



Configuring WMT Streaming Media Services on Standalone Content Engines

This chapter provides an overview of the Windows Media Technologies (WMT) streaming and caching services, and describes how to use the Content Engine CLI to configure these services on standalone Content Engines.

This chapter contains the following sections:

- [Overview of the Windows Media Services Streaming Solution, page 9-2](#)
- [Overview of the WMT Streaming and Caching Services, page 9-4](#)
- [Configuration Guidelines, page 9-10](#)
- [Configuring WMT RTSP Streaming and Caching Services on Standalone Content Engines, page 9-14](#)
- [Enabling WMT Licenses on Standalone Content Engines, page 9-17](#)
- [Configuring General WMT Settings on Standalone Content Engines, page 9-18](#)
- [Configuring Transparent Redirection of WMT Requests, page 9-31](#)
- [Enabling and Configuring WMT Caching on Standalone Content Engines, page 9-34](#)
- [Configuring Standalone Content Engines to Distribute VOD Files, page 9-35](#)
- [Configuring Standalone Content Engines to Deliver WMT Live Streams, page 9-37](#)
- [Displaying Information about the Current WMT Configuration, page 9-46](#)
- [Displaying Information about the WMT RTSP Server Configuration, page 9-46](#)
- [Using WMT Logging with Standalone Content Engines, page 9-47](#)



Note

Throughout this chapter the following terminology is used. The Windows Media streaming and caching services are collectively referred to as the *WMT feature*. Windows Media Player 9 Series clients are called *Windows Media 9 players*. Windows Media Services 9 servers are called *Windows Media 9 servers*. The Windows Media Services 9 RTSP backend server that is running on the Content Engine is called the *WMT RTSP server*. The term *WMT RTSP transparent redirection* is used to refer to RTSP transparent redirection (WCCP Version 2 services 80 and 83) of WMT RTSP requests from Windows Media 9 players.

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*. For information about using the WMT diagnostic tools for troubleshooting purposes, see the “[Troubleshooting with the WMT Diagnostic Tools](#)” section on page 22-11.

Overview of the Windows Media Services Streaming Solution

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

Table 9-1 describes the major components of Windows Media Services.

Table 9-1 Components of Microsoft Windows Media Services

Component	Description
Windows Media player	Desktop application that the end user runs to play requested digital media files (for example, Windows Media 6.4 players and Windows Media 7.0 players, Windows Media 9 players or Windows Media 10 players). These clients can take advantage of the VCR-like controls in the Windows Media player to pause the stream or to skip backward or forward (in the case of stored content [video on demand]).
Windows Right Manager and Encoder	Content-creation application.
Windows Media Server	Server and distribution application that uses an application-level protocol called Microsoft Media Server (for example, Windows Media 9 server and Windows Media 4.1 server) to send active streaming format (ASF) files across the Internet.

With WMS 9, Microsoft introduced a major change in the streaming protocol. Windows Media Services 9 Series by default uses a new RTSP-based protocol for streaming.

These are the streaming protocols currently used by Windows Media 9 players:

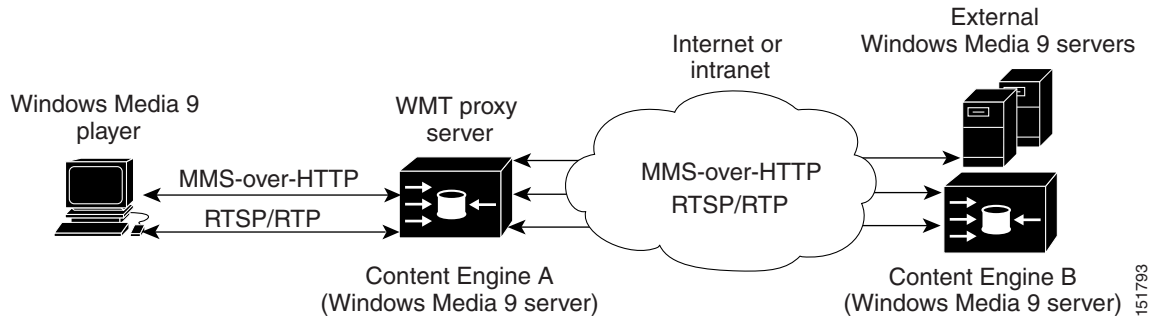
- Windows Media Services 9 Series RTSP/RTP-based protocol
- Windows Media Services 9 Series-over-HTTP

As Figure 9-1 shows, a Content Engine, which is running the ACNS 5.5.1 software, has full interoperability with the Windows Media 9 server and a Windows Media 9 player over all of these streaming protocols.



Note

Although the Windows Media Player will play a file using the HTTP protocol (from port 80), the Content Engine does not support HTTP streaming because the Content Engine does not buffer HTTP files.

Figure 9-1 Streaming Protocols Supported for WMS 9 in the ACNS 5.3.1 Software or Later**Note**

In the ACNS 5.3.1 software and later releases, RTSP/RTP is a supported streaming protocol. Consequently, RTSP requests from Windows Media 9 players are supported. Proxy caching (caching VOD files) and live splitting for WMT RTSP is supported. In centrally managed deployments (that is, Content Engines are registered with a Content Distribution Manager), managed live events are also supported. You use the Content Distribution Manager GUI to configure a managed live event. Standalone Content Engines do not support managed live events. End-to-end RTSP (from the client to the encoder) is not currently supported for managed live events. For information about configuring managed live events, see the *Cisco ACNS Software Configuration Guide for Centrally Managed Deployments, Release 5.5*.

When using direct proxy routing or WCCP redirection to route requests to standalone Content Engines, the unicast published URL can be the following:

- `rtsp://liveChannelOriginFqdn/program-name` (added in the ACNS 5.3.1 software release)

In the ACNS 5.3.1 software and later releases, the live stream source of a Windows Media live program can be one of the following

- `http://encoder:port-number`
- `rtsp://wmStreamingServer:port-number/file name` (RTSP support was added in the ACNS 5.3.1 software release)

For more information about WMT RTSP, see the [“About the WMT RTSP Protocol”](#) section on page 9-4.

The ACNS 5.2.1 software and later releases interoperate with the following software:

- Windows Media Services 9 (WMS 9) Series—Includes the Windows Media 9 player, Windows Media Encoder, and Windows Media 9 server.

In the ACNS 5.3.1 software to ACNS 5.4 software releases, WMT RTSP and WMT MMS are supported. In ACNS 5.5.x release, WMT MMS is not supported but WMT RTSP is supported.

- Windows Media Services 4.1 Series—Includes the Windows Media Player 4.1 Series, Windows Media Encoder, and Windows Media 4.1 server.

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.

Overview of the WMT Streaming and Caching Services

When the WMT feature is enabled on the Content Engine, the Content Engine provides a native (integrated) WMT server that delivers Microsoft's standard streaming formats (.ASF, .WMA, and .WMV files) through either unicast or multicast streams. The integrated WMT server has the ability to serve the streams to the clients by VOD, broadcast (live), and multicast. The WMT feature also allows a standalone Content Engine to support WMT transparent caching and WMT proxy caching.

The WMT feature on a standalone Content Engine is licensed software. To enable this feature on a Content Engine, you must have a WMT license key. You must specify a permanent 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 WMT license through the Cisco.com website. You specify the WMT license key as part of enabling the WMT feature on a standalone Content Engine. See the [“Enabling WMT Licenses on Standalone Content Engines” section on page 9-17](#).

About the WMT RTSP Protocol

The Real-Time Streaming Protocol (RTSP) is a standard Internet streaming control protocol (RFC 2326). It is an application level protocol for control over the delivery of data with real-time properties such as video and audio. RTSP has been widely adopted in the industry. For example, Apple Computer's QuickTime, RealNetworks' RealMedia, and the Cisco Streaming Engine all use RTSP as the streaming control protocol. In WMS 9, Microsoft added support for the RTSP protocol as the streaming control protocol. In earlier versions of WMS (for example, WMS 4.1), WMS used MMS as the streaming control protocol. In the ACNS 5.3.1 software and later releases, WMT RTSP support for WMS 9 is available (that is, support for RTSP requests from Windows Media 9 players).

**Note**

In ACNS 5.5 software release, MMS is not supported. However, MMS-over-HTTP is supported.

The WMT RTSP server, which is running on the Content Engine, uses the WMT RTSP protocol to serve the VOD request to the Windows Media 9 players. The WMT RTSP protocol is the IETF RTSP standard protocol plus Microsoft proprietary extensions. The WMT RTSP server also uses this protocol to support broadcasting. The standard listening port for RTSP services is port 554.

RTSP requests from Windows Media 9 players can be routed directly to the Content Engine or transparently redirected. To route such requests directly to the Content Engine, you must configure the Windows Media 9 players to point directly to the Content Engine. To transparently redirect such requests to the Content Engine, you must configure WMT RTSP transparent redirection (WCCP Version 2 services 80 and 83) on the Content Engine and the WCCP Version 2 router. For more information about configuring Windows Media 9 players to point directly to a Content Engine, see the [“Pointing Windows Media 9 Players Directly to a Standalone Content Engine for WMT RTSP Requests” section on page 4-43](#). For more information about configuring WMT RTSP transparent redirection, see the [“Configuring RTSP Transparent Redirection of WMT Requests” section on page 9-31](#).

**Tip**

For live streaming, Content Engines always obtain the live stream from an external WMT server; the Content Engine is never the originator of the live content. For a standalone Content Engine to deliver WMT live streams, you need WMT caching proxy and server capabilities on the standalone Content Engine. The WMT product is licensed software and requires a WMT license key. For more information about this license key, see the [“Enabling WMT Licenses on Standalone Content Engines” section on page 9-17](#).

How Standalone Content Engines Process WMT Requests

Standalone Content Engines can receive WMT requests directly from WMT clients, or from WCCP Version 2 routers or Layer 4 CSS switches (through WMT transparent redirection).

The actual protocol used is negotiated between the WMT client and the server. If both the client and the server are Windows Media Services 9 Series, then the RTSP protocol is used if the URL starts with `mms://`, and the HTTP protocol is used if the URL starts with `http://`. If either the client or the server is pre-WMS 9 (the client is a Windows Media 6.4 or 7.0 player, or the server is a Windows Media 4.1 server instead of a Windows Media 9 server), the MMS protocol is used.

In the case of MMS-over-HTTP with Windows Media Services 9 Series, a standalone Content Engine that is running the ACNS 5.3.1 software and later releases, supports the Fast Start and Fast Cache features for preloaded VOD, live-split, and on-demand (cache-hit) content from Windows Media 9 players over the following protocols: HTTP, RTSP, and MMS-over-HTTP. For more information on these features, see the [“Configuring Standalone Content Engines to Deliver WMT Live Streams”](#) section on page 9-37 and the [“Configuring Fast Cache on Standalone Content Engines”](#) section on page 9-29.



Note

Support for the Fast Start and Fast Cache features was added in the ACNS 5.2.1 software. In the ACNS 5.2.x software, these features are only available in MMS-over-HTTP streaming with Windows Media Services 9 Series. In the ACNS 5.3.1 software and later releases, the Fast Start and Fast Cache features are also available for RTSP requests from Windows Media 9 players (WMT RTSP requests).

In the ACNS 5.2.1 software and later releases, the WMT streaming module contains two sets of processes that handle client requests:

- The `mms_server` processes that handle MMS-over-HTTP
- The `mcast_mms` processes that handle MMS requests over IP multicast

In the ACNS 5.2.1 software and later releases, standalone Content Engines use the Fast Start and Fast Cache features to stream live stream-split content or on-demand (cache-hit) content to the client; a Windows Media 9 server cannot stream content to a Content Engine using Fast Start or Fast Cache.

About WMT Streaming and Caching Services with Standalone Content Engines

The term *WMT streaming and caching services* is used to refer collectively to the two groups of WMT services:

- WMT MMS services (supported in the ACNS 5.2.1 software to ACNS 5.4.x software releases)
- WMT RTSP services for WMS 9 clients and servers (supported in the ACNS 5.3.1 software and later releases)

[Table 9-2](#) lists the types of WMT streaming and caching services that are supported with standalone Content Engines that are running the ACNS 5.2.1 software and later releases.

Table 9-2 WMT Streaming and Caching Services with Standalone Content Engines

Operation	Description
WMT proxy caching	The Content Engine receives WMT requests directly from a Windows Media player. The Content Engine retrieves the requested content if it is not already stored in its local cache, stores a copy locally whenever possible, and sends the requested content to the client. For more information, see the “About WMT Proxy Caching” section on page 9-6.
WMT transparent caching	The Content Engine receives WMT requests that are transparently redirected to it by a WCCP Version 2 router or a Layer 4 switch. The Content Engine retrieves the requested content if it is not already stored in its local cache, stores a copy locally whenever possible, and sends the requested content to the client. For more information, see the “About WMT Transparent Caching” section on page 9-7.
Distribution of WMT live streams (common)	The Content Engine serves WMT live streams to all local users (Windows Media players) whose WMT traffic it receives. The WMT live streams can be unicast or multicast live feeds. The Content Engine splits the live feeds into multicast or unicast to relay the stream to the WMT client. For more information, see the “About Live Splitting with WMT” section on page 9-8.
Distribution of preloaded VOD files (rare)	VOD files are preloaded on the Content Engine for on-demand delivery of these files to the Windows Media players. VOD caching is similar to HTTP caching; however, VOD files are cached in a different file system (mediafs) on the standalone Content Engine. To configure a standalone Content Engine to distribute VOD files, follow these steps: <ol style="list-style-type: none"> 1. Preload the VOD files on this Content Engine, as described in the “Configuring Content Preloading for Standalone Content Engines” section on page 11-2. 2. Publish the URLs of the preloaded VOD files that clients can now access through their WMT media players.

About Caching Policies in WMT Streaming Media Caching

In contrast to HTTP caching, caching policies in WMT streaming media caching are much simpler, because streaming media is mostly large static content. The caching policy in WMT caching is straightforward. All responses are cacheable, including partial responses. All WMT requests result in communication between the Content Engine and the origin 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.

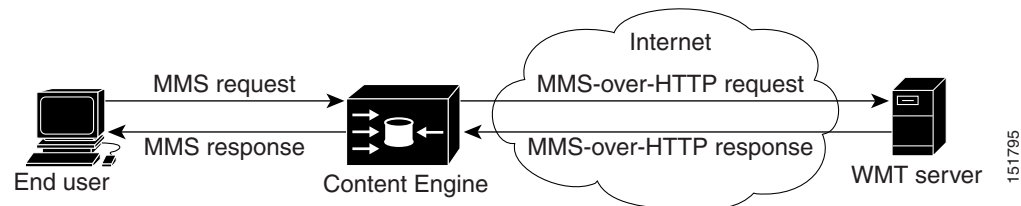
About WMT Proxy Caching

If direct proxy routing is being used to direct WMT requests to the standalone Content Engine, then you can configure the Content Engine to support WMT proxy caching. In direct proxy mode, the standalone WMT-enabled Content Engine accepts incoming WMT streaming requests directly from WMT clients

(end users who are using the Windows Media player to request WMT content) and acts on behalf of these clients, communicating with the origin WMT server. This type of caching is referred to as *WMT proxy caching*.

If the client is a Windows Media 6.4 or 7.0 player, the Content Engine accepts and serves the streaming requests over MMS-over-HTTP. (See [Figure 9-2](#).)

Figure 9-2 WMT MMS Proxy Caching (Direct Proxy Routing)



If the client is a Windows Media 9 player and the Content Engine is running the ACNS 5.3.1 software and later releases, the Content Engine can accept and serve the streaming request over RTSP and HTTP. (See [Figure 9-1](#).)



Note

If a firewall is positioned between a Content Engine and a requesting client, make sure that you assign the external IP address of the Content Engine when explicitly configuring the Windows Media player proxy settings on the end users' desktops to point to directly to this Content Engine.

You can use the Setup utility or the Content Engine CLI to enable and configure WMT proxy caching on a standalone Content Engine that is running the ACNS 5.2.1 software and later releases.

- For information about how to use the Setup utility to configure WMT proxy caching on a standalone Content Engine, see the [“Using the Setup Utility to Configure a Basic Configuration on a Standalone Content Engine”](#) section on page 4-21.
- For information about how to use the Content Engine CLI to configure WMT proxy caching, see the [“Enabling and Configuring WMT Caching on Standalone Content Engines”](#) section on page 9-34.

About WMT Transparent Caching

If WMT transparent redirection (WCCP or Layer 4 switching) is being used to direct WMT requests to a standalone Content Engine, then you can configure the Content Engine to support WMT transparent caching. In this case, the standalone Content Engine is acting as a transparent proxy server for clients who are requesting WMT content, and the Content Engine is not visible to these clients. After receiving a transparently redirected WMT request, the Content Engine retrieves the requested content if it is not already stored in its local cache, stores a copy locally whenever possible, and sends the requested content to the client media player.

You can use the Setup utility or the Content Engine CLI to enable and configure WMT transparent caching on a standalone Content Engine that is running the ACNS 5.2.1 software and later releases. If you use the Setup utility, you can only configure the Content Engine to accept redirected WMT requests from WCCP Version 2 routers. If you use the Content Engine CLI, you can configure the Content Engine to accept redirected WMT requests from Layer 4 switches as well as from WCCP Version 2 routers.

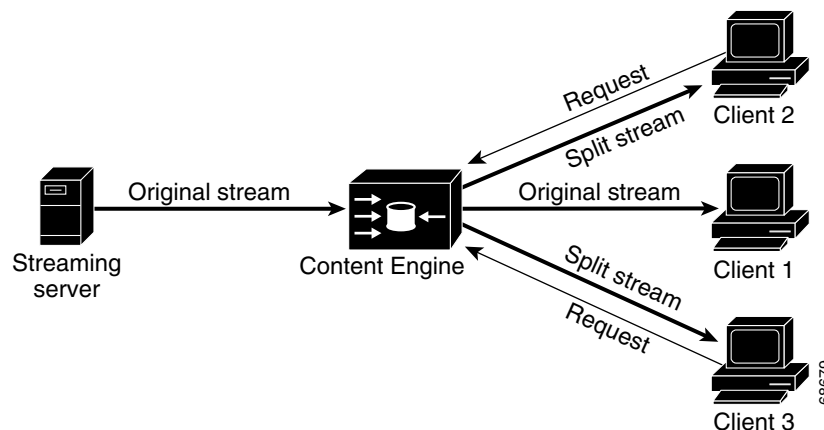
For information about how to use the Setup utility to configure WMT transparent caching on a standalone Content Engine, see the [“Using the Setup Utility to Configure a Basic Configuration on a Standalone Content Engine”](#) section on page 4-21. For information about how to configure WMT transparent caching on a standalone Content Engine through the Content Engine CLI, see the [“Enabling and Configuring WMT Transparent Caching on Standalone Content Engines”](#) section on page 9-35.

About Live Splitting with WMT

The WMT-enabled Content Engine also supports live splitting. By splitting requests for live streams. A single stream from the origin streaming server is split to serve each client that requested the stream. (See [Figure 9-3](#).) If a WMT client requests a publishing point on a remote streaming server without specifying an ASF file, the Content Engine dynamically creates an alias file that references the remote streaming server. All further requests to that remote streaming server are served by having the Content Engine split the stream and serve it to the WMT clients.

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 9-3 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

Live splitting is supported for different data packet transport protocols (HTTP and RTSP). In the ACNS 5.3.1 software and later releases, live splitting for two additional transport protocols, RTSPU and RTSPPT, is available. To display aggregated statistics about WMT live streams, enter the **show statistics wmt streamstat live EXEC** command.

About Proxy Authentication for a WMT-Enabled Content Engine

The WMT-enabled Content Engine supports both basic and NTLM authentication by the origin server. When a client requests content that needs user authentication, the Content Engine acts as an agent, conveying the authentication information to and from the client and server to authenticate the client. Once the client is authenticated, the content is streamed as usual. The authentication is performed for both cached content as well as noncached VOD content.

The following are the three types of proxy authentication methods:

- **Basic authentication**—An authentication scheme in which the server requests the client's identification in the form of an encoded username and password. If the authentication fails, the client is informed accordingly, in which case the client retries or disconnects. If the authentication is successful, then the streaming media is served to the client. This is supported in nontransparent proxy mode (direct proxy routing) as well as transparent proxy mode, over HTTP.
- **Windows NTLM authentication**—A connection-based challenge-response authentication scheme. Because the NTLM protocol authenticates every connection, the proxy cannot arbitrarily create new connections with the origin server, and the proxy must reuse connections initiated by the client.

A file is served from the cache only if it is a complete cache hit; that is, the complete file is present on disk. If the file is not a complete hit, then the entire file is fetched from the origin server in the case of NTLM. NTLM authentication is supported in nontransparent proxy mode (direct proxy routing) as well as transparent proxy mode, over HTTP. The proxy supports caching and delivery of Digital Rights Management (DRM)-protected Windows Media files. Access control lists (ACLs) enforced by the origin server are automatically enforced by the proxy.

- **Microsoft Digest authentication**—An authentication method in which an initial authentication of the client is performed when the server receives the first challenge response from the client. After the server verifies that the client has not been authenticated yet, it accesses the services of a domain controller (DC) to perform the initial authentication of the client. When the initial authentication of the client is successfully completed, the server receives a Digest session key. The server caches the session key and uses it to authenticate subsequent requests for resources from the authenticated client. This is a connection-based challenge-response authentication scheme similar to NTLM authentication. This authentication scheme is supported in nontransparent proxy mode (direct proxy routing) as well as transparent proxy mode over HTTP.



Note

Filtering based on user identification is also supported. The proxy only supports authentication by the origin server. Proxy authorization, or authentication of the user to use the proxy, will be supported in a future release. Live streams that are split to clients are also authenticated with the origin server in the ACNS 5.1 software and later releases.

In the ACNS 5.3.1 software and later releases, pass-through authentication support for WMT RTSP requests from Windows Media 9 players is available. For more information on this topic, see the [“Configuring Pass-Through Authentication for WMT Requests”](#) section on page 10-7.

Configuration Guidelines

This section provides some general guidelines for configuring Windows media streaming and caching services on standalone Content Engines, and then provides instructions for configuring describes how to configure Windows Media services for standalone Content Engines.

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

- **Windows Media Services 9 Series** is a set of streaming solutions for creating, distributing, and playing back digital media files on the Internet. WMT includes the end user application (Windows Media 9 players), the server and distribution application (Windows Media 9 server) and the encoder application (Windows Media Encoder).

- If the WMT proxy server fails to serve a request that uses MMS-over-HTTP, the Windows Media 9 player will bypass the proxy and serve the request from the origin server. Previous versions of the Windows Media players (Version 6.4 and 7.0) did not support this feature. Typically, proxy servers fail to serve a request for one of these reasons:
 - The requested media file exceeds the configured values in the Content Engine (bandwidth, maximum number of sessions, or maximum bit rate).
 - The URL fails to comply with the rules or URL filter configured in the Content Engine.
 - The proxy server is down.

Table 9-3 lists the supporting WMT incoming and outgoing proxy modes. As this table indicates, the modes vary based on the release of the ACNS 5.x software that is running on the Content Engine.

Table 9-3 Supported WMT Incoming and Outgoing Proxy Modes

Proxy Mode	ACNS 5.3.1 Software or Later	ACNS 5.2.1 Software or Earlier	Comment
Incoming proxy mode	<ul style="list-style-type: none"> • MMS transparent proxy (WCCP transparent redirection through services 81 and 82). <p>Note ACNS 5.5.x software does not support MMS.</p> <ul style="list-style-type: none"> • WMT RTSP transparent proxy (WCCP transparent redirection through services 80 and 83) 	WMT transparent proxy (WCCP transparent redirection through service 82 and 83)	For more information about WMT RTSP transparent redirection, see the “Configuring RTSP Transparent Redirection of WMT Requests” section on page 9-31.
Outgoing proxy mode	<ul style="list-style-type: none"> • MMS-over-HTTP proxy mode • RTSP proxy mode 	<ul style="list-style-type: none"> • MMS-over-HTTP proxy mode 	Use the wmt proxy outgoing global configuration command to configure a WMT outgoing proxy on the Content Engine. In the ACNS 5.3.1 software and later releases, support for an RTSP outgoing proxy server is also available (the wmt proxy outgoing rtsp host command) for WMT RTSP requests from Windows Media 9 players. For examples of how to configure an outgoing proxy server, see Step 10 of the “Configuring General WMT Settings on Standalone Content Engines” section on page 9-18

- You can configure numerous WMT features with the **wmt** global configuration command.

```
ContentEngine(config)# wmt ?
  accelerate                WMT streaming acceleration
  accept-license-agreement  Accept license; View by 'show wmt license-agreement'
  advanced                  WMT advanced configuration
  bandwidth                 WMT bandwidth configurations
  broadcast                  Broadcast live configuration.
  cache                     WMT cache config
  disallowed-client-protocols Specify disallowed wmt client protocols
  enable                    Enable WMT
  evaluate                  Start/continue 60-day evaluation of WMT.
  extended                  WMT extended configurations
  fast-cache                Fast-cache feature
  fast-start                 Fast-start feature
  http                      MMS over HTTP configurations
  incoming                  Configuration for incoming WMT requests
  l4-switch                  Configure layer-4 switch interoperability
  license-key                Required license key for WMT
  live-url-stripping         Strip live URL's ? and beyond
  max-concurrent-sessions   Maximum number of unicast clients that can be
                           served concurrently.
  multicast                  Multicast configuration and scheduling.
  proxy                      Out-going proxy configuration
  transaction-logs           WMT transaction log configuration
```

For an example of how to use the **wmt** global configuration command to configure WMT general settings, see the [“Configuring General WMT Settings on Standalone Content Engines”](#) section on page 9-18.

WMT Proxy Server Requirements

The following are requirements for a standalone Content Engine that will be functioning as a WMT proxy server:

- Interoperability is the most important requirement for WMT software components. The WMT proxy server is required to work with all versions of Microsoft Windows Media player, Windows Media Encoder, and third-party Windows Media applications.
- In order to support WMT transparent caching, WCCP Version 2 must be running on the standalone Content Engine.
- You must configure disk space to include mediafs storage with the **disk config** command before you can cache streaming media using WMT.
- The mediafs partitions is mounted on the standalone Content Engine. This is the storage partition that is used to store any WMT streaming media content that is cached on the Content Engine.
- The Content Engine is running the ACNS 5.2.1 software and later releases.
- You have a Microsoft WMT license key. The Microsoft WMT product is licensed software. To enable the licensed WMT product feature on a standalone Content Engine, you must have a WMT license key, which is supplied on a certificate shipped with the Content Engine. For information about how to specify the WMT license key, see the [“Enabling WMT Licenses on Standalone Content Engines”](#) section on page 9-17.



Note If you are downloading the ACNS 5.x software, you can purchase a WMT license though the Cisco.com website.

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



Note You must disable the WMT feature using the **no wmt enable** global configuration command before uninstalling the WMT license key on a standalone Content Engine.

- You have the IP address of the standalone Content Engine that will be configured as a WMT proxy server.
- You have the IP address of the WCCP Version 2-enabled routers if you want to use transparent WCCP redirection.

Checklist for Configuring WMT Streaming and Caching Services on Standalone Content Engines

Table 9-4 is a checklist of tasks for configuring WMT streaming and caching services on 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, as well as how to configure how WMT requests are routed to this standalone Content Engine.



Note The Setup utility allows you to enable WMT on a standalone Content Engine that is running the ACNS 5.2 .1 software and later releases, and then configure WMT proxy caching and WMT transparent caching on the Content Engine. For information on this topic, see the “[Configuring a Basic Configuration on Standalone Content Engines with the Setup Utility](#)” section on page 4-10.

Table 9-4 Checklist for Configuring WMT MMS Services with Standalone Content Engines

Task	Additional Information and Instructions
<ol style="list-style-type: none"> 1. Enable WMT on the standalone Content Engine. <ol style="list-style-type: none"> a. Accept the WMT license agreement. b. Accept the evaluation WMT license, or specify your Cisco permanent WMT license. c. Enable the licensed WMT feature on the standalone Content Engine. 	See the “ Enabling WMT Licenses on Standalone Content Engines ” section on page 9-17.
<ol style="list-style-type: none"> 2. Configure one or more of the following routing methods to direct client requests for Windows Media content to this standalone Content Engine: <ul style="list-style-type: none"> – Direct proxy routing (nontransparent) – Transparent redirection (WCCP Version 2 routing or Layer 4 switching) 	With direct proxy routing, the Windows Media players send their WMT requests directly to this Content Engine (nontransparent forward proxy server). With direct proxy routing, you must point the Windows Media players directly to the Content Engine, as described in the “ Pointing Windows Media Players Directly to a Standalone Content Engine for WMT MMS Requests ” section on page 4-45.

Table 9-4 Checklist for Configuring WMT MMS Services with Standalone Content Engines (continued)

Task	Additional Information and Instructions
3. For direct proxy routing, enable and configure WMT proxy caching on the Content Engine.	See the “ Enabling and Configuring Nontransparent WMT Proxy Caching on Standalone Content Engines ” section on page 9-34.
4. For transparent redirection, enable and configure WMT transparent caching on the Content Engine.	See the “ Enabling and Configuring WMT Transparent Caching on Standalone Content Engines ” section on page 9-35.
5. Choose which types of WMT streaming content that this standalone Content Engine will be distributing to clients: <ul style="list-style-type: none"> <li data-bbox="159 604 532 636">– Video-on-demand (VOD) files <li data-bbox="159 646 410 678">– Live WMT streams 	<p>For VOD files, see the “Configuring Standalone Content Engines to Distribute VOD Files” section on page 9-35.</p> <p>For live WMT streams, choose which methods this Content Engine will use to relay live WMT streams to Windows Media clients:</p> <ul style="list-style-type: none"> <li data-bbox="776 716 1360 747">• If multicast out will be used, then to go to task 6. <li data-bbox="808 762 841 793">or <li data-bbox="776 804 1312 835">• If unicast out will be used, then go to task 7.
6. Configure the Content Engine to relay live content to Windows Media clients through multicasting.	See the “ Configuring Standalone Content Engines to Deliver WMT Live Streams ” section on page 9-37.
7. Configure the Content Engine to relay live content to Windows Media clients through unicast.	<p>See the “Configuring Multicast-In Unicast-Out on Standalone Content Engines” section on page 9-43.</p> <p>See the “Configuring Unicast-In Unicast-Out on Standalone Content Engines” section on page 9-44.</p>

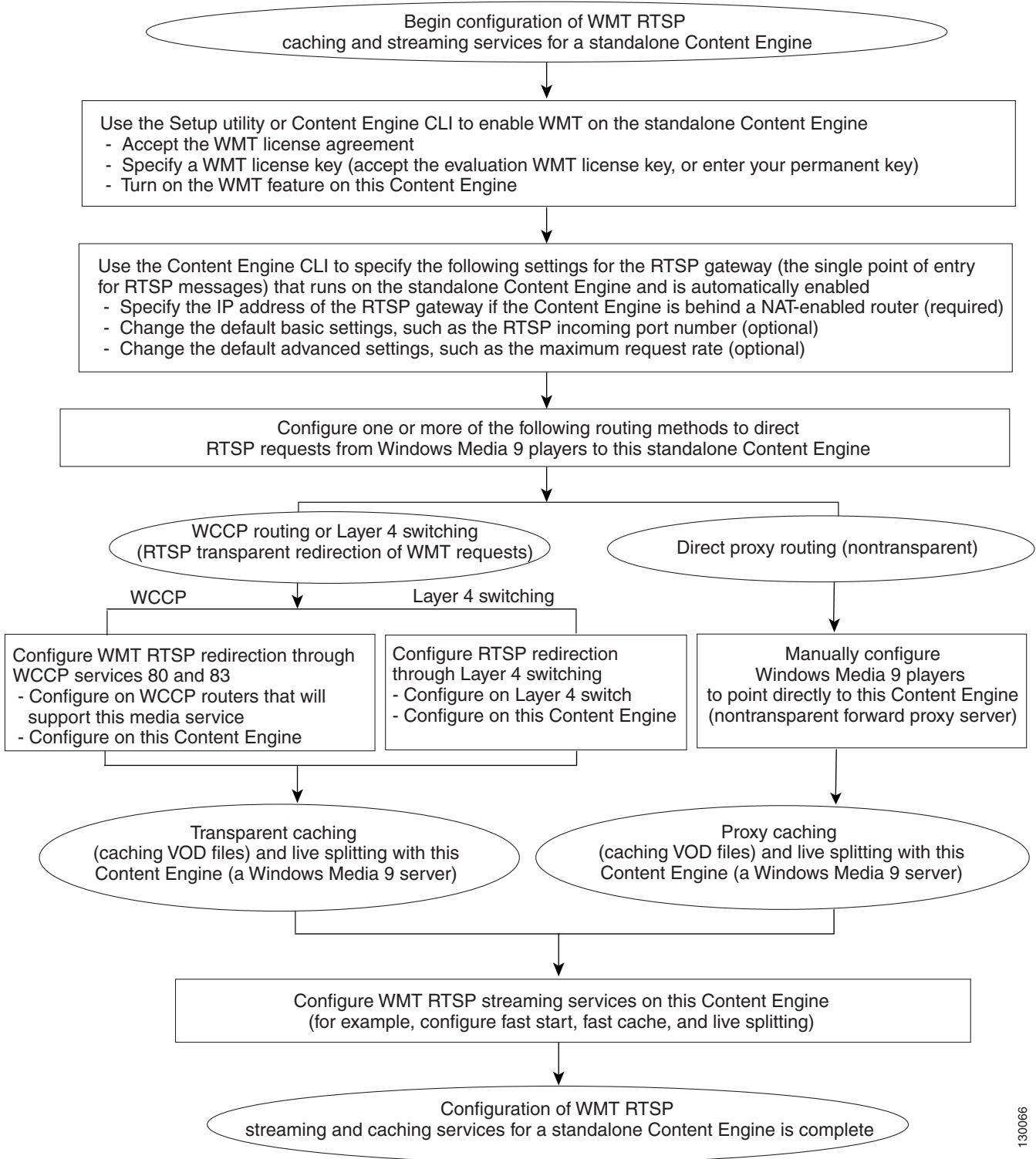
Configuring WMT RTSP Streaming and Caching Services on Standalone Content Engines

This section describes how to use the Content Engine CLI to configure WMT RTSP streaming and caching services on a standalone Content Engine that is running the ACNS 5.3.1 software and later releases.

[Figure 9-4](#) provides a detailed view on how to configure these services initially for standalone Content Engines. [Table 9-4](#) provides a checklist of tasks for configuring these services on a standalone Content Engine.

This section also describes how to perform the necessary configuration changes to the Windows Media 9 players (if direct proxy routing is to be used) and the necessary configuration changes to WCCP Version 2 routers (if WMT RTSP redirection through WCCP [services 80 and 83] will be used).

Figure 9-4 Configuring WMT RTSP Streaming and Caching Services with Standalone Content Engines



130066

After you have configured the WMT RTSP caching services as depicted in [Figure 9-4](#), you use the exact same procedure to configure WMT streaming services regardless of whether the MMS-over-HTTP or the RTSP protocol will be used to deliver the WMT streaming content (live WMT streams and VOD files) to the WMT clients.

Checklist for Configuring WMT RTSP Streaming and Caching Services on Standalone Content Engines

[Table 9-5](#) is a checklist of tasks for configuring WMT RTSP streaming and caching services on standalone Content Engines that are running the ACNS 5.3.1 software and later releases. This checklist includes the steps involved in configuring these services on a standalone Content Engine, as well as how to configure how the RTSP requests from Windows Media 9 players are routed to this standalone Content Engine that is functioning as a Windows Media 9 server.

Table 9-5 Checklist for Configuring WMT RTSP Services with Standalone Content Engines and Windows Media 9 Players

Task	Additional Information and Instructions
<ol style="list-style-type: none"> 1. Enable WMT on the standalone Content Engine. <ol style="list-style-type: none"> a. Accept the WMT license agreement. b. Accept the evaluation WMT license, or specify your Cisco permanent WMT license. c. Enable the licensed WMT feature on the standalone Content Engine. 	See the “Enabling WMT Licenses on Standalone Content Engines” section on page 9-17.
<ol style="list-style-type: none"> 2. If necessary, specify the RTSP gateway settings. <ol style="list-style-type: none"> a. If the Content Engine is behind a NAT-enabled router, you must specify the IP address of the RTSP gateway (required). b. You can also change the default basic and advanced RTSP gateway settings (optional). 	See the “Configuring the RTSP Gateway for Standalone Content Engines” section on page 8-14.
<ol style="list-style-type: none"> 3. Configure one or more of the following routing methods to direct RTSP requests from Windows Media 9 players to this standalone Content Engine: <ul style="list-style-type: none"> – Direct proxy routing (nontransparent) – WMT RTSP transparent redirection (WCCP Version 2 routing or Layer 4 switching) 	<p>With direct proxy routing, the Windows Media 9 players send their WMT RTSP requests directly to this Content Engine (nontransparent forward proxy server). With direct proxy routing, you must point the Windows Media 9 players directly to the Content Engine, as described in the “Pointing Windows Media 9 Players Directly to a Standalone Content Engine for WMT RTSP Requests” section on page 4-43.</p> <p>With WCCP routing or Layer 4 switching, you must configure the WCCP routers or Layer 4 switches and the Content Engine (transparent proxy server) for WMT RTSP transparent redirection, as described in the “Configuring RTSP Transparent Redirection of WMT Requests” section on page 9-31.</p>
<ol style="list-style-type: none"> 4. For direct proxy routing, enable and configure WMT proxy caching on the Content Engine. 	See the “Enabling and Configuring Nontransparent WMT Proxy Caching on Standalone Content Engines” section on page 9-34.
<ol style="list-style-type: none"> 5. For transparent redirection, enable and configure WMT transparent caching on the Content Engine. 	See the “Enabling and Configuring WMT Transparent Caching on Standalone Content Engines” section on page 9-35.

Table 9-5 Checklist for Configuring WMT RTSP Services with Standalone Content Engines and Windows Media 9 Players (continued)

Task	Additional Information and Instructions
<p>6. Choose which types of WMT streaming content that this standalone Content Engine will be distributing to clients:</p> <ul style="list-style-type: none"> – Video-on-demand (VOD) files – Live WMT streams 	<p>For VOD files, see the “Configuring Standalone Content Engines to Distribute VOD Files” section on page 9-35.</p> <p>For live WMT streams, choose which methods this Content Engine will use to relay live WMT streams to Windows Media clients:</p> <ul style="list-style-type: none"> • If multicast out will be used, then to go to task 7. or • If unicast out will be used, then go to task 8.
7. Configure the Content Engine to relay live content to Windows Media 9 players through multicasting.	See the “ Configuring Standalone Content Engines to Deliver WMT Live Streams ” section on page 9-37.
8. Configure the Content Engine to relay live content to Windows Media 9 players through unicast.	<p>See the “Configuring Multicast-In Unicast-Out on Standalone Content Engines” section on page 9-43.</p> <p>See the “Configuring Unicast-In Unicast-Out on Standalone Content Engines” section on page 9-44.</p>

Enabling WMT Licenses on Standalone Content Engines

Before enabling licenses for WMS 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. To display the system clock, use the **show clock EXEC** command. To set the system clock, use the **clock set EXEC** command.

To use the Content Engine CLI to enable Windows Media Services on a standalone Content Engine, follow these steps:

Step 1 View the WMT license agreement.

```
ContentEngine# show wmt license-agreement
```

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

```
ContentEngine# configure terminal
ContentEngine(config)# wmt accept-license-agreement
```

Step 3 Enter your Cisco license key for the WMT product.

```
ContentEngine(config)# wmt license-key licensekey
```

Alternatively, accept an evaluation WMT license.

```
ContentEngine(config)# wmt evaluate
```

Step 4 Enable the WMT feature on this Content Engine.

```
ContentEngine(config)# wmt enable
```

Step 5 When asked if you want to proceed, enter **yes** to proceed.

This operation needs to restart http proxy and real proxy (if running) for memory reconfiguration. Proceed? [no] **yes**

The next step is to choose one or more of the following routing methods to direct client requests for Windows Media content to this standalone Content Engine:

- WCCP routing or Layer 4 switch (WMT transparent redirection for WMT MMS requests, and WMT RTSP transparent redirection for WMT RTSP requests)
- Direct proxy routing (nontransparent)

With direct proxy routing, the Windows Media players send their requests directly to this Content Engine (acting as a nontransparent forward proxy server). For instructions on how to configure a Windows Media player on the end user desktops to point directly to this Content Engine as their proxy server, see the [“Pointing Windows Media Players Directly to a Standalone Content Engine for WMT MMS Requests”](#) section on page 4-45.

Configuring General WMT Settings on Standalone Content Engines

To configure the general WMT settings on a standalone Content Engine through the Content Engine CLI, follow these steps:

Step 1 Disallow specific WMT client protocols for streaming with the **wmt disallowed-client-protocols** global configuration command:

```
wmt disallowed-client-protocols {http | rtspt | rtspu}
```

In the ACNS 5.3.1 software release, the following changes were made to this command:

- The **rtspt** and **rtspu** options were added.
- The **tcp** and **udp** options are hidden for backward compatibility.

Parameter	Description
disallowed-client-protocols	Specifies disallowed WMT client protocols.
http	Disallows streaming over the HTTP protocol (http://).
rtspt	Disallows streaming over the rtspt protocol (rtspt://).
rtspu	Disallows streaming over the rtspu protocol (rtspu://).

Step 2 Configure the maximum number of unicast clients that a standalone Content Engine can support concurrently. The default is 2500 clients.

```
ContentEngine(config)# wmt max-concurrent-sessions number
```

number specifies the maximum number of incoming unicast requests that the Content Engine should serve concurrently. This limit is subject to physical resources on the Content Engine (1 to 8000).

Step 3 Specify the maximum bandwidth for preloading WMT content on the Content Engine.

```
ContentEngine(config)# wmt max-bandwidth incoming bitrate
```

With the ACNS 5.x software, you can preload WMT streaming media files that may have different bit rates at the URL specified for content preloading. You can also control WMT bandwidth and bit rates using the **wmt max-bandwidth** and **wmt max-bitrate** global configuration commands.

- Step 4** Specify the maximum bit rate per WMT stream that can be received by the Content Engine.

By default, there is no limit. This bit rate is called the *WMT incoming stream bitrate*.

```
ContentEngine(config)# wmt bitrate wmt incoming bitrate
```

bitrate specifies the WMT incoming stream bit rate in kilobits per second. This value can be from 0–2147483647.

- Step 5** Specify the maximum bit rate per WMT stream that can be served by the Content Engine. By default, there is no limit. This bit rate is called the *WMT outgoing stream bit rate*.

```
ContentEngine(config)# wmt bitrate wmt incoming bitrate
```

bitrate specifies the WMT outgoing stream bit rate in kilobits per second. This value can be from 0 to 2147483647.

For more information about configuring bandwidth and bit rates, see the “[Configuring Incoming and Outgoing WMT Bandwidth and Bit Rates](#)” section on page 9-23.

- Step 6** Specify the maximum size of a single object that the Content Engine should store in its WMT cache.

Use the **wmt cache max-obj-size** global configuration command to specify this value. The range of values is between 1 and 1,000,000 megabytes. The default value is 1024 megabytes.

```
ContentEngine(config)# wmt cache max-obj-size size
```

- Step 7** Enable WMT caching on the standalone Content Engine if it is not already enabled.

```
ContentEngine(config)# wmt cache enable
```

- Step 8** Enable WMT live URL stripping.

```
ContentEngine(config)# wmt live-url-stripping enable
```

- Step 9** Enable transparent redirection through a Layer 4 switch instead of through a WCCP router:

- a. On the Content Engine, enable transparent redirection of MMS requests through a Layer 4 switch.

```
ContentEngine(config)# wmt 14-switch enable
```

- b. On the Content Engine, enable transparent redirection of RTSP requests through a Layer 4 switch.

```
ContentEngine(config)# rtsp 14-switch enable
```

- Step 10** To configure a WMT outgoing proxy on the Content Engine, use the **wmt proxy outgoing** global configuration command.

Configure a Content Engine to send all of its MMS cache miss traffic to a specific MMS outgoing proxy server, or send its MMS-over-HTTP miss traffic to a specific HTTP outgoing proxy server, or send all of its WMT RTSP cache miss requests to a specific RTSP outgoing proxy server without using ICP or WCCP. In the ACNS 5.3.1 software and later releases, you can also configure an RTSP outgoing proxy server for WMT RTSP requests from Windows Media 9 players.

The command syntax is as follows:

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

where:

- **http** is the keyword for an outgoing MMS-over-HTTP proxy configuration.
- **rtsp** is the keyword for an outgoing RTSP proxy configuration.
- **host** is the keyword for the outgoing proxy server.
- *hostname* is the hostname or *ip-address* is the IP address of the outgoing proxy server.

In the following example, a Content Engine at a branch office is configured to use MMS-over-HTTP to send all its WMT cache miss requests to a central Content Engine at 172.16.30.30 through port 8080:

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

In the following example, a Content Engine at a branch office is configured to send all its MMS cache miss requests to a central Content Engine at 172.16.30.31 through port 1700:

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

In the following example, the Content Engine at a branch office is configured to use MMS-over-TCP (MMST) or MMS-over-UDP (MMSU) to send all its cache miss RTSP requests to a central Content Engine at 172.16.30.30 through port 8080:

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

In the following example, the Content Engine at a branch office is configured to use RTSP to send all its cache miss RTSP requests from Windows Media 9 players to a central Content Engine at 172.16.30.30 through port 8080:

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

Step 11 Decide which type of media file should be served by WMT.

Typically, Content Engines are shipped with a default list of filename extensions to be served by WMT. The default list in the Content Engine contains the following filename extensions: asf, none, nsc, wma, and wmv. The default list of filename extensions includes *none* in order to enable a Content Engine to serve media files without file extensions (for example, broadcast aliases or URLs of live encoders). The filename extension *nsc* is included in the list to enable a Content Engine to multicast media files.

- To add filename extensions to this list, use the **wmt http allow extension** global configuration command.
- To remove a filename extension from the list, use the **no wmt http allow extension** global configuration command.



Note In the ACNS 5.2.1 software release, the **wmt mms allow extension** EXEC command was replaced with the **wmt http allow extension** EXEC command. The **show wmt mms allow extension** EXEC command was also replaced with the **show wmt http allow extension** EXEC command.

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

- You cannot have more than 20 extensions in the list of allowed file extensions.
- File extensions must be alphanumeric, and the first character of every extension should be an alphabetic one.
- You cannot have more than 10 characters in a file extension.

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

```
ContentEngine(config)# wmt http allow extension mp3
ContentEngine(config)#
```

Step 12 View the file extensions included in the list after you add or delete file extensions.

```
ContentEngine(config)# exit
ContentEngine# show wmt http allow extension
```

The **show wmt http allow extension EXEC** command does not display anything if you have not modified the default list.

Step 13 (Optional) Disable one or more of the following three WMT streaming acceleration features that by default are enabled on the Content Engine:

- Live split
- Proxy-cache
- VOD

Use the appropriate **no wmt accelerate** global configuration command to disable the feature.

- To disable the acceleration of live splitting, enter the **no wmt accelerate live-split** command. To reenable this feature on the Content Engine, enter the **wmt accelerate live-split enable** command.
- To disable the acceleration of proxy caching, enter the **no wmt accelerate proxy-cache** command. To reenable this feature on the Content Engine, enter the **wmt accelerate proxy-cache enable** command.
- To disable the acceleration of serving VOD files to WMT clients, enter the **no wmt accelerate vod** command. To reenable this feature on the Content Engine, enter the **wmt accelerate VOD enable** command.

Step 14 Configure the WMT Fast Streaming features on the Content Engine:

- Specify the maximum burst bandwidth (in kilobits per second [kbps]) for the Fast Start feature. This value specifies the maximum burst bandwidth that a single player can use for accelerated initial buffering of the streaming content. For example:

```
ContentEngine(config)# wmt fast-start max-bandwidth 3000
```

The Fast Start feature allows the Windows Media 9 server to push the beginning portions of a stream to the Windows Media 9 player at the maximum available bandwidth. This feature is enabled on a Content Engine by default. The increased bandwidth that this feature initially uses to send data to the Windows Media 9 player can overburden a network if many players connect to the stream at the same time. The maximum burst bandwidth can be from 1 to 65535 kbps. The default is 3600. The maximum value is associated with the WMT license. By default, the Fast Start feature is enabled on the Content Engine. For more information, see the [“Configuring Fast Start on Standalone Content Engines” section on page 9-28](#).

- Specify the maximum delivery rate (maximum acceleration factor) for the Fast Cache feature. For example:

```
ContentEngine(config)# wmt fast-cache max-delivery-rate 5
```

The Fast Cache feature allows the stream rendering rate to be decoupled from the stream delivery rate on the network. This allows a Windows Media 9 server to send the stream content faster than the client’s rendering speed. The maximum delivery rate (that is, the Fast-Cache speed multiplier) can be from 1 to 65535. (Setting the **max-delivery-rate** value to 1 is equivalent to disabling the Fast

Cache feature.) By default, the Fast Cache feature is enabled on the Content Engine. For more information, see the [“Configuring Fast Cache on Standalone Content Engines” section on page 9-29](#).

Step 15 Configure the WMT advanced client features on the Content Engine:

- a. Specify the client maximum packet size (WMT maximum IP packet size) in bytes. The value can be from 512 to 2048 bytes. The default is 1500 bytes. For example:

```
ContentEngine(config)# wmt advanced client maximum-packet-size 1800
```

- b. Specify the maximum amount of time that the Content Engine is to wait for a response from a WMT client before timing out the connection. The value can be from 30 to 300 seconds. The default timeout is 120 seconds. For example:

```
ContentEngine(config)# wmt advanced client idle-timeout 100
```



Note These two **wmt advanced client** global configuration commands are available in the ACNS 5.3.1 software and later releases.

Step 16 Decide whether you want this Content Engine to forward its WMT logs to the upstream server (that is, a Windows Media server or another Content Engine.)

By default, Content Engines that are running the ACNS 5.3.1 software and later releases will forward their WMT logs to the upstream sever. This feature applies to all of the supported protocols (for example, HTTP and RTSP [RTSPT and RTPU]).

To disable this feature and configure the Content Engine to not forward its WMT logs to the upstream server, enter the **no wmt advanced server log-forwarding enable** global configuration command. To reenale this feature, enter the **wmt advanced server log-forwarding enable** global configuration command.

Step 17 Configure WMT logging on the Content Engine. For more information, see the [“Using WMT Logging with Standalone Content Engines” section on page 9-47](#).

Step 18 (Optional) Preload VOD files on the Content Engine for on-demand delivery of these files to Windows Media clients. For more information, see the [“Configuring Standalone Content Engines to Distribute VOD Files” section on page 9-35](#).

Step 19 (Optional) Configure the standalone Content Engine to deliver WMT live streams to the WMT clients. For more information, see the [“Configuring Standalone Content Engines to Deliver WMT Live Streams” section on page 9-37](#).

Step 20 Display statistics about WMT requests that are being serviced by this Content Engine. For more information, see the [“Displaying Information about the WMT RTSP Server Configuration” section on page 9-46](#).

Step 21 Configure URL filtering for WMT requests that are directed to this Content Engine. For more information, see [Chapter 11, “Configuring Content Preloading and URL Filtering on Standalone Content Engines.”](#)

Step 22 Configure rules for WMT requests that are directed to this Content Engine. For more information, see [Chapter 13, “Configuring the Rules Template on Standalone Content Engines.”](#)

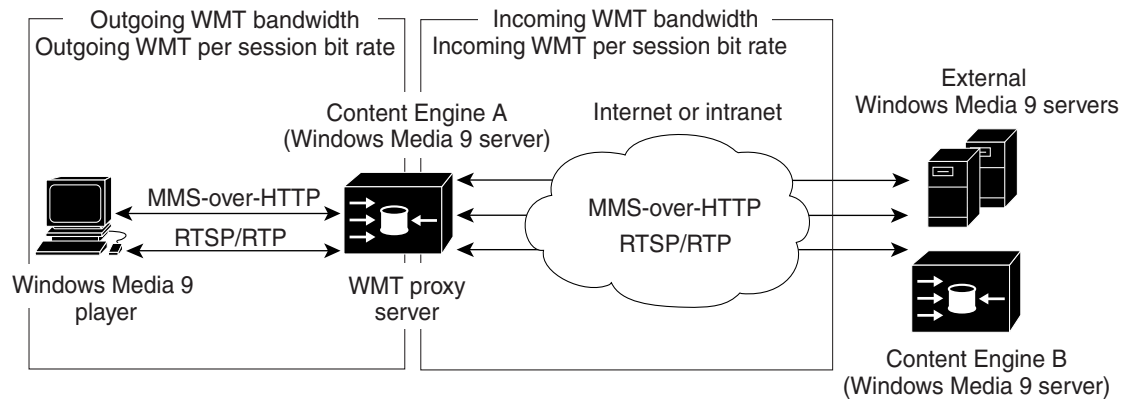
Step 23 Display statistics for the configured rules for WMT requests.

For example, enter the **show statistics rtsp EXEC** command to display statistics that are related to the configured RTSP and WMT RTSP rules. For more information, see the [“Displaying Statistics for Configured Rules” section on page 13-33](#).

Configuring Incoming and Outgoing WMT Bandwidth and Bit Rates

The bandwidth between the WMT client and the WMT proxy server (Content Engine A) is called *outgoing WMT bandwidth*. The bandwidth between the WMT proxy server (Content Engine A) and the Windows Media 9 server is called *incoming WMT bandwidth*. (See [Figure 9-5](#).)

Figure 9-5 Incoming and Outgoing Bandwidth and Bit Rates



Note

In the ACNS 5.3.1 software and later releases, you can configure bandwidth for MMS-over-HTTP and RTSP requests.

In addition to incoming and outgoing WMT bandwidth, there are incoming and outgoing WMT bit rates per session. When a WMT client requests media files for the first time, the WMT proxy server (that is, the Content Engine) caches the on-demand Windows Media files. All subsequent requests for the same file are served by the WMT proxy server from its cache. The WMT proxy server can also live split a broadcast; it can pull only one unicast stream for the origin streaming server (for example, an external Windows Media 9 server) and live split the broadcast to multiple WMT clients.

The bandwidth between the WMT proxy and the origin streaming server is called the *incoming bandwidth*. Because the bandwidth from the edge to the outside IP WAN is very limited, it is important that you specify a per-session limit (the maximum bit rate per request) for each service that is running on the Content Engine and that consumes incoming bandwidth (for example, the WMT streaming service), as well as an aggregate limit (the maximum incoming bandwidth.) The outgoing bandwidth needs to be controlled based on the WMT license that is configured on the Content Engine.

To specify a WMT incoming and outgoing bandwidth, use the **bandwidth wmt outgoing** and **bandwidth incoming** global configuration commands:

- To specify the outgoing WMT bandwidth in kbps, use the **bandwidth wmt outgoing kbps** global configuration command. This command sets the maximum bandwidth for WMT content that can be delivered to a client that is requesting WMT content. The range of values is between 0 and 2,147,483,647 Kbps.

If the specified outgoing bandwidth exceeds the limit specified by the WMT license then a warning message is displayed to inform you of this situation. However, the specified outgoing bandwidth setting is applied because you may have configured this setting before you enabled the initial WMT license or another WMT license that has a higher limit.

- To specify the incoming WMT bandwidth in Kbps, use the **bandwidth wmt incoming *kbps*** global configuration command. This command sets the maximum bandwidth for WMT content that can be delivered to a Content Engine from the origin streaming server or another Content Engine in the case of a cache miss. The specified bit rate is the maximum incoming WMT per session bit rate. The range of values is between 0 and 2,147,483,647 kbps. Incoming bandwidth applies to broadcast stations, multicast station, and VOD content from the origin server in the case of a cache miss.

Incoming bandwidth applies to the following:

- VOD content from the origin server in the case of a cache miss
- Broadcast stations in which the source for the broadcast station and multicast stations is a unicast (MMS-over-HTTP or RTSP-over-RTP) or a multicast. If the source is a multicast, the specified incoming bandwidth is not applied.
- Multicast stations in which the source of the multicast station is a unicast (RTSP) or a multicast.

To specify a WMT incoming and outgoing WMT per session bit rate, use **bitrate wmt incoming** and **bitrate wmt outgoing** global configuration commands:

- To specify the maximum incoming streaming bit rate per session that can be delivered to the WMT proxy server (a Content Engine) from the origin streaming server or another Content Engine in the case of a cache miss, use the **bitrate wmt incoming *bit-rate*** global configuration command. The specified bit rate is the maximum incoming WMT per session bit rate. The range of values is between 0 and 2,147,483,647 kbps. The default value is 0 (no bit rate limit).
- To set the maximum outgoing streaming bit rate per session that can be delivered to a client that is requesting WMT content, use the **bitrate wmt outgoing *bit-rate*** global configuration command. The specified bit rate is the maximum outgoing WMT per session bit rate. The range of values is between 0 and 2,147,483,647 kbps. The default value is 0 (no bit rate limit).

Outgoing bandwidth applies to the following:

- VOD content from the WMT proxy server on the Content Engine in the case of a cache miss.
- Broadcast stations and multicast stations that are configured on the Content Engine. The source for the broadcast station can be unicast (MMS [MMST and MMSU], MMS-over-HTTP, or RTSP-over-RTP) or multicast.

About Variable WMT Bit Rates

A content provider can create streaming media files at different bit rates to ensure that different clients who have different connections—for example, modem, DSL, or LAN—can choose a particular bit rate. The WMT caching proxy can cache multiple bit rate or variable bit rate (VBR) files, and based on the bit rate specified by the client, it serves the appropriate stream. Another advantage of creating variable bit rate files is that a single URL is all that must be specified for the delivery of streaming media.



Note

In the case of multiple bit rate files, the Content Engine that is acting as the WMT proxy server only retrieves the bit rate that the client has requested.

Configuring Subnet-Based Outgoing Bandwidth

In the ACNS 5.3.1 software and later releases, you can configure IP subnet-based bandwidth control for WMT requests. This feature allows you to specify the maximum bandwidth consumption for specific client IP subnets (that is, the aggregate bandwidth for the subnet). This bandwidth control feature is supported for WMT streaming through the following protocols: Windows Media 9 RTSP and HTTP.

You specify the rules for limiting subnet-based outgoing bandwidth in an XML configuration file. This configuration file is called the *advanced bandwidth configuration file*. For example, you may have three subnets (Subnet A that is the parent subnet, and Subnet B and C that are within Subnet A), and you specified three subnet-based bandwidth rules as follows:

- Rule A—Subnet A, 10.1.1.0/24, has been configured with an allow bandwidth of 10000 Kbps.
- Rule B—Subnet B, 10.1.1.0/25, has been configured with an allow bandwidth of 7000 Kbps.
- Rule C—Subnet C, 10.1.1.128/25, has been configured with an allow bandwidth of 5000 Kbps.

Even though the total allowed bandwidth of Subnet B and C is 12000 kbps (as defined by Rule B and C in the configuration file), the total bandwidth will not exceed 10000 kbps because of Rule A.

The following is an example of the format of the advanced bandwidth configuration file. This example also shows the required order of the lines in the advanced bandwidth configuration file:

XML Configuration File Format:

```
<?xml version="1.0"?>
<BandwidthSpec>
  <BandwidthRule>
    <ClientNetwork>10.77.140.133/32</ClientNetwork>
    <description>(Apply to PC jdoe-w2k)</description>
    <Allow limit="3000" service="wmt"/>
  </BandwidthRule>
  <BandwidthRule>
    <description>Comment (Apply to PCs in subnet 10.77.140.x)</description>
    <Allow limit="50000" service="wmt"/>
    <ClientNetwork>10.77.140.0/24</ClientNetwork>
  </BandwidthRule>
  <BandwidthRule>
    <Allow limit="1400" service="wmt"/>
    <ClientNetwork>10.1.1.1/32</ClientNetwork>
  </BandwidthRule>
  <Default limit="3000" service="wmt" />
</BandwidthSpec>
```

The following information applies to the format of this advanced bandwidth configuration file:

- The <description> tag is optional.
- The <ClientNetwork> - IPAddress/Netmask entry is a required field.
- If the <Allow limit> field is specified as **-1**, the bandwidth allowed is unlimited.
- The Service tag currently has only one supported option (the **wmt** option).
- The <Default> tag is optional. This tag is used to configure the default bandwidth. If none of the subnet bandwidth rules match, the default rule is applied if it is configured.

You use FTP to download this advanced bandwidth configuration file to the Content Engine so that the file is available in the local sysfs partition on the Content Engine.



Note

The **bandwidth wmt outgoing** global configuration command configures the total outgoing WMT bandwidth, which controls the total outgoing bandwidth used for WMT streaming; regardless of any subnet-based bandwidth configuration that is specified through the advanced bandwidth configuration file.

Specify the path of the advanced bandwidth configuration file, as follows:

```
ContentEngine(config)# bandwidth advanced config-file filename-path
```

Display the WMT server bandwidth allocation statistics, as follows:

```
ContentEngine # show statistics bandwidth advanced
```

The command output shows such information as the currently used bandwidth for each specified rule in the advanced configuration file and the currently available bandwidth for each specified rule in the configuration file.

Display statistics about bandwidth allocation errors, as follows:

```
ContentEngine # show statistics bandwidth advanced errors
```

The command output shows the time the allocation error occurred, the client from which the request was received, and the amount of allocated bandwidth that was requested.

Configuring a WMT Bandwidth Incoming Bypass List

To configure a WMT bandwidth incoming bypass list on standalone Content Engines, use the **wmt bandwidth incoming bypass-list** global configuration command.

The syntax of this command is as follows:

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

[Table 9-6](#) describes the command parameters.

Table 9-6 Parameters for the *wmt bandwidth incoming bypass-list* CLI Command

Parameter	Description
bandwidth	Configures WMT bandwidth settings on the Content Engine.
incoming	Allows bypassing of incoming bandwidth restrictions for broadcast alias and multicast stations.
bypass-list	Configures a list of up to four Content Engines that will be exempted from checking for incoming bandwidth.
<i>ip-address</i>	IP address of an exempt Content Engine.
<i>hostname</i>	Hostname of an exempt Content Engine.

The increased bandwidth that the Fast Start feature initially uses to send data to the Windows Media 9 player can overburden a network if many players connect to the stream at the same time. To specify the maximum burst bandwidth allowed for a single player, use the **wmt fast-start max-bandwidth** *number* global configuration command. For more information, see the [“Configuring Incoming and Outgoing WMT Bandwidth and Bit Rates”](#) section on page 9-23.

Configuring Fast Streaming Features on Standalone Content Engines

Windows Media Services 9 Series offers a set of Fast Streaming features that combine streaming, downloading, and caching features for web client acceleration improving the user experience by accelerating delivery of streaming content to the client. In versions earlier than Windows Media Services 9 Series, content was streamed at a constant bit rate to clients.

The ACNS 5.2.1 software and later releases the following Fast Streaming features are supported: Fast Start, Fast Cache, and Fast Reconnect for WMT requests. In the ACNS 5.3.1 software and later releases, this support is available for WMT RTSP requests. [Table 9-7](#) lists the Fast Streaming features that are supported by Content Engines that are running the ACNS 5.2.1 software and later releases.

Table 9-7 Fast Streaming Features Supported by Standalone Content Engines

Feature	More Information
Fast Start	See the “Configuring Fast Start on Standalone Content Engines” section on page 9-28.
Fast Cache	See the “Configuring Fast Cache on Standalone Content Engines” section on page 9-29.

[Table 9-8](#) lists the types of content that are supported with the Fast Start and Fast Cache features.

Table 9-8 Fast Start and Fast Cache Support for Standalone Content Engines

Feature	Preloaded VOD Files	Live Content
Fast Start	Yes	Yes
Fast Cache	Yes	Not applicable

With preloaded VOD files, if the Windows Media 9 server (the Content Engine) determines that the client (a Windows Media 9 player) supports the Fast Start and Fast Cache features, it uses the Fast Start and Fast Cache features to send the packets to the clients. Otherwise, the Windows Media 9 server transmits the packets at its regular speed.

With a cache hit, the Windows Media 9 server (the Content Engine) uses the Fast Start and Fast Cache features to serve the content when the content is cached or partially cached in its local storage.

With a cache miss, the Windows Media 9 server (the Content Engine) operates as a Windows Media 7.0 player and communicates with the remote origin server. Although the origin server supports Fast Start and Fast Cache, the packets must come from the remote server and will still be at the normal speed. After the content is cached on the Content Engine (cache hit or partial hit), the Fast Start and Fast Cache features are supported over MMS (MMS-over-HTTP and RTSP-over-RTP) from Windows Media 9 players.

With live content, the Windows Media 9 server (the Content Engine) needs to hold the content in its buffer for a few seconds. When the first client requests the live stream, the buffer fills up. This buffer is used to serve Fast Start packets to subsequent clients that request the same stream. The first client does not experience any benefit from the Fast Start feature. However, the first client triggers the process of having data come into the Content Engine’s buffer. Subsequent clients who request the same live content will see the benefit of Fast Start when the Content Engine pushes the buffered content to these clients at a faster pace.



Note

The Fast Start feature is only used by Windows Media 9 players that connect to a unicast stream.

A media player must fill its internal buffer (the default is 5 seconds) before it starts rendering the video. For example:

- Without the Fast Start feature, if the player's buffer is set to 5 seconds and the stream-encoding bit rate is 100 kbps, the server will send data at 100 kbps. In this case, it takes 5 seconds for the player to fill up the internal buffer. Consequently, it takes 5 seconds before users can view the video in their players.
- With the Fast Start feature, the server can push the data at a faster pace (for example, 500 kbps). In this case, it takes only 1 second for the player to fill up its internal buffer and to start rendering the video. Consequently, it takes only 1 second before users can start to view the video in their player.

Configuring Fast Start on Standalone Content Engines

Windows Media Services 9 Series introduced the Fast Start feature. This feature allows the server to push the beginning portions of a stream to the client at the maximum available bandwidth. This reduces the amount of time that is required to fill the player's internal buffer, and reduces the amount of time that users (the WMT clients) need to wait before they can start to view the stream in their player (Windows Media 9 players). The Windows Media 4.1 server and Windows Media 4.1 player do not support the Fast Start feature.



Note

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

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

Fast Start provides the following benefits to the users:

- Better playback by eliminating buffering time, while playing a single piece of content or switching seamlessly between on-demand clips or broadcast channels
- Fast forwarding or rewinding of content without additional delay or rebuffering
- Pre-buffering of data, making the Windows Media player resistant to playback errors due to lost packets or other network problems

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



Note

In the ACNS 5.2.x software, the Fast Start feature is only available for MMS-over-HTTP requests with Windows Media Services Version 9.0. In the ACNS 5.3.1 software and later releases, support for the Fast Start feature is also available for RTSP requests from a Windows Media 9 player.

Standalone Content Engines, which are running the ACNS 5.2.1 software and later releases, use the Fast Start feature for preloaded video-on-demand files, cache hits, and live content. This feature is not supported for cache misses and is used only by clients that connect to a unicast stream.

Use the **wmt fast-start** global configuration command to configure the Fast Start feature on a standalone Content Engine, which is running the ACNS 5.2.1 software and later releases.

The syntax for this command is as follows:

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

Table 9-9 describes the command parameters.

Table 9-9 Parameters for the wmt fast-start CLI Command

Parameter	Description
fast-start	Configures the Fast Start feature.
enable	Enables the Fast Start feature.
max-bandwidth	Maximum amount of bandwidth in kilobits per second (kbps) that a single player can use for accelerated initial buffering of the streaming content.
<i>number</i>	Maximum burst bandwidth allowed per player. The default is 3500.

The increased bandwidth that the Fast Start feature initially uses to send data to the player can overburden a network if many players connect to the stream at the same time. To specify the maximum burst bandwidth allowed for a single player, use the **wmt fast-start max-bandwidth number** global configuration command.

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

To configure Fast Start on a standalone Content Engine, follow these steps:

-
- Step 1** Enable Fast Start on the Content Engine by entering the following command:
- ```
ContentEngine(config)# wmt fast-start enable
```
- Step 2** Set the maximum burst bandwidth allowed per media player when Fast Start is used to serve packets to the media player by entering the following command:
- ```
ContentEngine(config)# wmt fast-start max-bandwidth number
```
- Step 3** Verify that Fast Start is enabled by entering the **show wmt EXEC** command.
-

Configuring Fast Cache on Standalone Content Engines

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

Fast Cache is useful in the following situations:

- When the network bandwidth available to the client exceeds the bandwidth required for the content; for example, clients that use a cable modem, Digital Subscriber Line (DSL) connection, or corporate intranets.
- When the network connectivity is intermittent or has high latency; for example, wireless networks.
- When the quality of the content received is of high importance; for example, businesses that provide pay-per-view movies.

With this feature, the media data can be delivered at a rate higher than the playback rate to the client for faster delivery. With the Fast Cache feature, the stream rendering rate is decoupled from the stream delivery rate on the network. This allows a Windows Media 9 server to send the stream content faster than the client's rendering speed. The extra data is then buffered at the Windows Media Player 9 Series client to allow the Windows Media 9 player that is running on this client to adapt better to fluctuations in network bandwidth later on.



Note

In the ACNS 5.2.x software, the Fast Cache feature is only available for MMS-over-HTTP requests with Windows Media Services Version 9.0. In the ACNS 5.3.1 software and later releases, support for the Fast Cache feature is also available for WMT RTSP requests from Windows Media 9 players.

The Fast Cache feature is applicable only for the RTSP and HTTP protocols.

A Windows Media 9 server informs a Windows Media 9 player that it supports the Fast Cache feature. The player then indicates to the server how fast it is for the Fast Start and Fast Cache features. When Fast Cache is configured on a Content Engine, which is running the ACNS 5.2.1 software and later releases, the Content Engine serves the content to a Windows Media 9 player using the smaller of the following two values:

- The bit rate specified in the client request
- The maximum delivery rate configured for Fast Cache in the Content Engine

The following example illustrates bandwidth control and Fast Cache speed adjustment. Client A is a client PC that is running the Windows Media 9 player, which has an IP address of 10.77.140.133, and is configured as follows:

```
<?xml version="1.0"?>
<BandwidthSpec>
  <BandwidthRule>
    <ClientNetwork>10.77.140.133/32</ClientNetwork>
    <description>(Apply to my PC)</description>
    <Allow limit="1100" service="wmt"/>
  </BandwidthRule>
  ...
</BandwidthSpec>
```

If Client A requests media content through RTSP (RTSP TCP mode) and the bit rate of the requested file is 500 kbps, because the available bandwidth for Client A is 1100 kbps the Fast Cache speed for Client A is restricted to 2 (because a speed of 2 would consume 1000 Kbps [500 x 2 = 1000 kbps]). Consequently, the bandwidth consumed by Client A is less than the total available bandwidth limit of 1100 kbps.

Standalone Content Engines, which are running the ACNS 5.2.1 software and later releases, use the Fast Cache feature for preloaded video-on-demand files and for cache hits. The Fast Cache feature is not supported for cache misses, and is not applicable for the delivery of live content.

To configure Fast Cache on a standalone Content Engine, follow these steps:

Step 1 Enable Fast Cache on the Content Engine by entering the following command:

```
ContentEngine(config)# wmt fast-cache enable
```

By default, the Fast Cache feature is enabled on a Content Engine. To reenable the Fast Cache feature on a standalone Content Engine, use the **wmt fast-cache enable** global configuration command.

Step 2 Set the maximum delivery rate allowed per media player when Fast Cache is used to serve packets to the player by entering the following command:

```
ContentEngine(config)# wmt fast-cache max-delivery-rate number
```



Note

The default maximum Fast Cache speed on the Content Engine is 5 (which can be changed with the **wmt fast-cache max-delivery-rate** global configuration command). To configure the maximum bit rate for the Fast Cache feature, use the **wmt fast-cache max-delivery rate** global configuration command to configure the Fast Cache speed multiplier. The value can be from 1 to 65535 (a value of 1 is equivalent to the Fast Cache feature being disabled). The maximum value for the bit rate is associated with the WMT license on the Content Engine.

When Fast Cache is configured, the Content Engine serves content using either the bit rate specified in the client request or the maximum delivery rate configured for Fast Cache in the Content Engine, whichever is smaller.

Step 3 Verify that Fast Cache is enabled by entering the **show wmt EXEC** command.

Configuring Transparent Redirection of WMT Requests

This section describes how to enable and configure a router and a standalone Content Engine for transparent redirection of RTSP requests:

Configuring RTSP Transparent Redirection of WMT Requests

With transparent redirection of WMT requests, a WCCP Version 2-enabled router or a Layer 4 switch transparently redirects WMT RTSP requests to the Content Engine (acting as a transparent proxy server). WMT RTSP transparent redirection is used to support WMT transparent caching on a standalone Content Engine that is running the ACNS 5.3.1 software and later releases. With this type of transparent redirection, you must configure WMT RTSP redirection on the WCCP Version 2-enabled routers or the Layer 4 switch as well as on the standalone Content Engine that will receive these redirected WMT MMS requests.

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

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

The following example shows how to use the Content Engine CLI to configure WMT RTSP transparent redirection through WCCP Version 2. This example assumes that you have enabled the licensed WMT feature on the standalone Content Engine, as described in the [“Enabling WMT Licenses on Standalone Content Engines”](#) section on page 9-17.

To configure WMT RTSP transparent redirection through WCCP Version 2, follow these steps:

Step 1 Enable WCCP Version 2 on the router (Router A).

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

Step 2 Enable the WCCP Version 2 services 80 and 83 on the router.

a. Enable service 80 (the rtsp redirection service) on Router A.

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

b. Enable service 83 (the wmt-rtspu redirection service) on Router A.

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



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

Step 3 Specify an interface on which the RTSP redirection services will run on Router A.

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

The following shows how to configure the outgoing interface to the Internet as Ethernet 0 on Router A:

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

Step 4 From interface configuration mode on Router A, enable WCCP redirection to service 80 and 83 on the specified router interface (in this case, the outgoing interface).

Specify the inbound or outbound interface for service 80 and service 83.

```
RouterA(config-if)# ip wccp 80 redirect out
RouterA(config-if)# ip wccp 83 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). This depends primarily on your network topology.

Step 5 Enable WMT RTSP redirection through WCCP on the standalone Content Engine that will be functioning as the transparent proxy server for redirected WMT requests from Router A:

a. Enable WCCP Version 2 on the Content Engine.

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

b. Create the numbered router list that you want to associate with service 80 and 83.

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

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

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

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

- Step 6** Enable transaction logging on the standalone Content Engine.

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



Tip You can configure standalone Content Engines to log usernames for any authenticated WMT RTSP requests. For more information, see the [“Enabling the Logging of Usernames to the WMT Transaction Log”](#) section on page 9-54.

- Step 7** Save the new configuration on the Content Engine.

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

- Step 8** Verify that WMT is now running on the Content Engine.

```
ContentEngine# show wmt
```

- Step 9** Configure WMT parameters (for example, configure the WMT bandwidth) as needed using CLI commands or the Content Engine GUI.

- Step 10** After starting the Windows Media 9 player, display all of the WMT statistics for this Content Engine:

```
ContentEngine# show statistics wmt all
```

Objects transported over HTTP are counted in the HTTP statistics, and not included in the **show statistics wmt all** command output. In the ACNS 5.3.1 software and later releases, the command output also includes WMT statistics for objects transported over RTSP if WMT RTSP transparent redirection has also been configured (services 80 and 83).

After configuring the routers and Content Engine to support WMT RTSP transparent redirection through WCCP Version 2, enable and configure WMT transparent caching on the Content Engine, as described in the [“Enabling and Configuring WMT Transparent Caching on Standalone Content Engines”](#) section on page 9-35.

To enable the Content Engine to receive RTSP requests that are transparently redirected to it by a Layer 4 switch, enter the following command:

```
ContentEngine(config)# rtsp 14-switch enable
```

The **rtsp 14-switch enable** global configuration command allows the Content Engine to receive RTSP requests that are transparently redirected to it by a Layer 4 switch (for example, a CSS switch). The Layer 4 switch intercepts the RTSP request from the client and transparently redirects that request to the Content Engine.

Enabling and Configuring WMT Caching on Standalone Content Engines

This section describes how to enable and configure the following types of WMT caching on standalone Content Engines:

- [Enabling and Configuring Nontransparent WMT Proxy Caching on Standalone Content Engines, page 9-34](#)
- [Enabling and Configuring WMT Transparent Caching on Standalone Content Engines, page 9-35](#)

Enabling and Configuring Nontransparent WMT Proxy Caching on Standalone Content Engines

With direct proxy routing, the client WMT media players send their requests directly to the Content Engine that is acting as a nontransparent forward proxy server. Direct proxy routing is used to support WMT proxy caching on a Content Engine. With direct proxy routing, you must point the client WMT media players directly to the Content Engine.

For more information about pointing the client WMT media players directly to the Content Engine, see the following sections:

- [Pointing Windows Media 9 Players Directly to a Standalone Content Engine for WMT RTSP Requests, page 4-43](#)
- [Pointing Windows Media Players Directly to a Standalone Content Engine for WMT MMS Requests, page 4-45](#)

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

-
- Step 1** Specify the maximum size of a single object that the Content Engine should store in its WMT cache. The range of values is between 1 and 1,000,000 megabytes. The default value is 1024 megabytes.

```
ContentEngine(config)# wmt cache max-obj-size size
```

- Step 2** Specify the external WMT server that the Content Engine is to use as an upstream WMT server (the outgoing HTTP proxy server for WMT) with the **wmt proxy outgoing** global configuration command.

For more information, see the [“Configuring Standalone Content Engines to Distribute VOD Files” section on page 9-35](#).

- Step 3** Enable WMT caching on the standalone Content Engine if it is not already enabled.

```
ContentEngine(config)# wmt cache enable
```

Enabling and Configuring WMT Transparent Caching on Standalone Content Engines

With WCCP routing or Layer 4 switching, you must configure the WCCP Version 2-enabled routers or Layer 4 switches and the Content Engine (transparent proxy server) for WMT RTSP transparent redirection to accept redirected WMT requests from Windows Media 9 players. For more information, see the “[Configuring Transparent Redirection of WMT Requests](#)” section on page 9-31.

To use the Content Engine CLI to enable and configure WMT transparent caching on a standalone Content Engine, follow these steps:

-
- Step 1** Enable WMT caching on the standalone Content Engine if it is not already enabled.
- ```
ContentEngine(config)# wmt cache enable
```
- Step 2** Specify the maximum size of a single object that the Content Engine should store in its WMT cache. The range of values is between 1 and 1,000,000 megabytes. The default value is 1024 megabytes.
- ```
ContentEngine(config)# wmt cache max-obj-size size
```
- Step 3** Specify the list of routers from which this Content Engine will accept redirected WMT requests.
- ```
ContentEngine(config)# wccp wmt router-list number
```
- Step 4** If you have not yet created a list of routers from which you want this Content Engine to accept redirected WMT requests, then create the router list now:
- ```
ContentEngine(config)# wccp router-list number
```
- In the following example, there are two WCCP Version 2-enabled routers associated with router list 1. These routers have the IP addresses 172.16.25.25 and 172.16.26.26.
- ```
ContentEngine(config)# wccp router-list 1 172.16.25.25 172.16.26.26
```
- Step 5** If you have not yet enabled the router list (for example, router list 1) that includes the WCCP Version 2-enabled routers that will redirect WMT requests to this Content Engine, then enable it.
- ```
ContentEngine(config)# wccp wmt router-list-num number
```
- Step 6** Enable WCCP Version 2 on the Content Engine, if it is not already enabled.
- ```
ContentEngine(config)# wccp version 2
```
- Step 7** Specify the external WMT server that the Content Engine should use as its upstream WMT server by using the **wmt proxy outgoing** global command.
- 

## Configuring Standalone Content Engines to Distribute VOD Files

You can preload VOD files on the Content Engine for on-demand delivery of these files to Windows Media clients. VOD caching is similar to HTTP caching; however, VOD files are cached in a different file system (mediafs) on the standalone Content Engine. WMT transparent caches and WMT proxy caches both support VOD caching.

To configure a standalone Content Engine to distribute VOD files to Windows Media clients, follow these steps:

- 
- Step 1** Preload the VOD files on this Content Engine.
- a. Enable content preloading on the Content Engine.
  - b. Use a preload URL list file to indicate which Windows Media content is to be preloaded on the Content Engine.
  - c. Configure bandwidth control for preloading.
  - d. Schedule or force an immediate preloading of the content.
- Step 2** Publish the URLs of the preloaded VOD files that clients can now access through their Windows Media players.
- 

For instructions on how to preload files on a standalone Content Engine, see the [“Configuring Content Preloading for Standalone Content Engines” section on page 11-2](#). For information about how you can verify that the preloaded VOD files are being cached and properly distributed to clients, see the next section, [“Verifying That Preloaded VOD Files Are Cached and Properly Distributed to Windows Media Clients.”](#)

## Verifying That Preloaded VOD Files Are Cached and Properly Distributed to Windows Media Clients

This section describes how you can verify that a standalone Content Engine has stored the preloaded VOD files in its cache, and is distributing these files to Windows Media clients upon request. This example assumes the following:

- Preloading has been configured on the Content Engine, the preload URL list includes some Windows Media files, and the Content Engine has completed the preload operation. For more information on this topic, see the [“Configuring Content Preloading for Standalone Content Engines” section on page 11-2](#).
  - A Windows Media player on at least one of your client desktops has been configured to point directly to the Content Engine (the nontransparent forward proxy server for this WMT client). For information on this topic, see the [“Pointing Windows Media 9 Players Directly to a Standalone Content Engine for WMT RTSP Requests, page 4-43”](#).

To point to a VOD source and verify that both WMT proxy caching and WMT transparent caching are working properly on the standalone Content Engine when the clients are Windows Media 6 or 7 players, follow these steps:

- 
- Step 1** Launch the Windows Media 6 or 7 player from one of your client’s personal computers (Client A) that is either configured to point directly to the Content Engine (nontransparent forward proxy server) or configured not to point directly to the Content Engine.
- Step 2** From the Windows Media player, choose **File > Open URL**.
- Step 3** Enter a URL that points to a Windows Media streaming file (for example, a \*.asf or \*.wmv file) that has been preloaded on the Content Engine.

The specified preloaded video should start playing in the Windows Media player on the client’s desktop.

- Step 4** Verify the statistics on the Windows Media player by clicking **Edit > Statistics Advanced**.
- Step 5** From any of the WCCP Version 2-enabled routers that are configured to redirect WMT requests to this Content Engine (acting as a transparent proxy server), check that WMT packets are being redirected on that router by entering the **show ip wccp EXEC** command.
- Step 6** View the WMT caching statistics for this standalone Content Engine:
- ```
ContentEngine# show stat wmt savings
ContentEngine# show stat wmt request
ContentEngine# show stat wmt usage
```
- Step 7** Check the WMT transaction log on this standalone Content Engine.
- ```
ContentEngine# type-tail "/local1/logs/export/working.log"
```
- 

## Configuring Standalone Content Engines to Deliver WMT Live Streams

Based on the capabilities and limitations of the network, standalone Content Engines can receive live WMT streams and then deliver WMT streaming content through multicast out or unicast out.

This section describes how to configure standalone Content Engines to deliver WMT live streams, and includes the following sections:

- [Configuring Standalone Content Engines to Multicast Live WMT Streams, page 9-37](#)
- [Configuring an Alternative Source URL \(Source Failover\) for a WMT Multicast, page 9-42](#)

## Configuring Standalone Content Engines to Multicast Live WMT Streams

You can configure standalone Content Engines to send live content to Windows Media clients through multicasting or unicast. This section describes how to configure standalone Content Engines to relay live content through multicasting:

- [Configuring Multicast-In Multicast-Out on Standalone Content Engines, page 9-38](#)
- [Configuring Unicast-In Multicast-Out on Standalone Content Engines, page 9-39](#)
- [Defining WMT Multicast Stations and Multicast Schedules on Standalone Content Engines, page 9-40](#)
- [Starting and Stopping WMT Multicast Stations, page 9-41](#)



### Note

You must enable WMT on the Content Engine before you can use the **wmt multicast** and **wmt broadcast** global configuration commands. See the “[Configuring WMT RTSP Streaming and Caching Services on Standalone Content Engines](#)” section on page 9-14.

---

To enable WMT multicasting for unicast-in multicast-out (“[Configuring Unicast-In Multicast-Out on Standalone Content Engines](#)” section on page 9-39) and multicast-in multicast-out (“[Configuring Multicast-In Multicast-Out on Standalone Content Engines](#)” section on page 9-38) on a standalone Content Engine, use the following command:

```
wmt multicast {schedule-start name minute hour day month | station-configuration
name dest_addr dest_port media_source [play-forever] }
```

To configure the Time To Live (TTL) for a WMT multicast, use the **wmt multicast time-to-live** *tvl* global configuration command. The TTL value is specified as the number of hops. The value can be from 0 to 255 hops. The default is five hops. For example:

```
ContentEngine(config)# wmt multicast time-to-live 10
```

For information about how to configure standalone Content Engines to relay live content to clients through unicast, see the “[Configuring an Alternative Source URL \(Source Failover\) for a WMT Multicast](#)” section on page 9-42.

## Configuring Multicast-In Multicast-Out on Standalone Content Engines

The multicast-in multicast-out multicast receive feature allows you to receive multicast WMT streams delivered through IP multicasting, and then send them to end users through another delivery channel (unicast or multicast). The two WMT multicast-out features combined enable you to receive and deliver WMT streaming media content through IP multicasting, and to do conversions from multicast to unicast (and vice versa).

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

To configure a standalone Content Engines to use multicast-in multicast-out to relay live WMT streams to Windows Media clients, follow these steps:

---

**Step 1** Configure a multicast station on the Content Engine by using the **wmt multicast station-configuration** global configuration command:

The syntax of this command is as follows:

- **station-configuration** configures the WMT multicast station on the Content Engine.
- *name* specifies the name of the WMT multicast station.
- *dest\_addr* is the destination IP address (multicast IP address) of the WMT multicast station.
- *dest\_port* is the destination port (1–65535) of the WMT multicast station.
- *media\_source* is the media source of the multicast.

In the following example, a multicast station named *acme* is configured and used by the Content Engine as the multicast source file. Its Class D multicast IP address is 233.33.33.34, and the multicast port is 6667. The multicast stream stops playing once the end of the *source.nsc* file is reached, unless the **play-forever** option is specified.

```
ContentEngine(config)# wmt multicast station-configuration acme 233.33.33.34
6667 http://172.16.30.31/source.nsc
```

Note that in the ACNS 5.3 software and later releases, the source TCP port on the multicast must be unique from any other multicast WMT station configured. Even if you use a different source server, the source TCP ports must be unique across all multicast stations.

**Note**

In the ACNS 5.3.1 software and later releases, `rtsp://`, `rtspu://`, and `rtsp://` URLs for WMS 9 (that is if the client is a Windows Media 9 player and the server is the Windows Media 9 server running on the Content Engine) as well as `http://` URLs are supported.

**Step 2** Start the multicast.

```
ContentEngine# wmt multicast-station start acme
ContentEngine#
```

**Step 3** Open your WMT media player and choose **File > Open URL**.

**Step 4** Enter the following URL:

```
http://ContentEngineIPAddress/acme.nsc
```

**Step 5** Click **OK**.

The Windows Media client should receive the media file specified in [Step 1](#).

## Configuring Unicast-In Multicast-Out on Standalone Content Engines

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

**Tip**

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

The unicast-in multicast-out multicast delivery feature allows you to distribute streaming media efficiently by allowing different devices on the IP multicast to receive a single stream of media content from the Content Engine simultaneously. This can save significant network bandwidth consumption, because a single stream is sent to many devices, rather than sending a single stream to a single device every time that this stream is requested.

This multicast delivery feature is enabled by setting up a multicast address on the Content Engine to which different devices, configured to receive content from the same channel, can subscribe. The delivering device sends content to the multicast address set up at the Content Engine, from which it becomes available to all subscribed receiving devices.

To configure a standalone Content Engines to use unicast-in multicast-out to send live WMT streams to WMT clients, follow these steps:

**Step 1** Configure a multicast station on the Content Engine by using the **wmt multicast station-configuration** global configuration command.

In the following example, a multicast station named test1 is configured and used by the Content Engine as the multicast source file. Its Class D IP address is 239.33.33.33, and the multicast port is 3333. The **play-forever** option is used. When the input source.asf file is a VOD file, this option automatically restarts playback of the file from the beginning of the source.asf file once the end of this file has been reached. This source file source.asf can be located on any Windows WMT server.

```
ContentEngine(config)# wmt multicast station-configuration test1 239.33.33.33
3333 rtsp://172.16.30.31/source.asf play-forever
```

**Step 2** Start the multicast.

```
ContentEngine# wmt multicast-station start test1
ContentEngine#
```

**Step 3** Open your WMT media player and choose **File > Open URL**.

**Step 4** Enter the following URL:

```
http://ContentEngineIPAddress/test1.nsc
```

**Step 5** Click **OK**.

The Windows Media player should retrieve the multicast description .nsc file and join the multicast station that is specified in Step 1.

The use of port 80 is implied in the URL for WMT multicasting, such as the following:

```
http://ContentEngineIPAddress:80/test1.nsc.
```

## Defining WMT Multicast Stations and Multicast Schedules on Standalone Content Engines

To configure a WMT multicast station on a standalone Content Engine, use the **wmt multicast station-configuration** global configuration command.



### Note

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

The **wmt multicast station-configuration** *name dest\_addr dest\_port media\_source* command specifies a multicast station name, a multicast IP address, port number, and media source for the multicast station created. Each WMT multicast station needs a multicast IP address. You must enter a valid Class D IP address multicast address in the range 224.0.0.0 to 239.255.255.255, except for the reserved IP ranges based on RFC 1700 and related documents as follows:

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

**Note**

You must choose a multicast IP address that does not conflict internally within the same multicast-enabled network configuration. For a complete table of unusable multicast address ranges, see [Table B-8](#) in the “Unusable Multicast Address Assignments” section on [page B-11](#).

The destination port of the WMT multicast station is specified by the **dest\_port** option. Valid options are 1 through 65535. However, the multicast-enabled network may impose certain restrictions on your choice of port. Normally, port numbers less than 1024 should be avoided, but the Content Engine does not enforce any restrictions.

The **media\_source** option determines the source of the multicast. The source can be any valid WMT URL. In other words, if you can play the URL on your Windows Media player, then you can make this URL the source of your multicast. The **play-forever** option configures the stream to loop and restart. The default is to play the stream once and stop.

For example:

```
ContentEngine(config)# wmt multicast station-configuration acme 239.33.33.33
3333 rtsp://172.16.30.31/source.asf play-forever
```

In this example:

- The name of the WMT multicast station is acme.
- The multicast IP address of the WMT multicast station is 239.33.33.33.
- The destination port of the WMT multicast station is 3333.
- The source of the multicast is rtsp://172.16.30.31/source.asf, and it will play forever.

To configure multicasting schedules for WMT multicast stations, use the **wmt multicast station-configuration station-name schedule-start** global configuration command.

The **schedule-start name minute** option creates a scheduling option to allow the Content Engine to start a multicast at a specified time.

[Table 9-10](#) describes the command parameters for the **wmt multicast station-configuration station-name schedule-start** global configuration command.

**Table 9-10** Parameters for the **wmt multicast station-configuration schedule-start** Command

| Parameter             | Description                                                                                                                                                                                                                                         |
|-----------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <i>station-name</i>   | Name of the WMT multicast station for which you are creating the schedule.                                                                                                                                                                          |
| <b>schedule-start</b> | Configures an automatic start schedule.                                                                                                                                                                                                             |
| <i>minute</i>         | Start time minute (0–59).                                                                                                                                                                                                                           |
| <b>now</b>            | Start the WMT multicast station now. If you specify this option, the multicast station will be started immediately. If the station is running and the Content Engine is reloaded, the station will be automatically started again after the reload. |

## Starting and Stopping WMT Multicast Stations

In the ACNS 5.3.1 software and later releases, the ability to resume a WMT multicast automatically after a Content Engine is reloaded is supported. To support this feature, the **wmt multicast station-configuration station-name schedule-start now** global configuration command was added in the ACNS 5.3.1 software release. If you specify this command for a specific multicast station, the

multicast station will be started immediately and then automatically started again after the Content Engine is reloaded. The reason for introducing this new global configuration command is that the **wmt multicast-station start** *station-name* EXEC command is not persistent across reboots (that is, if a multicast station is running before the Content Engine is reloaded, it will not continue to run after the reload).

The **no wmt multicast station-configuration** *station-name* **schedule-start now** global configuration command, which was added in the ACNS 5.3.1 software release, works exactly like the **wmt multicast-station stop** *station-name* EXEC command. You can use either command to stop a specific WMT multicast station.

The **wmt multicast-station start** EXEC command only works if you have configured a multicast station first, using the **wmt multicast station-configuration** *station-name* global configuration command.

For example, after using the **wmt multicast station-configuration** *station-name* global configuration command to configure a multicast station, you can start or stop the multicast station by using the **wmt multicast-station** EXEC command:

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

Table 9-11 describes the command parameters for the **wmt multicast-station** EXEC command.

**Table 9-11 Starting and Stopping WMT Multicast Stations**

| Parameter                | Description                                                                                                                                                                                                             |
|--------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>multicast-station</b> | Sets the WMT multicast stations to start or stop.                                                                                                                                                                       |
| <b>start</b>             | Starts a WMT multicast station. If you use this option to start a multicast station and it is running before the Content Engine is reloaded, the station will not continue to run after the Content Engine is reloaded. |
| <i>station-name</i>      | Name of the WMT multicast station to be started.                                                                                                                                                                        |
| <b>stop</b>              | Stops a WMT multicast station.                                                                                                                                                                                          |
| <i>station-name</i>      | Name of the WMT multicast station to be stopped.                                                                                                                                                                        |

The following examples demonstrate the **start** and **stop** options on the multicast station named acme.

```
ContentEngine# wmt multicast-station start acme
ContentEngine# wmt multicast-station stop acme
```

## Configuring an Alternative Source URL (Source Failover) for a WMT Multicast

In the ACNS 5.3.1 software and later releases, you can configure an alternative source URL (source failover) for a WMT multicast with the **wmt multicast station-configuration** *station-name* **failover** global configuration command:

```
ContentEngine(config)# wmt multicast station-configuration acme failover ?
 alternate-source Alternate source url
 retry-count No. of retries for all sources
 retry-interval Sleep interval between retries
```

Use the **alternate-source** option to specify a fully-qualified alternative source URL for the WMT multicast.

To specify the number of retries for all of the sources, use the **retry-count** option. The retry count can be from 0 to 2000. To specify the retry interval (that is, the amount of time that the Content Engine is to wait between retries), use the **retry-interval** option. The retry interval can be from 0 to 60 minutes.

## Configuring Standalone Content Engines to Unicast Live WMT Streams

You can configure standalone Content Engines to send live content to WMT clients through multicasting or unicast. This section describes how to configure standalone Content Engines to relay live content through unicast:

- [Configuring Multicast-In Unicast-Out on Standalone Content Engines, page 9-43](#)
- [Configuring Unicast-In Unicast-Out on Standalone Content Engines, page 9-44](#)

These sections also describe how to configure a WMT broadcast alias on a standalone Content Engine for unicast-out.

### Configuring Multicast-In Unicast-Out on Standalone Content Engines

The multicast-in unicast-out feature allows you to create a broadcasting publishing point to deliver an incoming stream live to requesting clients using multicast as the source of the streaming media. To configure a multicast-in unicast-out broadcast on a standalone Content Engine, use the **wmt broadcast {alias-name name source url}** global configuration command. With this command, you create a broadcasting alias to deliver an incoming stream live to requesting clients, using multicast as the source of the streaming media.

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

To configure a standalone Content Engines to use multicast-in unicast-out (unicasting out) to relay live WMT streams to WMT clients, follow these steps:

---

**Step 1** Configure a WMT broadcast alias on the Content Engine:

```
ContentEngine(config)# wmt broadcast alias-name myunicast source
http://172.16.30.31/station.nsc
ContentEngine(config)#
```

In this step a unicast publishing point with the alias name myunicast is configured with a multicast source station.nsc file. This source is a server sending out WMT multicast streams. The source of an alias in the format http://server/file.nsc signals the Content Engine to treat this source as a multicast input source.

**Step 2** Open your WMT media player and choose **File > Open URL**.

**Step 3** Enter the following URL:

```
rtsp://ContentEngineIPAddress/myunicast
```

**Step 4** Click **OK**.

The WMT media player should receive the media source file specified in Step 1. In this situation, an RTSP URL is used to access the streaming media, and only the alias name is specified instead of the \*.nsc files as in the multicast-out situations.

This converts the multicast stream to unicast and sends it to the requester (the WMT client).

---

## Configuring Unicast-In Unicast-Out on Standalone Content Engines

The unicast-in unicast-out feature provides a point-to-point connection between the client and the Content Engine. The advantage of unicasting when streaming media over a network is that only a single stream needs to be pulled over the network between the origin server and Content Engine, but that stream can be delivered to multiple clients in a nonmulticast environment. A server running Windows Media Services can provide a unicast video stream to multiple clients through a single stream delivered to the Content Engine. Unicast-in unicast-out is typically used to broadcast live events.

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

You can configure unicast-in unicast-out in the following ways:

- By live splitting without any configuration.

In this case, the Content Engine acts as a proxy. When clients request the same unicast URL, the Content Engine proxy automatically splits the stream from the source to the clients.

- By configuring the Content Engine with a broadcast alias.

In this case, a client makes the request to the Content Engine as if it were the Windows Media Server, and the Content Engine checks to see whether the incoming stream is present. If it is, then the Content Engine joins the stream and splits it to the new client. If the request is the first client request for this stream, the Content Engine sends the request out to the server and then serves it to the new client.

To configure a standalone Content Engine to use unicast-in unicast-out (unicast out) to relay live WMT streams to clients, follow these steps:

- 
- Step 1** From the Content Engine GUI, choose **Caching > WMT-Streaming**. The WMT Streaming window appears.
  - Step 2** Click **WMT Config**. The WMT Configurations window opens.
  - Step 3** Click the **Broadcast Unicast Publishing** link. The WMT Broadcast Unicast Publishing window appears.
  - Step 4** In the Alias Name field, enter a broadcast alias for the live broadcast configuration (for example, broadcast1).
  - Step 5** In the Source field, enter the broadcast source for the live broadcast configuration using the following format:  
`<protocol>://server-name:port-num/path/file-name`

The variables are as follows:

- *protocol* is either HTTP or RTSP.
- *server-name* is the name of the server.
- *port-num* is the port number. The default is port 8080 for HTTP and port 554 for RTSP.
- *path* is the full pathname.
- *file-name* is a media filename if the file is in the content root directory.

For example:

```
rtsp://wms.company.com/cotv
```

wms.company.com is the name of the Windows Media Server, and cotv is the name used when the broadcast alias is created.

**Step 6** Click **Update** to save the settings.

**Step 7** Open your WMT player and choose **File > Open URL**. Enter the following URL:

```
rtsp://ContentEngineIPAddress/broadcast1
```

- *ContentEngineIP address* is the IP address or domain name of the Content Engine.
- **broadcast1** is the broadcast alias specified in [Step 4](#).

**Step 8** Click **OK**.

The WMT player should receive the media source file specified in [Step 5](#). In this situation, an RTSP URL is used to access the streaming media, and only the broadcast alias (for example, broadcast1) is specified instead of the \*.nsc files in the multicast-out situations. This converts the multicast stream to unicast and sends it to the WMT client.

## Clearing WMT Streams on Standalone Content Engines

To clear WMT streams on a standalone Content Engine, use the **clear wmt** EXEC commands:

```
ContentEngine# clear wmt ?
 incoming Clear all incoming WMT streams
 outgoing Clear all outgoing WMT streams
 stream-id Stream Id of the WMT stream to be cleared
```

[Table 9-12](#) describes these the **clear wmt** EXEC commands.

**Table 9-12** CLI Commands for Clearing WMT Streams on Standalone Content Engines

| Command                   | Description                                                                                                                                                  |
|---------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>clear wmt incoming</b> | Clears all incoming WMT streams from the Content Engine. Also stops all of the Content Engine's WMT processes that are associated with incoming WMT streams. |

**Table 9-12** CLI Commands for Clearing WMT Streams on Standalone Content Engines

|                                      |                                                                                                                                                                  |
|--------------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>clear wmt outgoing</b>            | Clears all outgoing WMT streams from the Content Engine. Also stops all of the Content Engine's WMT processes that are associated with the outgoing WMT streams. |
| <b>clear wmt stream-id <i>id</i></b> | Clears the WMT streams that have the specified stream ID. Also stops the Content Engine's WMT process that is associated with the specified stream ID.           |

In the ACNS 5.2.1 software and later releases, the error logs will log an error message when WMT streams are cleared and the associated processes are stopped on the Content Engine. For more information, see the [“Logging the Clearing of WMT Streams on Standalone Content Engines”](#) section on page 9-56.

## Displaying Information about the WMT RTSP Server Configuration

In the ACNS 5.3.1 software and later releases, by entering the **show rtsp server wmt** EXEC command, you can display information about the WMT RTSP server that runs on the Content Engine. The following sample output shows a sample configuration for the WMT RTSP server that is running on the standalone Content Engine:

```
Content Engine# show rtsp server wmt
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
```

## Displaying Information about the Current WMT Configuration

To display the current WMT configuration for a standalone Content Engine, enter the **show wmt** EXEC command. To display the current WMT configuration for a standalone Content Engine, enter the **show wmt** EXEC command.

## Displaying WMT Statistics

To display statistics about WMT requests, use the **show statistics wmt** EXEC commands.

```
ContentEngine# show statistics wmt ?
all Display all Windows Media statistics
bytes Display unicast bytes statistics
errors Display errors statistics
multicast Display multicast statistics
requests Display unicast request statistics
rule Display rule template statistics
savings Display unicast savings statistics
streamstat Display Windows Media streaming connections
urlfilter Display urlfiltering statistics for mms and rtsp requests
usage Display concurrent usage statistics
ContentEngine#
```

In the ACNS 5.3.1 software and later releases, the output of the **show statistics wmt** EXEC commands includes information about WMT RTSP requests. For example, the output from the **show statistics wmt** EXEC commands was changed as follows:

- RTSP-related information was added to the **show statistics wmt all** command output.
- Information about RTSPT and RTSPU were added in the transport protocol portion of the **show statistics wmt bytes** command output.
- RTSPT and RTSPU errors were added to the **show statistics wmt errors** command output.
- The **show statistics wmt requests** command output includes the RTSPT and RTSPU protocols as well as Fast Start and Fast Cache data.

In the ACNS 5.3.1 software and later releases, you can display aggregated live statistics by entering the **show statistics wmt streamstat live** EXEC command.

## Using WMT Logging with Standalone Content Engines

This section describes how to use the WMT logging features, and covers the following topics:

- [Using WMT Multicast Logging, page 9-48](#)
- [Using WMT Transaction Logging, page 9-48](#)
- [Using WMT Error Logging, page 9-55](#)

## Using WMT Multicast Logging

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

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

To provide a log of multicast statistics to multicast server administrators, use the **log** option of the **wmt multicast station-configuration** global configuration command:

```
wmt multicast {station-configuration name dest_addr dest_port media_source
[log {local | webserver webserver_url}]}
```

- To enable logging of multicast URLs, specify the **log** option.
- To configure logging of multicast URLs to a local disk, specify the **local** option.
- To configure logging of multicast URLs to a web server, specify the **webserver** option and enter the URL to identify the location of the web server.

The variables for this command are as follows:

- *name* is the name of the WMT multicast station.
- *dest\_addr* is the WMT multicast station destination IP address.
- *dest\_port* is the WMT multicast station destination port (1–65535).
- *media\_source* is the WMT multicast media source (for example, `http://live/live`).
- *webserver\_url* specifies the fully qualified webserver URL.

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

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

```
10.1.101.2 2003-05-11 13:39:21 - asfm://233.0.4.5:4000 0 30 1 200
{5DC90EEB-CEB1-467C-9F7A-BCF5EEDE3FF} 10.1.0.3055 en-US - -
wmplayer.exe 10.1.0.3055
Windows_2000 10.0.0.2195 Pentium 0 152543 65389
asfm UDP WINDOWS_MEDIA_AUDIO_V2
MICROSOFT_MPEG-4_VIDEO_CODEC_V3 http://172.16.192.91/cisco.nsc
- 166245 - 176 0 0 0 0 0 0 1
0 100 233.0.4.5 - - -
```

## Using WMT Transaction Logging

For some companies, streaming media is a source of revenue, and therefore needs to be tracked closely. Because these companies charge their customers to stream on-demand content and live broadcasts, they must rely on logged information to track what content a particular customer viewed, how long they viewed it, and the viewing quality. Consequently, the accuracy and reliability of transaction logging is very important to these companies.

The Windows Media Services 9 Series provides a more robust logging model than Windows Media Services Version 4.1. In the ACNS 5.2.1 software, support for Windows Media Services 9 logging was added.

In the ACNS 5.2.1 software and later releases, the following logging formats are supported for WMT transaction logging:

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

**Note**

In the ACNS 5.1 software and earlier releases, only the standard Windows Media Services 4.1 and the extended Windows Media Services 4.1 logging formats were supported.

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

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

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

Windows Media players connect to a Windows Media server using the following protocols:

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

Depending on the version of the Windows Media player, logs are sent in different formats, such as text, binary, or XML. See [Table 9-13](#).

**Table 9-13 Log Formats Accepted by Windows Media Services 9**

| Protocol | Player and Distributor                                                                                                                                                                                                                                                   | Log Type                                                          |
|----------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------------------------------------------------------------------|
| HTTP/1.0 | Windows Media players earlier than Version 9.0 (for example, Windows Media 6.4 or 7.0 players)<br><br>Content Engine (caching and proxy server) is running Windows Media Services Version 9.0 and streaming from a WMT server that is running Windows Media Services 4.1 | World Wide Web Consortium (W3C) standard space-delimited text log |

**Table 9-13** Log Formats Accepted by Windows Media Services 9 (continued)

| Protocol | Player and Distributor                                                                                                                                                    | Log Type          |
|----------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------------------|
| HTTP/1.1 | Windows Media 9 players<br>Distribution server is running Windows Media Services 9.0<br>Content Engine (caching and proxy server) is running Windows Media Services 9.0   | XML structure log |
| RTSP     | Windows Media 9.0 players<br>Distribution server is running Windows Media Services 9.0<br>Content Engine (caching and proxy server) is running Windows Media Services 9.0 | XML structure log |

In the ACNS 5.2.1 software and later releases, XML logging for MMS-over-HTTP is supported. The posted XML log file from the Windows Media player to the Content Engine (Windows Media server) can be parsed and saved to the normal WMT transaction logs that are stored on the Content Engine.

**Note**

In the ACNS 5.3.1 software and later releases, support for WMS 9 logging for WMT RTSP requests from Windows Media 9 players is available.

## Specifying the Format of the WMT Transaction Logs

To specify the format for the WMT transaction logs on standalone Content Engines, use the **wmt transaction-logs format** global configuration command that is supported in the ACNS 5.2.1 software and later releases.

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

By default, the standard Windows Media Services 4.1 logging format is used (no Content Engine-specific details are logged).

[Table 9-14](#) describes the command parameters.

**Table 9-14** Parameters for the wmt transaction-logs format CLI Command

| Parameter               | Description                                                                                                  |
|-------------------------|--------------------------------------------------------------------------------------------------------------|
| <b>transaction-logs</b> | Configures the logging format of the WMT transaction logs.                                                   |
| <b>format</b>           | Sets the format for WMT transaction logs.                                                                    |
| <b>extended</b>         | Specifies the WMT extended format for transaction logs. Enables username logging in the WMT transaction log. |

**Table 9-14** Parameters for the *wmt transaction-logs format CLI Command (continued)*

| Parameter     | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                           |
|---------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>wms-4</b>  | <p>Sets WMT to generate transaction logs in extended Windows Media Services 4.1 format.</p> <p>When this option is used, the Content Engine uses the standard Windows Media Services 4.1 format to generate the transaction log but also includes the following three additional fields in the transaction log:</p> <ul style="list-style-type: none"> <li>• CE_action (cache hit or cache miss)</li> <li>• CE-bytes (number of bytes sent from the Content Engine for a cache hit)</li> <li>• username (username of the person who made the WMT request when Microsoft NL LAN Manager (NTLM) authentication, Microsoft Negotiate authentication, Microsoft Digest authentication, and basic authentication are used)</li> </ul> <p><b>Note</b> Microsoft Negotiate authentication is an authentication method in which the WMS Negotiate Authentication plug-in is used to authenticate the client. This method of authentication uses the client's logon credentials. It uses the encrypted password and username that the user entered during the login process.</p> <p>Microsoft Digest authentication is an authentication method in which an initial authentication of the client is performed when the server receives the first challenge response from the client. After the server verifies that the client has not been authenticated yet, it accesses the services of a domain controller (DC) to perform the initial authentication of the client. When the initial authentication of the client is successfully completed, the server receives a Digest session key. The server caches the session key and uses it to authenticate subsequent requests for resources from the authenticated client.</p> |
| <b>wms-90</b> | <p>Sets WMT to generate transaction logs in extended Windows Media Services 9 format.</p> <p>When this option is used, the Content Engine uses the standard Windows Media Services 9 format to generate the transaction log but also includes the following three additional fields in the transaction log:</p> <ul style="list-style-type: none"> <li>• CE_action (cache hit or cache miss)</li> <li>• CE-bytes (number of bytes sent from the Content Engine for a cache hit)</li> <li>• username (username of the person who made the WMT request when NTLM authentication, Microsoft Negotiate authentication, Microsoft Digest authentication, and basic authentication are used)</li> </ul>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                     |
| <b>wms-41</b> | Sets WMT to generate transaction logs in the standard Windows Media Services 4.1 format.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                              |
| <b>wms-90</b> | Sets WMT to generate transaction logs in the standard Windows Media Services 9 format.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                |

To log the username to the WMT transaction log, you must enable the extended WMT logging feature on the Content Engine, use the **wmt extended transaction-log enable** global configuration command. For more information, see the section, [““Enabling the Logging of Usernames to the WMT Transaction Log” section on page 9-54.”](#)

## Extended Windows Media Services 9.0 Logging Format

ACNS 5.5 software forwards two types of logs to the Windows Media Server:

- Logs sent by the Content Engine for its own connection to the Window Media Server.
- Logs forwarded from the Windows Media Player (client) for its playback. These logs are of two types: combination logs and render logs.

For more information, refer to the Windows Media Service 9.0 Logging Model documentation at the following URL:

<http://www.microsoft.com/windows/windowsmedia/howto/articles/LoggingModel.aspx>

Extended Windows Media 9.0 transaction logs are displayed using the following format:

```
c-ip date time c-dns cs-uri-stem c-starttime x-duration c-rate
c-status c-playerid c-playerversion c-playerlanguage cs(User-Agent)
cs(Referer) c-hostexe c-hostexever c-os
c-osversion c-cpu filelength filesize avgbandwidth protocol transport audiocodec
videocodec channelURL sc-bytes c-bytes s-pkts-sent c-pkts-received
c-pkts-lost-client c-pkts-lost-net c-pkts-lost-cont-net
c-resendreqs c-pkts-recovered-ECC
c-pkts-recovered-resent c-buffercount c-totalbuffertime c-quality s-ip s-dns
s-totalclients s-cpu-util CE-action CE-bytes Username
```

Table 9-15 describes the fields shown in this example.

**Table 9-15** Field Descriptions for Windows Media Services 9.0 Logs

| Field           | Descriptions                                                                                                                                                                                                                                                                                                                                                                                                               |
|-----------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| c-ip            | IP address of the client computer. A client that is not connected properly provides a client proxy server IP address, not the client IP address.                                                                                                                                                                                                                                                                           |
| date            | Date (according to Greenwich mean time) when an entry is generated in the log file.                                                                                                                                                                                                                                                                                                                                        |
| time            | Time (according to Greenwich mean time) when an entry is generated in the log file.                                                                                                                                                                                                                                                                                                                                        |
| c-dns           | DNS name of the client computer.                                                                                                                                                                                                                                                                                                                                                                                           |
| cs-uri-stem     | Name of the file that is playing, an .asf file for unicast and an .asx file for multicast.                                                                                                                                                                                                                                                                                                                                 |
| c-starttime     | Time stamp (in seconds) of the stream when an entry is generated in the log file.                                                                                                                                                                                                                                                                                                                                          |
| x-duration      | Length of time a client played content before a client event (fast forward [FF], rewind [REW], pause, stop, or jump to marker). A log entry is generated whenever one of these client events occurs.                                                                                                                                                                                                                       |
| c-rate          | Mode of Windows Media player when the last command event was sent. <ul style="list-style-type: none"> <li>• 1 = Windows Media player was paused or stopped during a play, fast-forward, rewind, or marker jump operation.</li> <li>• -5 = Windows Media player was rewound from a play, stop, or pause operation.</li> <li>• 5 = Windows Media player was fast-forwarded from a play, stop, or pause operation.</li> </ul> |
| c-status        | Codes that describe client status. Mapped to HTTP/1.1 and RTSP client status codes described in Request for Comments (RFC) 2068 and RFC 2326. Windows Media Services includes the extensible client status codes 480 (simultaneous client connections exceeded the maximum client limit of the server) and 483 (stream exceeded maximum file bit rate limit of the server).                                                |
| c-playerid      | Globally unique identifier (GUID) of the player.                                                                                                                                                                                                                                                                                                                                                                           |
| c-playerversion | Version number of the player.                                                                                                                                                                                                                                                                                                                                                                                              |

**Table 9-15** *Field Descriptions for Windows Media Services 9.0 Logs (continued)*

| <b>Field</b>            | <b>Descriptions</b>                                                                                                                                                                                                           |
|-------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| c-playerlanguage        | Language country code of the client computer.                                                                                                                                                                                 |
| cs(User-Agent)          | Browser type used if Windows Media player was embedded in a browser.                                                                                                                                                          |
| cs(Referer)             | URL of the web page in which Windows Media player was embedded (if it was embedded).                                                                                                                                          |
| c-hostexe               | Host application; for example, a web page in a browser (iexplore.exe), a Microsoft Visual Basic applet (vb.exe), or standalone Microsoft Windows Media player (mplayer2.exe).                                                 |
| c-hostexever            | Host application version number.                                                                                                                                                                                              |
| c-os                    | Operating system of the client computer.                                                                                                                                                                                      |
| c-osversion             | Operating system version number of the client computer.                                                                                                                                                                       |
| c-cpu                   | CPU type of the client computer.                                                                                                                                                                                              |
| filelength              | Length of the file (in seconds). This value is 0 for a live stream.                                                                                                                                                           |
| filesize                | Size of the file (in bytes). This value is 0 for a live stream.                                                                                                                                                               |
| avgbandwidth            | Average bandwidth (in bits per second) at which the client was connected to the server.                                                                                                                                       |
| protocol                | Protocol used to access the stream: HTTP or ASFM (multicast protocol).                                                                                                                                                        |
| transport               | Transport protocol used to deliver the stream (UDP, TCP, or UDP over IP multicast).                                                                                                                                           |
| audiocodec              | Audio codec used in the stream.                                                                                                                                                                                               |
| videocodec              | Video codec used to encode the stream.                                                                                                                                                                                        |
| channelURL              | URL to the .nsc file. A unicast client information log file records a dash (-) for this field.                                                                                                                                |
| sc-bytes                | Bytes sent by the server to the client.                                                                                                                                                                                       |
| c-bytes                 | Number of bytes received by the client from the server. For unicast, the c-bytes value and sc-bytes value must be identical. If not, packet loss has occurred.                                                                |
| s-pkts-sent             | Total number of packets sent by the server.                                                                                                                                                                                   |
| c-pkts-received         | Number of packets from the server (s-pkts-send) that are received correctly by the client on the first try.                                                                                                                   |
| c-pkts-lost-client      | Number of packets lost during transmission from server to client and not recovered at the client layer through error correction or at the network layer through User Datagram Protocol (UDP) resends.                         |
| c-pkts-lost-net         | Number of packets lost on the network layer.                                                                                                                                                                                  |
| c-pkts-lost-cont-net    | Maximum number of continuously lost packets on the network layer during transmission from server to client.                                                                                                                   |
| c-resendreq             | Number of client requests to receive new packets. This field contains a value only if the client is using UDP resend.                                                                                                         |
| c-pkts-recovered-ECC    | Number of packets repaired and recovered on the client layer. Packets repaired and recovered at the client layer are equal to the difference between c-pkts-lost-net and c-pkts-lost-client.                                  |
| c-pkts-recovered-resent | Number of packets recovered because they were resent using UDP.                                                                                                                                                               |
| c-buffercount           | Number of times the client buffered while playing the stream.                                                                                                                                                                 |
| c-totalbuffertime       | Time (in seconds) the client used to buffer the stream. If the client buffers the stream more than once before a log entry is generated, c-totalbuffertime is the total amount of time the client spent buffering the stream. |

Table 9-15 Field Descriptions for Windows Media Services 9.0 Logs (continued)

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

## Enabling the Logging of Usernames to the WMT Transaction Log

If the Content Engine is configured to use the extended format of WMT transaction logging and the extended WMT logging feature is enabled, then the Content Engine logs usernames for any authenticated WMT requests. Usernames are logged not only for NTLM authentication but also for Negotiate, Digest, and basic authentication.



### Note

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

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

To enable the logging of usernames for any authenticated WMT request on standalone Content Engines, follow these steps:

- Step 1** Configure the Content Engine to use the extended Windows Media Services 4.1 or Windows Media Services 9 format for transaction logging by using the **wmt transaction-logs format extended** global configuration command.

For more information, see the [“Specifying the Format of the WMT Transaction Logs” section on page 9-50](#).

- Step 2** Enable the Content Engine to log the usernames for any authenticated WMT request.

```
Content Engine(config)# wmt extended transaction-log enable
```

## Windows Media Transaction Log Forwarding

Windows Media transaction logs are forwarded to a Windows Media Server or an upstream Content Engine only if log forwarding is enabled on both the Content Engine (by using the **wmt advanced server log-forwarding enable** global configuration command, which is enabled by default) and the Windows Media Server 9 (using the WMS Client logging plugin, which disabled by default). Log forwarding is supported for the RTSP protocol only.

To enable log forwarding on the Windows Media Server Version 9, follow these steps:

- 
- Step 1** From the Windows Media Services Administration GUI, choose your Windows Media publishing point, and in the details pane on the right, click the Properties tab.
  - Step 2** In the Category pane, choose **Logging**, and in the Plug-in pane, double-click **WMS Client Logging**. The WMS Client Logging Properties dialog box appears.
  - Step 3** Choose the **Log Entries** tab and check the Sessions played from a player cache or a cache/proxy server check box.
  - Step 4** Click **Apply**, and then click **OK**.
  - Step 5** Disable the WMS Client Logging plugin, and then re-enabled it for log forwarding to take effect for that publishing point.
- 

## Using WMT Error Logging

In the ACNS 5.2.1 software release, WMT error logging was enhanced. More information is now logged about the following events:

- When a WMT client is abruptly disconnected
- When any WMT streams are cleared on the Content Engine

Error logs are in the same format and location as syslogs. The WMT log messages are logged to `/local1/errolog/wmt_errorlog.current`.

You can configure the Content Engine for WMT error logging by using the **debug wmt error EXEC** command. This command debugs WMT level 1 functionality.

```
ContentEngine# debug wmt error ?
 client-ip Debug request from a specific client
 server-ip Debug request to a specific server
```

- Specify the **client-ip** *cl-ip-address* option to debug the request from a specific client IP address to level 1 (show error).
- Specify the **server-ip** *sv-ip-address* option to debug the request from a specific server IP address to level 1 (show error).

There is also a **debug wmt trace EXEC** command that debugs WMT level 2 functionality (show error and trace). Content Engine performance is affected when you run the **debug wmt trace** command. Consequently, we recommend that the **debug wmt trace** command be used only at the direction of Cisco Systems technical support personnel.

## Logging WMT Client Disconnects

When a WMT client is disconnected abruptly, the following information is logged in ACNS software error logs:

- Reasons for the client disconnect (for example, the request was blocked by the rules, the maximum incoming or outgoing bit rate limit was reached, the maximum incoming or outgoing bandwidth limit was reached).
- Client information (for example, client IP address, server IP address, the requested URL, client protocol, version of the client media player, the number of packets that the client received, and the number of packets that the server sent).

## Logging the Clearing of WMT Streams on Standalone Content Engines

In the ACNS 5.2.1 software release, the error logs were enhanced to log a message when WMT streams are cleared and the associated processes are stopped on the Content Engine.

See [Table 9-12](#) for a description of the Content Engine CLI commands, which you can use to clear WMT streams on a standalone Content Engine and which result in a message being sent to the error logs.