



Introduction

This chapter provides an overview of the Cisco Application and Content Networking System (ACNS) solution. This chapter contains the following sections:

- [Overview of the ACNS Software Solution, page 1-1](#)
- [Overview of Standalone Content Engine Capabilities, page 1-6](#)



Note

The term *standalone Content Engines* is used throughout this guide to refer to Content Engines that ACNS administrators have intentionally not registered with a Content Distribution Manager (if there is one in the network) so that they can configure, manage, and monitor these Content Engines as standalone devices. This guide focuses solely on deploying, managing, and monitoring standalone Content Engines that are running the ACNS 5.5 software. Multiple standalone Content Engines can be deployed (for example, you can deploy clusters of standalone Content Engines). You can configure standalone Content Engines through the Content Engine CLI, Content Engine GUI, or the Setup utility that was introduced in the ACNS 5.2.1 software release.

For information about how to deploy, manage, and monitor Content Engines that are registered with a Content Distribution Manager, see the *Cisco ACNS Software Configuration Guide for Centrally Managed Deployments, Release 5.5*.

Overview of the ACNS Software Solution

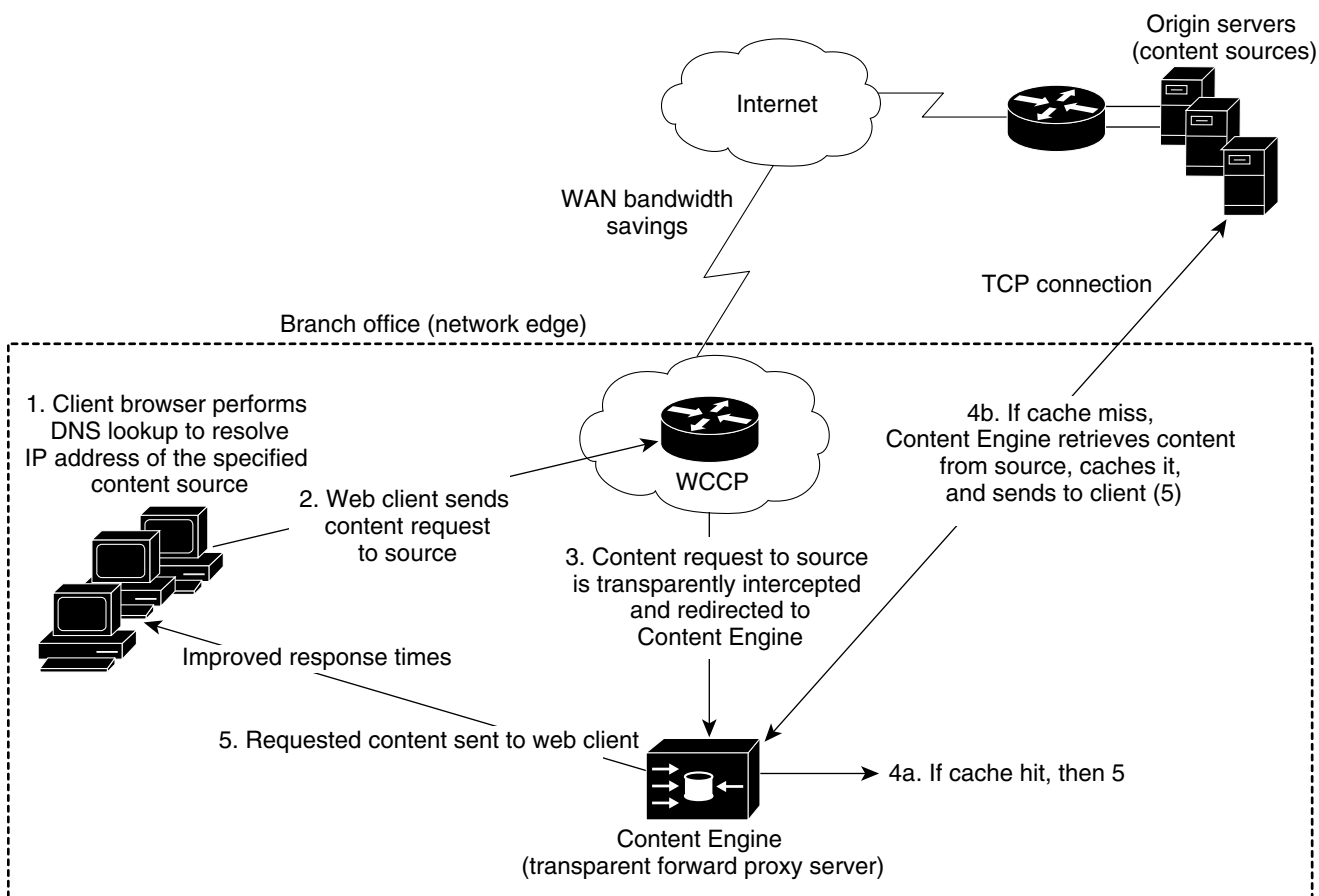
With the advent of e-business applications such as e-commerce, e-learning, knowledge sharing, and corporate communications, networks can experience uncontrollable bottlenecks in the flow of traffic. The ACNS solution helps enterprises and Internet service providers (ISPs) protect their networks from these uncontrollable bottlenecks and efficiently distribute rich media files to their end users. Designed for affordability and ease of installation, the ACNS solution enables high-impact, high-bandwidth rich media such as high-quality streaming video to be quickly deployed with minimal administration. Streaming is a technology that allows content to be accessed or viewed before all the media packets have been received, with caching, the content must be received in its entirety before it can be accessed.

ACNS software supports multiple content routing methods and allows Content Engine caches to be populated with content in multiple ways. Content Engines with ACNS software installed accelerate content delivery by caching frequently accessed content (transparently or proxy-style) and then locally fulfilling content requests rather than traversing the Internet or intranet to a distant server.

By caching content locally, Content Engines minimize redundant network traffic that traverses WAN links. As a result, WAN bandwidth costs either decrease or grow less quickly. This bandwidth optimization increases network capacity for additional users or traffic and for new services such as Voice over IP (VoIP).

For example, enterprises can deploy one or more Content Engines at each branch office, configure access control and filtering on each of these Content Engines, and then push content to these Content Engines. The Content Engine at each branch office uses specific policies to determine if a content request should be denied or granted. If access to the content is granted, the Content Engine checks its local cache for a copy of the content. If the content is already stored in its local cache, the Content Engine sends the client the cached content; otherwise, it retrieves the content from the source (origin server), caches the content, and sends the cached content to the client. When the Content Engine receives subsequent client requests for the same content, it sends the client the cached content instead of retrieving the same content from the origin server again. See Figure 1-1 for a sample deployment.

Figure 1-1 Sample Enterprise Deployment for Standalone Content Engines at Branch Offices



In this sample deployment, client requests are transparently redirected to the Content Engine at the branch office by a router that is running Cisco’s Web Cache Communication Protocol (WCCP). Other possible routing methods include transparent redirection through a Layer 4 Cisco Content Services Switch [CSS] switch or direct proxy routing (web clients are explicitly configured to send their requests directly to the Content Engine).

**Note**

The Setup utility expedites the process of getting standalone Content Engines up and running with a set of commonly used caching services (listed in [Table 4-2](#)).

Types of ACNS Network Devices

[Table 1-1](#) shows the three different types of ACNS network devices.

Table 1-1 **Types of ACNS Network Devices**

Device	Description	More Information
Content Distribution Manager	Is a centralized content and device management station that includes the configuration and monitoring of ACNS 5.x network devices (Content Routers, Content Engines that are registered with the Content Distribution Manager), content acquisition and distribution, and services. Allows you to centrally manage devices (Content Engines and Content Routers) as groups instead of individually.	See the <i>Cisco ACNS Software Configuration Guide for Centrally Managed Deployments, Release 5.5</i> .
Content Engine	Serves the requested content to the client. Content Engines can be deployed in one of two ways: <ul style="list-style-type: none"> Standalone Content Engines (the focus of this guide) or Content Engines that are registered with a Content Distribution Manager (see the <i>Cisco ACNS Software Configuration Guide for Centrally Managed Deployments, Release 5.5</i>). 	For an overview of standalone Content Engines, see the “About Standalone Content Engines” section on page 1-4.
Content Router	Redirects content requests to the registered Content Engine that is closest to the client. This type of request redirection is referred to as <i>content routing</i> . With content routing, a Content Engine must be registered with a Content Distribution Manager. Standalone Content Engines do not support content routing.	See the <i>Cisco ACNS Software Configuration Guide for Centrally Managed Deployments, Release 5.5</i> .

**Note**

The ACNS software device mode determines whether the ACNS device is functioning as a Content Distribution Manager, Content Engine, or Content Router. You can specify the device mode through the **device mode** global configuration command. The default device mode is Content Engine.

As the following sample deployments indicate, all three types of ACNS network devices do not need to be present in order for the ACNS 5.x network to function:

- Deployment A—A single standalone Content Engine (no Content Distribution Manager or Content Router)
- Deployment B—Several standalone Content Engines (no Content Distribution Manager or Content Router)
- Deployment C—One Content Distribution Manager, several Content Engines that are registered with a Content Distribution Manager, and no Content Router
- Deployment D—One Content Distribution Manager, several Content Engines that are registered with a Content Distribution Manager, and one or more Content Routers

About Standalone Content Engines

Content Engines accelerate any HTTP-deliverable and streaming media-type content by storing and delivering content close to the end users (web clients) on their local networks. The term *web client* denotes an end user who is using a browser or media player to request content or information. For more information about the supported web clients, see the [“Web Clients Supported by Standalone Content Engines” section on page 1-12](#).

Standalone Content Engines can be deployed in one of the following ways:

- Inside an enterprise firewall on an internal network
- At the edge of the enterprise network (in branch offices)



Note

Multiple standalone Content Engines can be deployed (for example, you can deploy clusters of standalone Content Engines).

Standalone Content Engines can service content requests from the following types of clients:

- Web browsers (for example, Microsoft Internet Explorer)
- Streaming media players (for example, Windows Media players, RealMedia players [RealPlayer and RealOne players]).

Client requests for content can be routed to standalone Content Engines in one or more of the following ways:

- Direct proxy routing in which the client browsers and media players are configured to send their requests directly to the Content Engine, which acts as a nontransparent forward proxy server
- Transparent redirection (routers and switches transparently intercept web requests and send them to the Content Engine for inspection and manipulation)
 - Cisco WCCP routing
 - Layer 4 switching

With transparent redirection, the WCCP-enabled router or a Layer 4 switch transparently intercepts the client request and redirects it to the Content Engine instead of to the intended server. The Content Engine poses as the intended server, acknowledges the request, and establishes the connection with the client. The client believes that it has established a connection with the intended server even though it is actually using a connection with the Content Engine. For more information about transparent redirection, see the [“Deploying Caching and Streaming Services in Transparent Mode” section on page 3-6](#).

One or more routing methods can be used in a single environment. For example, you can configure a standalone Content Engine to use direct proxy routing for HTTP or FTP-over-HTTP requests but use transparent redirection for Windows Media Technologies (WMT) requests. The routing method that is used to route content requests to the standalone Content Engine determines the types of content services that the Content Engine can support. For more information, see the “[Caching and Streaming Services with Standalone Content Engines](#)” section on page 1-9.

**Note**

Content routing is not supported on standalone Content Engines. If content routing is used, you must register the Content Engines with the Content Distribution Manager. For information about content routing, see the *Cisco ACNS Software Configuration Guide for Centrally Managed Deployments, Release 5.5*.

Types of Content Served in ACNS Networks

Content is the fundamental element of the ACNS network; it represents all the data that the ACNS network handles. Content can be a file or a media stream, and may be on-demand, preloaded, pre-positioned, or live. Content can be classified based on how the content is acquired, distributed, or served.

[Table 1-2](#) describes the different types of content that can be served in an ACNS 5.x network.

**Note**

Pre-positioned content is only supported on Content Engines that are registered with a Content Distribution Manager. Although pre-positioned content is not supported on standalone Content Engines, *preloaded* content is supported on standalone Content Engines.

Table 1-2 *Types of Content Served in an ACNS Network*

Type of Content	Description
On-demand	<p>Content that is acquired, cached, and delivered because of a user request (client-triggered demand), as shown in Figure 1-1. This type of caching is called <i>demand-pull caching</i>. You can configure demand-pull caching on a standalone Content Engine when it is operating in transparent mode (Content Engine receives requests through transparent redirection) or nontransparent mode (Content Engine receives requests directly from the web client).</p> <p>Cached content that is retrieved through HTTP is stored in the cache file system (cfs) storage partition on the Content Engine. Cached content that is retrieved through the two streaming protocols (WMT and RTSP) is stored in the media file system (mediafs) storage partition on the Content Engine.</p>
Preloaded	<p>Content that is retrieved and stored on a standalone Content Engine because you scheduled a retrieval of specific content in anticipation of user requests for that content. The following types of content can be preloaded on a standalone Content Engine: HTTP URLs and FTP-over-HTTP URLs. All configured HTTP and FTP-over-HTTP apply to the preloaded objects. During the preload process, the standalone Content Engine scans web sites several link levels down for content, retrieves the specified content, and stores it locally. When the Content Engine receives future requests for this content, it serves the content from its local storage, which results in WAN bandwidth savings as well as accelerated delivery of the content to the web client.</p> <p>For more information about preloading content on standalone Content Engines, see the “Configuring Content Preloading for Standalone Content Engines” section on page 11-2.</p>

Table 1-2 *Types of Content Served in an ACNS Network (continued)*

Type of Content	Description
Pre-positioned	<p>Content that is retrieved and distributed through a network of Content Engines that are registered with a Content Distribution Manager because the ACNS network administrator has configured acquisition and distribution of content on these Content Engines in anticipation of user requests.</p> <p>Used as a means of distributing content to populate Content Engines in a centrally managed ACNS network environment. Bandwidth-intensive content objects, such as Java applets, Macromedia Flash animations, Shockwave programs, and other file formats can be managed and scheduled for distribution to Content Engines during off-peak hours.</p> <p>For information about managing pre-positioned content, see the <i>Cisco ACNS Software Configuration Guide for Centrally Managed Deployments, Release 5.5</i>.</p>
Live	<p>Content stream (typically streaming media) that is being broadcast from an origin server. (This content is acquired as a live streaming broadcast from either a satellite or a terrestrial broadcasting source.) You configure the policies associated with obtaining the live multimedia stream, such as the program listing URL (playlist), the maximum bit rate, and so forth, as well as the distribution policies, such as priority, schedule, and maximum bandwidth.</p> <p>For information about configuring a standalone Content Engine to deliver WMT live streams, see the “Configuring Standalone Content Engines to Deliver WMT Live Streams” section on page 9-39.</p>

The remainder of this chapter focuses on standalone Content Engine deployments only. The routing methods and content services that are supported by a Content Engine vary based on whether the Content Engine is a standalone Content Engine or is a Content Engine that is registered with a Content Distribution Manager. For information about Content Engines that are registered with a Content Distribution Manager, see the *Cisco ACNS Software Configuration Guide for Centrally Managed Deployments, Release 5.5*.

Overview of Standalone Content Engine Capabilities

The section provides an overview of standalone Content Engine deployments, and includes the following sections:

- [Proxying Services with Standalone Content Engines, page 1-7](#)
- [Caching and Streaming Services with Standalone Content Engines, page 1-9](#)
- [Filtering and Access Control with Standalone Content Engines, page 1-15](#)
- [Monitoring and Troubleshooting Features with Standalone Content Engines, page 1-17](#)

For sample deployment scenarios, see Chapter 3, “Deployment Scenarios for Standalone Content Engines.”

Proxying Services with Standalone Content Engines

Standalone Content Engines can be deployed as proxy servers to provide the following core proxying services:

- Forward proxy caching
- Reverse proxy caching

**Note**

A proxy server is an intermediary server that accepts requests for content from clients. If the proxy server already has a copy of the requested content in its local storage (cache), it services these requests from its own local storage; otherwise, it forwards these requests to the origin server or other proxy servers. A proxy server functions as both a client and a server. It acts as a server to the web client that is requesting the content, and acts as a client to the servers (for example, the origin server or other proxy servers [for example, the specified upstream proxy servers]) that it connects to. A proxy server is commonly called a *proxy*.

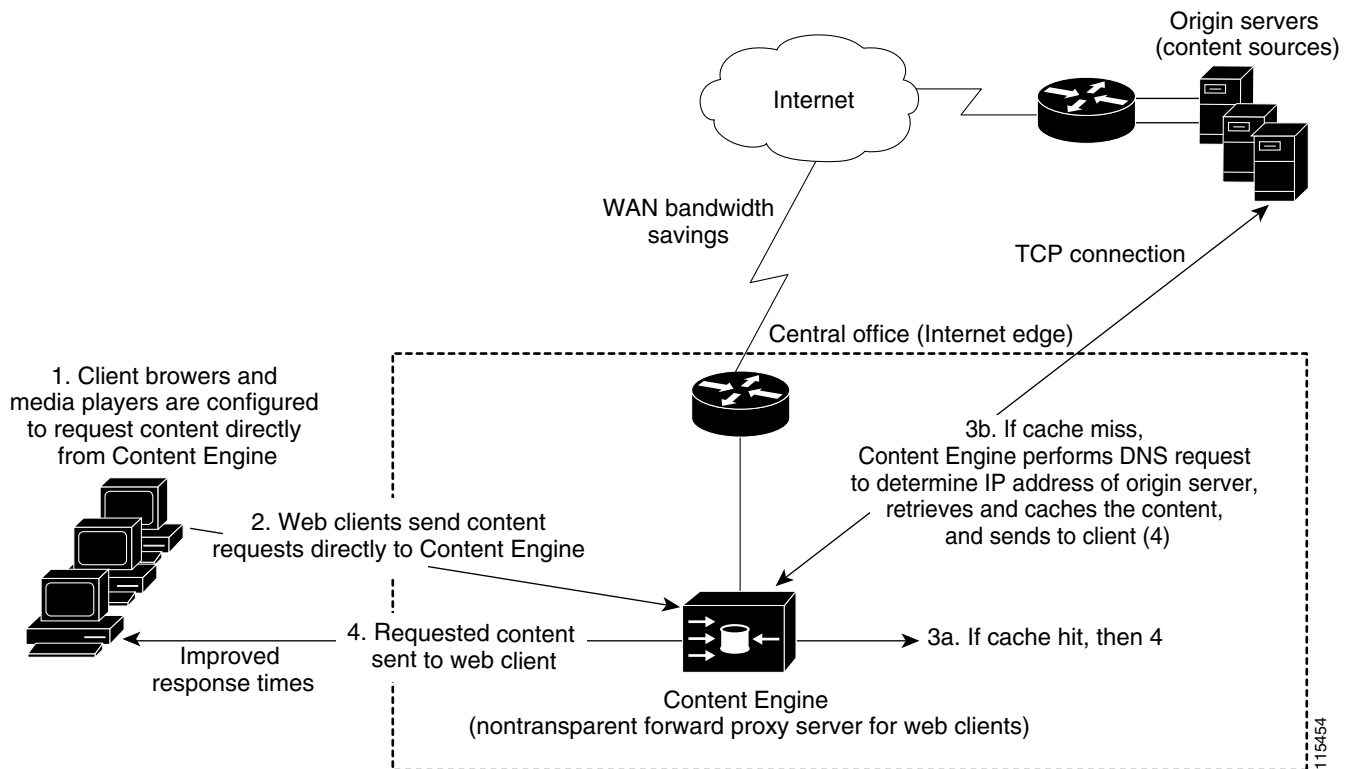
Forward Proxy Caching

With forward proxy caching, the standalone Content Engine acts as a proxy server for web clients. The Content Engine (forward proxy server) provides internal clients access to the Internet through a firewall.

If the client browsers and media players are explicitly configured to send their content requests to the Content Engine (forward proxy server), this is referred to as *direct proxy routing*. When direct proxy routing is used to direct client requests to the Content Engine, the Content Engine is operating in nontransparent mode; the clients are aware that their requests are being directed to the Content Engine. The Content Engine uses specific policies and rules to determine whether a client should be granted or denied access to the requested Internet content. This type of forward proxying service is usually provided as part of a larger Internet security solution in enterprise environments. By implementing this service, enterprises enable their end users (web clients) to go outside the firewall without compromising the integrity of the company's private network.

The direct proxy routing method is the most straightforward routing method. This routing method is typically used when the user desktops are tightly controlled. Consequently, direct proxy routing is generally used in enterprise environments as opposed to service provider environments. See [Figure 1-2](#) for a sample deployment of forward proxy caching when direct proxy routing is used to direct client requests to the Content Engine. See [Table 1-5](#) for a list of supported caching and streaming services when direct proxy routing is used to route requests to the Content Engine. For more information about deploying caching and streaming services with direct proxy routing, see the “[Deploying Caching and Streaming Services in Nontransparent Mode](#)” section on page 3-3.

Figure 1-2 Sample Deployment of Forward Proxy Caching with Direct Proxy Routing



You can also deploy a Content Engine as a transparent forward proxy server that provides the same benefits as direct proxy routing but does not require any configuration changes to client desktops. In this type of deployment, clients are not aware that their content requests are being redirected to the Content Engine (transparent forward proxy server). The transparent redirection method can be implemented through a WCCP-enabled router or a Layer 4 switch.

Although the transparent redirection method demands an understanding of network topology and traffic patterns, organizations generally prefer to use this method because it does not require any configuration changes to the client desktops. However, there may be a legacy requirement for using direct proxy routing. There also may be cases in which organizations need to use direct proxy routing for a particular service (for example, HTTPS proxy caching) because they are not allowed to make the necessary configuration changes to the WCCP-enabled router or switch at the branch office. For more information, see the [“Overview of Transparent Reverse Proxy Caching”](#) section on page 3-10.

Reverse Proxy Caching

With reverse proxy caching, the standalone Content Engine acts as a proxy server for the servers in a web server farm, and the web clients are not aware that their HTTP requests are being transparently redirected to the Content Engine (transparent reverse proxy server). In such deployments, the Content Engine acts as a proxy for the servers in the web server farm.

**Note**

A major difference between reverse proxy caching and forward proxy caching is that with reverse proxy caching the Content Engine is configured to cache only specific content (for example, only content from the web servers in the web server farm), with forward proxy caching, the Content Engine is configured to cache content whenever possible.

Reverse proxy caching also enables a standalone Content Engine to provide external clients access to internal content (for example, content on a company's intranet) through a firewall. Typically, this reverse proxy caching is used for secure web publishing. In a reverse proxy cache configuration, the proxy server is configured with an Internet-routable IP address. Web clients are directed to the proxy server based on DNS resolution of a domain name. To a web client, the reverse proxy server appears to be the origin web server.

Some significant advantages to deploying reverse proxy caching on standalone Content Engines are as follows:

- It provides an alternative to web server expansion by offloading the processing of static images from the server farm. By having the reverse proxy server transparently handle inbound requests for content, web traffic is significantly reduced.
- It provides a possible way of replicating content to geographically dispersed areas by deploying Content Engines in these areas.
- It does not require any client configuration changes (you do not need to configure the client browsers to point to the Content Engine that is functioning as the reverse proxy server).

Reverse proxy caching is a WCCP service (service 99). Consequently, both the router and the Content Engine must be configured to run the reverse proxy service. See [Table 1-6](#) for a list of supported caching and streaming services when transparent redirection is used to direct client requests to the Content Engine. For more information about deploying the reverse proxy caching service, see the [“Overview of Transparent Reverse Proxy Caching”](#) section on page 3-10.

Caching and Streaming Services with Standalone Content Engines

The ACNS software supports several types of caching and streaming media services. The types of services that you can configure on a standalone Content Engine vary depending on how the content request is routed to the Content Engine.

[Table 1-3](#) lists the streaming media solutions supported by the ACNS 5.x software, including Microsoft Windows Media Technologies (WMT), and Real-Time Streaming Protocol (RTSP) solutions from RealNetworks, Inc., Apple Computer, and Cisco Systems, Inc.

Table 1-3 ACNS 5.x Software Streaming Media Solutions

Solutions	Description of Solution	Standalone Content Engines	Content Engines Registered with Content Distribution Manager
Microsoft WMT	Microsoft solution for streaming media. Windows Media Services 9 (WMS 9) supports the RTSP/RTP protocol (IETF standard RTSP protocol plus proprietary extensions). In the ACNS 5.3 software release, support for WMS 9-over-RTSP was added.	WMT proxy and server are installed on the Content Engine. Windows Media clients request streaming media content.	Same as standalone Content Engine.
RealNetworks RealMedia	RealNetworks, Inc. solution for streaming media that uses RealNetworks RTSP protocol (IETF standard RTSP protocol plus proprietary extensions).	RealProxy is the only RealMedia component that can be installed. RealProxy uses the RealNetworks RTSP protocol to serve content to RealMedia clients (RealPlayer and RealOne players).	Same as standalone Content Engines except that all of the RealMedia components (for example, RealProxy and RealSubscriber) can be installed on the Content Engine because it is registered with a Content Distribution Manager.
Apple QuickTime	Apple Computer solution for streaming media that uses the IETF standard RTSP protocol.	CLI-based live and rebroadcast programs can be played back from the QuickTime player.	Cisco Streaming Engine runs on the Content Engine and delivers QuickTime-compliant content (for example, MOV and MPEG-1 content) to QuickTime clients.
Cisco Streaming Engine	Cisco RTSP-based streaming media solution that uses the IETF standard RTSP protocol.	Cisco Streaming Engine can be enabled on a stand-alone Content Engine for live and video-on-demand (VOD) streaming services for local users.	Cisco Streaming Engine runs on the Content Engine and delivers pre-positioned content. Also used to support IP/TV integration for live and video-on-demand (VOD) streaming services for local users.

**Note**

See [Table 1-5](#) and [Table 1-6](#) for a list of supported caching and streaming services when direct proxy routing or transparent redirection is used to direct client requests to a standalone Content Engine.

RTSP is a standard Internet streaming control protocol (RFC 2326). It is an application-level protocol that controls the delivery of streaming media data with real-time properties, such as video and audio, and has been widely adopted as a streaming media protocol in the caching and ACNS environments. Apple QuickTime, RealNetworks RealProxy, and the Cisco Streaming Engine are all backend RTSP servers that use RTSP as a streaming media protocol. On standalone Content Engines, only RealProxy is supported.

The RealNetworks RealProxy feature uses the RealNetworks RTSP protocol, which includes proprietary extensions to the standard IETF standard RTSP protocols. The RealProxy feature enables a standalone Content Engine to support RealMedia transparent and proxy caching. RealProxy also enables the Content Engine to stream cached VOD files to RealMedia players, and to support live splitting.

The RealProxy feature is licensed RealNetworks software. To enable this feature on a Content Engine, you must have a RealProxy license key, which is supplied on a certificate shipped with the Content Engine. If you are downloading the ACNS 5.x software, you can purchase a RealProxy license through the Cisco.com website. For more information, see the “[Configuring RealMedia Services](#)” section on page 8-9. For more information on RTSP streaming media services with standalone Content Engines, see [Chapter 8, “Configuring RealMedia Services on Standalone Content Engines.”](#)

You can use the Cisco Streaming Engine to pre-position RTSP-based content on Content Engines that are registered with a Content Distribution Manager. The Cisco Streaming Engine uses the standard RTSP protocol to serve QuickTime-compliant content (for example, MOV and MPEG-1 content) to clients.

Cisco IP/TV is a member of the Cisco content networking product family. Cisco IP/TV consists of IP/TV Program Manager, one or more IP/TV Broadcast Servers, and IP/TV Viewer or the QuickTime web plug-in. The central management platform for the IP/TV network, IP/TV Program Manager, offers a simple browser interface to schedule and set policies for live and rebroadcast events and VOD files, as well as recording capabilities. IP/TV Broadcast Servers offer real-time encoding, and delivery of live, scheduled, and on-demand video.

When the Cisco Streaming Engine is used in conjunction with the IP/TV 5.1 software and later releases, it provides both live and VOD streaming services to local users. The following two examples show how the Cisco IP/TV solution (the IP/TV Release 5.1 and later releases) can be used in conjunction with the ACNS 5.1 software and later releases, and Content Engines that are registered with a Content Distribution Manager:

- Record live events created with IP/TV and deliver the content using the ACNS 5.1 software and later releases.
- Extend the reach of IP/TV live or rebroadcast events to multicast islands through the ACNS 5.1 software and later releases. The term *multicast islands* refers to non-multicast-enabled networks or environments with a combination of a unicast WAN and a multicast LAN.

In the ACNS 5.1 software, a fourth device mode (Program Manager) was added for certain devices to support IP/TV integration into ACNS software networks (the ACNS 5.1 software and later releases). For example, you can configure the Content Engine CE-565 or CE-7305 model as an IP/TV Program Manager by specifying Program Manager as the device mode. When you launch the Setup utility on a device that supports device mode changes, you are asked to specify the device mode. To configure a device as a Program Manager through the Setup utility, specify **PM** as the device mode:

```
What is the mode of the device (CE/CR/CDM/PM) [CE]: PM
```

Cisco IP/TV and the Cisco Streaming Engine are not supported on standalone Content Engines. For information about deploying the Cisco Streaming Engine, see the *Cisco ACNS Software Configuration Guide for Centrally Managed Deployments, Release 5.5*. For more information about IP/TV, see the Cisco IP/TV 5.x product documentation.

Supported Protocols for Caching and Streaming

The interaction between a web client (browser or media player) and a web server uses the existing standard application-layer Internet protocols such as HTTP and RTSP. The Content Engine has to be able to serve web objects to the web client using all of these web access protocols.

Table B-1 describes the network protocols that a Content Engine running the ACNS 5.5.x software can use to serve content to the web client. Table B-2 lists the streaming media protocols, control channels, the corresponding data format, and transport types that can be used to deliver streaming media files with standalone Content Engines.

Support for HTTP, FTP, TFTP, HTTPS, and the IETF standard RTP/RTSP protocols is included as part of the ACNS software product (the ACNS 5.1 software and later releases). Support for the following two features require a separate license:

- The WMT feature, which offers Windows Media Services, requires a WMT license. For more information, see the “[Configuring WMT RTSP Streaming and Caching Services on Standalone Content Engines](#)” section on page 9-14.
- The RealNetworks RealProxy feature, which uses RealNetworks’ RTSP protocol that includes proprietary extensions to the standard IETF standard RTSP protocol, requires a RealProxy license. For more information, see the “[Configuring RealMedia Services](#)” section on page 8-9.

Web Clients Supported by Standalone Content Engines

Table 1-4 lists the web clients that can communicate with standalone Content Engines that are running the ACNS 5.4.x software.

Table 1-4 Web Clients Supported by Standalone Content Engines

Client Protocol	Client
HTTP protocol <ul style="list-style-type: none"> • HTTP • HTTPS-over-HTTP • MMS-over-HTTP 	All Internet browsers including Microsoft Internet Explorer, Netscape, and all others that conform to HTTP 1.0 or HTTP 1.1 specifications. Windows Media players, Version 6.x and 7.x, can use MMS-over-HTTP to request content from a Content Engine.
FTP-over-HTTP	Client browsers issuing FTP-over-HTTP requests (support was added in the ACNS 5.1 software release).
Native FTP	FTP clients (for example, Reflection X clients, WS-FTP clients, or Unix or DOS command-line FTP programs) issuing FTP native requests to the Content Engine that is acting as a nontransparent proxy server for these clients’ FTP native requests (support was added in the ACNS 5.3.1 software release).
Trivial File Transfer Protocol (TFTP)	TFTP clients (support was added in the ACNS 5.1 software release).

Table 1-4 Web Clients Supported by Standalone Content Engines (continued)

Client Protocol	Client
RealNetworks proprietary RTSP protocol	RealMedia players use this protocol (the standard IETF RTSP protocol plus proprietary RealNetworks extensions) to request content from the Content Engine The following media players are collectively referred to as <i>RealMedia players</i> : <ul style="list-style-type: none"> • RealPlayer (Version 8.x and later) from RealNetworks, Inc. • RealOne player
Microsoft proprietary RTSP protocol	Windows Media 9 players use this protocol (the standard IETF RTSP protocol plus proprietary Microsoft extensions) in TCP mode to request content from the Content Engine (support was added in the ACNS 5.3.1 software release).

Standalone Content Engines can deliver streaming media as live content or as on-demand content (for example, VOD files) to RealMedia players and WMT players. Standalone Content Engines do not support requests from QuickTime players or Cisco IP/TV Viewer.

Supported Caching and Streaming Services with Direct Proxy Routing

Table 1-5 lists the supported caching and streaming services when direct proxy routing is used to direct client request to a standalone Content Engine. An asterisk (*) indicates that the particular caching service can be configured through the Setup utility as well as through the Content Engine CLI.

Table 1-5 Caching and Streaming Services with Direct Proxy Routing

Services	More Information
Conventional Caching	
HTTP forward proxy caching*	Configuring Nontransparent HTTP Forward Proxy Caching on Standalone Content Engines
FTP-over-HTTP caching	Configuring Nontransparent FTP-over-HTTP Caching on Standalone Content Engines
FTP native caching	Configuring Nontransparent FTP Native Caching
HTTPS proxy caching	Configuring HTTPS Caching for Standalone Content Engines
WMT Caching and Streaming	
WMT proxy caching*	Enabling and Configuring WMT Caching on Standalone Content Engines
Streaming of live WMT streaming	Configuring Standalone Content Engines to Deliver WMT Live Streams
Streaming of preloaded VOD files	Configuring Standalone Content Engines to Distribute VOD Files
RTSP Caching and Streaming	
RealMedia proxy caching*	Configuring Direct Proxy Routing and RealMedia Proxy Caching
RealProxy streaming of cached VOD files and live splitting	Configuring RealMedia Services

Supported Caching and Streaming Services with Transparent Redirection

Table 1-6 lists the caching and streaming services that are supported when the standalone Content Engine receives the request through transparent redirection. An asterisk (*) indicates that the particular caching service can be configured through the Setup utility as well as through the Content Engine CLI.

Table 1-6 Caching and Streaming Services with Transparent Redirection

Services	More Information
Conventional Caching	
HTTP reverse proxy caching*	Configuring HTTP Reverse Proxy Caching for Standalone Content Engines
FTP native caching	Configuring Transparent FTP Native Caching
HTTPS transparent caching	Configuring HTTPS Transparent Caching for Standalone Content Engines
HTTP transparent caching*	Configuring Transparent HTTP Forward Proxy Caching for Standalone Content Engines
DNS caching	Configuring the DNS Caching Service (Service 53) for Standalone Content Engines
WMT Caching and Streaming	
WMT transparent caching*	Enabling and Configuring WMT Transparent Caching on Standalone Content Engines
Streaming of WMT live streams	Configuring Standalone Content Engines to Deliver WMT Live Streams
Streaming of preloaded VOD files	Configuring Standalone Content Engines to Distribute VOD Files
RTSP Caching and Streaming	
RealMedia transparent caching*	Configuring RTSP Transparent Redirection and Caching of RealMedia Requests
RealProxy streaming of cached VOD files and live splitting	Configuring RealMedia Services



Note

RealProxy is enabled through the Content Engine CLI or through the Setup utility. RealProxy is enabled with a default configuration file. To change the RealProxy default configuration, you must use the RealNetworks RealSystem Administrator GUI. You can use the Content Engine CLI to restore the RealProxy default configuration file on a standalone Content Engine by entering the **rtsp real-proxy default-configuration EXEC** command. For more information, see the “[Configuring RealMedia Services](#)” section on page 8-9.

Filtering and Access Control with Standalone Content Engines

Other core capability of standalone Content Engines is to the ability to filter and control access to web content. Content Engines can be configured to use their local database or a remote authentication, authorization, and accounting (AAA) server to authenticate and authorize client requests. With the ACNS software (the ACNS 5.2.1 software and later releases), AAA accounting through TACACS+ is also available. (See [Chapter 18, “Configuring AAA Accounting on Standalone Content Engines.”](#))

You can use URL filtering and the Rules Template to block access to any URL or modify the actual content stream (for example, rewrite certain headers). Through the use of access control lists (ACLs), filters can be applied to specific addresses, groups of addresses, or groups of users. In addition to these policies, there are bandwidth limitations and resource controls that determine whether a client request will be accepted at all.

**Note**

The access control and filtering services that are supported on a standalone Content Engine vary depending on the content protocol (for example, access control is supported for HTTP, HTTPS, and FTP-over-HTTP requests, but ICAP is only supported for HTTP and FTP-over-HTTP). See [Table B-5](#) for a list of the access control and filtering content services that are supported with standalone Content Engines that are running the ACNS 5.4.x software.

In the ACNS 5.2.3 software and later releases, you can also configure a Content Engine to bypass URL filtering for certain HTTP and HTTPS requests. For more information on this feature, see the [“Configuring Content Engines to Bypass URL Filtering for Specific HTTP and HTTPS Requests”](#) section on page 11-41.

After the standalone Content Engine receives a client request for content, it performs the following tasks:

- Authenticates the web client, asks the client to provide a username and password so that it can consult an AAA server, and checks whether the client is allowed to access the web. This type of authentication and authorization is referred to as *content authentication*. For more information, see the [“Authentication, Authorization, and Accounting with Standalone Content Engines”](#) section on page 1-16.
- Passes the request through a filter such as Websense or SmartFilter to make sure that the requested object is not objectionable content. For more information, see [Chapter 11, “Configuring Content Preloading and URL Filtering on Standalone Content Engines.”](#)

**Note**

The ACNS 5.x software relies on third-party software to implement content filtering. Supported filtering software includes Websense, N2H2, and SmartFilter.

- Compares content against configured rules, which might rewrite certain headers, redirect the request, or otherwise manipulate the request. See [Table 13-2](#) for a list of the supported rule actions per protocol. For information about how to configure rules on a standalone Content Engine, see [Chapter 13, “Configuring the Rules Template on Standalone Content Engines.”](#)
- Checks to see whether the requested content is already in its cache. If so, then the Content Engine serves the object directly from its local cache, rather than from the origin web server, which saves bandwidth to the Internet.
- If the requested content is not already in its cache, the Content Engine retrieves it from the Internet on the client’s behalf, and caches the content for future use if appropriate. The ACNS 5.x software supports this functionality by supporting the web access protocols (including HTTP) and all streaming protocols listed in [Table B-1](#).

Standalone Content Engines, which are running the ACNS 5.1 software and later releases, also support IP packet filtering, which controls access to specific interfaces (services) on the Content Engine. You can configure IP ACLs that determine whether or not IP packets are allowed to cross specific interfaces on a Content Engine. For example, you can use IP ACLs to control access to content serving and management services on the Content Engine. For more information, see [Chapter 19, “Creating and Managing IP Access Control Lists for Standalone Content Engines.”](#)

Authentication, Authorization, and Accounting with Standalone Content Engines

The ACNS 5.x software provides authentication, authorization, and accounting (AAA) support for users who have external access servers (for example, RADIUS or TACACS+ servers), and for users who need a local access database with AAA features.

- *Authentication* (or *login*) is the action of determining who the user is. It checks the username and password.
- *Authorization* (or *configuration*) is the action of determining what a user is allowed to do. It permits or denies privileges for authenticated users in the network. For example, if you log in to a standalone Content Engine with a superuser administrator account (for example, the predefined admin account), you have the highest level of access privileges and can perform any administrative task such as the following:
 - Configure the standalone Content Engine.
 - Obtain statistical information that the standalone Content Engine has collected.
 - Reload the device.



Note Generally, authentication precedes authorization and is not mandatory.

- *Accounting* is the action of keeping track of administrative user activities for system accounting purposes. In the ACNS 5.2.1 software and later releases, AAA accounting through TACACS+ is supported. For more information, see [Chapter 18, “Configuring AAA Accounting on Standalone Content Engines.”](#)

In ACNS 5.x environments, there are two main types of authentication and authorization:

- Content authentication—Controls end user access to content that is served by Content Engines. For more information on this topic, see [Chapter 10, “Configuring Content Authentication and Authorization on Standalone Content Engines.”](#)
- Administrative login authentication—Controls administrative login authentication methods (local, RADIUS, or TACACS+) to process administrator requests to log on to the Content Engine for monitoring, configuration, or troubleshooting purposes.

An administrator can log in to a standalone Content Engine through the console or the Content Engine GUI. To process these administrative login requests, the Content Engine checks the specified authentication database to verify the user’s username and password and to determine the access rights that this particular administrator should be granted during this login session. When the Content Engine receives a login request, the Content Engine can check its local database or a remote third-party database (the TACACS+ database or the RADIUS database) to verify the username and password and to determine the access privileges of the administrator.

You can configure any combination of these authentication and authorization methods to control administrative login access to a standalone Content Engine:

- Local authentication and authorization
- RADIUS
- TACACS+

By default, the Content Engine uses the local login authentication method as the primary method to process administrative login requests. When you enable local authentication with one or more other authentication methods, local authentication is always attempted first if the priority flags are not set. You cannot specify different administrative login authentication methods for console and Telnet connections. For more information, see [Chapter 17, “Configuring Administrative Login Authentication and Authorization on Standalone Content Engines.”](#)

**Note**

Content authentication and authorization is independent of administrative login authentication and authorization.

See [Table B-4](#) for a list of the caching, filtering, and authentication mechanisms supported by standalone Content Engines that are running the ACNS 5.4.x software.

Monitoring and Troubleshooting Features with Standalone Content Engines

It is important that you monitor your Content Engines in order to gauge their performance and to identify any signs that you need to tune their configurations or deploy additional Content Engines. Several tools are available to monitor the performance of standalone Content Engines that are running the ACNS 5.4.x software. This set of tools includes the Cisco Discovery Protocol (CDP), the Simple Network Management Protocol (SNMP), and the ACNS software alarms. For more information on this topic, see the [“Monitoring Standalone Content Engines” section on page 21-2](#).

In addition to monitoring the performance of a Content Engine, transaction monitoring is supported.

**Note**

The term *transaction* refers to a completed successful or failed request for a web resource by a client. Standalone Content Engines that are running the ACNS 5.x software can record all errors and access activities for reporting purposes.

In the ACNS 5.x software, each content service module (for example, the HTTP module, the WMT server, the FTP proxy process, and the TFTP server) on the Content Engine provides logs of the requests that were serviced. Logging for the following types of requests is provided: HTTP requests, HTTPS requests, FTP requests, WMT requests, RTSP streaming requests, and TFTP requests.

For each content transport protocol, there is a corresponding **show protocol-name statistics EXEC** command that displays the statistics for that protocol.

Typically, Content Engine administrators are interested in what types of requests have been made of the Content Engine and what the results of these requests were. For example, if streaming media is a source of revenue for a company, then the company needs a way to track which customer is accessing which content, how long a user viewed the content, and at what viewing quality. Because these companies charge their customers to stream on-demand content and live broadcasts, they must rely on logged information as the basis for billing their customers for their content access services.

The software logs that record requests that are serviced by a Content Engine are called *transaction logs*. Typical fields in the transaction log are the date and time when a client request was made, the URL that was requested, whether it was a cache hit or a cache miss, the type of request, the number of bytes transferred, and the source IP address.

Transaction logs are generally used for the following purposes:

- Problem identification and solving
- Load monitoring
- Billing
- Statistical analysis
- Security problems
- Cost analysis and provisioning

In the ACNS 5.2.1 software and later releases, Windows Media Services 9 logging is supported. The Windows Media Services 9 Series provides a more robust logging model than Windows Media Services Version 4.1.

You can log data in a predefined format (for example, Squid, Extended Squid, or Apache) or a custom transaction log format that allows you to log additional fields. The contents of the transaction logs can be periodically exported to an external server using FTP. You can also configure log rotation policies.

**Note**

Only one logging format type can be active at a time. When transaction logging is enabled through the Content Engine GUI, the Squid log format is used.

The ACNS 5.x software also supports sanitized logging. If the sanitized logging feature is enabled, the logging of web resource requests do not include (or obfuscate) the identity of the web client. The IP address and usernames of clients in the transaction log file are disguised. For more information, see the [“Sanitizing Transaction Logs” section on page 21-42](#).

In the ACNS 5.2.1 software and later releases, the ability to send HTTP transaction log messages to a remote syslog server is supported. This feature allows you to monitor the remote syslog server for HTTP request authentication failures in real time. For more information, see the [“Monitoring HTTP Request Authentication Failures in Real Time” section on page 21-48](#). For more information about the ACNS 5.x software transaction logs, see the [“Monitoring Transactions with Standalone Content Engines” section on page 21-27](#).

In the ACNS 5.2.3 software and later releases, you can configure a Content Engine to monitor the performance of specific URLs. The Content Engine maintains statistics about the various response characteristics for each of the monitored URLs. For more information on this topic, see the [“Monitoring Critical Disk Drives on Standalone Content Engines” section on page 21-17](#).