



# Configuring RealMedia Services on Standalone Content Engines

---

This chapter describes how to configure RealMedia streaming and caching services on standalone Content Engines, including how to configure the Real-Time Streaming Protocol (RTSP) gateway that runs on the Content Engine. For information about how to configure Windows Media Technologies (WMT) streaming and caching services (WMT RTSP streaming and caching services) on standalone Content Engines, see [Chapter 9, “Configuring WMT Streaming Media Services on Standalone Content Engines.”](#)

This chapter contains the following sections:

- [Overview of the RealMedia Streaming Solution, page 8-2](#)
- [Configuring RealMedia Services, page 8-9](#)
- [Verifying RealProxy Configurations for Standalone Content Engines, page 8-23](#)
- [Restoring the RealProxy Factory-Default Configuration on Standalone Content Engines, page 8-29](#)
- [Restarting RealProxy on Standalone Content Engines, page 8-30](#)
- [Disabling RealMedia Caching on Standalone Content Engines, page 8-30](#)
- [Uninstalling the RealProxy License Key, page 8-31](#)
- [Displaying RealProxy Statistics for Standalone Content Engines, page 8-31](#)

For background information about streaming media services, see the [“Understanding Some Basic ACNS Streaming Media Concepts” section on page 2-10](#). For complete syntax and usage information for the CLI commands used in this chapter, see the *Cisco ACNS Software Command Reference, Release 5.5* publication. For information about how to configure streaming media services for Content Engines that are registered with a Content Distribution Manager, see the *Cisco ACNS Software Configuration Guide for Centrally Managed Deployments, Release 5.5*.



## Note

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

---

# Overview of the RealMedia Streaming Solution

RealMedia is the streaming media solution from RealNetworks, Inc. RealMedia uses the RealNetworks RTSP protocol (IETF standard RTSP protocol plus proprietary extensions) to deliver streaming media content to RealMedia clients (for example, RealPlayer 8.0 and RealOne players).

RealMedia has two main software components: RealProxy and RealSubscriber. With registered Content Engines, you can enable and run RealProxy and RealSubscriber; however, standalone Content Engines only support RealProxy. You cannot enable and run RealSubscriber on a standalone Content Engine.

The RealProxy software from RealNetworks, Inc., included as a software option in the ACNS 5.x software allows you to deploy the following RealMedia services on standalone Content Engines:

- Live splitting (distributing live feeds)
- Streaming video-on-demand (VOD) files in an RTSP-based format
- Caching VOD files

Table 8-1 describes RealMedia supported services for standalone Content Engines.

**Table 8-1 RealMedia Streaming and Caching Services Supported for Standalone Content Engines**

Service	Description
RealMedia proxy caching of VOD files	The standalone Content Engine is functioning as a nontransparent proxy server for end users who are using a RealMedia player to request content. After receiving an RTSP request directly from a RealMedia player, the Content Engine retrieves the requested VOD file if it is not already stored in its local cache, stores a copy locally, and sends the requested content to the RealMedia player. See the <a href="#">“Configuring Direct Proxy Routing and RealMedia Proxy Caching”</a> section on page 8-21.
RealMedia transparent caching of VOD files	The standalone Content Engine is functioning as a transparent proxy server for end users who are using a RealMedia player to request content. After receiving a redirected RTSP request from a WCCP Version 2 router or Layer 4 switch, the Content Engine retrieves the requested content, stores a copy locally, and sends the requested content to the RealMedia player. See the <a href="#">“Configuring RTSP Transparent Redirection and Caching of RealMedia Requests”</a> section on page 8-17.
RealProxy live splitting	The Content Engine serves RTSP live streams to all local users (RealMedia players) whose requests are directed to it. The RTSP live streams can be unicast live feeds or multicast live feeds. The Content Engine splits the live feeds into a multicast or unicast to relay the stream to the RealMedia client. Live streams are not files so they cannot be cached but VOD files can be cached. See the <a href="#">“About Live Splitting and Caching VOD Files with RealProxy”</a> section on page 8-6.



**Note**

In the ACNS 5.2.1 software release, support for Synchronized Multimedia Integration Language (SMIL) files in RealProxy was added. For more information, see the [“About RealProxy SMIL File Support”](#) section on page 8-8.

Table 8-2 describes the RealProxy features and benefits for standalone Content Engines.

**Table 8-2 RealProxy Features and Benefits**

RealProxy Feature	Description	Benefits
Proxy for RealMedia players (for example, RealPlayer 8.0 or RealOne players)	RealProxy makes requests for content on behalf of the RealMedia clients.	Manages traffic inside the firewall by coordinating requests for similar content. Masks end user IP addresses.
Splitting support for live broadcasts	RealProxy splits a single inbound live broadcast feed to multiple RealMedia clients.	Reduces inbound bandwidth usage to a single stream of content during a live event.
Caching of RealSystem G2 and Progressive Network Audio (PNA) content	RealProxy caches all proxied streaming media traffic from RealNetworks servers. RealProxy caches content locally after authentication with the origin streaming server.	Significantly reduces inbound bandwidth usage by eliminating redundant file transmissions across the network.
Authentication and accounting	RealProxy authenticates every content request with the origin streaming server before delivering the cached content to the clients.	Retains access to general usage data for the broadcaster. Appropriately authenticates users. Guarantees the freshest content for end users.
Aggregate bandwidth thresholds	RealProxy thresholds allow you to specify the maximum bandwidth for inbound and outbound RTSP traffic (cached content and live content).	Provides control over aggregate bandwidth usage within the network and prevents stress on mission-critical applications.
Proxy routing	RealProxy can tier proxies and manage bandwidth at lower nodes in the network. Parent proxies can be chosen based on logical sets of rules on the downstream proxy.	Allows you to proxy route requests, providing an additional level of control.

The Content Engine can be configured to accept transparently redirected content requests as well as traditional proxy-style requests from RealMedia players. The redirection of RTSP traffic to the Content Engine media cache is enabled through the Content Engine GUI or CLI. The RealProxy software is configured with the RealSystem administrator GUI, accessed from the RealProxy window of the Content Engine GUI. (See the [“Configuring RealProxy with the RealSystem Administrator GUI” section on page 8-21](#).)

The RealProxy feature on a standalone Content Engine is licensed software. For more information on this topic, see the [“About the RealProxy License Key” section on page 8-9](#).

## About the RTSP Gateway and Backend RTSP Servers

The RTSP gateway is the single point of entry for RTSP messages. The RTSP gateway is automatically enabled and runs on the standalone Content Engine. The RTSP gateway listens on the standard RTSP port (default port 554) and funnels incoming RTSP traffic through it to enforce rules-based implementation and URL filtering of RTSP requests.

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

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

The possible actions that the RTSP gateway can take on an RTSP streaming request are as follows:

- Tunnels the request to the RealProxy server, an RTSP backend server that is running on the same Content Engine as the RTSP gateway.
- Transfers the socket fd to the Windows Media server that is running on the same Content Engine as the RTSP gateway.

If the user-agent is Windows Media Player, then instead of tunneling the RTSP request to the Windows Media server (the `mms_server` process), the RTSP gateway transfers the socket fd (kick fd) to the Windows Media server that is running on the Content Engine. This method allows Windows Media clients to communicate directly with the Windows Media server for subsequent RTSP requests, which eliminates the RTSP gateway process from subsequent RTSP request flows.

- Redirects the request

The RTSP gateway checks if the request matches any of the configured rules. Rules may decide if a particular request does not need to go through authentication and authorization. If the RTSP request is successfully authenticated and authorized, URL filtering is performed on the request. If the request is a WCCP-transparently redirected request and the Content Engine cannot handle the redirect, the RTSP gateway redirects the request and adds bypass entries.

- Rejects the request



---

**Note** The RTSP gateway supports local list URL filtering (good site and bad site lists) for RTSP requests. URL filtering with third-party software (for example, SmartFilter, Websense, and N2H2) is not supported for RTSP requests.

---

If the RTSP request is blocked as a result of URL filtering, then the client receives a 403 Forbidden error message. Otherwise, the RTSP gateway sends the RTSP request to the appropriate backend RTSP server based on the properties of the incoming request, including such properties as the client player, final destination, and media file type. The RTSP gateway encodes the URL so that the backend RTSP servers know the original URL before unified name space (UNS) translation.

On standalone Content Engines that are running the ACNS 5.2 software and earlier releases, the RealProxy server is the only backend RTSP server that can be enabled on the Content Engine. In the ACNS 5.3.1 software and later releases, you can enable the RealProxy server or the WMT RTSP server as backend RTSP servers on standalone Content Engines.

Table 8-3 lists the supported backend RTSP servers for standalone Content Engines that are running the ACNS 5.3.1 software and later releases.

**Table 8-3 Backend RTSP Servers Supported on Standalone Content Engines**

RTSP Streaming Solution	RTSP Backend Servers	RTSP-Based Clients	Protocol Used to Service RTSP-based Requests
RealMedia from RealNetworks, Inc.	RealProxy server	RealMedia players	RealProxy server uses RealNetworks RTSP proprietary protocol over RTP to stream RTSP content to RealMedia players.
Windows Media 9 Series from Microsoft Corporation	Windows Media 9 RTSP server (WMT RTSP server)	Windows Media 9 players	WMT RTSP server uses the IETF standard RTSP protocol (plus proprietary Microsoft extensions) to stream content to Windows Media 9 players.



**Note**

For Content Engines that are registered with a Content Distribution Manager, you can also enable RealSubscriber and Cisco Streaming Engine as backend RTSP servers that run on the Content Engine. For information about how to configure Cisco Streaming Engine and RealSubscriber for registered Content Engines, see the *Cisco ACNS Software Configuration Guide for Centrally Managed Deployments, Release 5.5*.

The following sample output from the **show rtsp all** EXEC command shows a sample configuration for Content Engine 24, a standalone Content Engine that is has been configured to use the RTSP gateway and two RTSP backend servers (the WMT RTSP server and the RealProxy server) to service RTSP requests from RealMedia and Windows Media 9 players:

```
ContentEngine 24# show rtsp all

RTSP Gateway Configuration
-----
RTSP Gateway ip-address 209.165.202.128
RTSP Gateway incoming port 554
RTSP Gateway incoming request rate limit is 40 requests/sec
RTSP Gateway initial setup delay is 10 sec
RTSP Gateway L4-switch is enabled
RTSP Gateway Transparent Interception (WCCP):
    Not configured.

WMT RTSP Server/Proxy Configuration
-----
WMT version: ce507-001.000
WMT license key is installed
WMT evaluation is not enabled
WMT end user license agreement accepted
WMT is enabled
WMT disallowed client protocols: none
WMT outgoing bandwidth configured is 1 Kbits/sec
WMT incoming bandwidth configured is 56000 Kbits/sec
WMT max sessions configured: 2500
WMT max sessions platform limit: 2500
WMT max sessions enforced: 2500 sessions
WMT max outgoing bit rate allowed per stream: 2 Kbits/sec
WMT max incoming bit rate allowed per stream: 3 Kbits/sec
WMT debug level: 0
WMT L4 switch is enabled
WMT debug client ip not set
```

```

WMT debug server ip not set
WMT fast-start is enabled
WMT fast-start max. bandwidth per player is 65 (Kbps)
WMT fast-cache is enabled
WMT fast-cache acceleration factor is 5
WMT Extended Transaction Log is not enabled
WMT Transaction Log format is Windows Media Services 4.1 logging

```

```

Real Proxy Configurations
-----
Real Proxy version: ce507-9.0.2.794
Real Proxy is not enabled
Real Proxy evaluation is not enabled
Real Proxy license key not installed
Real Proxy end user license agreement accepted
Real Proxy is configured to use 30% of MEDIAFS partition
Real Proxy incoming bandwidth enforced is 0 kbps
Real Proxy outgoing bandwidth enforced is 0 kbps

```

**Note**


---

For registered Content Engines, the command output from the **show rtsp server EXEC** command would also include configuration information about RealSubscriber and the Cisco Streaming Engine (two additional RTSP backend servers that can be enabled on a registered Content Engine).

---

Each backend RTSP server that is running on the Content Engine performs its own transaction logging. For example, RealProxy uses RealProxy transaction logs while the WMT RTSP server uses the WMT transaction logs. By default, transaction logging is enabled on a Content Engine.

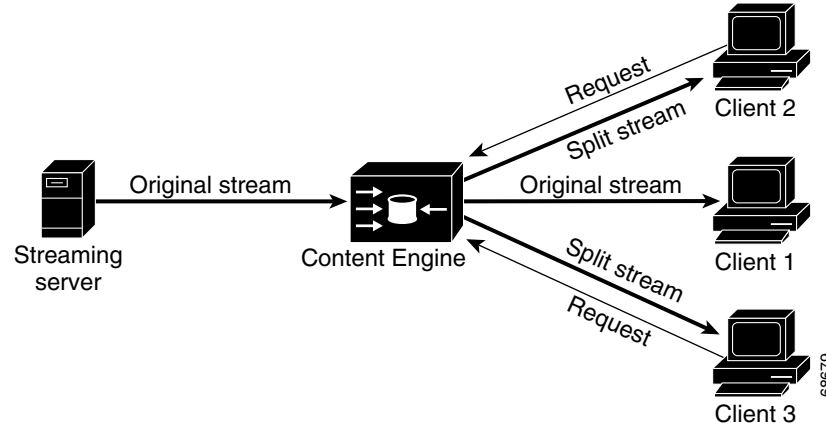
For more information about the RTSP gateway, see the [“Configuring the RTSP Gateway for Standalone Content Engines” section on page 8-14](#). For information about how to configure the WMT RTSP server and WMT RTSP services, see the [“Configuring WMT RTSP Streaming and Caching Services on Standalone Content Engines” section on page 9-14](#).

## About Live Splitting and Caching VOD Files with RealProxy

When RealProxy is enabled on a standalone Content Engine, the Content Engine will serve as the stream splitting point for all local users (RealMedia players) whose RealMedia requests is directed to the Content Engine. All subsequent requests to that origin streaming server (the Helix Universal Server) are served by the Content Engine, which splits the stream and serves it to the RealMedia clients.

[Figure 8-1](#) shows how the Content Engine splits one unicast live stream into multiple unicast streams in the local network. When the first client (Client 1) that requested the original stream disconnects from the network, the Content Engine continues to serve the other clients (Client 2 and Client 3) until all clients disconnect from the network.

**Figure 8-1** How a Standalone Content Engine Supports Live Splitting



By having the Content Engine perform the live splitting, you potentially save considerable network bandwidth between the client and the origin streaming server, because the Content Engine is closer to the clients.



**Note**

RealProxy cannot cache live broadcasts (also referred to as *live clips* and *live streams*) because there is no actual downloadable file to cache. However, RealProxy supports live splitting in order to allow RealMedia clients to share the broadcast, which saves WAN bandwidth.

When the Content Engine receives a request for a RealMedia VOD file (.rm file) that is not already stored in its local cache, the Content Engine retrieves the requested VOD file from the origin streaming server (the Helix Universal Server), caches the VOD file if RealMedia caching is enabled on the Content Engine, and streams the requested VOD file to the RealMedia client.

In both operations (streaming of VOD files and live splitting), the RealMedia request can be directed to the Content Engine through one of two methods:

- Transparent redirection through WCCP Version 2 or Layer 4 switching
- Direct proxy routing (explicitly configuring the RealMedia player proxy settings)

Configuration for both of these operations is the same on the Content Engine and routers; the only difference is the source (a VOD file or a live stream).

All incoming RTSP requests are directed to the RTSP gateway on the Content Engine. The RTSP gateway decides which backend RTSP server on the Content Engine (for example, RealProxy) to direct the request to. For more information about the RTSP gateway, see the [“Configuring the RTSP Gateway for Standalone Content Engines”](#) section on page 8-14.

For more information about RealMedia caching for standalone Content Engines, see the [“Configuring RealMedia Services”](#) section on page 8-9.

## About Caching Policies in RealMedia Caching

In contrast to HTTP caching, caching policies in RealMedia caching are much simpler, because streaming media are mostly large static content. All responses other than live streams are cacheable (VOD files can be cached), including partial responses. All RealMedia requests result in communication between the Content Engine and the origin streaming server (the Helix Universal Server), even if the request is a cache hit.

By establishing the streaming control session, the Content Engine can verify that its cached content is fresh, and the client can access the content. Because streaming objects are typically very large in size, the overhead of establishing the control session with the server is minimal and does not reduce the bandwidth savings from the cache hits.

**Note**

---

Live streams are not cached.

---

## About Access Control

If a RealMedia client requests a cached stream, before the client is allowed to play the stream, the RealProxy server that is running on the Content Engine uses the RealNetworks proprietary RTSP protocol to send the request to the origin streaming server (Helix Universal Server) for permission. If the origin streaming server denies the request, the RealMedia client is not allowed to receive the requested stream.

## About RealProxy SMIL File Support

SMIL is a simple but powerful markup language that allows you to coordinate multiple clips. SMIL also allows you to define how, when, and where you want the multiple clips to be played.

A client browser can automatically launch the media player on the client desktop when the video and presentation material is packaged in a media-index file such as an SMIL file. The browser is typically configured so that the moment it retrieves an SMIL file, it automatically launches the RealMedia player on the client desktop, and passes the SMIL file to the client RealMedia player. Media-index files can contain either relative links to media files or absolute links to media files.

**Note**

---

The .asx file is used by the WMT streaming solution, and the SMIL file is used by the RealNetwork streaming solution. For more background information on media-index files, see the [“How a Client Media Player Issues a Request”](#) section on page 2-14.

---

SMIL files are used for the following main purposes:

- To describe the overall layout of the SMIL-based presentation.
- To act as the macro meta file for the SMIL-based presentation.

Rather than encoding presentations in a single file, SMIL allows content creators to encode pieces of the presentation in separate files and then use SMIL to control the interaction of these separate files. The SMIL file points to the source of the media and data files, as well as the source of the more specific meta files that comprise the SMIL-based presentation.

- To establish the overall timeline of the SMIL-based presentation.

With a SMIL-based presentation, each element in the presentation can be encoded and transmitted separately, with synchronization control. Consequently, content creators can optimize the delivery of their SMIL-based presentations by specifying the least bandwidth-intensive format for transmittal. In addition to reducing the bandwidth requirements, SMIL also expedites the process of editing an SMIL-based presentation after the presentation is completed. For example, a content creator can use SMIL to easily delay the audio track of a completed SMIL-based presentation (for example, have the audio not start for 5 seconds after the presentation begins) without having to edit the actual audio file.

In the ACNS 5.2.1 software and later releases, SMIL file support for RealProxy is supported. SMIL support is provided under the following conditions:

- Case 1—The SMIL file and its contents are pre-positioned on a registered Content Engine.
- Case 2—The SMIL file and its contents are not pre-positioned on the Content Engine.
- Case 3—The SMIL files have absolute URLs, and each URL is pointing to a different server.

All of the three cases are supported on Content Engines that meet the following requirements:

- The Content Engine is running the ACNS 5.2 software and later releases.
- RealProxy is enabled on the Content Engine.
- The Content Engine is registered with a Content Distribution Manager.

Only Cases 2 and 3 are supported on standalone Content Engines (Content Engines that are not registered with a Content Distribution Manager) that meet the following requirements:

- The Content Engine is running the ACNS 5.2.1 software and later releases.
- RealProxy is enabled on the Content Engine.

## About the RealProxy License Key

The RealNetworks, Inc. RealProxy product is licensed software. To activate the licensed RealProxy feature on a standalone Content Engine, you must have a RealProxy license key. You must specify a permanent RealProxy license key that is supplied on a certificate shipped with the Content Engine, or use an evaluation key for a temporary period. If you are downloading the ACNS 5.x software, you can purchase a RealProxy license through the Cisco.com website. You specify the RealProxy license key as part of enabling the RealProxy feature on a standalone Content Engine. See the [“Enabling RealProxy on Standalone Content Engines”](#) section on page 8-13.

## Configuring RealMedia Services

The Content Engine can be configured to accept transparently redirected RTSP requests as well as traditional proxy-style RTSP requests from RealMedia players. RealProxy also supports live splitting and caching of RealMedia VOD files (.rm files).

The Setup utility allows you to enable the licensed RealProxy feature on a standalone Content Engine that is running the ACNS 5.2.1 software and later releases, and then enable RealMedia proxy caching and RealMedia transparent caching on the Content Engine. With the Setup utility, you can configure the Content Engine to accept redirected RTSP requests from a WCCP Version 2-enabled router. With the Content Engine CLI, you can configure the Content Engine to accept redirected RTSP requests from a WCCP Version 2-enabled router or a Layer 4 switch. RealProxy is configured with the RealSystem administrator GUI, which is accessed from the RealProxy window of the Content Engine GUI. (For information about access the RealSystem administrator GUI, see the [“Configuring RealProxy with the RealSystem Administrator GUI”](#) section on page 8-21.)

**Note**

For information about how to use the Setup utility to enable RealMedia caching on a standalone Content Engine, see the [“Configuring a Basic Configuration on Standalone Content Engines with the Setup Utility”](#) section on page 4-10.

This section describes how to configure RealMedia streaming and caching services for a standalone Content Engine through the Content Engine CLI and the RealSystem Administrator GUI. The following topics are described in this section:

- [Enabling RealProxy on Standalone Content Engines](#), page 8-13
- [Configuring the RTSP Gateway for Standalone Content Engines](#), page 8-14
- [Configuring RTSP Transparent Redirection and Caching of RealMedia Requests](#), page 8-17
- [Configuring Direct Proxy Routing and RealMedia Proxy Caching](#), page 8-21
- [Configuring RealProxy with the RealSystem Administrator GUI](#), page 8-21

When configuring RealMedia streaming and caching services with standalone Content Engines, note the following important points:

- In order to support RealMedia transparent caching, WCCP Version 2 must be running on the standalone Content Engine. WCCP Version 2 must also be running on the routers that will be redirecting RTSP requests to the Content Engine.
- You must configure disk space to include mediafs storage with the **disk config** command before you can run cache streaming media using RealProxy.

The mediafs partitions is mounted on the standalone Content Engine. This is the storage partition that is used to store any RTSP streaming media content that is cached on the Content Engine.

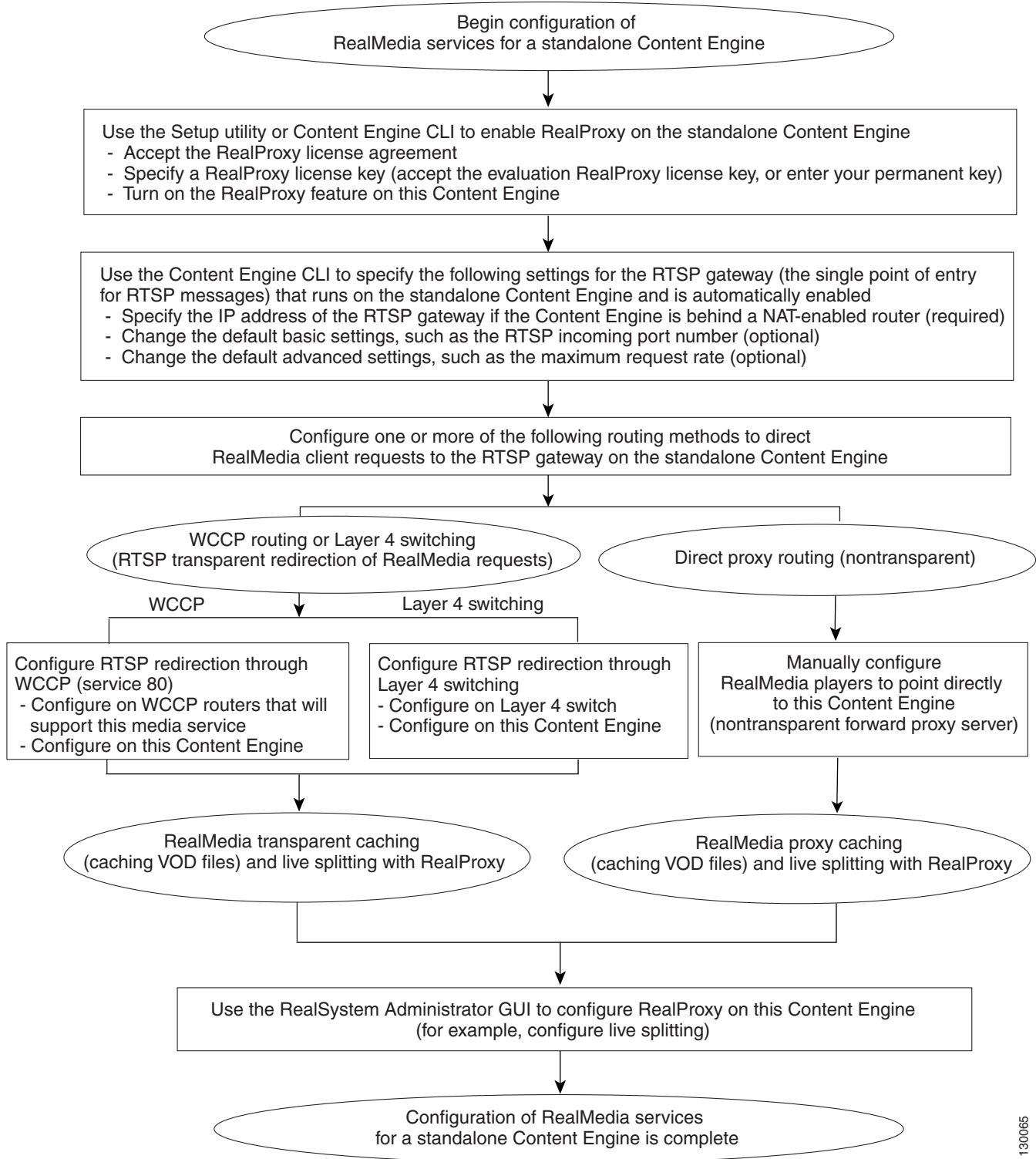
- You have the IP address of the WCCP Version 2-enabled routers if you want to use transparent WCCP redirection.
- You have the IP address of the standalone Content Engine that you want to enable and run RealProxy on.
- You have a RealProxy license key. For information about the RealProxy license, see the [“About Live Splitting and Caching VOD Files with RealProxy”](#) section on page 8-6.
- Live broadcasts are live streams and not files, and therefore cannot be cached. However, RealProxy can split live broadcast (live splitting) to conserve network bandwidth. For more information about live splitting, see the [“About Live Splitting and Caching VOD Files with RealProxy”](#) section on page 8-6.

**Note**

Content Engines that use a Content Service Switch (CSS) to load balance streaming traffic cannot stream UDP traffic (such as RTSPU), because the Content Service Switch does not support UDP traffic.

[Figure 8-2](#) provides a detailed view of how to configure RealMedia streaming and caching services initially for standalone Content Engines.

Figure 8-2 Configuration of RealMedia Streaming and Caching Services



130065

Table 8-4 is a checklist of tasks for configuring RealMedia services for standalone Content Engines that are running the ACNS 5.2.1 software and later releases. This checklist includes the steps involved in configuring these services on a standalone Content Engine, including the configuration of one or more routing methods to direct RTSP requests from RealMedia players to the Content Engine.

**Table 8-4 Checklist for Configuring RealMedia Streaming and Caching Services for Standalone Content Engines**

Task	Additional Information and Instructions
<p>1. Enable the RealProxy feature on the standalone Content Engine.</p> <ul style="list-style-type: none"> <li>a. Accept the RealProxy license agreement.</li> <li>b. Accept the evaluation RealProxy license, or specify your Cisco permanent RealProxy license.</li> <li>c. Enable the licensed RealProxy feature on the standalone Content Engine.</li> </ul>	<p>See the <a href="#">“Enabling RealProxy on Standalone Content Engines”</a> section on page 8-13.</p>
<p>2. If necessary, specify the RTSP gateway settings.</p> <ul style="list-style-type: none"> <li>a. If the Content Engine is behind a NAT-enabled router, you must specify the IP address of the RTSP gateway (required).</li> <li>b. You can also change the default basic and advanced RTSP gateway settings (optional).</li> </ul>	<p>See the <a href="#">“Configuring the RTSP Gateway for Standalone Content Engines”</a> section on page 8-14.</p>
<p>3. Configure one or more of the following routing methods to direct content requests from RealMedia players to the RTSP gateway on this standalone Content Engine:</p> <ul style="list-style-type: none"> <li>– Direct proxy routing (nontransparent)</li> <li>– RealMedia RTSP transparent redirection (WCCP Version 2 routing or Layer 4 switching)</li> </ul>	<p>With direct proxy routing, the RealMedia client players (for example, RealPlayer or RealOne player) send their content requests directly to this Content Engine (acting as a nontransparent forward proxy server). With direct proxy routing, you must point the RealMedia clients directly to the Content Engine. See the <a href="#">“Configuring Direct Proxy Routing and RealMedia Proxy Caching”</a> section on page 8-21.</p> <p>With RealMedia RTSP transparent redirection, you must configure the WCCP routers or Layer 4 switches and the Content Engine (transparent proxy server) for this type of transparent redirection. See the <a href="#">“Configuring RTSP Transparent Redirection and Caching of RealMedia Requests”</a> section on page 8-17.</p>
<p>4. Use the RealSystem administrator GUI to configure RealProxy (for example, configure live splitting with RealProxy).</p> <p><b>Tip</b> Live broadcasts are live streams are not files; therefore, they cannot be cached. However, RealProxy can split live broadcasts (live splitting) to conserve network bandwidth.</p>	<p>See the <a href="#">“Configuring RealProxy with the RealSystem Administrator GUI”</a> section on page 8-21.</p>

## Enabling RealProxy on Standalone Content Engines

The RealNetworks, Inc. RealProxy product is licensed software. To activate the licensed RealProxy feature on a standalone Content Engine, you must have a RealProxy license key. For more information about the RealProxy license, see the [“About the RealProxy License Key”](#) section on page 8-9.

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

To use the Content Engine CLI to enable the licensed RealProxy feature on a standalone Content Engine, follow these steps:

---

**Step 1** View the RealProxy license agreement.

```
ContentEngine# show rtsp proxy media-real license-agreement
```

**Step 2** After reading the license agreement, enter global configuration mode and accept the license agreement.

```
ContentEngine# configure terminal
ContentEngine(config)# rtsp proxy media-real accept-license-agreement
```

**Step 3** Enter your Cisco license key for the licensed RealProxy feature.

```
ContentEngine(config)# rtsp proxy media-real license-key licensekey
```

Alternatively, accept an evaluation RealProxy license.

```
ContentEngine(config)# rtsp proxy media-real evaluate
```

**Step 4** Enable the licensed RealProxy feature on this Content Engine.

```
ContentEngine(config)# rtsp proxy media-real enable
```

---

**Note**

For information about uninstalling the RealProxy license on standalone Content Engines, see the [“Uninstalling the RealProxy License Key”](#) section on page 8-31.

---

The next step is to specify the RTSP gateway settings, if necessary. Because the RTSP gateway is automatically enabled on the Content Engine with a default configuration (see [Table 8-5](#)), you only need to change the default RTSP gateway settings in the following situations:

- If the Content Engine is behind a NAT-enabled router, you must specify the IP address of the RTSP gateway. By default, there is no IP address specified for the RTSP gateway.
- If you want to change any of the default settings, including the port that the RTSP gateway is to listen on for incoming requests (port 554 is the default).

For information about how to change the default RTSP gateway settings, see the [“Configuring the RTSP Gateway for Standalone Content Engines”](#) section on page 8-14.

Otherwise, the next step is to configure one or more of the following routing methods to direct content requests from RealMedia players to this standalone Content Engine:

- Direct proxy routing (nontransparent)

With direct proxy routing, the RealMedia players send their requests directly to this Content Engine (nontransparent forward proxy server). For instructions on how to configure a RealMedia player on the end user desktops to point directly to this Content Engine as its proxy server, see the [“Pointing RealMedia Players Directly to a Standalone Content Engine”](#) section on page 4-46.

- WCCP routing or Layer 4 switch (RealMedia RTSP transparent redirection)

By default, Layer 4 switching is not enabled and WCCP transparent redirection is not configured on the RTSP gateway. (See [Table 8-5](#).) To enable transparent redirection of RealMedia RTSP requests through Layer 4 switching or WCCP Version 2, complete the process described in the [“Configuring RTSP Transparent Redirection and Caching of RealMedia Requests”](#) section on page 8-17.

## Configuring the RTSP Gateway for Standalone Content Engines

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



### Note

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

In the ACNS 5.3.1 software release, the RTSP gateway was expanded to enable it to switch and tunnel RTSP requests from Windows Media 9 players to a WMT RTSP-based server. Consequently, you can enable RealProxy or the WMT RTSP-based server as a backend RTSP server on standalone Content Engines that are running the ACNS 5.3.1 software and later releases. For Content Engines that are registered with a Content Distribution Manager, you can also enable RealSubscriber and Cisco Streaming Engine as backend RTSP servers that run on the Content Engine.

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

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

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

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

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

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



**Note**

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

## Default RTSP Gateway Settings

The RTSP gateway is automatically enabled on the Content Engine, and cannot be disabled with a command. [Table 8-5](#) lists the default settings for the RTSP gateway.

**Table 8-5**      *Default Settings for the RTSP Gateway*

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

## Configuring Basic Settings for the RTSP Gateway

By default, the RTSP gateway is always enabled on a Content Engine, and cannot be disabled by entering a CLI command. As [Table 8-5](#) shows, the RTSP gateway has a set of default settings. You only need to change these default settings under the following conditions:

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

To configure the basic settings for the RTSP gateway on a standalone Content Engine, use the **rtsp** global configuration commands.

**rtsp** {**ip-address** *rtsp-gateway-ip-address* | **L4-switch enable** | **port incoming** *port*} command.

[Table 8-6](#) describes the command parameters.

**Table 8-6** Parameters for the *rtsp* Command

Parameter	Description
<b>ip-address</b>	Configures the IP address of the RTPS gateway.
<i>rtsp-gateway-ip-address</i>	IP address of the RTSP gateway that is running on the Content Engine. By default, no IP address is specified.
<b>L4-switch</b>	Configures Layer 4 switch interoperability with RTSP.
<b>enable</b>	Enables Layer 4 switch interoperability with RTSP.
<b>port</b>	Configures the RTSP gateway port.
<b>incoming</b>	Configures the port on which the RTSP gateway listens for incoming RTSP requests.
<i>port</i>	Port number on which the RTSP gateway is to listen for incoming RTSP requests. You can specify a single incoming port number. The port number can be from 1 to 65535. The default is port 554.

To configure the basic RTSP gateway parameters on a standalone Content Engine, follow these steps:

- 
- Step 1** Specify the incoming port (the port on which the RTSP gateway is to listen for incoming RTSP requests).
- ```
ContentEngine(config)# rtsp port incoming port-number
```
- Step 2** Specify the IP address of the RTSP gateway.
- ```
ContentEngine(config)# rtsp ip-address rtsp-gateway-ip-address
```
- Step 3** Configure transparent redirection of RTSP requests through WCCP Version 2 or Layer 4 switching.
- For more information, see the [“Configuring RTSP Transparent Redirection and Caching of RealMedia Requests”](#) section on page 8-17.
-

## Configuring Advanced Options for the RTSP Gateway

If the Content Engine is running the ACNS 5.2.1 software and later releases, you can use the **rtsp advanced** global configuration command to configure the following three advanced options for the RTSP gateway:

- Bypass gateway
- Maximum initial setup delay
- Maximum request rate

The syntax for the command is as follows:

```
rtsp advanced {bypass-gateway media-real | max-initial-setup-delay time_delay |
max-request-rate number}
```

Table 8-7 describes the command parameters.

**Table 8-7** Parameters for the *rtsp advanced* CLI Command

Parameter	Description
<b>advanced</b>	Configures the advanced options for the RTSP gateway that is running on the Content Engine.
<b>bypass-gateway</b>	Sets the bypass gateway feature that enables the specified types of RTSP requests (for example, RealMedia requests) to bypass the RTSP gateway.
<b>media-real</b>	However, if the Content Engine is registered with a Content Distribution Manager, you can specify the following additional options for Content Engines because the Cisco Streaming Engine and RealSubscriber are supported as RTSP-based backend servers: the <b>bypass-gateway cisco-streaming-engine</b> option and the <b>bypass-gateway real-subscriber</b> option.
<b>max-initial-setup-delay</b>	Sets the maximum delay that is allowed between the TCP accept and the first RTSP message from the client. The unit of measurement is seconds. The default is 10 seconds.
<i>time_delay</i>	Maximum time delay allowed in seconds (0-2147483647). The default value is 10 seconds.
<b>max-request-rate</b>	Sets the maximum number of incoming requests per second that the RTSP gateway allows.
<i>number</i>	Maximum number of incoming requests per second that the RTSP gateway allows. The default value is 40 requests per second.

## Configuring RTSP Transparent Redirection and Caching of RealMedia Requests

If RTSP transparent redirection through WCCP Version 2 or Layer 4 switching is being used to direct content requests from RealMedia players to a standalone Content Engine, you can configure the Content Engine to support RealMedia transparent caching for VOD files. In this case, the standalone Content Engine is acting as a transparent proxy server for the end users who are using RealMedia players to request streaming media content. After receiving a transparently redirected RTSP request from a RealMedia player, the Content Engine retrieves the requested content from the origin streaming server if it is not already stored in its local cache, stores a copy locally whenever possible (VOD files can be cached but not live streams), and sends the requested content to the RealMedia player.

**Note**

---

The term *RealMedia RTSP transparent redirection* is used to refer to RTSP transparent redirection of RealMedia requests (WCCP service 80). In contrast, the term *WMT RTSP transparent redirection* is used to refer to RTSP transparent redirection of WMT requests from Windows Media 9 players. For information about how to configure WMT RTSP transparent redirection, see the [“Configuring RTSP Transparent Redirection of WMT Requests” section on page 9-32](#).

---

The Content Engine listens for redirected RTSP requests on the standard RTSP port (default port 554). To intercept RealMedia RTSP traffic on multiple ports, you must configure a user-defined WCCP service (services 90 to 97). For information on this topic, see the [“Configuring Standalone Content Engines to Support User-Defined WCCP Services” section on page 6-15](#).

With RealMedia RTSP transparent redirection, a Layer 4 switch or WCCP Version 2-enabled router redirects RealMedia requests to the Content Engine (acting as a transparent proxy server). RTSP transparent redirection is used to support RealMedia transparent caching on a standalone Content Engine.

To enable transparent redirection of RTSP requests through Layer 4 switching, enter the **rtsp L4-switch enable** global configuration command. After you enter the command, a message appears indicating that Layer 4 switching for RTSP has been enabled on the Content Engine:

```
ContentEngine(config)# rtsp L4-switch enable
Turn on l4 switch
```

To configure RealMedia RTSP transparent redirection through WCCP Version 2, you must perform these tasks:

- Configure RealMedia RTSP redirection on the WCCP Version 2 routers that will support this RTSP streaming service.
- Configure RTSP redirection on the standalone Content Engine that will be receiving these transparently redirected RTSP requests.

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

**Note**

---

To configure RealMedia RTSP transparent redirection on multiple ports, you must configure a user-defined WCCP service (services 90 to 97) on the WCCP Version 2-enabled routers and the Content Engine. For more information about configuring such WCCP services, see the [“Configuring Standalone Content Engines to Support User-Defined WCCP Services” section on page 6-15](#).

---

The following example shows how to configure RealMedia RTSP transparent redirection (service 80) on a standalone Content Engine and a router with WCCP Version 2, and enable RealProxy transparent caching on the Content Engine. This example assumes that you have enabled the licensed RealProxy feature on the standalone Content Engine, as described in the “[Enabling RealProxy on Standalone Content Engines](#)” section on page 8-13.

To configure RealMedia RTSP transparent redirection (the rtsp service [service 80]) and RealMedia transparent caching for standalone Content Engines, follow these steps:

- Step 1** Enable WCCP Version 2 on the router. (WCCP Version 1 does not support service 80.)

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

- Step 2** Enable service 80 on the WCCP Version 2-enabled router.

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

- Step 3** Specify the interface on which the RTSP redirection service will run.

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

The following shows how to configure the router to run service 80 on the Fast Ethernet interface:

```
Router(config)# interface fastEthernet 0/0
```

- Step 4** Configure the router to use the outbound interface for service 80.

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



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

- Step 5** End the configuration session on the router.

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

- Step 6** Enable RTSP redirection through WCCP on the standalone Content Engine. This is the Content Engine that will act as the transparent proxy server for redirected RTSP requests from these WCCP Version 2-enabled routers.

- a. Create the numbered router list that you want to associate with service 80.

In the following example, there is a single WCCP Version 2-enabled router associated with router list 1. This router has an IP address of 10.1.3.1.

```
ContentEngine(config)# wccp router-list 1 10.1.3.1
```

- b. Enable the router list (router list 1) that you just created in the previous step (Step a.).

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

WCCP configuration for RTSP succeeded. Please remember to configure WCCP service 80 on the corresponding router.

You have already configured service 80 on the corresponding router (the router with an IP address of 10.1.1.1) in [Step 2](#) through [Step 5](#) of this procedure.

- c. Enable WCCP Version 2 on the Content Engine that will accept redirected RTSP requests from the WCCP Version 2-enabled routers that are listed in router list 1.

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

- Step 7** If transaction logging is not currently enabled on the Content Engine, enable it.

```
ContentEngine(config)# transaction-log enable
```

- Step 8** Save the new configuration on the Content Engine.

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

- Step 9** Display the list of WCCP services that are currently configured on Content Engine A to verify that service 80 (the rstp service) is listed.

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

The partial sample output that is shown above indicates that this WCCP service is now enabled on the Content Engine, along with the FTP service and the web-cache service. For a descriptive list of the supported WCCP services, see [Table B-3](#).

- Step 10** Verify that WCCP transparent redirection is now enabled on the Content Engine.

```
ContentEngine# show rtsp
```

- Step 11** If necessary, specify the RTSP gateway settings:

- a. If the Content Engine is behind a NAT-enabled router, you must specify the IP address of the RTSP gateway (required).
- b. You may also want to change the default basic and advanced RTSP gateway settings.

For more information, see the “[Configuring the RTSP Gateway for Standalone Content Engines](#)” section on page 8-14.

- Step 12** After configuring the routers and Content Engine to support RTSP redirection through WCCP Version 2, you can configure RealMedia transparent caching on the Content Engine.

- a. Enable the RealProxy product feature on the Content Engine, if it is not already enabled.

```
ContentEngine(config)# rtsp proxy media-real enable
```

- b. Use the RealSystem administrator GUI to configure RealProxy (for example, for live splitting). For more information, see the “[Configuring RealProxy with the RealSystem Administrator GUI](#)” section on page 8-21.

---

For an example of how to verify whether RealMedia transparent caching is working properly for standalone Content Engines, see the “[Example 1—Verifying the Configuration for RealMedia VOD Caching](#)” section on page 8-24.

## Configuring Direct Proxy Routing and RealMedia Proxy Caching

If direct proxy routing is being used to direct content requests from RealMedia players directly to the standalone Content Engine, then you can configure the Content Engine to support RealMedia proxy caching for VOD files. With RealMedia proxy caching, the standalone Content Engine is functioning as a nontransparent proxy server for the RealMedia players. After receiving a content request directly from a RealMedia player (for example, a RealPlayer or RealOne player), the Content Engine retrieves the requested VOD file from the origin streaming server if it is not already stored in its local cache, stores a copy locally, and sends the requested streaming media content to the RealMedia player.

The following example assumes that you have enabled the licensed RealProxy feature on the standalone Content Engine, as described in the [“Enabling RealProxy on Standalone Content Engines”](#) section on page 8-13.

To use the Content Engine CLI to configure RealMedia proxy caching on a standalone Content Engine, follow these steps:

---

**Step 1** Configure the RealMedia players to send their requests directly to this Content Engine.

By default, the RTSP gateway on the Content Engine listens for incoming RTSP requests on port 554. If you entered the **rtsp port incoming *port-number*** global configuration command to specify another port as the incoming RTSP port (for instance, you configured the RTSP gateway to listen on port 575 instead of port 554), you must configure the RealMedia players to send their requests directly to the configured RTSP incoming port. For more information, see the [“Pointing RealMedia Players Directly to a Standalone Content Engine”](#) section on page 4-46.



**Note** If a firewall is positioned between a Content Engine and a requesting client, make sure that you specify the external IP address of the Content Engine as the proxy server when you configure the RealMedia proxy settings on client desktops. See the [“Pointing RealMedia Players Directly to a Standalone Content Engine”](#) section on page 4-46.

---

**Step 2** Use the RealSystem administrator GUI to configure RealProxy (for example, for live splitting) on the Content Engine.

For more information, see the next section, [“Configuring RealProxy with the RealSystem Administrator GUI.”](#)

---

## Configuring RealProxy with the RealSystem Administrator GUI

RealProxy is a licensed product from RealNetworks, Inc. You use the Setup utility or the Content Engine CLI to enable the licensed RealProxy feature on a standalone Content Engine. You can also use the Setup utility or the Content Engine CLI to enable RealMedia proxy caching and RealMedia transparent caching on a standalone Content Engine after RealProxy has been enabled on the Content Engine. However, you perform RealProxy configuration through the RealNetworks RealSystem administrator GUI.

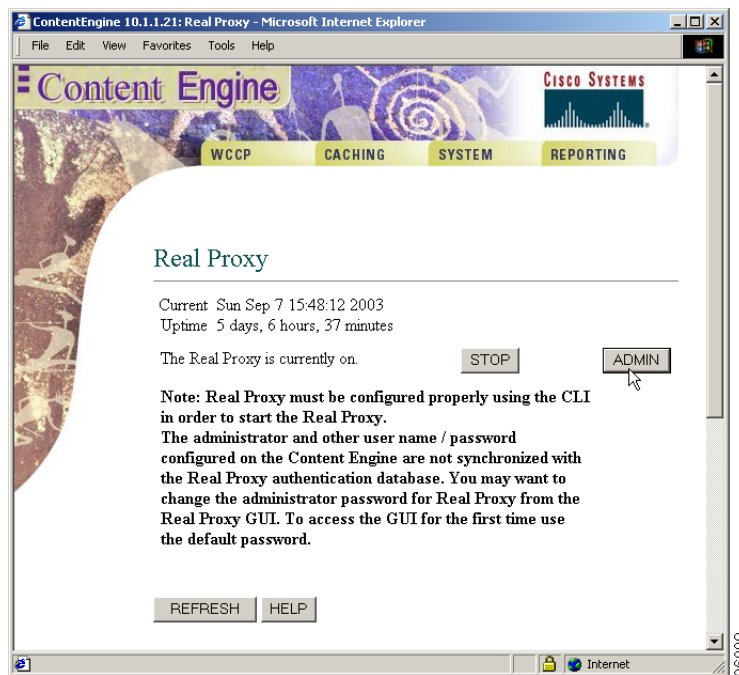
To access the RealSystem administrator GUI, follow these steps:

- Step 1** From the Content Engine GUI, choose **Caching > Real Proxy**. The Content Engine RealProxy window appears. (See [Figure 8-3](#).)



**Note** To access the Content Engine GUI, enter the Content Engine IP address in secure mode and append the default port number 8003 as the URL address in your browser of choice. For example, enter **https://ContentEngineIPAddress:8003** as the URL.

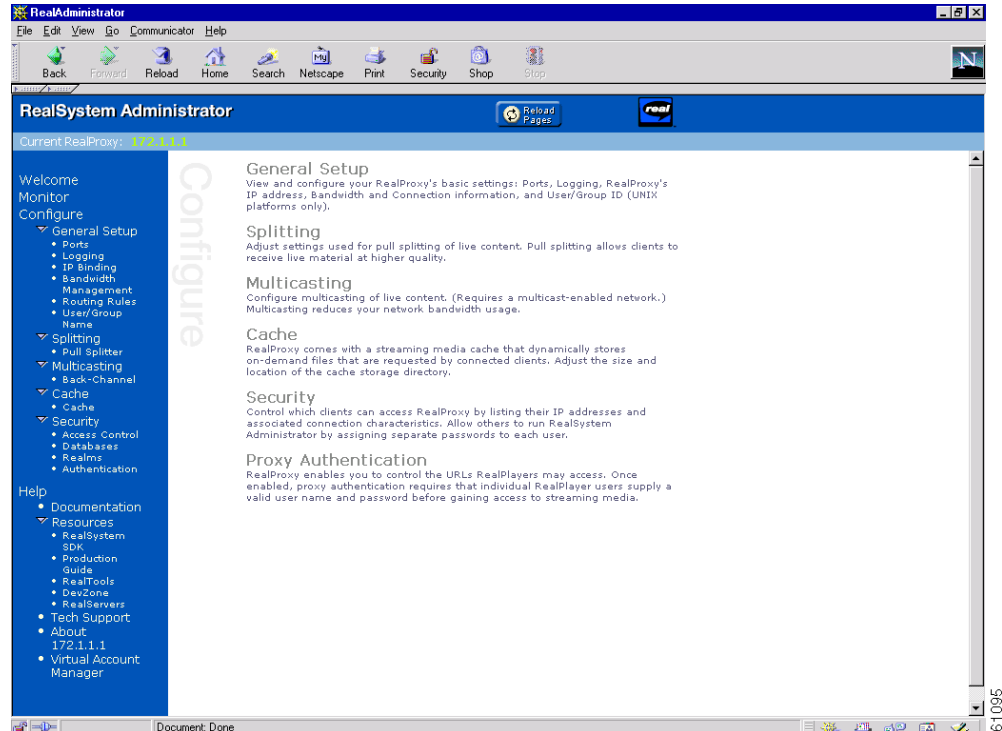
**Figure 8-3** Content Engine RealProxy Window



**Note** The **ADMIN** button only appears in the Content Engine RealProxy window if the RealProxy software has been installed and enabled on the Content Engine.

- Step 2** In the Content Engine RealProxy window, click the **ADMIN** button.
- Step 3** Use **admin** as the default username and **diamond** as the password to access the RealSystem administration GUI from the Content Engine RealProxy window. The main window for the RealSystem administrator GUI (see [Figure 8-4](#)) appears.

Figure 8-4 RealSystem Administrator GUI



- Step 4** Use the RealSystem administrator GUI to configure the licensed RealProxy feature on this Content Engine. For example, use this GUI to configure RealProxy live splitting and caching of VOD files.

After configuring RealProxy, you should verify that the RealProxy configuration is working properly. For some examples of how to verify the RealProxy configurations for live splitting and VOD caching, see the next section, “[Verifying RealProxy Configurations for Standalone Content Engines.](#)”

## Verifying RealProxy Configurations for Standalone Content Engines

This section provides two examples of how to verify the configuration of RealProxy on standalone Content Engines:

- [Example 1—Verifying the Configuration for RealMedia VOD Caching, page 8-24](#)
- [Example 2—Verifying the Configuration for RealProxy Live Splitting, page 8-27](#)

## Example 1—Verifying the Configuration for RealMedia VOD Caching

This first example shows how to verify the configuration of the RealMedia transparent caching service on a standalone Content Engine. This example assumes the following:

- RealProxy has been enabled on the Content Engine. (See the “[Enabling RealProxy on Standalone Content Engines](#)” section on page 8-13.)
- There is a single standalone Content Engine (Content Engine A) that has WCCP Version 2 enabled on it.
- There is a single WCCP Version 2-enabled router (Router A) that is configured to redirect content requests from RealMedia players to Content Engine A (transparent proxy server).
- The RealMedia transparent caching service is enabled on Content Engine A, and Content Engine A is configured to accept redirected RTSP requests from Router A.
- The RealMedia players (in this case, RealPlayer) on Client A and Client B are not configured to point directly to Content Engine A.
- Client A and Client B are on the same subnet.

To verify that the RealMedia transparent caching service (RealMedia transparent caching through WCCP Version 2) is working properly, follow these steps:

- 
- Step 1** From the Client A desktop, use RealPlayer to request a RealMedia streaming video file (.rm file).
- a. From the RealPlayer menu, choose **File > Open URL**.
  - b. Specify a URL that points to a RealMedia streaming video file (for example, **rtsp://origin-streaming-server-ip-address/gm1\_real\_02\_00500.rm**).




---

**Note** Request the video file more than once. The origin streaming server from which you are requesting content (for example, the .rm video files) must be on a different subnet than Client A and Client B, so that the RealPlayer requests from these client desktops are routed to Router A.

---

The requested video file should start playing in RealPlayer on Client A.

- Step 2** Check the statistics on RealPlayer.
- a. From RealPlayer, choose **Tools > Playback Statistics**.
  - b. In RealPlayer, click the Streams tab to bring it to the front.
  - c. In the Streams tab, check whether UDP is shown as the transport protocol that is being used. If UDP is shown as the transport protocol, this indicates that the stream is being delivered to RealPlayer in a streaming fashion instead of reverting to HTTP when the streaming proxy or server is not available.
- Step 3** From the Client A desktop, use RealPlayer to replay the same streaming video file that you just requested.

- Step 4** On the WCCP-enabled router, check the number of redirected RTSP packets from RealMedia players. The following is a sample output:

```
Router# show ip wccp 80
Global WCCP information:
  Router information:
    Router Identifier:      10.1.3.1
    Protocol Version:      2.0

  Service Identifier: 80
    Number of Cache Engines: 1
    Number of routers:      1
    Total Packets Redirected: 6
    Redirect access-list:    -none-
    Total Packets Denied Redirect: 0
    Total Packets Unassigned: 0
    Group access-list:      -none-
    Total Messages Denied to Group: 0
    Total Authentication failures: 0
```

- Step 5** On the Content Engine, check the RealMedia caching statistics.

- a. From Content Engine A, display the RealMedia caching saving statistics for this Content Engine.

```
ContentEngine# show statistics rtsp proxy media-real savings
Media Cache Statistics - Savings
-----
```

	Requests	Bytes
Total:	17	16666028
Hits:	11	3656524
Miss:	6	13009504
Savings:	64.7 %	21.9 %

- b. From Content Engine A, display the number of RealMedia requests that this Content Engine has received. (See the sample output.) Verify that the Content Engine is processing RealMedia requests.

```
ContentEngine# show statistics rtsp proxy media-real requests
Media Cache Statistics - Requests
-----
```

	Total	% of Requests
Total Received Requests:	17	-
Demand Cache Hit:	11	64.7
Demand Cache Miss:	6	35.3
Demand Pass-Through:	0	0.0
Live Split:	0	0.0
Live Pass-Through:	0	0.0

- Step 6** From the Client B desktop, use RealPlayer to request the same video file that you requested earlier from the Client A desktop.

This will allow you to check whether Content Engine A is storing a copy of the requested VOD in its local cache instead of retrieving the video file again from the origin streaming server.

- The number of cache hits displayed in the output of the **show statistics rtsp proxy media-real savings** EXEC command should increase as you use RealPlayer on Client B to request the same video file that you just requested from RealPlayer on Client A.
- The number of RealMedia requests displayed in the output of the **show statistics rtsp proxy media-real requests** EXEC command should increase as you use RealPlayer on Client B to request the same VOD file that you just requested from Client A.

**Step 7** On Router A, open a console or Telnet session.

**Step 8** On Router A, display statistics and status information for the rtsp service (service 80).

```
Router# show ip wccp 80
```

The statistics should show a number greater than 0 for packets redirected.

**Step 9** Verify that packets are being redirected to Content Engine A from Router A:

- If Router A shows that there are packets being redirected to Content Engine A, then RealMedia transparent caching (transparent redirection of RTSP requests through service 80) is operating properly on Content Engine A and Router A.
- If Router A shows that no packets are being redirected to Content Engine A, then RealMedia transparent caching is not operating properly. In this case, you should troubleshoot the problems with your configuration of the rtsp service. The following are some examples of how to do this.
  - a. From Content Engine A, display the list of WCCP services that are currently configured on Content Engine A. The following is a sample output.

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

Verify that the rtsp service (service 80) is listed.

- b. From Content Engine A, display a list of WCCP-enabled routers that recognize Content Engine A. The following is a sample output:

```
ContentEngineA# show wccp routers
Routers Seeing this Content Engine
  Router Id      Sent To
  10.0.0.0       10.1.1.1
Routers not Seeing this Cache Engine
  10.1.1.1
Routers Notified of but not Configured
-NONE-
```

Check the command output to determine if Router A is on the list of WCCP-enabled routers that recognizes Content Engine A.

- c. From Content Engine A, display WCCP generic routing encapsulation (GRE) packet-related information for Content Engine A.

```
ContentEngineA# show wccp gre
```

Check the command output to view the number of redirected packets that Content Engine A has rejected and accepted. Verify that the number of accepted packets is increasing as you continue to request VOD files that are on streaming servers that are on different subnets than the requesting client (RealPlayer on the Client A and B desktops).

- Step 10** You can also use the RealSystem administrator GUI to monitor RealProxy statistics when RealMedia clients are connected.
- From the Content Engine GUI, choose **Caching > Real Proxy**. The Content Engine RealProxy window appears. (See [Figure 8-3](#).)
  - Click the **Admin** button.
  - When asked, enter the username and password. The default username is admin. The default password is diamond. After entering a valid username and password, the RealSystem Administrator main window appears. (See [Figure 8-4](#).)
  - Use the RealSystem Administrator GUI to monitor RealProxy statistics for the RealMedia requests that are being serviced by RealProxy on this Content Engine.

- Step 11** By default, transaction logging is enabled on the Content Engine. Verify that the RealMedia transactions that are being serviced by this Content Engine are being tracked in the RealProxy transaction log.

```
ContentEngine# type-tail /local1/logs/real-proxy/rproxy-transaction-log-filename
```

Depending upon where the sysfs is mounted, RealProxy logs are logged to a working log on the local disk in one of these files:

- /local1/logs/real-proxy/working.log
- /local2/logs/real-proxy/working.log

You can specify the interval at which the working log should be cleared by moving the data to an archive log. The archive log files are located on the local disk in the /local1/logs/ or /local2/logs/ directory depending upon where the sysfs is mounted.

Because multiple archive files are saved, the filename includes the time stamp when the file was archived. Because the files can be exported to an FTP/SFTP server, the filename also contains the IP address of this Content Engine. For more information about using transaction logs, see the “[Using ACNS Software Transaction Logs](#)” section on page 21-31.

- Step 12** Check the system log file for RealProxy error messages.

```
ContentEngine# type-tail syslog.txt
```

RealProxy generates error messages and writes them to the RealProxy log file. These error messages are captured by the ACNS software and passed to the system log file. See [Table 21-9](#) for correspondence between the RealProxy error codes and corresponding syslog priority levels. For more information about system logging, see the “[Monitoring the Performance of Specific URLs](#)” section on page 21-52.

---

## Example 2—Verifying the Configuration for RealProxy Live Splitting

This second example shows how to verify the configuration for RealProxy live splitting (live splitting through RealProxy and a WCCP Version 2-enabled router) on a standalone Content Engine. When RealProxy is enabled on a standalone Content Engine, the Content Engine serves as the stream splitting point for all local users (RealMedia players) whose RealMedia traffic is directed to the Content Engine. All further requests to that origin streaming server are served by the Content Engine, which splits the stream and serves it to the RealMedia players. Live broadcasts (live streams) are not files and therefore cannot be cached. For more background information about RealProxy live splitting, see the “[About Live Splitting and Caching VOD Files with RealProxy](#)” section on page 8-6.

In this example, the RealMedia players on the Client A, B, and C desktops are configured to point to a live stream that is set up to play continuously. WCCP Version 2 is used to redirect the request for this live stream to the Content Engine. A WCCP Version 2-enabled router transparently intercepts the request for the live stream, and redirects the request to the Content Engine (Content Engine A) that is running RealProxy.

This example assumes the following:

- RealProxy is enabled on Content Engine A. (See the [“Enabling RealProxy on Standalone Content Engines”](#) section on page 8-13.)
- WCCP Version 2 is also enabled on Content Engine A.
- There is a single WCCP Version 2-enabled router (Router A) that is configured to redirect content requests from RealMedia players to Content Engine A (transparent proxy server).
- The RealMedia players (in this case, RealPlayer) on Clients A, B, and C are not configured to point directly to Content Engine A.
- Clients A, B, and C are on the same subnet.

To verify that RealProxy live splitting is working properly, follow these steps:

- 
- Step 1** Create a live stream on an origin streaming server (a Helix Universal Server) that supports the RealNetworks proprietary RTSP protocol.
- Use either an encoder or the Simulated Live Transfer Agent (SLTA) tool.
- SLTA is a Helix Universal Server utility to stream a prerecorded clip as if it were a live event.
- The origin streaming server (Helix Universal Server) on which you are creating the live stream should be on a different subnet from that of Clients A, B, and C so that the RealMedia requests from these clients are routed to Router A.
- Step 2** From the Client A desktop, use RealPlayer to request the live stream (live event).
- a. From the RealPlayer menu, choose **File > Open URL**.
  - b. Specify a live event alias (for example, `rtsp://origin-streaming-server-ip-address/broadcast/live`).
- The requested live event should start playing in RealPlayer on Client A.
- Step 3** From the Client B and Client C desktops, use RealPlayer to request the same live stream that you requested earlier with RealPlayer on the Client A desktop.
- Step 4** Use the same verification process that is described in the [“Example 1—Verifying the Configuration for RealMedia VOD Caching”](#) section on page 8-24 to verify the following:
- The WCCP-enabled router (Router A) is redirecting the RealPlayer requests for live streams to the Content Engine A.
  - Content Engine A is serving the live streams with bandwidth savings to Client A and Client B.
  - Transactions related to these live streams (for example, which clients are viewing this particular live event, and the length of time that the client viewed this live event) are being logged in the RealProxy transaction log on Content Engine A.

For example, use this process:

- a. Open a console or Telnet session on Router A. On Router A, enter the **show ip wccp 80 EXEC** command to display statistics and status information for service 80 for Router A. The statistics should show a number greater than 0 for packets redirected.
- b. From Content Engine A, enter the **show statistics rtsp proxy media-real requests EXEC** command to display the number of RealMedia requests that Content Engine A has received. The statistic should show a number greater than 0 for live split requests.

---

## Restoring the RealProxy Factory-Default Configuration on Standalone Content Engines

On standalone Content Engines, RealProxy is enabled through the Content Engine CLI or through the Setup utility. To change the RealProxy default configuration, you must use the RealNetworks RealSystem administrator GUI, which you can access from the Content Engine GUI. However, in the ACNS 5.3.3 software and later releases, you can use the Content Engine CLI (the **rtsp real-proxy restore factory-default EXEC** command) to restore the RealProxy factory-default configuration. The default configuration is the RealProxy configuration file and the database that contains the RealProxy license keys on a standalone Content Engine.

In the ACNS 5.3.3 software release, the **rtsp real-proxy default-configuration EXEC** command was replaced with the **rtsp real-proxy restore factory-default EXEC** commands. In the ACNS 5.3.1 software and earlier releases, when you entered the **rtsp real-proxy default-configuration EXEC** command, only the RealProxy configuration files were restored to the default settings; the database that contains the RealProxy license key settings was not restored to the factory defaults.



### Note

On Content Engines that are registered with a Content Distribution Manager, you can use the Content Distribution Manager or the CLI to restore the RealNetworks (RealProxy and RealSubscriber) license key settings to the factory default. For information about how to restore the factory-default settings for RealNetworks license keys on Content Engines that are registered with a Content Distribution Manager, see the *Cisco ACNS Software Configuration Guide for Centrally Managed Deployments, Release 5.5*.

To restore the RealProxy factory-default configuration on a standalone Content Engine, follow these steps:

- 
- Step 1** Enter the **rtsp real-proxy restore factory-default EXEC** command, and enter **yes** when asked if you want to proceed:

```
ContentEngine# rtsp real-proxy restore factory-default
User would lose the current real proxy configuration. Do you want to proceed? [yes/no] yes
Restart Real Proxy to load the factory defaults configuration.
```

**Step 2** Restart RealProxy on the Content Engine to load the restored RealProxy factory default configuration on the Content Engine.

For more information, see the [“Restarting RealProxy on Standalone Content Engines”](#) section on page 8-30.

**Note**

For information on how to access the RealSystem administrator GUI from the Content Engine GUI, see the [“Configuring RealProxy with the RealSystem Administrator GUI”](#) section on page 8-21.

## Restarting RealProxy on Standalone Content Engines

To restart RealProxy on a standalone Content Engine, follow these steps:

**Step 1** Stop RealProxy on the Content Engine.

```
ContentEngine(config)# no rtsp proxy media-real enable
```

**Step 2** Restart RealProxy on the Content Engine.

```
ContentEngine(config)# rtsp proxy media-real enable
```

## Disabling RealMedia Caching on Standalone Content Engines

To disable RealMedia caching on a standalone Content Engine, follow these steps:

**Step 1** Access the RealSystem Administrator GUI window by clicking the **Admin** button in the RealProxy window of the Content Engine GUI. (See [Figure 8-3](#).) (You must enable the RealProxy before you can access the **Admin** button in this window.)

**Note**

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

**Step 2** Choose **Configure > Cache**.

**Step 3** In the Enable Caching field, choose **No**.

**Step 4** Click **Apply**.

**Step 5** Stop RealProxy on the Content Engine.

```
ContentEngine(config)# no rtsp proxy media-real enable
```

**Step 6** Restart RealProxy on the Content Engine.

```
ContentEngine(config)# rtsp proxy media-real enable
```

## Uninstalling the RealProxy License Key

If the RealProxy license key is no longer needed on the Content Engine because the licensed RealProxy feature is not needed, you can uninstall the RealProxy license key by entering the **no rtsp rproxy media-real license-key** global configuration command. After a license key is uninstalled on one Content Engine, it can be used on another Content Engine if that Content Engine supports the licensed RealProxy feature.



### Note

The licensed RealProxy feature must be disabled using the **no rtsp proxy media-real enable** command before uninstalling the RealProxy license key on a standalone Content Engine.

## Displaying RealProxy Statistics for Standalone Content Engines

You can use the **show statistics rtsp proxy media-real EXEC** commands to display RealProxy statistics for standalone Content Engines. The displayed statistics relate only to objects transported over RTSP that were requested by a RealMedia client. Objects transported over HTTP are counted in the HTTP statistics. Streaming objects requested by other clients or transported over protocols bypass the Content Engine.

```
ContentEngine# show statistics rtsp proxy media-real requests
Media Cache Statistics - Requests

```

	Total	% of Requests
Total Received Requests:	0	-
Demand Cache Hit:	0	0.0
Demand Cache Miss:	0	0.0
Demand Pass-Through:	0	0.0
Live Split:	0	0.0
Live Pass-Through:	0	0.0

```
ContentEngine# show statistics rtsp proxy media-real savings
Media Cache Statistics - Savings

```

	Requests	Bytes
Total:	0	0
Hits:	0	0
Miss:	0	0
Savings:	0.0 %	0.0 %

You can also obtain detailed configuration, statistics, and reporting of RealProxy status through the RealNetworks RealSystems administrator GUI. The Content Engine GUI has a RealProxy window (Figure 8-3). The **ADMIN** button is active in the Content Engine Management GUI when RealProxy is

installed and enabled on the Content Engine. You will be provided with a default username and password to access this administrator window from the Content Engine GUI. For more information, see the [“Configuring RealProxy with the RealSystem Administrator GUI” section on page 8-21](#).