enable a user-defined WCCP service (service 91) on a WCCP Version 2-enabled router:

```
Router# configure terminal
Router(config)# ip wccp 91
Router(config)# interface ethernet 0
Router(config-if)# ip wccp 91 redirect out
```

- Use the **show ip wccp EXEC** command on the WCCP Version 2-enabled router to view values associated with WCCP variables.
- The Time To Live (TTL) value of routers servicing a cluster must be 15 seconds or less.

- Service groups consist of up to 32 Content Engines and 32 WCCP-enabled routers.
- All Content Engines in a cluster must include all WCCP-enabled routers that are servicing the cluster in their configuration. If a Content Engine within a cluster does not include one or more of the routers in its configuration, the service group detects the inconsistency and the Content Engine is not allowed to operate within the service group.

**Note**

A Content Engine and a WCCP-enabled router cannot be separated by a firewall. The firewall handles only packet traffic toward the origin web server and does not handle packet traffic sent to the client by the Content Engine on behalf of the server.

Many WCCP Version 2 features also require a configuration of certain options in the **wccp** global configuration command. Refer to the *Cisco ACNS Software Command Reference, Release 5.5* publication for more details on the **wccp** global configuration command. If you do not know how to configure a router or a switch, refer to the software documentation supplied with the devices. Further information on WCCP Version 2 commands and router configuration examples are in the Cisco IOS software online documentation.

Configuring Standalone Content Engines to Support User-Defined WCCP Services

A user-defined WCCP service is a WCCP Version 2 service in which port numbers can be configured to redirect traffic to a standalone Content Engine. The Content Engine is functioning as a transparent forward proxy server.

WCCP Version 2 allows you to configure up to eight user-defined WCCP services (services 90 to 97) that support multiple ports (up to eight ports per WCCP service). In order to configure these services, you must create one port list for each service that will be used (for example, create port list number 1 for service 90). The port list contains the port numbers that the WCCP Version 2-enabled router will support WCCP redirection on for that particular user-defined WCCP service. When configuring these user-defined WCCP services, you must specify whether the traffic is to be redirected to the caching application, the HTTPS caching application, or the streaming application on the Content Engine.

To configure the Content Engine to cache web traffic using multiple ports, configure the Content Engine and WCCP Version 2-enabled routers to run a user-defined WCCP service. Use these user-defined web services to support WCCP redirection of HTTP, MMS, HTTPS, and RTSP requests on multiple ports (up to eight ports per service) for the standard WCCP services (for example, the predefined https-cache, rtsp, mmst, and reverse-proxy services) that ordinarily only support a single port.

You can use the Content Engine GUI or CLI commands to configure a user-defined WCCP service on a Content Engine. From the Content Engine GUI, select **WCCP > Services**. Use the displayed Services window to configure a Content Engine to support a user-defined WCCP service (services 90 to 97). For more information about how to use the Services window, click the **HELP** button in the window.

From the Content Engine CLI, use the **wccp service-number** global configuration command to configure the Content Engine to support a user-defined WCCP service. As the following example shows, you can define a mask and router list for each user-defined WCCP service (for example, service 95) that you create.

```
ContentEngine(config)# wccp service-number 95 ?
mask                Specify mask used for CE assignment
router-list-num     Router list number
```

Use the **router-list-num** option to specify the list of WCCP Version 2-enabled routers that the Content Engine should accept redirected requests from for the specified WCCP service.

```
ContentEngine(config)# wccp service-number 95 router-list-num 1
```



Note The **router-list-num** option is not required when you are configuring Layer 2 redirection using the **http l4-switch** command or using policy-based routing. (See the “[Configuring Layer 4 Switching as a Redirection Method](#)” section on page 6-50.)

Use the **port lists** option to specify which ports the Content Engine is to listen on for redirected traffic from the specified WCCP service. The following example shows how to configure the Content Engine to listen on the ports listed in port list number 5 for redirected traffic for service 95 from the WCCP Version 2-enabled routers on router list 1.

```
ContentEngine(config)# wccp service-number 95 router-list-num 1 port-list-num 5
```

Use the **applications** option to specify whether the redirected traffic is to be directed to the caching application, the HTTPS caching application, or the streaming application on the Content Engine.

```
ContentEngine(config)# wccp service-number 95 router-list-num 1 port-list-num 1
application ?
cache                Direct traffic to the caching application
https-cache         Direct traffic to the https caching application
streaming           Direct traffic to the streaming media application
```

If you use the Content Engine GUI to enable and configure WCCP on the standalone Content Engine, then you must specify the designated router list for each service in each of the following Content Engine GUI windows: the Web Cache window (**WCCP > Web Cache**), the Reverse Proxy window (**WCCP > Reverse Proxy**), the Custom Web Cache window (**WCCP > Custom Web Cache**), and the WCCP Services window (**WCCP > Services**).

For an example of how to configure user-defined WCCP services (services 90 to 97) for standalone Content Engines, see the “[Example 4—Configuring Multiple WCCP Version 2 Services on Standalone Content Engines](#)” section on page 6-46.

Displaying WCCP Configuration Information for Standalone Content Engines

There are several Content Engine CLI commands that you can use to display WCCP-related configuration information (for example, a list of currently configured WCCP services) on standalone Content Engines.

To display a list of the WCCP services that are currently configured on a standalone Content Engine, enter the **show wccp services EXEC** command:

```
ContentEngine# show wccp services
Services configured on this Content Engine
  Web Cache
  Reverse Proxy
```

```
RTSP
WMT
MMSU
DNS
FTP
RTSPU
HTTPS Cache
WCCPv2 Service 90
WCCPv2 Service 91
WCCPv2 Service 92
WCCPv2 Service 93
WCCPv2 Service 94
WCCPv2 Service 95
WCCPv2 Service 96
WCCPv2 Service 97
```

For a list of supported WCCP services, see [Table B-3](#).

To display a list of Content Engines for the currently configured WCCP services, enter the **show wccp content-engines EXEC** command.

To display router-related information about the WCCP services that are configured on a standalone Content Engine, enter the **show wccp EXEC** command.

Disabling and Reenabling WCCP Flow Redirection on Standalone Content Engines

By default, WCCP flow redirection is enabled on a standalone Content Engine.

To reenabling WCCP flow redirection on a standalone Content Engine, use the **wccp flow-redirect enable** global configuration command.

```
ContentEngine(config)# wccp flow-redirect enable
```

To disable WCCP flow redirection, use the **no** form of this command.

```
ContentEngine(config)# no wccp flow-redirect enable
```

For more information, see the [“Configuring WCCP Flow Protection”](#) section on page 15-9.

Shutting Down WCCP on Standalone Content Engines

To prevent broken TCP connections, the Content Engine performs a proper shutdown of WCCP after you enter the **reload** or **no wccp version** command. The Content Engine does not reboot until either all connections have been serviced or the maximum wait time (specified with the **wccp shutdown max-wait** command [by default, 120 seconds]) has elapsed for WCCP Version 2.

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

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

You can use the Content Engine CLI or GUI to shut down WCCP on a standalone Content Engine. From the Content Engine GUI, check the **Clean WCCP shutdown** check box in the main window of the Content Engine GUI (see [Figure 4-17](#)), and then click the **REBOOT** button in the same window.

To use the Content Engine CLI to perform a proper shutdown of WCCP on a standalone Content Engine, follow these steps.

Step 1 Specify the time to wait for a proper shutdown.

```
Content Engine(config)# wccp shutdown max-wait seconds
```

seconds is the maximum period in seconds (0–86400) that the Content Engine waits before it performs a proper shutdown of WCCP after you have entered a **no wccp version** command. The default is 120 seconds. This command is only supported for WCCP Version 2. The following example shows how to configure the Content Engine to wait 1000 seconds:

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

Step 2 Shut down WCCP Version 2 on the Content Engine.

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

The Content Engine waits 1000 seconds before it shuts down WCCP Version 2. A countdown message appears, indicating how many seconds remain before WCCP will be shut down on the Content Engine.

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

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

Configuring a Router for WCCP Transparent Redirection

This section describes how to perform the following tasks on a router:

- [Setting a Password for a WCCP Version 2-Enabled Router, page 6-20](#)
- [Performing a General WCCP Version 2 Configuration on a Router, page 6-20](#)
- [Enabling WCCP on a Router, page 6-20](#)
- [Enabling a WCCP Version 2 Router to Support WCCP Service Groups, page 6-20](#)
- [Enabling Packet Redirection on Outbound or Inbound Interfaces Using WCCP, page 6-24](#)
- [Bypassing the Content Engine with Router Access Lists, page 6-26](#)

**Note**

Use the **ip wccp** command on the WCCP Version 2-enabled router to specify the WCCP service that you want the router to run. The WCCP service is specified by its service number or name. For a complete list of supported WCCP services, service numbers, and names, see [Table B-3](#). For information about how to configure WCCP services on a router, see the “[Configuring WCCP Services on a Router](#)” section on [page 6-27](#).

Setting a Password for a WCCP Version 2-Enabled Router

You must set a password for a WCCP Version 2-enabled router that the standalone Content Engine will access, as follows:

```
Router(config)# ip wccp web-cache password [0-7] password
```

where:

- **password** is the string that directs the WCCP Version 2-enabled router to apply MD5 authentication to messages received from the specified service group. Messages that are not accepted by the authentication are discarded.
- *0-7* is the optional value that indicates the HMAC MD5 algorithm 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 security for the connection between the router and the Content Engine.

Performing a General WCCP Version 2 Configuration on a Router

The following example shows a general WCCP Version 2 configuration session on a router:



Note

You must enter the **ip wccp version 2** command on all WCCP Version 2 router configurations to enable redirection using WCCP Version 2.

```
Router# configure terminal
Router(config)# ip wccp version 2
Router(config)# interface ethernet0
Router(config-if)# ip wccp web-cache redirect out
```

Enabling WCCP on a Router

To enable WCCP on a router, enter the **ip wccp version** global configuration command. For example, the following command enables WCCP Version 2 on the router:

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



Note

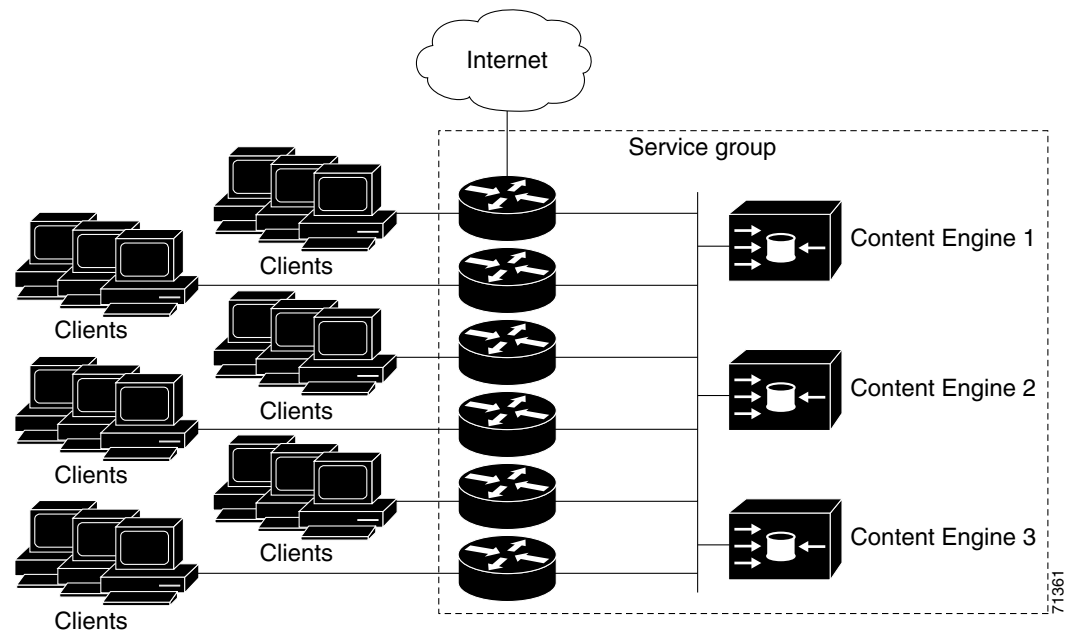
The **ip wccp** global configuration command must be used to enable WCCP on each router that is included on the router list.

Enabling a WCCP Version 2 Router to Support WCCP Service Groups

WCCP Version 2 enables a set of Content Engines in a Content Engine cluster to connect to multiple routers. The Content Engines in a cluster and the WCCP Version 2-enabled routers connected to the Content Engine cluster that are running the same service are known as a *service group*. Standalone Content Engines (Content Engines that are not registered with a Content Distribution Manager) can be part of a Content Engine cluster.

Through communication with the Content Engines, the WCCP Version 2-enabled routers are aware of the available Content Engines. Routers and Content Engines become aware of one another and form a service group using WCCP Version 2. See [Figure 6-3](#).

Figure 6-3 Service Groups with WCCP Version 2



Once the service group has been established, one of the Content Engines is designated to determine the lead assignments among the Content Engines in the Content Engine cluster. The type of supported WCCP services varies depending on whether WCCP Version 1 or Version 2 is used. All WCCP services that are listed in [Table B-3](#) except for the standard web-cache service (service 0) require WCCP Version 2.

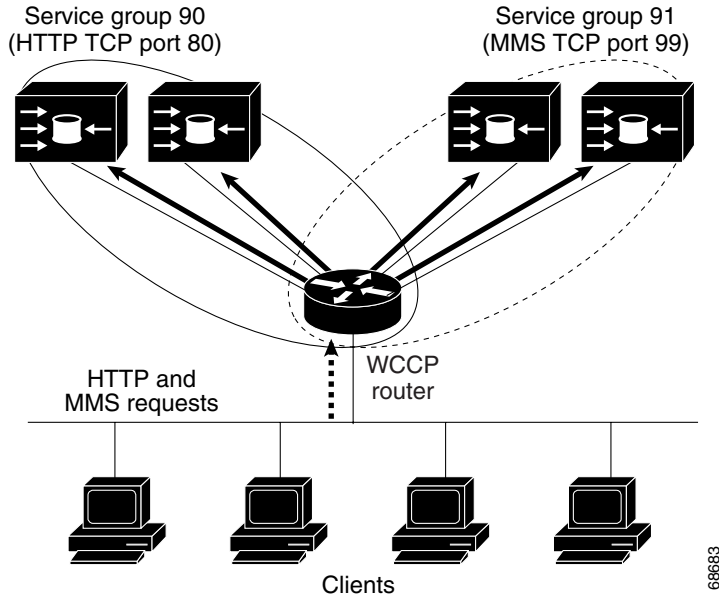
WCCP uses the concept of a service group to define caching-related services for a WCCP Version 2-enabled router and Content Engines in a cluster. WCCP also redirects user requests from clients requesting these caching-related services to these clusters in real time. In transparent caching through WCCP, you can configure a WCCP Version 2-enabled router to redirect requests to the Content Engine that is functioning as a transparent caching engine.

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

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

In [Figure 6-4](#), the two Content Engines on the left handle only HTTP traffic through port 80 and are defined as members of service group 90. The two Content Engines on the right handle only Microsoft Media Server (MMS) requests through port 99 and are defined as members of service group 91.

Figure 6-4 WCCP Version 2 Service Groups



The custom web-cache and reverse-proxy services (service 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.

With eight user-defined services using a maximum number of eight ports each, the maximum number of ports that can be specified for transparent redirection is 64.

WCCP can also handle asymmetric packet flows and always maintains a consistent mapping of web servers to caches regardless of the number of switches or routers used in a WCCP service group (up to 32 routers or switches communicating with up to 32 Content Engines in a cluster).

To direct a WCCP Version 2-enabled router to enable or disable support for a WCCP service group, use the **ip wccp** global configuration command. To remove the ability of a router to control support for a WCCP service group, use the **no** form of this command.

```
ip wccp {web-cache | service-number} [group-address groupaddress] [redirect-list access-list]
[group-list access-list] [password [0-7] password]
```

Table 6-4 describes the **ip wccp** command parameters.

Table 6-4 Parameters of the ip wccp Command

web-cache	Enables the web-cache service.
<i>service-number</i>	Identification number of the WCCP service being controlled by a WCCP Version 2-enabled router. The service number can be from 0 to 99. For a list of WCCP service numbers, see Table B-3.
group-address	(Optional) Directs the WCCP Version 2-enabled router to use a specified multicast IP address for communication with the WCCP service group.
<i>groupaddress</i>	(Optional) Multicast address used by the WCCP Version 2-enabled router to determine which Content Engine should receive redirected messages.
redirect-list	(Optional) Directs the WCCP Version 2-enabled router to use an access list to control traffic redirected to this WCCP service group.

Table 6-4 Parameters of the ip wccp Command (continued)

<i>access-list</i>	(Optional) String (not to exceed 64 characters) that is the name of the access list that determines which traffic is redirected to a Content Engine.
group-list	(Optional) Directs the WCCP Version 2-enabled router to use an access list to determine which Content Engines are allowed to participate in the WCCP service group.
<i>access-list</i>	(Optional) String (not to exceed 64 characters) that is the name of the access list that determines which Content Engines are allowed to participate in the WCCP service group.
password	(Optional) String that directs the WCCP Version 2-enabled router to apply MD5 authentication to messages received from the specified service group. Messages that are not accepted by the authentication are discarded.
<i>0-7</i>	(Optional) Value that indicates the HMAC MD5 algorithm used to encrypt the password. This value is generated when an encrypted password is created for the Content Engine.
<i>password</i>	(Optional) Password name that is combined with the HMAC MD5 value to create security for the connection between the WCCP Version 2-enabled router and the Content Engine.

The following example shows how to configure a WCCP Version 2-enabled router to run the WCCP reverse proxy service (service 99), using (listening to) the multicast address 172.31.0.0:

```
Router(config)# ip wccp 99 group-address 172.31.0.0
```

**Note**

Use the **ip wccp group-listen** command to configure an interface on a WCCP Version 2-enabled router to enable or disable the reception of IP multicast packets for the WCCP feature.

WCCP Version 2 provides authentication that enables you to control which WCCP Version 2-enabled routers and Content Engines become part of the WCCP service group. You use passwords and the HMAC MD5 algorithm by the **ip wccp password [0-7] password** command to control service group membership.

Enabling WCCP Redirection on the Router

Before you use WCCP Version 2, you must configure IP on the interface connected to the Internet and the interface connected to the Content Engine. The interface connected to the Content Engine must be an Ethernet or Fast Ethernet interface.

To enable a router interface to use WCCP Version 2 to redirect web traffic to a standalone Content Engine that is running the ACNS 5.x software, follow these steps:

-
- Step 1** Enable the router to use WCCP.
- ```
Router# configure terminal
Router(config)# ip wccp version 2
```
- Step 2** (Optional) Specify a redirect access list.

Only packets that match this access list are redirected to the Content Engine. If you do not specify a redirect access list, all packets are redirected to the Content Engine.

```
Router(config)# ip wccp redirect-list [number | name]
```

**Step 3** Enter interface configuration mode by specifying an interface name and number.

The following example shows how to specify the Ethernet 0 interface:

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

**Step 4** Configure the router interface that is connected to the Internet to redirect web traffic to the standalone Content Engine.

```
Router(config-if)# ip wccp web-cache redirect [in | out]
```




---

**Note** All WCCP-enabled routers support the **out** option but only certain routers support the **in** option. Consequently, we recommend that you specify the **out** option whenever possible. For more information, see the next section, “[Enabling Packet Redirection on Outbound or Inbound Interfaces Using WCCP.](#)”

---

**Step 5** (Optional) Configure the router to use the fast switching path on the interface if the client and a Content Engine are located on the same network.

```
Router(config-if)# ip route-cache same-interface
```

**Step 6** Exit configuration mode.

```
Router(config-if)# end
```

**Step 7** Save the running configuration to the startup configuration, which is stored in nonvolatile memory.

```
Router # copy running-config startup-config
```

---

## Enabling Packet Redirection on Outbound or Inbound Interfaces Using WCCP

Redirection can be specified for either outbound or inbound interfaces. Inbound traffic can be configured to use Cisco Express Forwarding (CEF), distributed Cisco Express Forwarding (dCEF), fast forwarding, or process forwarding.

Configuring WCCP for redirection of inbound traffic on interfaces allows you to avoid the overhead associated with CEF forwarding for outbound traffic. If you enable redirection on an outbound interface, this can cause all packets that arrive at all interfaces to take the slower switching path. If you enable redirection on an inbound interface, only those packets arriving at that interface will take the configured feature path; packets arriving at other interfaces will use the faster default path. However, not all routers support redirection on their inbound interfaces.

Configuring WCCP for inbound traffic also allows packets to be classified before the routing table lookup, which provides faster redirection of packets.

To enable packet redirection on an outbound or inbound interface using WCCP, use the **ip wccp redirect** interface configuration command as described in Table 6-5. To disable WCCP redirection, use the **no** form of this command.

**ip wccp {web-cache | service-number} redirect {out | in}**

**Table 6-5 Parameters of the ip wccp redirect Command**

| Parameter             | Description                                                                                                                                                                                 |
|-----------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>web-cache</b>      | Enables the web-cache service.                                                                                                                                                              |
| <i>service-number</i> | Identification number of the WCCP service group being controlled by a WCCP Version 2-enabled router. The number can be from 0 to 99. See Table B-3 for a list of supported service numbers. |
| <b>redirect</b>       | Enables packet redirection checking on an outbound or inbound interface.                                                                                                                    |
| <b>out</b>            | Specifies packet redirection on an outbound interface.                                                                                                                                      |
| <b>in</b>             | Specifies packet redirection on an inbound interface.                                                                                                                                       |

The **ip wccp redirect** interface command has the potential to affect the **ip wccp redirect exclude in** command. If you have **ip wccp redirect exclude in** set on an interface and you subsequently configure the **ip wccp redirect in** command, the **exclude in** command is overridden. The opposite is also true: configuring the **exclude in** command overrides the **redirect in** command.

To disable caching for certain clients, follow these steps:

**Step 1** Set the access list used to enable redirection.

```
Router# configure terminal
Router(config)# ip wccp web-cache redirect-list access-list number
```

**Step 2** Create an access list that enables or disables traffic redirection to the Content Engine.

```
Router(config)# access-list access-list number deny host host-address
```

**Step 3** Set the access list to enable access to any host.

```
Router(config)# access-list access-list number permit ip any
```

The following example shows a configuration session in which reverse proxy packets on Ethernet interface 0 are being checked for redirection and redirected to a Content Engine:

```
Router# configure terminal
Router(config)# ip wccp 99
Router(config)# interface ethernet 0
Router(config-if)# ip wccp 99 redirect out
```

The following example shows a configuration session in which HTTP traffic arriving on Ethernet interface 0/1 is redirected to a Content Engine:

```
Router# configure terminal
Router(config)# ip wccp web-cache
Router(config)# interface ethernet 0/1
Router(config-if)# ip wccp web-cache redirect in
```

## Bypassing the Content Engine with Router Access Lists

By default, all HTTP packets are redirected to the Content Engine. A WCCP Version 2-enabled router can be configured with access lists to permit or deny redirection of traffic to a standalone Content Engine. In the following example, traffic conforming to the following criteria are not redirected by the router to the Content Engine:

- Originating from the host 10.1.1.1 destined for any other host
- Originating from any host destined for the host 10.255.1.1

```
Router# configure terminal
Router(config)# ip wccp web-cache redirect-list 120
Router(config)# access-list 120 deny ip host 10.1.1.1 any
Router(config)# access-list 120 deny ip any host 10.255.1.1
Router(config)# access-list 120 permit ip any
```

Traffic not explicitly permitted is implicitly denied redirection. The **access-list 120 permit ip any** command explicitly permits all traffic (from any source en route to any destination) to be redirected to the Content Engine. Because criteria matching occurs in the order in which the commands are entered, the global **permit** command is the last command entered. For further information on access lists, refer to Cisco IOS software documentation.

Use the **ip wccp redirect-list** global configuration command to limit the redirection of packets to those matching an access list. Use this command to specify which packets should be redirected to the Content Engine.

When WCCP is enabled but the **ip wccp redirect-list** command is not used, all web-related packets are redirected to the Content Engine. When you specify the **ip wccp redirect-list** command, only packets that match the access list are redirected.

The **ip wccp** global configuration command and the **ip web-cache redirect** interface configuration command are the only commands required to start redirecting requests to the Content Engine using WCCP. To instruct an interface on the WCCP-enabled router to check for appropriate outgoing packets and redirect them to a Content Engine, use the **ip web-cache redirect** interface configuration command. When the **ip wccp** command is enabled but the **ip web-cache redirect** command is disabled, the WCCP-enabled router is aware of the Content Engine but does not use it.

Some websites use the source IP address of packets for authentication. The Content Engine uses its own IP address when sending requests to websites. Thus, the requests from the Content Engine may not be authenticated. Use the **ip wccp redirect-list** command to bypass the Content Engine in these cases.

**ip wccp redirect-list** {*number* | *name*}

where:

- *number* is the standard or extended IP access list number from 1 to 199.
- *name* is the standard or extended IP access list name. This argument is only available in Cisco IOS Release 11.2 P.

# Configuring WCCP Services on a Router

This section describes how to configure the following WCCP services on a router that is running WCCP Version 2:

- [Configuring the Standard Web-Cache Service \(Service 0\) on a Router, page 6-27](#)
- [Configuring the DNS Caching Service \(Service 53\) on a Router, page 6-28](#)
- [Configuring the FTP-Native Caching Service \(Service 60\) on a Router, page 6-29](#)
- [Configuring the HTTPS-Cache Service \(Service 70\) on a Router, page 6-29](#)
- [Configuring the RTSP Service \(Service 80\) on a Router, page 6-30](#)
- [Configuring the WMT-RTSPU Service \(Service 83\) on a Router, page 6-31](#)
- [Configuring User-Defined WCCP Services \(Services 90–97\) on a Router, page 6-31](#)
- [Configuring the Custom-Web-Cache Service \(Service 98\) on a Router, page 6-32](#)
- [Configuring the Reverse-Proxy Service \(Service 99\) on a Router, page 6-33](#)

Remember that after you configure the WCCP service on the router, you must also configure the Content Engine to accept the redirected requests. See “[Configuring Standalone Content Engines for WCCP Transparent Redirection](#)” section on page 6-9.

## Configuring the Standard Web-Cache Service (Service 0) on a Router

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

The following is an example of how to enable WCCP Version 2 on a router, and then configure the standard web-cache service (service 0) on the router:

---

**Step 1** Enable WCCP Version 2 on the router.

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

**Step 2** Enable the standard web-cache service (service 0) on the router.

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

**Step 3** Specify the interface on which the standard web-cache service will run. Typically, this interface carries the traffic that is going out to the Internet.

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

In the following example, Ethernet interface 0/1 on the router is configured to run the standard web-cache service:

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

- Step 4** Configure the router to check the HTTP traffic that arrives on the interface that the standard web-cache service is configured on (for example, Ethernet interface 0/1). The router checks this traffic to determine whether it should redirect these packets to the standalone Content Engine. This Content Engine is functioning as a transparent forward proxy server that will accept redirected HTTP requests on port 80 from this WCCP Version 2-enabled router.

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

---

Remember that you must also configure the standalone Content Engine to accept redirected HTTP requests on port 80 by configuring the standard web-cache service on the Content Engine (transparent HTTP forward proxy caching). For more information on this topic, see the [“Configuring the Standard Web-Cache Service \(Service 0\) for Standalone Content Engines”](#) section on page 7-18.

## Configuring the DNS Caching Service (Service 53) on a Router

The DNS caching service (service 53) is a predefined WCCP Version 2 caching service. This service permits WCCP Version 2-enabled routers to redirect client requests transparently to a Content Engine so that the Content Engine can resolve the DNS name. After the Content Engine resolves the DNS name, the Content Engine stores it locally so that it can use these resolved names for future DNS requests.

To configure the DNS caching service (service 53) on a router, follow these steps:

- Step 1** Enable WCCP Version 2 on the router.

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

- Step 2** Enable the DNS caching service (service 53) on the router.

```
Router(config)# ip wccp 53
```

- Step 3** Specify the router interface on which the DNS caching service will run.

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

- Step 4** Configure the router to use the outbound interface for the DNS caching service.

```
Router(config-if)# ip wccp 53 redirect out
```

---

Remember that you must also configure the DNS caching service (service 53) on the standalone Content Engine before the Content Engine can accept redirected DNS requests from WCCP Version 2 routers. For information on this topic, see the [“Configuring DNS Caching for Standalone Content Engines”](#) section on page 7-62.

## Configuring the FTP-Native Caching Service (Service 60) on a Router

The ftp-native caching service (service 60) is a predefined WCCP Version 2 caching service. This service permits WCCP Version 2-enabled routers to redirect FTP native requests transparently to a single port on the Content Engine. The Content Engine retrieves the requested FTP content, stores a copy locally (native FTP caching), and serves the requested content to the FTP client.

**Note**

In the ACNS 5.3.1 software release, the name of this service was changed from “ftp” to “ftp-native” to clearly differentiate between FTP native requests and FTP-over-HTTP requests. Service 60 (the ftp-native caching service) only applies to transparent redirection of FTP native requests and does not apply to FTP-over-HTTP requests.

To configure the ftp-native caching service (service 60) on a router, follow these steps:

- 
- Step 1** Enable WCCP Version 2 on the router.
- ```
Router# configure terminal  
Router(config)# ip wccp version 2
```
- Step 2** Enable the ftp-native caching service (service 60) on the router.
- ```
Router(config)# ip wccp 60
```
- Step 3** Specify the interface on which the ftp-native caching service will run.
- ```
Router(config)# interface type number
```
- Step 4** Configure the router to use the outbound interface for the ftp-native caching service.
- ```
Router(config-if)# ip wccp 60 redirect out
```
- 

Remember that you must configure the ftp-native caching service on the standalone Content Engine before the Content Engine can accept redirected FTP-native requests from WCCP Version 2 routers. For more information on this topic, see the [“Configuring Transparent FTP Native Caching” section on page 7-54](#).

## Configuring the HTTPS-Cache Service (Service 70) on a Router

The https-cache service (service 70) is a predefined WCCP Version 2 web-caching service. This service permits WCCP Version 2-enabled routers to redirect HTTPS traffic transparently to a standalone Content Engine on port 443.

To configure the https-cache service (service 70) on a router, follow these steps:

- 
- Step 1** Enable WCCP Version 2 on the router.
- ```
Router# configure terminal  
Router(config)# ip wccp version 2
```
- Step 2** Enable the https-cache service (service 70) on the router.
- ```
Router(config)# ip wccp 70
```
- Step 3** Specify the interface on which the https-cache service will run.

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

In the following example, the Ethernet 0 interface on the router is configured to run the https-cache service.

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

**Step 4** Configure the router to use the outbound interface for the https-cache service.

```
Router(config-if)# ip wccp 70 redirect out
```

---

Remember that you must also configure the standalone Content Engine for HTTPS transparent caching before it can accept redirected HTTPS requests from WCCP Version 2-enabled routers. For more information on this topic, see the [“Configuring HTTPS Transparent Caching for Standalone Content Engines”](#) section on page 7-27.

## Configuring the RTSP Service (Service 80) on a Router

The rtsp service (service 80) is a predefined WCCP Version 2 media-caching service. This media-caching service that permits WCCP Version 2-enabled routers to redirect RTSP client requests transparently to a single port on a Content Engine (RealMedia transparent caching).

The Content Engine listens for redirected RTSP requests on the standard RTSP port (default port 554). To intercept RTSP traffic on ports other than the default port (port 554), configure a user-defined WCCP service (services 90 to 97). To configure transparent interception of RTSP requests from RealMedia clients, you only need to configure the rtsp service (service 80) on the WCCP Version 2-enabled router.

In contrast, you must configure the rtsp service (service 80) as well as the wmt-rtspu service (service 83) on the WCCP Version 2-enabled router to configure transparent interception of WMT RTSP requests. For information about configuring service 83 on a router, see the [“Configuring the WMT-RTSPU Service \(Service 83\) on a Router”](#) section on page 6-31.

To configure the rtsp service (service 80) on a router, follow these steps:

---

**Step 1** Enable WCCP Version 2 on the router.

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

**Step 2** Enable the rtsp service (service 80) on the router.

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

**Step 3** Specify the interface on which the rtsp service will run.

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

**Step 4** Configure the router to use the outbound interface for the rtsp service.

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

---

**Note**

Remember that you must configure RealMedia transparent caching on the standalone Content Engine before it can accept redirected RealMedia RTSP requests from WCCP Version 2-enabled routers. For more information on this topic, see [Chapter 8, “Configuring RealMedia Services on Standalone Content Engines.”](#)

## Configuring the WMT-RTSPU Service (Service 83) on a Router

The wmt-rtspu service (service 83) is a predefined WCCP Version 2 media-caching service. This service permits WCCP Version 2-enabled routers to redirect RTSP client requests from Windows Media 9 players transparently to a single port on a Content Engine (that is, acting as a transparent proxy server, which is configured for WMT RTSP transparent caching). The wmt-rtspu service was added in the ACNS 5.3.1 software release, and is also called the rtspu service.

The Content Engine listens for redirected RTSP requests on the standard RTSPU port (default port 5005). To intercept WMT RTSP traffic on ports other than the default port (port 5005), configure a user-defined WCCP Version 2 service (services 90 to 97).

To configure the wmt-rtspu service (service 83) on a router, follow these steps:

- 
- Step 1** Enable WCCP Version 2 on the router.
- ```
Router# configure terminal  
Router(config)# ip wccp version 2
```
- Step 2** Enable the wmt-rtspu service (service 83) on the router.
- ```
Router(config)# ip wccp 83
```
- Step 3** Specify the interface on which the wmt-rtspu service will run.
- ```
Router(config)# interface type number
```
- Step 4** Configure the router to use the outbound interface for the wmt-rtspu service.
- ```
Router(config-if)# ip wccp 82 redirect out
```
- 

You must also configure the rtsp service (service 80) on the WCCP router to support transparent redirection of WMT RTSP requests to a Content Engine.

Remember that you must also configure WMT RTPS transparent caching on the standalone Content Engine before it can accept redirected WMT RTSP requests from WCCP Version 2-enabled routers. For more information on this topic, see the [“Configuring WMT RTSP Streaming and Caching Services on Standalone Content Engines”](#) section on page 9-14.

## Configuring User-Defined WCCP Services (Services 90–97) on a Router

To configure a router to use WCCP Version 2 to support a user-defined WCCP service (services 90 to 97), follow these steps:

- 
- Step 1** Enable WCCP Version 2 on the router.
- ```
Router# configure terminal
```

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

- Step 2** Enable the WCCP feature for the user-defined service (for example, service 90).

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

- Step 3** Specify the interface on which service 90 will run on the router.

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

- Step 4** Configure the Content Engine to use the outbound interface for service 90.

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

Remember that you must also configure the user-defined WCCP service (for example, service 90) on the standalone Content Engine before the Content Engine can accept redirected proxy packets from WCCP Version 2-enabled routers. For more information on this topic, see the [“Configuring Standalone Content Engines to Support User-Defined WCCP Services”](#) section on page 6-15.

Configuring the Custom-Web-Cache Service (Service 98) on a Router

The custom-web-cache service (service 98) is a predefined WCCP Version 2 web-caching service. This service permits WCCP Version 2-enabled routers to redirect HTTP traffic to a Content Engine on multiple ports other than port 80. The Content Engine is functioning as a transparent forward proxy server. This WCCP service allows you to configure the Content Engine to listen on multiple ports (up to eight ports) for WCCP redirected HTTP requests without having to configure a user-defined WCCP service (services 90 to 97).

To configure a router to use WCCP Version 2 to support the custom-web-cache service (service 98), follow these steps:

- Step 1** Enable WCCP Version 2 on the router.

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

- Step 2** Enable the custom-web-cache service (service 98) on the router.

```
Router(config)# ip wccp 98
```

- Step 3** Specify the interface on which the custom-web-cache service will run.

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

In the following example, the Ethernet 0 interface is configured to run the custom-web-cache service:

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

- Step 4** Configure the router to use the outbound interface for the custom-web-cache service.

```
Router(config-if)# ip wccp 98 redirect out
```

Remember that you must also configure the custom-web-cache service on the standalone Content Engine before the Content Engine can accept redirected web cache proxy packets from WCCP Version 2-enabled routers on multiple ports. For more information on this topic, see the [“Configuring the Custom Web-Cache Service \(Service 98\) for Standalone Content Engines”](#) section on page 7-20.

Configuring the Reverse-Proxy Service (Service 99) on a Router

The reverse-proxy service (service 99) is a predefined WCCP Version 2 service. This service permits WCCP Version 2-enabled routers to redirect reverse proxy packets to a standalone Content Engine that is functioning as a transparent reverse proxy server.

To configure a router to use WCCP Version 2 to support the reverse-proxy service (service 99), follow these steps:

- Step 1** Enable WCCP Version 2 on the router.

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

- Step 2** Enable the reverse-proxy service (service 99) on the router.

```
Router(config)# ip wccp 99
```

- Step 3** Specify the interface on which the reverse-proxy service will run.

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

In the following example, the Ethernet 0 interface is configured to run the reverse-proxy service:

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

- Step 4** Configure the router to use the outbound interface for the reverse-proxy service. The router will check the reverse proxy packets on Ethernet interface 0 to determine if it should transparently redirect these packets to the Content Engine (that is acting as a transparent reverse proxy server).

```
Router(config-if)# ip wccp 99 redirect out
```

Remember that you must also configure the reverse-proxy service (service 99) on the standalone Content Engine before the Content Engine can accept redirected reverse proxy packets from WCCP Version 2 routers. For more information on this topic, see the [“Configuring HTTP Reverse Proxy Caching for Standalone Content Engines”](#) section on page 7-23.

Clearing WCCP Statistics on a Router

Use the **clear ip wccp EXEC** command to clear the WCCP statistics maintained on the WCCP Version 2 router, either for a particular service or for all the services.

```
clear ip wccp { web-cache | service-number }
```

where:

- **web-cache** specifies that the router should remove statistics for the web-cache service.
- *service-number* specifies that the router should remove statistics for the specified service. The service group number can be from 0 to 99.

For example, direct the router to clear the statistics for the reverse-proxy service (service 99) by entering the following command on the WCCP Version 2-enabled router:

```
Router# clear ip wccp web-cache 99
```

Configuring WCCP Layer 2 Support

WCCP on a router or switch can take advantage of switching hardware that either partially or fully implements the traffic interception and redirection functions of WCCP in hardware at Layer 2. This allows the Content Engine to perform a Layer 2 or MAC address rewrite redirection if it is directly connected to a compatible Cisco switch. This redirection processing is accelerated in the switching hardware, which makes this method a more efficient method than Layer 3 redirection using GRE.

The Content Engine must have a Layer 2 connection with the switch. Because there is no requirement for a GRE tunnel between the switch and the Content Engine, the switch can use a cut-through method of forwarding encapsulated packets using the **l2-redirect** option in the CLI.

Two load-balancing schemes exist between WCCP Version 2-enabled routers or switches and Content Engines when the Layer 2 forwarding method is chosen:

- Hash assignment
For the Catalyst 6000 and 6500 series switches, this load-balancing method is called WCCP Layer 2 Policy Feature Card (PFC) redirection. This method is intended to achieve forwarding performance of up to 3 gigabits per second using a combination of the Supervisor Engine 1A and the Multilayer Switch Feature Card 2 (MSFC2).
- Mask assignment
This type of load-balancing is called the WCCP Layer 2 Policy Feature Card 2 (PFC2) redirection. It uses a combination of the Supervisor Engine 2 and the MSFC2.

You can use the Content Engine GUI or CLI commands to specify the load-balancing schemes for a specific WCCP service on a Content Engine. All WCCP services supported by the Content Engine use such CLI commands as **wccp custom-web-cache**, **wccp media-cache**, **wccp reverse-proxy**, **wccp service-number**, **wccp web-cache**, **wccp wmt**, and **wccp rtsp** to support either the hash or the mask assignment load-balancing method with Layer 2 forwarding. You can specify one load-balancing method (hashing and masking) per WCCP service in a Content Engine cluster. For example, if you define three WCCP services for Content Engine Cluster A, two of the services in Cluster A could be using the hash load-balancing method. The third service in Cluster A could be using the mask load-balancing method.

**Note**

You can only enable Layer 2 redirection with the mask assignment load-balancing method through the Content Engine CLI (this is not supported through the Content Engine GUI).

For information about how to configure the hash load-balancing method, see the next section, “[Configuring Layer 2 Forwarding with the Hash Load-Balancing Method](#).” For information about how to configure the mask load-balancing method, see the “[Configuring Layer 2 Forwarding with the Mask Load-Balancing Method](#)” section on page 6-36.

Configuring Layer 2 Forwarding with the Hash Load-Balancing Method

Both types of packet-forwarding methods (layer GRE and layer 2 redirection) support hashing as a load-balancing method. Hashing allows you to specify how redirected traffic should be load balanced among multiple Content Engines in a Content Engine cluster.

When configuring user-defined WCCP Version 2 services on a Content Engine, you can configure hashing parameters (for example, hash on source IP address) for that particular user-defined WCCP Version 2 service. The default hashing assignment for user-defined WCCP services (Services 90 to 97) is hash on destination IP addresses. You use the **wccp service-number** global configuration command to change the default hashing assignment for any of the user-defined WCCP services.

The following example shows how you can use the **wccp service-number** command to configure a user-defined WCCP service (in this case, Service 90) to hash on source IP addresses instead of destination IP addresses:

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

[Table 6-6](#) lists the default hashing assignments. The default hashing assignment for predefined WCCP services is fixed and cannot be changed.

Table 6-6 Default Hashing Assignments for WCCP Version 2 Services

WCCP Service Type	Service Number (Identifier)	Default Hashing Assignment for Service
User-defined WCCP services	90 to 97	Default hashing assignment is hash on destination IP addresses
Predefined WCCP services		
web-cache	0	Hash on destination IP addresses
dns caching	53	Hash on source port
ftp-native	60	Hash on destination IP addresses
https-cache	70	Hash on source IP addresses
rtsp	80	Hash on destination IP addresses
mmst	81	Hash on destination IP addresses
mmsu	82	Hash on destination IP addresses
wmt-rtspu	83	Hash on destination IP addresses

Table 6-6 Default Hashing Assignments for WCCP Version 2 Services (continued)

WCCP Service Type	Service Number (Identifier)	Default Hashing Assignment for Service
custom-web-cache	98	Hash on destination IP addresses
reverse-proxy	99	Hash on source IP addresses and source port

The following example shows how to configure a Content Engine to receive Layer 2 redirected traffic from a Catalyst 6500 series switch with a Multilayer Switch Feature Card (MSFC) and Supervisory Engine 1A (SUP 1A) using a hash assignment method for load balancing. To configure the Content Engine, follow these steps:

Step 1 Enable WCCP Version 2 on the Content Engine.

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

Step 2 Create a router list on the Content Engine. In the following example, router list 1 is created and contains only a single WCCP Version 2-enabled router (the router with an IP address of 172.16.55.1).

```
ContentEngine(config)# wccp router-list 1 172.16.55.1
```

Step 3 Configure the standard web-cache service (service 0) on the Content Engine.

Configure this WCCP service to use the router list you just created in Step 2. Enter the **l2-redirect** option to specify Layer 2 redirection as the packet-forwarding method (as opposed to GRE) for this service. Because the mask assignment method is not specified, the default hash assignment method is used to load balance redirected requests.

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

Step 4 Use the **show wccp services detail EXEC** command to display the configuration so that you can verify it.

```
ContentEngine# show wccp services detail
```

Step 5 Write the running configuration to nonvolatile memory.

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

Configuring Layer 2 Forwarding with the Mask Load-Balancing Method

Both types of packet-forwarding methods (GRE and Layer 2 redirection) support masking as a load-balancing method. Use the **wccp service-name mask** global configuration command to specify the different masks (for example, the destination IP mask) on the Content Engine.

Use the **wccp service-name mask** global configuration command to change the default masks for a particular WCCP Version 2 service. For example, use the **wccp https-cache mask** global configuration command to configure masks for transparently redirected HTTPS requests (the https-cache service).

```
ContentEngine(config)# wccp https-cache mask ?
dst-ip-mask      Specify sub-mask used in packet destination-IP address
dst-port-mask    Specify sub-mask used in packet destination-port number
src-ip-mask      Specify sub-mask used in packet source-IP address
src-port-mask    Specify sub-mask used in packet source-port number
```

```
wccp https-cache {mask {[dst-ip-mask hex_num] [dst-port-mask port_hex_num]
[src-ip-mask hex_num] [src-port-mask port_hex_num]}}
```

Table 6-7 describes the command parameters.

Table 6-7 Parameters of the wccp https-cache CLI Command

Parameter	Description
mask	Sets the mask used for Content Engine assignment. Configure at least one mask. You can configure up to four masks.
dst-ip-mask	(Optional) Sets the mask used to match the destination IP address of the redirected packet.
<i>hex_num</i>	IP address mask defined by a hexadecimal number (for example, 0xFC000000). The range is 0x00000000 to FC000000.
dst-port-mask	(Optional) Sets the mask used to match the destination port number of the redirected packet.
<i>port_hex_num</i>	Source port mask defined by a hexadecimal number (for example, 0xFC00). The port range is 0 to 65535.
src-ip-mask	(Optional) Sets the mask used to match the source IP address of the redirected packet.
<i>hex_num</i>	IP address mask defined by a hexadecimal number (for example, 0xFC000000). The range is 0x00000000 to FC000000.
src-port-mask	(Optional) Sets the mask used to match the source port number of the redirected packet.
<i>port_hex_num</i>	Source port mask defined by a hexadecimal number (for example, 0xFC00). The port range is 0 to 65535.

To view the mask configuration for a specific WCCP Version 2 service, use the **show wccp masks service-name EXEC** command:

```
ContentEngine(config)# show wccp masks ?
  custom-web-cache  Custom web caching service
  dns               DNS caching service
  ftp-native        Native FTP caching service
  https-cache       HTTPS caching service
  reverse-proxy     Reverse Proxy web caching service
  rtsp              Media caching service
  service-number    Custom-service number
  web-cache         Standard web caching service
  wmt-rtspu        WMT RTSPU service
```

The following example shows how to configure a Content Engine to receive Layer 2 redirected traffic from a Catalyst 6500 series switch with a Multilayer Switch Feature Card 2 and Supervisor Engine 2 (MSFC2/SUP 2). To configure the Content Engine, follow these steps:

Step 1 Enable WCCP Version 2 on the Content Engine.

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

Step 2 Create a router list on the Content Engine. In the following example, router list 1 is created and contains only a single WCCP Version 2-enabled router (the router with an IP address of 172.16.55.1):

```
ContentEngine(config)# wccp router-list 1 172.16.55.1
```

Step 3 Configure the web-cache service on the Content Engine. Configure this WCCP service to use the router list created in Step 2. Enter the **l2-redirect** option to specify Layer 2 redirection as the packet-forwarding method (as opposed to GRE). Enter the **mask-assign** option to specify mask assignment as the load-balancing method (as opposed to the default hash assignment method) for this WCCP service.

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

Step 4 Display the configuration so that you can verify it.

```
ContentEngine# show wccp services detail
```

Step 5 Write the running configuration to nonvolatile memory.

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

Examples of Configuring WCCP Services for Standalone Content Engines

The section provides the following examples of how to configure WCCP services for standalone Content Engines using WCCP Version 2:

- [Example 1—Configuring the Web-Cache Service with WCCP Version 1, page 6-39](#)
- [Example 2—Configuring the Web-Cache Service with WCCP Version 2, page 6-43](#)
- [Example 3—Configuring the HTTPS Transparent Caching Service with WCCP Version 2, page 6-45](#)
- [Example 4—Configuring Multiple WCCP Version 2 Services on Standalone Content Engines, page 6-46](#)

Note the following important points when configuring the standard web-cache service (service 0) with either WCCP Version 1 or WCCP Version 2:

- The Content Engines must not have their packets encrypted or compressed and should be part of the “inside” Network Address Translation (NAT) firewall if one is present.
- A Content Engine and a WCCP-enabled router cannot be separated by a firewall. The firewall handles only packet traffic toward the origin web server and does not handle packet traffic sent to the client by the Content Engine on behalf of the server.
- Placing the Content Engine beyond a web cache redirect-enabled interface and along the route to the server will not cause the IP route cache to be populated with an entry.
- You can also use the Content Engine GUI to configure the standard web-cache service on the Content Engine. However, you must always use the CLI to configure the standard web-cache service on the router.
- To use a WCCP-enabled router to support the web cache service, an IP address must be configured on the interface connected to the Internet, and that interface must be connected to the Content Engine. Use the **show ip interface EXEC** command on the router to see whether the interfaces on the router that are configured for IP are currently usable.

The Cisco IOS software automatically enters a directly connected route in the routing table if the interface is usable. A usable interface is one through which the software can send and receive packets. If the software determines that an interface is not usable, it removes the directly connected routing entry from the routing table. Removing the entry allows the software to use dynamic routing protocols to determine backup routes to the network (if any).

If the interface can provide two-way communication, the line protocol is marked “up.” If the interface hardware is usable, the interface is marked “up.”

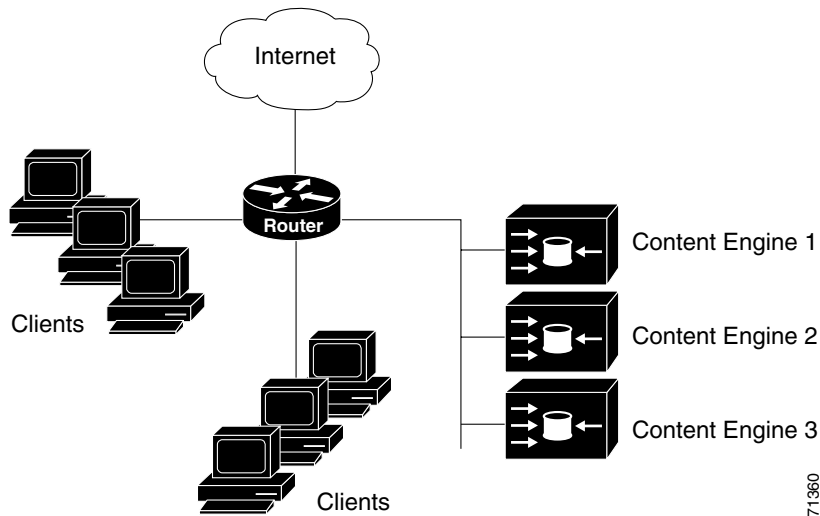
- If you specify an optional interface type, you will see information about that specific interface only.
- If you specify no optional arguments, you will see information about all of the interfaces.

When an asynchronous interface is encapsulated with Point-to-Point Protocol (PPP) or Serial Line Internet Protocol (SLIP), IP fast switching is enabled. Entering the **show ip interface EXEC** command on an asynchronous interface encapsulated with PPP or SLIP displays a message indicating that IP fast switching is enabled.

Example 1—Configuring the Web-Cache Service with WCCP Version 1

You can configure a single WCCP-enabled router and one or more Content Engines to run the standard web-cache service (service 0) using WCCP Version 1. [Figure 6-5](#) shows a sample WCCP Version 1 network configuration that consists of a cluster of three Content Engines that are being serviced by a single WCCP Version 1-enabled router.

Figure 6-5 Content Engine Network Configuration Using WCCP Version 1

**Note**

With WCCP Version 1, only a single WCCP-enabled router services a Content Engine cluster, becoming the default home router for the cluster. With WCCP Version 1, this single router that is servicing the cluster is the device that performs all the IP packet redirection.

When WCCP Version 1 is used, the following sequence of events occurs between the Content Engines and the single WCCP-enabled router (home router) that services these Content Engines:

1. Each Content Engine records the IP address of the WCCP-enabled router servicing the Content Engine cluster.
2. The Content Engines then transmit their IP addresses to the WCCP-enabled router, indicating their presence to one another in the Content Engine cluster.
3. The WCCP-enabled router then replies to the Content Engines, establishing that each can connect to others in the cluster, and providing a *view* (a list) of Content Engine addresses in the cluster, indicating that all can recognize one another.
4. Once the view has been established, one Content Engine is designated the lead and indicates to the WCCP-enabled router how IP packet redirection should be performed.

The lead Content Engine is defined as the one that has the lowest IP address in the cluster, and is seen by the WCCP Version 1-enabled router (home router) that is servicing the cluster.

The following example describes how to use the Content Engine CLI to enable and configure the standard web-cache service (service 0) using WCCP Version 1. With WCCP Version 1, you can only configure one service (the standard web-cache service) and a single WCCP-enabled router (home router). Even if there is a cluster of Content Engines, only a single WCCP Version 1-enabled router services a cluster of Content Engines, becoming the default home router for the cluster.

In this example, IP access lists are used to control which web-related packets are redirected to the standalone Content Engine. This example also shows how you can verify that the web-cache service is operating properly after it has been configured on a single router and the standalone Content Engine.

- Step 1** To use a WCCP-enabled router for the standard web-cache service, an IP address must be configured on the interface connected to the Internet, and the interface must be connected to the standalone Content Engine.

Use the **show ip interface EXEC** command on the router to check whether the interfaces on the router that are configured for IP are usable.

- Step 2** Determine whether WCCP is currently enabled on the router by entering the **show ip wccp EXEC** command.

- Step 3** Enable the router to use WCCP Version 1, and then configure the router to use WCCP Version 1 to redirect web-related packets that do not have a destination of 192.168.196.51 to the standalone Content Engine.

```
Router# configure terminal
Router(config)# ip wccp version 1
Router(config)# access-list 100 deny ip any host 192.168.196.51
Router(config)# access-list 100 permit ip any any
Router(config)# ip wccp web-cache redirect-list 100
Router(config)# interface Ethernet 0
Router(config)# ip wccp web-cache redirect out
Router(config-if)# end
Router#
%SYS-5-CONFIG_I: Configured from console by console.
```

- Step 4** Enable WCCP Version 1 on the standalone Content Engine.

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

- Step 5** Point the standalone Content Engine to the home router by entering the **wccp home-router ip-address** global configuration command. This may also be the IP address of the IP default gateway.

In the following example, the home router has an IP address of 192.168.51.102:

```
ContentEngine(config)# wccp home-router 192.168.51.102
```



Note You can also use the Content Engine GUI (choose **WCCP > Enable WCCP**) to configure the WCCP Version 1 web cache service on a standalone Content Engine. For more information about the Enable WCCP window, click the **HELP** button in the window.

- Step 6** Configure the standard web-cache service on the Content Engine, as described in the [“Configuring the Standard Web-Cache Service \(Service 0\) for Standalone Content Engines”](#) section on page 7-18.

- Step 7** Verify that the standard web-cache service is now enabled on this standalone Content Engine by entering the **show wccp EXEC** command.

```
ContentEngine# show wccp services
Services configured on this Content Engine
    Web Cache
ContentEngine#
```

You can also display other WCCP information on the standalone Content Engine, by using other options of the **show wccp EXEC** command on the Content Engine. For example, display WCCP generic routing encapsulation packet-related information on the Content Engine, by specifying the **gre** option of the **show wccp** command.

- Step 8** Verify that WCCP is enabled on the router and that the router is aware of the standalone Content Engine that you have configured as a web cache by entering the **show ip wccp web-cache** command on the WCCP-enabled router.

In the following example, the **show ip wccp web-cache** command is entered immediately after the home router has been configured. After a few seconds, the state of the standalone Content Engine that has an IP address of 192.168.25.3 changes from “NOT Usable” to “Usable,” as seen in the second output.

```
Router# show ip wccp web-cache
```

```
WCCP Web-Cache information:
  IP Address:          192.168.25.3
  Protocol Version:    1.0
  State:               NOT Usable
  Initial Hash Info:   FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF
                      FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF
  Assigned Hash Info:  00000000000000000000000000000000
                      00000000000000000000000000000000
  Hash Allotment:     0 (0.00%)
  Packets Redirected: 0
  Connect Time:       00:00:06
```

```
Router# show ip wccp web-cache
```

```
WCCP Web-Cache information:
  IP Address          192.168.25.3
  Protocol Version:   0.3
  State:              Usable
  Initial Hash Info:  FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF
                      FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF
  Assigned Hash Info: FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF
                      FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF
  Hash Allotment:    256 (100.00%)
  Packets Redirected: 0
  Connect Time:     00:00:31
```

- Step 9** On the WCCP Version 1-enabled router, enter the **debug ip wccp events EXEC** command to view information about significant WCCP events.

The following example is sample output of the **debug ip wccp events EXEC** command when a Content Engine is added to the list of available web caches for this WCCP Version 1-enabled router:

```
Router# debug ip wccp events
```

```
WCCP-EVNT: Built I_See_You msg body w/1 usable web caches, change # 0000000A
WCCP-EVNT: Web Cache 192.168.25.3 added
WCCP-EVNT: Built I_See_You msg body w/2 usable web caches, change # 0000000B
WCCP-EVNT: Built I_See_You msg body w/2 usable web caches, change # 0000000C
```

- Step 10** On the WCCP Version 1-enabled router, enter the **debug ip wccp packets** command to view information about every WCCP packet that was received or sent by this router.

The following example is sample output of the **debug ip wccp packets** command. The router is sending keepalive packets to the standalone Content Engine at 192.168.25.3. Each keepalive packet has an identification number associated with it. When the Content Engine receives a keepalive packet from the router, it sends a reply with the identification number back to the router.

```
Router# debug ip wccp packets
```

```

WCCP-PKT: Received valid Here_I_Am packet from 192.168.25.3 w/rcvd_id 00003532
WCCP-PKT: Sending I_See_You packet to 192.168.25.3 w/ rcvd_id 00003534
WCCP-PKT: Received valid Here_I_Am packet from 192.168.25.3 w/rcvd_id 00003533
WCCP-PKT: Sending I_See_You packet to 192.168.25.3 w/ rcvd_id 00003535
WCCP-PKT: Received valid Here_I_Am packet from 192.168.25.3 w/rcvd_id 00003534
WCCP-PKT: Sending I_See_You packet to 192.168.25.3 w/ rcvd_id 00003536
WCCP-PKT: Received valid Here_I_Am packet from 192.168.25.3 w/rcvd_id 00003535
WCCP-PKT: Sending I_See_You packet to 192.168.25.3 w/ rcvd_id 00003537
WCCP-PKT: Received valid Here_I_Am packet from 192.168.25.3 w/rcvd_id 00003536
WCCP-PKT: Sending I_See_You packet to 192.168.25.3 w/ rcvd_id 00003538
WCCP-PKT: Received valid Here_I_Am packet from 192.168.25.3 w/rcvd_id 00003537
WCCP-PKT: Sending I_See_You packet to 192.168.25.3 w/ rcvd_id 00003539

```



Tip To clear the router's counter for packets redirected by WCCP, enter the **clear ip wccp EXEC** command on the WCCP-enabled router.

Example 2—Configuring the Web-Cache Service with WCCP Version 2

The following example shows how to use the Content Engine CLI to configure the standard web-cache service (service 0) when the clients and the standalone Content Engine are on the same subnet and WCCP Version 2 is being used instead of WCCP Version 1:

Step 1 Ensure that WCCP Version 2 is enabled on each router that will be added to the router list in Step 3.

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

Step 2 Enter the **ip wccp web-cache password** global configuration command 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 [0-7] [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.
- 0-7 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.

Step 3 On the standalone Content Engine that you want to configure for the standard web-cache service (HTTP transparent caching using WCCP Version 2), create a router list (for example, router-list 1).

```
ContentEngine(config)# wccp router-list 1 10.10.10.1
```

Step 4 Enable the standard web-cache service on the Content Engine, and specify which WCCP Version 2-enabled routers (router list) will support this particular service for this Content Engine (that is, will redirect HTTP requests to this Content Engine on port 80).

Associate this WCCP service with the router list you just created. Assign the Layer 2 redirection option. If the mask assignment method is not specified, the default load-balancing method is the hash assignment method.

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

**Tip**

You can also use the Content Engine GUI (choose **WCCP > Web Cache** from the Content Engine GUI) to configure the WCCP Version 2 web cache service on a standalone Content Engine. If you use the Content Engine GUI to configure WCCP Version 2 on a Content Engine, then you must specify the designated router list for each WCCP service in each of the following Content Engine GUI windows: the Web Cache window (**WCCP > Web Cache**), the Reverse Proxy window (**WCCP > Reverse Proxy**), the Custom Web Cache window (**WCCP > Custom Web Cache**), and the WCCP Services window (**WCCP > Services**).

Step 5 Enable WCCP Version 2 on the Content Engine.

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

Step 6 Exit global configuration mode.

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

Step 7 Write the running configurations to nonvolatile memory on the Content Engine.

```
ContentEngine# write memory
```

Step 8 Now that the router has been configured to run WCCP Version 2, monitor WCCP on the router.

```
Router# show ip wccp
```

or

```
Router# show ip wccp {web-cache | 90-99}
```

Step 9 Query the WCCP-enabled router for information about the Content Engines that the router has detected in a specific service group. The information can be displayed for WCCP services ranging in value from 90 to 99.

```
Router# show ip wccp {web-cache | 90-99} detail
```

Step 10 Determine whether any **ip wccp direct** commands are configured on an interface.

```
Router# show ip interface
```

Step 11 Display which devices in a particular WCCP service group were detected and which Content Engines are not visible to all other routers to which the current router is connected. The information can be displayed for service groups 90 to 99.

```
Router# show ip wccp {web-cache | 90-99} view
```

Example 3—Configuring the HTTPS Transparent Caching Service with WCCP Version 2

The following example shows how to use the Content Engine CLI to configure a user-defined WCCP Version 2 service and HTTPS transparent caching on a standalone Content Engine. In this example, service number 95 is the user-defined service.

- Step 1** Configure service 95 so that the Content Engine will accept HTTPS requests that are being transparently intercepted and redirected to it by multiple WCCP Version 2-enabled routers on ports other than the default port.

You must first create a router list and a port list that service 95 is to use (for example, create port list 1 for service 95). The port list contains the port numbers that the WCCP Version 2-enabled router will support WCCP redirection for service 95.

```
ContentEngine# configure terminal
ContentEngine(config)# wccp service-number 95 router-list-num 1
ContentEngine(config)# wccp service-number 95 router-list-num 1 port-list-num 1
```

- Step 2** When configuring service 95, you must specify whether the traffic is to be redirected to the caching application, the HTTPS caching application, or the streaming application on the Content Engine.

```
ContentEngine(config)# wccp service-number 95 router-list-num 1 port-list-num 1
application ?
  cache          Direct traffic to the caching application
  https-cache    Direct traffic to the HTTPS caching application
  streaming       Direct traffic to the streaming media application
```

- Step 3** Specify that the Content Engine is to accept traffic that is redirected to its HTTPS application. The WCCP Version 2-enabled routers on router list number 1 will redirect HTTPS traffic to the HTTPS caching application on the Content Engine. The Content Engine will listen for such WCCP redirected requests on the ports that are specified in port list 1.

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

In the ACNS 5.2.1 software and later releases, the accept-all mode is supported for the https-cache service. The accept-all mode supports the filtering of HTTPS traffic. This mode works the same way as the traditional WCCP services (for example, the web-cache service that intercepts all web traffic by default).

By default, the Content Engine accepts all HTTPS traffic.

```
ContentEngine(config)# wccp https-cache ?
  accept-all    Accept all HTTPS traffic by default
  mask           Specify mask used for CE assignment
  router-list-num Router list number
```



Note If the `wccp https-cache accept-all` global configuration command is used, the HTTPS cache (the Content Engine that is configured for HTTPS transparent caching) will work in accept all mode (it will intercept all HTTPS traffic); otherwise, the HTTPS cache works in accept only mode as in the ACNS 5.1.x software.

Step 4 Enable WCCP Version 2 on the Content Engine.

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

Example 4—Configuring Multiple WCCP Version 2 Services on Standalone Content Engines

The following example shows how to use the Content Engine CLI to configure 16 WCCP Version 2 services on a standalone Content Engine:

Step 1 Configure a router list that lists the WCCP Version 2-enabled routers that will support the 16 WCCP Version 2 services (eight user-defined services and eight predefined services). In this case, router list 1 has only a single router (the WCCP Version 2-enabled router with an IP address of 10.1.202.1)

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

Step 2 Configure eight port lists (port lists number 1 through 8).

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 port 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
```

Step 3 Enable the standard web-cache service (service 0) on the Content Engine, and associate router list 1 with this first predefined WCCP service.

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

The Content Engine will listen on port 80 for redirected HTTP requests from the routers on router list 1.



Note The term *HTTP requests* is used to refer collectively to HTTP, FTP-over-HTTP, and HTTPS-over-HTTP requests.

Step 4 Enable the reverse-proxy caching service (service 99) on the Content Engine, and associate router list 1 with this second predefined WCCP Version 2 service.

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

The Content Engine will listen on port 80 for redirected reverse proxy requests from the routers on router list 1.

- Step 5** Enable the custom-web-cache service (service 98) on the Content Engine, and associate router list 1 and port 31 with this third predefined WCCP Version 2 service.

```
ContentEngine(config)# wccp custom-web-cache router-list-num 1 port 31
```

The Content Engine will listen on port 31 for redirected HTTP requests from the WCCP Version 2-enabled routers on router list 1.

- Step 6** Enable the rtsp service (service 80) on the Content Engine, and associate router list 1 with this fourth predefined WCCP Version 2 service.

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

The Content Engine will listen on the standard RTSP port (default port 554) for redirected RTSP requests from the WCCP Version 2-enabled routers on router list 1.

- Step 7** Enable the WMT services (services 81 and 82) on the Content Engine, and associate router list 1 with this fifth predefined WCCP Version 2 service.

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

After specifying this command, the Content Engine will listen on the default port (port 1755) for redirected WMT requests from the WCCP Version 2-enabled routers on router list 1.

- Step 8** Enable the dns caching service (service 53) on the Content Engine, and associate router list 1 with this sixth predefined WCCP Version 2 service.

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

The Content Engine will listen on port 80 for redirected DNS requests from the WCCP Version 2-enabled routers on router list 1.

- Step 9** Enable the ftp-native caching service (service 60) on the Content Engine, and associate router list 1 with this seventh predefined WCCP Version 2 service.

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

The Content Engine will listen on port 80 for redirected FTP native requests from the WCCP Version 2-enabled routers on router list 1. This is for FTP native caching (as opposed to FTP-over-HTTP caching that is involved when the Content Engine receives FTP-over-HTTP requests directly from a client browser and caches the requested content).

- Step 10** Enable the https-caching service (service 70) on the Content Engine, and associate router list 1 with this eighth predefined WCCP Version 2 service.

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

In the ACNS 5.1 software, HTTPS requests could only be SSL-terminated on the Content Engine in WCCP mode. In the ACNS 5.1 software, only HTTPS requests to specific sites (HTTPS origin servers that the Content Engine was specifically configured to support) were SSL-terminated in WCCP mode. In the ACNS 5.1 software, the Content Engine would bypass HTTPS requests that were directed to HTTPS servers that it had not been explicitly configured to support. For more information about SSL termination, see the [“About SSL Termination of HTTPS Client Requests” section on page 7-25](#).

In the ACNS 5.1.x software, only one interception mode (the accept-only mode) was supported for the https-cache service. With the accept-only mode, you had to configure the Content Engine to accept only redirected requests that were directed to specific HTTPS servers, as follows:

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

or

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

In both of the preceding examples, the Content Engine will only accept the redirected HTTPS traffic if the HTTPS server is configured on the Content Engine (using the **https server** global configuration command).

In the ACNS 5.2.1 software and later releases, the Content Engine SSL terminates HTTPS requests in WCCP mode and in manual proxy mode if the requested HTTPS servers are configured on the Content Engine, and tunnels the rest of the HTTPS traffic. For more information about tunneling of HTTPS requests, see the [“About Tunneling of HTTPS Client Requests” section on page 7-25](#). For specific requested content to be cached, you must import the proper certificates and keys for these HTTPS servers into the Content Engine and configure the Content Engine to cache these servers. For standalone Content Engines, this is performed through the Content Engine CLI, as described in the [“Configuring Certificates and Private Keys for HTTPS Caching” section on page 7-32](#).

- Step 11** 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. Specify 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
```

The Content Engine will 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 example because you want the WCCP routers to redirect the traffic to the caching application (as opposed to the streaming application) on the Content Engine.

- Step 12** 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
```

The Content Engine will 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.

- Step 13** 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
```

The Content Engine will listen on the ports listed in port list 3 (port 34) for redirected requests from the routers on router list 1.

- Step 14** 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
```

The Content Engine will listen on the ports listed in port list 4 (port 35) for redirected requests from the routers in router list 1.

- Step 15** 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
```

The Content Engine will listen on the ports listed in port list 5 (port 36) for redirected requests from the routers on router list 1.

- Step 16** 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
```

The Content Engine will listen on the ports listed in port list 6 (port 37) for redirected requests from the routers on router list 1.

- Step 17** 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
```

The Content Engine will listen on the ports listed in port list 7 (port 38) for redirected requests from the routers on router list 1.

- Step 18** 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
```

The Content Engine will listen on the ports listed in port list 8 (port 39) for redirected requests from the routers on router list 1.

- Step 19** Enable WCCP Version 2 on the Content Engine.

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

- Step 20** Disable the WCCP slow start feature on the Content Engine.

```
ContentEngine(config)# no wccp slow-start enable
```

For information about the WCCP slow start feature, see the [“Configuring WCCP Slow Start” section on page 15-10](#).

Configuring Layer 4 Switching as a Redirection Method

To configure transparent redirection when Layer 4 switching (a Content Services Switch [CSS] switch) is being used to redirect requests transparently to a standalone Content Engine, keep these important points in mind:

- The CSS switch supports transparent proxy caching as well as reverse proxy caching. The CSS switch provides several load-balancing methods depending on how you want to distribute data over the Content Engines (for example, entire URL, URL string, entire domain name, or domain string). The CSS switch also builds a list of known cacheable objects. The list may be modified, but much of the work is reduced by the Content Engine caching capabilities.
- You can configure the CSS switch to dynamically analyze the content and determine if it is cacheable or not. If it is cacheable, the CSS switch directs it to the cache service. If it is not cacheable, the CSS switch sends it directly to the origin web server.
- If all cache servers are unavailable in a transparent cache configuration, the CSS switch allows all client requests to progress to the origin web servers.

To configure transparent caching with a CSS switch, complete the following tasks:

1. Enable transparent caching on the CSS switch.
2. Enable the standalone Content Engine to accept redirected traffic from the CSS switch.
3. Enable transparent caching on the standalone Content Engine.

The following sample workflow shows how to use CLI commands to configure transparent caching using a CSS switch and a standalone Content Engine. In this example, `serv1` is configured as a transparent caching service using a CSS switch named `CS150` and a standalone Content Engine named `CE100`. Ensure that you have configured interfaces, services, owners, VLANs and content rules prior to configuring caching with the CSS switch.



Note

Refer to the *Content Services Switch Basic Configuration Guide* for further information on how to configure these attributes on the CSS switch. For a complete description of each command, refer to the *Content Services Switch Command Reference*.

Example of Configuring Transparent Caching Using Layer 4 Switching

The following example shows how to enable transparent caching using a Layer 4 CSS switch (as opposed to a WCCP-enabled router) and a standalone Content Engine. To enable transparent caching, follow these steps:

Step 1 On the CCS switch, add service `serv1` reserved for transparent caching.

```
CS150 (config) # add service serv1
CS150 (config-service[serv1]) #
```

Step 2 Specify transparent caching as the service type for `serv1`.

```
CS150 (config-service[serv1]) # type transparent cache
```

Step 3 Create an extension qualifier list (EQL) in which you specify which content types the CSS switch is to cache.

```
CS150 (config) # eql graphics
CS150 (config-eql[graphics]) #
```

- Step 4** Describe the EQL by entering a quoted text string with a maximum length of 64 characters.
- ```
CS150(config-eql[graphics])# description "This EQL specifies cacheable graphic files"
```
- Step 5** Specify the extension for content that you want the CSS switch to cache. Enter a text string containing from 1 to 8 characters.
- ```
CS150(config-eql[graphics])# extension jpeg
```
- You can also provide a description of the extension type here. Enter a text string enclosed by quotation marks. The maximum length is 64 characters.
- ```
CS150(config-eql[graphics])# extension jpeg "This is a graphics file"
CS150(config-eql[graphics])# exit
CS150(config)#
```
- Step 6** Specify the EQL in a content rule to match all content requests with the desired extension.
- ```
CS150(config-owner-content[cisco.com-rule1])# url "/" eql graphics
```
- Step 7** Configure the load-balancing method for the cache content rule. The default is round-robin.
- ```
CS150(config-owner-content[cisco.com-rule1])# balance domain
```
- Step 8** Specify a failover type (**bypass**, **linear**, **next**) to define how the CSS switch handles content requests when a service fails. The default is linear.
- ```
CS150(config-owner-content[cisco.com-rule1])# failover bypass
```
- Step 9** Display the EQL configuration.
- ```
CS150(config-owner-content[cisco.com-rule1])# show eql
```
- Step 10** Display the content rule to show the cache configuration.
- ```
CS150(config-owner-content[cisco.com-rule1])# show rule
```
- Step 11** Exit configuration mode on the CSS switch.
- ```
CS150(config-owner-content[cisco.com-rule1])# end
```
- Step 12** Save the configuration. The CSS switch is now configured for transparent caching services.
- ```
CS150(config-owner-content[cisco.com-rule1])# copy running-config startup-config
```
- Step 13** Configure the standalone Content Engine to transparently receive Layer 4 redirected traffic from Layer 4-enabled switches such as the CSS switch.
- ```
CE100(config)# http 14-switch enable
```



**Note** The **http 14-switch** command enables transparent redirection on HTTP port 80 only. If you want to intercept traffic on a different port, you must configure a WCCP service without a router, that contains the alternative port that you want to use.

If the version of ACNS that you are using does not accept the **wccp** command without the **router-list num** option, you can use a dummy router address. The Content Engine will accept redirected traffic and will send WCCP announcements to the configured router. You can avoid having the Content Engine send WCCP announcements to the configured router by later removing the dummy router list from the configured WCCP service; however, this configuration will be lost after you reload.

**Step 14** Exit configuration mode on the standalone Content Engine.

```
CE100(config)# exit
```

**Step 15** Write the running configuration to nonvolatile memory.

```
CE100# write memory
```

---