



Understanding the ACNS 5.2 Network

The Cisco Application and Content Networking System (ACNS) software solution expedites the delivery of strategic applications from the corporate data center to branch offices and local points of sale. Cisco ACNS software combines the technologies of caching, pre-positioning, and live and on-demand streaming to accelerate delivery of web applications, object files, live events, and video, enabling organizations of all sizes to reduce costs, drive up productivity, and increase revenues.

Cisco ACNS software runs on the Cisco Content Engine hardware platform. Content Engines in an ACNS network save bandwidth and protect networks by acting as forward proxies that authenticate, filter, and cache all traffic between an organization's internal network and the Internet. Content Engines also perform reverse proxy caching to reduce the load on expensive back-end web servers.

This chapter describes the basic concepts of the Cisco ACNS 5.2 network. It describes the function and relationships of the different device modes that make up the ACNS network and explains how content is acquired, distributed, stored, and delivered to the end user.

This chapter contains the following sections:

- [ACNS Network Overview, page 1-1](#)
- [Content Overview, page 1-4](#)
- [Content Request Interception and Router Redirection Overview, page 1-5](#)
- [Content Services Overview, page 1-7](#)
- [Content Acquisition and Distribution Architecture, page 1-9](#)
- [Content Replication Overview, page 1-12](#)
- [ACNS Network Topology Considerations, page 1-13](#)

ACNS Network Overview

Cisco ACNS software, Release 5.2 is the basis for the network solution described in this publication. Cisco ACNS software operates in four different device modes. These device modes are Content Engine, Content Distribution Manager, Content Router, and Cisco IP/TV Program Manager. An ACNS network consists of at least one Content Distribution Manager, one or more Content Engines, and one or more optional Content Routers.

Cisco ACNS software allows content services to be configured, reconfigured, and monitored centrally from the Content Distribution Manager graphical user interface (GUI), Cisco's web-based application management tool.

Content Distribution Managers

The primary role of a Content Distribution Manager is to perform centralized content and device management. In the ACNS 5.2 network, the Content Distribution Manager manages both content acquisition and distribution and also manages policy settings and configurations on individual Content Engines. Through the Content Distribution Manager GUI, the network administrator can specify what content is to be distributed and to whom. The Content Distribution Manager also allows the administrator to monitor network nodes and apply changes, such as software upgrades, to groupings of nodes from a central location.

Content Engines

The primary role of a Content Engine in the ACNS network is to serve client requests for content. Content Engines also play a major role in content request routing and in channel distribution of content. The ACNS network deploys Content Engines either inside an enterprise firewall on an internal network, or at the edge of the enterprise network.

Content Engines can be managed centrally through the Content Distribution Manager or locally as separate standalone content caches. (Refer to the *Cisco ACNS Software Configuration Guide for Locally Managed Deployments* publication.)

Content Routers

The primary role of a Content Router in the ACNS network is to redirect client requests for content to the closest Content Engine containing that content. In the ACNS 5.2 network, Content Routers use the Domain Name System (DNS) to ensure that the Content Router receives all requests for content. Once the Content Router receives these requests, it can redirect the end user to the content. (See the [“Configuring Content Request Routing Using a Content Router or Routing Content Engine”](#) section on page 4-42 for more details.)

An alternative to Content Router routing is to use an enterprise router that supports Web Cache Communication Protocol (WCCP) and is configured to intercept and route requests for content. WCCP detects client requests and routes the request to a Content Engine within the same network. WCCP does not require the presence of a Content Router. (See the [“Configuring WCCP Transparent Routing”](#) section on page 4-2.)

IP/TV Program Manager

IP/TV Program Manager Release 5.2 is a Linux-based application running on the Cisco Content Engine hardware. IP/TV Program Manager is accessed from a browser and is used by the system administrator or broadcast administrator to set up and manage IP/TV scheduled or on-demand programs, channels, recordings, and file transfers among IP/TV Servers.

You can access IP/TV Program Manager from Netscape 4.5x through 4.7x, or Microsoft Internet Explorer 5.x or later. The browser must have Java and JavaScript enabled.

Deployment Scenarios

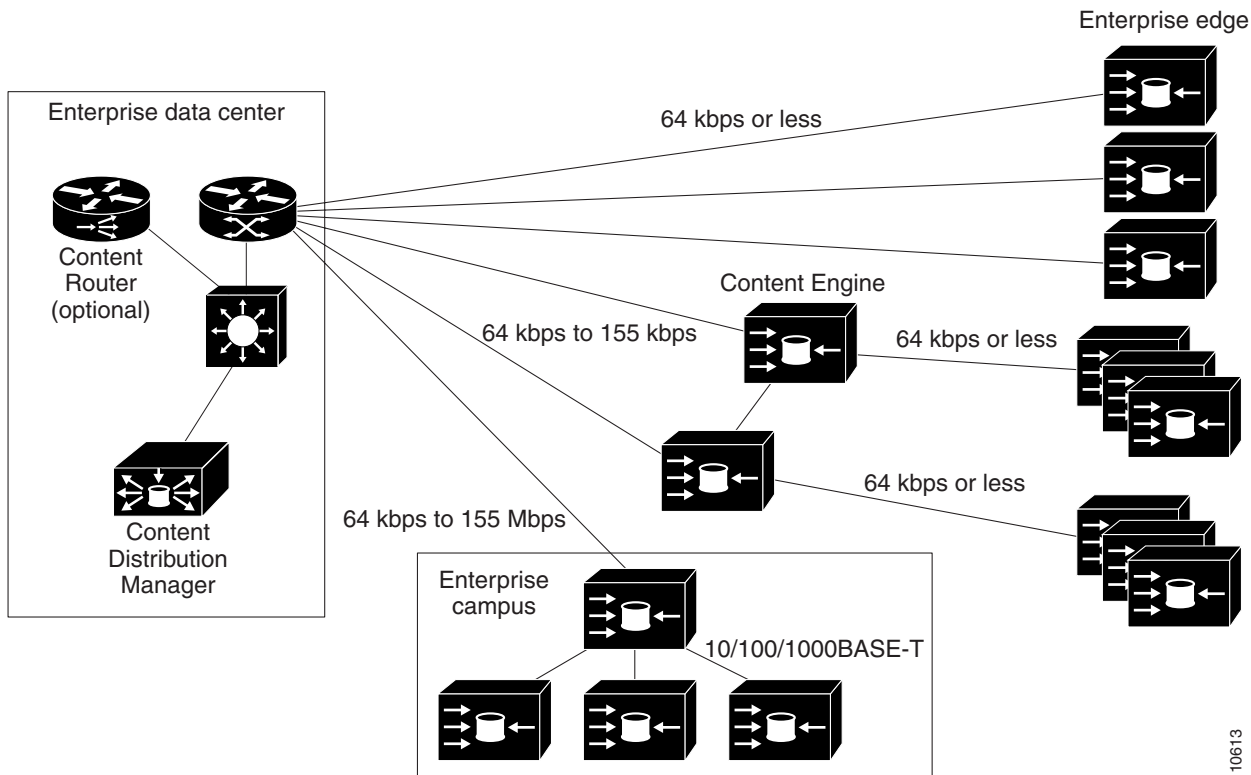
All types of devices do not need to be present in the ACNS network in order for Cisco ACNS software to function. For example, your ACNS 5.2 software might be deployed in any one of the following scenarios:

- One Content Engine only
- Many standalone Content Engines
- One Content Distribution Manager and many managed Content Engines
- One Content Distribution Manager and many managed Content Engines and Content Routers

Cisco ACNS 5.2 software is especially beneficial for deployment in an enterprise or Internet service provider (ISP) network where there is a requirement to both pre-position and cache content. Typically, such a network uses topologies that contain both high-speed and low-speed links. Cisco ACNS 5.2 software allows companies to use off-peak hours to move moderate or high-bandwidth content to the edge of their networks over low-speed links.

Figure 1-1 shows a typical enterprise intranet. Branch offices at the enterprise edge connect to the enterprise data center using low-speed (56-kbps, 128-kbps or 512-kbps) links. In addition, branch offices in the same region may connect to a regional headquarter data center, which then connects to corporate headquarters using limited-bandwidth links.

Figure 1-1 Typical Enterprise ACNS Network Topology



In this example, each branch office has a Content Engine. The corporate data center has a Content Distribution Manager that is managed by the central Information Technology (IT) staff. Content Routers might also be deployed in the data center as well.

Content Overview

Content is the fundamental element of the ACNS network; it represents all the data that the ACNS network handles. Content can be classified on the basis of how it is acquired, distributed, or served. The ACNS 5.2 network serves three classes of content: on-demand content, pre-positioned content, and live content.

This section explains some of the terminology that is used to discuss content in this publication.

On-Demand Content

On-demand content is acquired and cached by the Content Engine, and then delivered by the Content Engine in response to a client request. When the first client request is made for a piece of content, the Content Engine pulls the data from the origin web server and serves it to the client. The content is also stored (or cached) on the Content Engine. Subsequent requests for the same content are served to the client directly from the Content Engine cache.

Pre-Positioned Content

Pre-positioned content is a means of distributing content to populate Content Engines in a centrally managed ACNS network environment. Bandwidth-intensive content objects, such as Java applets, Flash animations, Shockwave programs, and other file formats can be managed and scheduled for distribution to Content Engines during off-peak hours.

The ACNS network administrator uses the Content Distribution Manager GUI to configure a number of channels to specify the destination Content Engines for content delivery and assigns a Content Engine (called a root Content Engine) that is generally (but not necessarily) located in close proximity to an origin web server, to fetch the content according to attributes that are set out in an XML file called a manifest file. The manifest file (see [Appendix A, “Creating Manifest Files”](#)) is used to specify the location from which the pre-positioned content objects should be fetched and the frequency with which the content should be checked for updates. The root Content Engine distributes the content to other Content Engines that serve as content caches and distribution gateways along the enterprise edge.

Pre-Positioned Versus Preloaded Content

Whereas pre-positioned content is associated with a centrally managed ACNS network environment, preloaded content (cache preloading) is configured on a Content Engine-by-Content Engine basis and is not generally associated with a centrally managed ACNS network.

Content Engines can be configured to preload specific content items using HTTP or FTP. Websites are scanned several link levels down for content. Preloaded content can be configured with specified bandwidth limits for better control of network usage.

Content preloading is done by configuring the Content Engine to create a cache request for all the content located at the origin web server that stores the primary content. At a time specified in the **pre-load schedule** command, the Content Engine verifies that its content is still current and updates any content that has changed.

Live Content

Live content is acquired as a live streaming broadcast from either a satellite or terrestrial broadcasting source. The ACNS network administrator configures 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.

Creating and Managing Content in an ACNS Network

Responsibilities for creating and managing content in an ACNS network are divided between webmasters and network administrators. This publication addresses both areas of responsibility.

Creating websites and adding content to the ACNS network are the responsibilities of the webmaster, whereas managing the content and the network is the responsibility of the ACNS network administrator. Webmasters are generally not involved with network topologies, configurations, WAN bandwidths, or Content Engine locations.

Enterprise companies can have multiple webmasters who are responsible for content by business unit, geography, content type, and so forth. Webmasters are authorized to add content to the ACNS network by creating one or more manifest files for channels associated with their website. Webmasters define and schedule acquisition jobs through the manifest file. Content is fetched according to the time defined in the manifest file. (See [Appendix A, “Creating Manifest Files.”](#))

To associate one or more manifest files with the same logical content, the website and all of its associated channels must be registered and predefined in the Content Distribution Manager GUI. A network administrator is the person responsible for registering and defining content providers, websites, and channels through the Content Distribution Manager GUI. The administrator points the channel to a manifest file, assigns Content Engines to the channel, and designates a root Content Engine. (See [Chapter 5, “Configuring the ACNS Network for Content Distribution.”](#))

The ACNS network administrator also assigns channels and allocates disk space. The administrator allocates both the total amount and the per-channel amount of disk space for the website and assigns each channel to a set of device groups. The administrator also schedules bandwidth allocation for each root Content Engine. (See [Chapter 10, “Working with Device Configurations,”](#) and [Appendix B, “Configuring Disk Space.”](#))

Content Request Interception and Router Redirection Overview

Cisco ACNS 5.2 software supports three methods of request handling to deliver content to the end user:

- WCCP transparent interception
- Direct proxy routing (with or without dynamic proxy autoconfiguration)
- DNS-style routing using a Content Router

**Note**

An ACNS 5.2 network can combine both WCCP and routing using a Content Router.

WCCP Transparent Interception

The Content Engine supports WCCP Version 2 redirected requests. When configured for WCCP routing, the Content Engine receives requests from its assigned router and compares the content request against the content currently stored in its cache. If the Content Engine has the requested content (cache hit), the request is served from this cached storage. If the Content Engine does not have the content (cache miss), the content is requested from the origin web server and served to the requesting client through the Content Engine. No proxy configuration is required at the end user's browser. (See the [“Configuring WCCP Transparent Routing”](#) section on page 4-2.)

Direct Proxy Routing

In this mode, end user web browsers need to be explicitly configured to use the IP address or host name of the Cisco Content Engine, and there is no need for additional hardware, such as Layer 4 switches, Content Routers, or WCCP-enabled routers to intercept user requests.

If the Content Engine has the requested content, the request is serviced from this cached storage. If the content is not in the Content Engine's storage, the content is requested from the origin web server and served to the requesting client by way of the Content Engine. (See the [“Configuring Direct Proxy Routing”](#) section on page 4-33 for more information.)

Dynamic Proxy Autoconfiguration

The dynamic proxy autoconfiguration (PAC) routing feature uses a Content Engine that is enabled as a PAC file server. This routing method can be deployed in an ACNS 5.2 network containing a Content Distribution Manager and at least one Content Engine. If end users are behind a NAT device or firewall and the administrator wants to direct the end users to different proxies based on their IP addresses, then there must be a PAC file server behind the NAT device as well. You do not need a Content Router or a routing-enabled Content Engine.

When using the PAC file server feature, the administrator writes one PAC file with ACNS software-defined macros that serves as a template. The PAC file server substitutes the closest proxies for the macros, where the closest proxy is determined by coverage zone information.

The PAC feature differs from proxy mode configuration in the following ways:

- Dynamic PAC supports multiple PAC files.
- The user can specify the names of the PAC files instead of being limited to preset names.
- The PAC file server dynamically adds a set of the nearest proxies to a PAC file template based on coverage zone information and the source IP address of the requester.
- The Content Distribution Manager acquires the PAC file templates and distributes them to Content Engines that are PAC file servers, rather than having the Content Engines themselves acquire the PAC file templates.

Content Router Routing

Content Router routing uses HTTP or RTSP redirection through a Content Router. The Content Router determines which Content Engine is best suited to deliver the desired content to the client by comparing the source IP address of the client end system against a table of address ranges assigned to Content Engines, known as the *coverage zone*. The coverage zone provides information on the proximity

of client end systems to Content Engines based on each client's IP address. The Content Router can then choose the closest, best-suited Content Engine to serve the website request to the client. (See the [“Configuring Content Request Routing Using a Content Router or Routing Content Engine”](#) section on page 4-42 for more information.)

If several Cisco Content Engines are located behind a Network Address Translation (NAT) device or firewall, the Content Router can issue the redirect response to a routing-enabled Content Engine behind the NAT device. The routing Content Engine has the ability to issue a redirect request to another Content Engine that is also located behind the NAT device or firewall. This Content Engine looks at its coverage zone file and issues a subsequent redirect response to the appropriate Content Engine that is configured to serve the client's request. (To enable routing on a Content Engine, see the [“Configuring a Content Engine as a Routing Content Engine”](#) section on page 4-53.)

Content Services Overview

The ACNS 5.2 network combines two types of content services using a single software base: content pre-positioning and content caching with filtering and access control. This section explains the objectives of these services and provides examples that are based on some typical needs and requirements of an enterprise network.

Content Pre-Positioning

When pre-positioning content, the administrator can specify a group of content objects to be “pushed” to the remote branch offices in off-peak times, such as at night. This is advantageous because after the content is pre-positioned at a nearby Content Engine in the branch office, for example, it can be accessed using a higher-bandwidth connection than the typical low-bandwidth link between branch offices and remote corporate headquarters. Pre-positioning can be particularly advantageous for streaming objects.

For example, the training department at <http://www.bigcorp.com> might prepare an online training course that includes multiple videos and slides. The course might be posted at an internal website. If there is no ACNS network, an employee from a remote office taking the course online must wait a long time for the video and slides to load. If, however, the administrator uses an ACNS network to pre-position the videos and slides on a Content Engine at the local branch office, then the employee can take the course as if the website were on the local LAN.

Content Caching and Access Control

Content caching, filtering, and access control services address the need to accomplish the following objectives:

- Reduce bandwidth usage by caching frequently accessed Internet content.
- Authenticate, monitor, log, and control web access from end users to implement corporate policies regarding work environment and system security.

Both objectives are made possible because of the Content Engine's special position as an “in-line” device between end users and the Internet.

For example, BigCorp Enterprise wants to make sure that the following policy, security, and budgetary objectives are met:

- Only full-time employees can access the web.
- No employee can access objectionable content through the corporate network.
- Viruses such as “Code Red” are prevented from being propagated inside the corporate network.
- Monthly payments to the company’s ISP are reduced.

BigCorp Enterprise can achieve all of these objectives by placing Content Engines at the network edge, where the corporate network interfaces with the Internet, and by placing Content Engines at remote branch offices. The Content Engines can then intercept content requests made by end users and enforce the above policies.

A Content Router might also be placed at the BigCorp Enterprise data center to aid in redirecting requests to Content Engines in the ACNS network.

User Authentication and Content Filtering

Content Engines can be configured to perform a number of authentication and content filtering services. Once the Content Engine receives a request, it does a number of things, including the following tasks:

- Authenticates the client, asks the client to provide a username and password so that it can consult an authentication, authorization, and accounting (AAA) server, and checks whether the client is allowed to access the web.
- Passes the request through a “filter,” such as Websense or SmartFilter, to make sure that the requested object is not objectionable content.

ACNS 5.2 software relies on third-party software to implement content filtering. Supported filtering software includes Websense, N2H2, and SmartFilter.

- Passes the request through “rules,” which might rewrite certain headers, redirect the request, or otherwise manipulate the request.
- Checks to see whether the requested content is already in the Content Engine cache. If so, then the Content Engine serves the object directly, rather than from the origin server, thus saving bandwidth to the Internet.
- If the requested content is not already in the cache, then the Content Engine fetches the content from the Internet on the client’s behalf, and caches the content for future use if appropriate.

This functionality is typically called “proxy” functionality. Cisco ACNS 5.2 software supports Content Engine proxy functionality by supporting all web access protocols (including HTTP) and all streaming protocols.



Note

For more information, see the [“Authentication Support” section on page 6-39](#) and the [“Configuring Request Authentication” section on page 13-3](#).

About Device Groups

To facilitate configuring content services for large numbers of Content Engines, ACNS 5.2 software supports device groups.

A *device group* is a set of similar devices (such as Content Engines) that share common qualities and capabilities. Some common qualities might include disk capacity, bandwidth capacity, or routing properties.

By grouping devices into device groups, you can configure settings and channel assignments for the entire group of devices at one time. Settings that can be configured through device groups include Content Engine properties, bandwidth settings, and content services.

**Note**

Content services that require a license key must be enabled on the Content Engine before you can configure service settings for a device group. You cannot enable software licenses on a device group. Licenses must be enabled on an individual Content Engine basis. (See the [“Licensing and Enabling Streaming Servers” section on page 9-1.](#))

Content Engines can be assigned to multiple device groups when the device group overlap feature is enabled in the Content Distribution Manager GUI. (See the [“Enabling Device Group Overlap” section on page 10-28.](#))

The Content Distribution Manager GUI allows you to change settings on individual devices to override the device group settings, and gives you the option of reverting back to the device group settings, if you wish.

To create and modify device groups, see the [“Working with Device Groups” section on page 10-25.](#)

Content Acquisition and Distribution Architecture

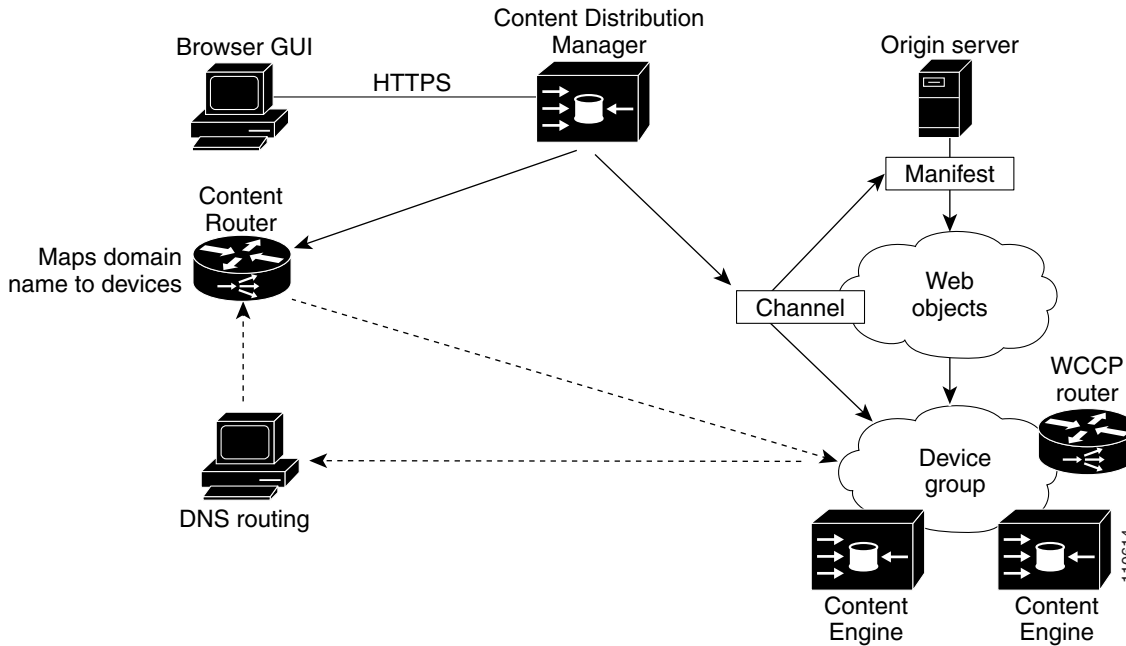
This section discusses the ACNS network content acquisition and distribution architecture. It describes how content arrives at a Content Engine and discusses the role of content providers, websites, channels, and manifests in fetching and distributing content to the various Content Engines in the ACNS network.

Understanding Basic Concepts and Terms

Content is made available for distribution by *content providers*. A content provider is a person, group of persons, or organization that is responsible for providing content for an enterprise organization. Content is usually stored on file servers or web servers that are not part of the ACNS network of managed devices. In this publication, the server that stores the content is called the *origin server*. Content from the content provider is often placed on the origin server by a webmaster, who uses the content to build a *website* or multiple websites.

Content from a single website is mapped to a set of devices or device groups by means of a *channel*. A channel is a configuration used to define how content is to be acquired, distributed, and stored in advance of a client request. Content objects associated with a specific channel have a common domain (host) name; in other words, the content in a specified channel resides in a single location on an origin server. Each channel is restricted to a single website and maps domain names to devices one to one for Content Router DNS interception. (See [Figure 1-2.](#))

Figure 1-2 One-to-One Channel Mapping to Devices



When a Content Engine is assigned to a channel, it is also by definition subscribed to the channel's website. A Content Engine that is subscribed to a website can handle content requests for this website through a Content Router and cache the content from this website. A Content Engine must be assigned to a channel in order to accept and distribute *pre-positioned content* for that channel. Channel assignment is also required for the Content Router to be able to serve content from the website of that channel.

Table 1-1 defines some important ACNS software concepts and terms.

Table 1-1 ACNS Software Concepts and Terms

Term	Definition
Locations	<p>Groupings of well-connected (often physically connected) or interchangeable Content Engines with adjacent IP addresses and usually found in the same proximate geographic location.</p> <p>Locations organize and group Content Engines into virtual networks for distribution of content through channels. Locations are configured in the Content Distribution Manager GUI; each location can have zero to many Content Engines assigned to it. (See the “Creating and Modifying Locations” section on page 5-3.)</p>
Level	<p>Hierarchical arrangement of locations defined by the child-to-parent relationships between locations.</p> <ul style="list-style-type: none"> Each location can have zero to one location as its parent. Each location can have zero to many locations as its children. Locations with no parents are placed at level 1. A location whose parent is at level 1 is placed at level 2, and so on. An ACNS 5.2 network can have up to 4 levels and support up to 200 Content Engines per location and up to 200 child locations per parent location. <p>To configure location levels, see the “Creating and Modifying Locations” section on page 5-3.</p>

Table 1-1 ACNS Software Concepts and Terms (continued)

Term	Definition
Location tree	<p>Set of locations organized in the form of a tree structure. The structure is defined by the location levels.</p> <p>An ACNS network can consist of one or many location trees. (See the “Viewing the Location Tree” section on page 10-8.)</p>
Channels	<p>Groups of content to which Content Engines can be assigned. Content Engines in the same location can be assigned to different channels.</p> <ul style="list-style-type: none"> • When configuring a channel in the Content Distribution Manager GUI, the administrator specifies a list of Content Engines which belong to the channel and which replicate the content of the channel, and a root Content Engine to acquire the content from the origin server and publish it. • For any given channel, there is only one publisher of content (the root Content Engine) and multiple receivers of that content (the Content Engines that are assigned to that channel). The location that contains the root Content Engine for a given channel is called the <i>root location</i>. A channel can have only one configured root Content Engine. Other Content Engines in the root location can act as backup publishers, if the configured root Content Engine fails. <p>To configure channels in the Content Distribution Manager GUI, see the “Creating and Modifying Channels” section on page 5-10.</p>
Root Content Engine	<p>Content Engine designated to download the content from the origin server and forward it to the Content Engines that are assigned to the content channel.</p> <p>You designate a Content Engine to be the root Content Engine for a channel through the Channels tab of the Content Distribution Manager GUI. (See the “Designating the Root Content Engine” section on page 5-14.)</p>
Forwarder	Content Engine that has been chosen as a distribution host by another Content Engine from a another location.
Location Leader	Content Engine that acts as a forwarder to other Content Engines within its own location.
Content provider	Person, group, or organization that is responsible for providing content for an enterprise or organization.
Origin server	Server that stores the source content.
Website	<p>Collection of content objects from a single origin server. Websites can be classified as either routable or nonroutable. Routable websites are controlled and operated by the enterprise corporation where the content is owned. Routable website domain names are fully qualified domain names (FQDNs) that are recognizable to Content Routers. Routable websites support all ACNS software routing and edge intercept mechanisms: WCCP interception, direct proxy routing, and Content Router routing. (See the “Content Request Interception and Router Redirection Overview” section on page 1-5.)</p> <p>Nonroutable websites are not controlled or operated by the enterprise corporation, and the content is not owned by the enterprise. For example, www.cnn.com or www.yahoo.com are nonroutable websites. Nonroutable websites support WCCP interception and proxy configuration only.</p>
Manifest file	XML file consisting of explicit item descriptions and crawler items that is used by the root Content Engine to acquire content from an origin server.
Multicast cloud	Group of multicast-enabled Content Engines configured to communicate multicast session information with one another.

Acquiring Content

Before content is distributed, it is acquired from the origin server by a *root* Content Engine. The root Content Engine is the one Content Engine that is authorized to go directly to the origin server for content. The root Content Engine then publishes the content to other Content Engines in the channel. You assign the root Content Engine by designating it as such in the Content Distribution Manager GUI. In general, the root Content Engine should be in the same general location as the origin server.

The root Content Engine is assigned a *manifest file* that specifies the content for a single channel. The manifest file is an XML file consisting of explicit item descriptions and crawler items (items to be acquired by crawling a website or FTP server) with prefix constraints or match rules. The manifest file is defined by the webmaster or the ACNS network administrator acting as an agent for the webmaster, and controls both content acquisition and content distribution. The manifest file is assigned to one or more root Content Engines (a primary and a list of backups) through the Content Distribution Manager GUI.

Distributing Content Through Channels

Channels form logical routes for content to travel from an origin server through the root Content Engine to all the other Content Engines in the channel. Logical routes for content distribution are based on the device *location* hierarchy, or *location tree*. (See [Table 1-1](#) for an explanation of these terms.)

The content distribution route follows the general tree structure of the manually constructed ACNS network location tree, where content is distributed from the root of the tree (root Content Engine) toward the branches; however, only the Content Engines assigned to a particular channel can be a part of that channel distribution tree. The channel distribution tree is constructed specifically for each channel.

Content is forwarded from Content Engine to Content Engine through the channel distribution tree until all Content Engines in the channel have received their subscribed content.

Content Replication Overview

Content is forwarded (or replicated) either by unicast pull (transmission initiated by a client request for the content) or, if it is enabled, by multicast push (transmission initiated in accordance with a preconfigured program or schedule). Unicast content forwarding involves communication between a single sender and single receiver, whereas multicast replication involves communication between a single sender and a selected group of receivers.

Unicasting

Unicasting is a method of data transmission in which each Content Engine or client receives its own copy of a file or a data stream from a single sender. For example, unicasting is used when only one Content Engine downloads a set of files or a data stream. In another scenario, several clients request the file or the data stream and each receives a new copy at the same time; for example, a proxy Content Engine receives the unicast stream and splits the stream, which is then fanned out to the multiple downstream clients that requested it. An advantage of unicasting is that each client can receive the entire data set at any time. The disadvantage of unicasting is that existing line bandwidths cannot be used optimally. (See the [“Configuring WMT Live Splitting for Unicast and Multicast Transmissions”](#) section on page 9-19 for information on configuring unicast transmissions.)

Multicasting

Multicasting allows efficient distribution of content to multiple Content Engines and is useful when many end users are interested in the same content. ACNS software supports Pragmatic General Multicast (PGM)-based multicast replication, using either satellite or multicast-enabled terrestrial infrastructures. (PGM is a reliable multicast protocol that enables PGM receivers to report loss of data and request retransmission by the PGM sender.)

In ACNS 5.x software, the administrator configures the ACNS network for multicasting by configuring a *multicast cloud* in the Content Distribution Manager GUI. The multicast cloud consists of one sender Content Engine, an optional backup sender (available in ACNS 5.1 software and later releases) for multicast-to-multicast failover, and at least one receiver Content Engine in a hub-spoke topology. All the Content Engines in one cloud share a unique advertising address, allowing them to communicate multicast session information. Content Engines that are assigned to the multicast cloud must be enabled for multicasting.

The administrator enables the multicasting feature in the Content Distribution Manager GUI by entering a license key (purchased from Cisco Systems) for each Content Engine that is to be multicast-enabled (see the [“Enabling Content Engines for Multicasting”](#) section on page 5-31).

The multicast-enabled Content Engines must then be assigned to one or more multicast-enabled channels (see the [“Adding and Removing Content Engines from Channels”](#) section on page 5-15 and the [“Assigning and Removing Multicast Clouds from Channels”](#) section on page 5-41). To implement multicast distribution, a multicast-enabled channel should have at least one multicast enabled cloud associated with it. In addition, the multicast-enabled Content Engines in the cloud must subscribe to the channel. You can enable a channel for multicasting when you first create the channel, or you can modify the channel properties for multicasting (see the [“Creating and Modifying Channels”](#) section on page 5-10).

When configuring the multicast cloud, the administrator specifies a range of addresses by entering a start IP address and an end IP address. Once a multicast cloud is configured, the multicast address range is used to provide each channel associated with it a unique data channel multicast address. When a channel is assigned to a multicast cloud (see the [“Assigning and Removing Multicast Clouds from Channels”](#) section on page 5-41), an unused IP address is automatically selected from this range to ensure that the address is used by only one channel and by only one multicast cloud. Since different multicast clouds may be associated with the same channel, the multicast address used for each channel needs to be different in each multicast cloud.

For more information on IP multicasting, including assigning IP multicast addresses, see [Appendix C, “IP Multicast Addressing.”](#)

ACNS Network Topology Considerations

The ACNS network administrator creates the ACNS network by defining the network topology location and relationship parameters in the Content Distribution Manager GUI. In the GUI, the administrator names the device locations, assigns the parent-child relationship to the locations, creates channels, and assigns Content Engines to channels. Procedures for using the Content Distribution Manager GUI to set up device locations and relationships are covered in [Chapter 5, “Configuring the ACNS Network for Content Distribution.”](#)

Before creating the ACNS network, the administrator should be prepared to answer the following questions:

- What does my network look like?
- Where is my origin web server located?

- Where do I want the content to go?
- How many Content Engines do I need at each location to support fan-out of child locations and end users?
- How many Content Engines do I need for failover and redundancy?