
# Performing Other Basic Tasks for Standalone Content Engines

After you have done a basic configuration on a standalone Content Engine, you can perform other basic tasks such as setting the system clock, managing login accounts, managing and monitoring disks. This chapter describes how to use the Content Engine CLI to perform the following basic tasks for standalone Content Engines:



**Note** For information about how to log in to the Content Engine, see the "Logging in to Standalone Content Engines" section on page 4-50.

# Showing Inventory

Cisco Content Engines are embedded with following three identification items:

- Product ID (PID)
- Version ID (VID)
- Serial number (SN)

This identity information is stored in non-volatile memory. Each Content Engine has a unique device identifier (UDI). The UDI = PID + VID + SN.

The UDI is electronically accessed by the product operating system or network management application to enable identification of unique hardware devices. Consequently, the data integrity of the UDI is vital to customers. This means that the UDI that is programmed into the Content Engine's non-volatile memory is equivalent to the UDI that is printed on the product label and on the carton label. This UDI is also equivalent to the UDI that can be viewed through any electronic means, and in all customer-facing systems and tools.

In the ACNS 5.2.1 software and later releases, enter the **show inventory** EXEC command to view the Content Engine's UDI. Currently, there is only CLI access to the UDI; there is no SNMP access to the UDI information.

All of the Content Engine models that are supported in the ACNS 5.2.1 software and later releases support this show inventory feature.

On newer Content Engine models, you can display the Content Engine's UDI by using a single command, the **show inventory** EXEC command.

```
CE-565# show inventory
PID: CE-565-K9 VID: 0 SN: serial number
```

*serial number* is the serial number of the Content Engine. If the version number is available, then it is displayed; otherwise, a zero (0) is displayed (as shown in the example).

On older Content Engine models (for example, the CE-507 or CE-2636), you must use the **show tech-support** and **show inventory** EXEC commands to display the Content Engine's UDI.

```
CE-507# show inventory
Please look at 'sh tech-support' for information!
CE-507# show tech-support
```

**Note** See the *Release Notes for Cisco ACNS Software, Release 5.5* for a list of hardware platforms that are supported in the ACNS 5.5 software release.

# Managing Administrative Login Accounts

A Content Engine that is running the ACNS software comes with a single predefined superuser login account (root administrator). This predefined administrative login account can be used to access the Content Engine GUI initially in order to perform a basic configuration on a standalone Content Engine and then add other login accounts.

**Note** The username for this predefined superuser account is admin and the default password is default. If these defaults have been changed by another ACNS system administrator, you need to obtain the new username and password.

You must assign a privilege profile to each new administrative login account that you create on your standalone Content Engine. Privilege profiles determine which tasks ACNS software administrators can perform, and the level of access granted to them based on the administrative login account that they used to log in to the Content Engine.

The following are the two types of predefined privilege profiles:

- Normal-level administrator—Privilege level of zero (0). Has read access, and can see some of the Content Engine configuration settings.
- Superuser administrator—Privilege level of 15. Has administrative privileges such as running the Setup utility, creating new administrative login accounts, and modifying any of the Content Engine configuration settings.

You can use the Content Engine GUI or CLI to change the password for this predefined superuser account or to create additional login accounts for other ACNS system administrators. After you have used the predefined superuser login account to perform a basic configuration on a standalone Content Engine, it is recommended that you change the password for this superuser login account. For information about adding or modifying an administrative login account, see the next section, "Setting the System Clock."

The ACNS 5.x software also supports various login authentication methods (local, RADIUS, or TACACS+ authentication). This enables you to configure a standalone Content Engine to use one or more of these authentication methods when it processes an administrative login request. For more information, see Chapter 17, "Configuring Administrative Login Authentication and Authorization on Standalone Content Engines."

**Note** Content authentication and authorization, which controls end users' access to the requested content that is served through a standalone Content Engine, is independent of the administrative login authentication and authorization for the Content Engine that controls the level of access for users who log in to the Content Engine for configuration, monitoring, or troubleshooting purposes. For information about content authentication and authorization, see Chapter 10, "Configuring Content Authentication and Authorization on Standalone Content Engines."

# Setting the System Clock

If you have an outside source on your network that provides time services (such as a Network Time Protocol [NTP] server), you do not need to set the system clock manually. When setting the clock, enter the local time. The Content Engine calculates Coordinated Universal Time (UTC) based on the time zone specified in the **clock timezone** global configuration command.

Two clocks exist in the system: the software clock and the hardware clock. The software uses the software clock. The hardware clock is used only at bootup to initialize the software clock.

To set or clear clock functions or update the calendar, use the **clock** EXEC command.

**clock** {**read-calendar** | **set** *time day month year* | **update-calendar**}

where:

| | |
|---|---|
| **read-calendar** | Reads the calendar and updates the system clock. |
| **set** | Sets the time and date of the software clock. |
| *time* | Current time in hh:mm:ss format (hh: 00–23; mm: 00–59; ss: 00–59). |
| *day* | Day of the month (1–31). |
| *month* | Month of the year (January, February, March, April, May, June, July, August, September, October, November, December). |
| *year* | Year (1993–2035). |
| **update-calendar** | Updates the calendar with the system clock. |

For example:

```
ContentEngine# clock set 13:32:00 01 February 2000
```

To set the daylight saving time and time zone for display purposes, use the **clock** global configuration command. To set and display the local and UTC current time of day without an NTP server, use the **clock timezone** global configuration command with the **clock set** EXEC command. The **clock timezone** parameter specifies the difference between UTC and local time, which is set with the **clock set** EXEC command. The UTC and local time are displayed with the **show clock detail** EXEC command.

An accurate clock and timezone setting is required for the correct operation of HTTP proxy caches.

# Displaying the Standard Time Zones

The ACNS system has several predefined standard time zones. Some of these time zones have built-in daylight saving time information while others do not. For example, if you are in an eastern region of the United States (US), you must use US/Eastern time zone, which includes daylight saving time information and will adjust the clock automatically every April and October. The system includes about 1500 standard time zone names.

In the ACNS 5.2.x software and earlier releases, there was no restriction on these reserved standard time zone names. You could overload these standard names in various ways. For example, you could use the US/Pacific time zone but entering the **clock summertime** EXEC command to define a different daylight saving time schedule. In the ACNS 5.3.1 software and later releases, strict checking is supported. The **clock summertime** command is now disabled when a standard time zone is configured. You can only configure daylight saving time if the time zone is a customized zone (that is, not a standard time zone).

The **show clock standard-timezones all** EXEC command allows you to browse through all standard timezones and choose from these predefined time zones.

You can choose a customized name that does not conflict with the predefined names of the standard time zones. Most predefined names of the standard time zones have two components, a region name and a zone name. You can list time zones by several criteria, such as regions and zones.

**show clock standard-timezones** {**all** | **regions** | **zones** *region name* | **details** *complete name* }

The following example shows a portion of the output from the **show clock standard-timezones all** EXEC command. As the following example shows, all of the standard time zones (approximately 1500 time zones) are listed. Each time zone is listed on a separate line.

```
ContentEngine # show clock standard-timezones all
Africa/Abidjan
Africa/Accra
Africa/Addis_Ababa
Africa/Algiers
Africa/Asmera
```

The following example shows a portion of the output from the **show clock standard-timezones region** EXEC command. As the example shows, all first level time zone names or directories are listed. All 1500 time zones are organized into directories by region.

```
ContentEngine # show clock standard-timezones regions
Africa/
America/
Antarctica/
Arctic/
Asia/
Atlantic/
Australia/
Brazil/
CET
.
.
.
US/
UTC
Universal
W-SU
WET
Zulu
```

The following example shows a portion of the output from the **show clock standard-timezones zones** EXEC command. As the following example shows, this command lists the name of every time zone that is within the specified region (for example, the US region).

```
ContentEngine # show clock standard-timezones zones US
Alaska
Aleutian
Arizona
Central
East-Indiana
Eastern
Hawaii
Indiana-Starke
Michigan
Mountain
Pacific
Samoa
```

The following sample shows a portion of the output from the **show clock standard-timezones details** EXEC command. As the following example shows, this command shows details about the specified time zone (for example, the US/Eastern time zone), The command output also includes the standard offset from the Greenwich mean time (GMT).

```
ContentEngine # show clock standard-timezones details US/Eastern
US/Eastern is standard timezone.
Getting offset information (may take a while) ...
Standard offset from GMT is -300 minutes (-5 hour(s)).
It has built-in summertime.
Summer offset from GMT is -240 minutes. (-4 hour(s)).
```

# Configuring Banners for Standalone Content Engines

In the ACNS 5.3.1 software and later releases, you can configure the following three types of banners in any ACNS software device mode:

- motd banner—Sets the message of the day. This message is the first message that is displayed when a login is attempted.
- login banner—Displayed after the motd banner but before the actual login prompt appears.
- exec banner—Displayed after the EXEC CLI shell has started.

**Note** All three of these banners are effective on a console, Telnet, or a Secure Shell (SSH) Version 2 session. When you run an SSH Version 1 client and log in to the Content Engine, the motd and login banners are not displayed.

The **banner** global configuration commands allow you to configure the different types of banners for standalone Content Engines.

**banner** {**motd** | **login** | **exec**} {**message** *text message* | **<cr>**}

The following example shows how to use the **banner motd message** global configuration command to configure the motd banner. In this example, the motd message consists of a single line of text.

```
ContentEngine(config)# banner motd message This is an ACNS 5.4 device
```

The following example shows how to use the **banner motd message** global command to configure a MOTD message that is longer than a single line. In this case, the Content Engine translates the "\n" portion of the message to a new line when the MOTD message is displayed to the user.

```
ContentEngine(config)# banner motd message "This is an ACNS device. \nAccess is
restricted.\n"
```

After you configure the banners, enter the **banner enable** global configuration command to enable banner support on the Content Engine. Enter the **show banner** EXEC command to display information about the configured banners.

This example shows how to configure and enable banner support on a standalone Content Engine (Content Engine A). In this example, the user logs in to Content Engine A through an SSH session.

**Step 1** Administrator A uses the **banner message** commands to configure the motd, login, and exec banners on Content Engine A, as follows:

**a.** Configure the motd message. In this example, the motd message is longer than a single line. In this case, Content Engine A translates the "\n" portion of the message to a new line in the motd message that is displayed to the user (in this case, Administrator B).

```
ContentEngine (config)# banner motd message "This is the motd message.
\nThis is an ACNS 5.4 device\n"
```

**b.** Configure a login message. This example shows how to configure a motd message that is longer than a single line. In this case, Content Engine A translates the "\n" portion of the message to a new line in the login message that is displayed to the user (in this case, Administrator B).

```
ContentEngine(config)# banner login message "This is login banner.
\nUse your password to login\n"
```

**c.** Configure an interactive banner. The **banner exec** command is almost identical to the **banner motd message** command except that with the **banner exec** command the banner content is obtained from the command line input that the user enters after being prompted for the input.

```
ContentEngine (config)# banner exec message "This is the EXEC banner.
\nUse your ACNS username and password to log in to this Content Engine.\n"
```

**Step 2** Administrator A enables banner support on Content Engine A.

```
ContentEngine (config)# banner enable
```

**Step 3** Administrator A enters the **show banner** EXEC commands to display information about the configured banners. For example, the **show banner motd** EXEC command is used to display information about the configured message of the day (motd) banner.

**Step 4** Another ACNS administrator (Administrator B) uses an SSH session to log in to Content Engine A. For example:

```
% ssh admin@ce
```

After Administrator B logs in to the SSH session, Administrator B will see a login session that includes a motd banner as well as a login banner that asks Administrator B to enter a login password. For example:

```
This is the motd banner.
This is an ACNS 5.4 device
This is login banner.
Use your password to login.

Cisco Content Engine

admin@ce's password:
```

After Administrator B enters a valid login password, the EXEC banner is displayed, and Administrator B is asked to enter the ACNS username and password. For example:

```
Last login: Fri Oct  1 14:54:03 2004 from client
System Initialization Finished.
This is the EXEC banner.
Use your ACNS username and password to log in to this Content Engine.
```

After Administrator B enters a valid ACNS username and password, the Content Engine CLI is displayed. The CLI prompt varies depending on the privilege level of the login account. In this example, because Administrator B entered a username and password that had administrative privileges (privilege level of 15), the EXEC mode CLI prompt is displayed.

```
ContentEngine#
```

# Adding or Modifying Administrative Login Accounts

After you have used the predefined superuser login account to perform a basic configuration on a standalone Content Engine, we recommend that you change the password for this superuser login account.

From the Content Engine GUI, choose **System > Users**. Use the displayed Users window to modify a password for this predefined superuser login account. You can also use the Users window to create additional administrative login accounts (normal administrative user or superuser accounts). For information about using this window, click the **HELP** button to access context-sensitive help. For more information about accessing the Content Engine GUI, see the "Logging in to the Content Engine GUI" section on page 4-55.

To use the Content Engine CLI to add or modify an ACNS system administrative login account on a standalone Content Engine, follow these steps:

**Step 1** Access the Content Engine CLI in global configuration mode.

**Step 2** Configure the entries for the group name-based access list.

```
ContentEngine(config)# username name {cifs-password {0 plainword |1 lancrypto ntcrypto}|
password {0 plainword | 1 cryptoword} uid uid} | privilege {0|15}}
```

Table 5-1 describes the parameters for the **username** global configuration command.

***Table 5-1 Parameters for the username CLI Command***

| Parameter | Description |
|---|---|
| **username** | Sets the username for the administrative login account. |
| *name* | Username for the administrative login account. |
| **cifs-password** | Sets the Windows file-sharing user password. |
| **0** | Specifies that an unencrypted Windows file sharing password will follow. |
| *plainword* | Clear-text user Windows file sharing password. |
| **1** | Specifies that a hidden Windows file sharing password will follow. |
| *lancrypto* | Encrypted password for LAN Manager networks. |
| *ntcrypto* | Encrypted password for Windows NT networks. |
| **password** | Sets the user password for the administrative login account. |
| **0** | Specifies that an unencrypted user password will follow. |
| **1** | Specifies that a hidden user password will follow. |
| *cryptoword* | Encrypted user password. |
| **uid** | Sets the user ID for the password. |

*Table 5-1 Parameters for the username CLI Command (continued)*

| | |
|---|---|
| *uid* | Text password user ID (2001–65535). |
| **uid** | Sets the user ID for an encrypted password. |
| *uid* | Encrypted password user ID (2001–65535). |
| **privilege** | Sets the user privilege level of an ACNS network administrative login account. |
| **0** | User privilege level for a normal-level administrator (the ACNS system administrative users who are not superusers). |
| **15** | User privilege level for the ACNS system administrators who have superuser access. |

This example shows how you can use the **username** EXEC command to modify passwords and privilege levels for administrative login accounts on a standalone Content Engine:

```
ContentEngine# show user username jrdoe
Uid                 : 2003
Username            : jrdoe
Password            : ghQ.GyGhP96K6
Privilege           : normal user

ContentEngine(config)# username jrdoe privilege 15
User's privilege changed to super user (=15)

ContentEngine# show user username jrdoe
Uid                 : 2003
Username            : jrdoe
Password            : ghQ.GyGhP96K6
Privilege           : super user
```

# Configuring Disk Space

Disk space in ACNS software is allocated on a per-file system basis, rather than on a per-disk basis. You can configure your overall disk storage allocations according to the kinds of client protocols you expect to use and the amount of storage that you need to provide for each of the functions, as described in Table 5-2. Use the **disk add** EXEC command to add a single disk with the specified partitions.

*Table 5-2 Cisco ACNS Software Disk Storage for Standalone Content Engines*

| Disk Storage Type | Function |
|---|---|
| sysfs (system file system) | Stores log files, including transaction logs, syslogs, and internal debugging logs. Also can store image files and configuration files. For more information about sysfs, see the next section, "Creating Disk Space for the Sysfs." |
| cfs (cache file system) | Caches HTTP and FTP objects. |
| mediafs (media file system) | Caches content from streaming proxy servers, such as WMT and RealProxy. |

Enter the **show disks** EXEC command to view information about the current disk configuration of a standalone Content Engines.

```
ContentEngine# show disks
Local disks:
  SYSFS                29.9GB      96.8%
  CFS                   0.0GB       0.0%
  MEDIAFS               0.0GB       0.0% (from-unused-cdnfs)
  CDNFS                 1.0GB       3.1%
  FREE                  0.0GB       0.0%

Note: CDNFS and MEDIAFS amounts are reported in terms of actual usable amounts
of storage for applications.  Due to internal filesystem overhead of
approximately 3%, the reported amounts may be smaller than what you configured.
CDNFS space is allocated with higher priority than MEDIAFS, so if you
configured MEDIAFS and CDNFS, then MEDIAFS will be reduced by the amount
of the total CDNFS and MEDIAFS overhead.  If you have not configured MEDIAFS,
then CDNFS will be reduced by the amount of the overhead.

Network-attached disks:
  NONE
ContentEngine#
```

**Note** Standalone Content Engines do not have a CDNFS partition because this partition is used to store pre-positioned content, which a standalone Content Engine does not support.

Use the **disk config sysfs** EXEC command to configure disk resources for standalone Content Engine.

In the ACNS 5.2.x software and earlier releases, the CDNFS, MEDIAFS, and SYSFS partitions use the ext2 file system. With ext2 file systems, if the system crashed or if the system is not shut down properly a file system check of these partitions takes a long time. If there are sector failures on the disk, the time to perform a file system check with an ext2 file system increases even more. In the ACNS 5.3.1 software and later releases, the ext3 file system is used instead of the ext2 file system. By migrating to the ext3 file system, the amount of time required to perform a file system check of the CDNFS, MEDIAFS, and SYSFS partitions is decreased, which increases the availability of the Content Engine. If you are upgrading from an earlier release of the ACNS software, the ext2 file system is automatically converted to the ext3 file system when you upgrade to the ACNS 5.3.1 software and later releases.

In the ACNS 5.2.1 software and later releases, the ability to monitor Content Engine disk drives is supported. Disk status is now recorded in flash (non-volatile storage). When an error on a Content Engine disk device occurs, a message is written to the system log (syslog) if the sysfs partition is still intact, and an SNMP trap is generated if SNMP is configured on the Content Engine.

In addition to tracking the state of critical disk drives, you can define a disk device error-handling threshold on the Content Engine. If the number of disk device errors reaches the specified threshold, the corresponding disk device is automatically marked as "bad." The ACNS system does not stop using the bad disk device immediately; it stops using the bad disk drive after the next reboot.

If the specified threshold is exceeded, the Content Engine either records this event or reboots. If the automatic reload feature is enabled and this threshold is exceeded, then the ACNS system automatically reboots the Content Engine. For more information about specifying this threshold, see the "Specifying the Disk Error-Handling Threshold" section on page 21-18. For more information about monitoring critical disks, see the "Monitoring Critical Disk Drives on Standalone Content Engines" section on page 21-17.

In the ACNS 5.3.1 software and later releases, the ability to monitor proactively the health of disks with Self Monitoring, Analysis, and Reporting Technology (SMART) is supported. SMART provides you with hard drive diagnostic information and information about impending disk failures. For more information, see the "Proactively Monitoring Disk Health with SMART" section on page 21-20.

## Creating Disk Space for the Sysfs

If you are initially configuring a standalone Content Engine, you must create disk space for the system file system (sysfs) by using the **disk config sysfs** EXEC command.

To configure disk space on a standalone Content Engine, follow these steps:

**Step 1** Exit configuration mode, if you have not already done so.

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

**Step 2** Configure the disk space for the sysfs. For example, to configure the sysfs for 5 GB, enter this command:

```
ContentEngine# disk config sysfs 5GB
```

**Step 3** Reload the Content Engine for the disk configuration to take effect.

```
ContentEngine# reload
```

**Tip** For the new disk space configuration to take effect, you must first reboot the software. If you encounter an error message, reenter your disk configuration and use the **reload** EXEC command on the Content Engine for the disk configuration to be applied.

# Removing All Disk Partitions on a Single Disk Drive

Use the **disk delete-partitions** EXEC command to remove all disk partitions on a single disk drive.

**Caution** The disk delete-partitions EXEC command will erase everything on the specified disk.

Typically, this command is used when you want to add a new disk drive that was previously used with another operating systems (for example, a Microsoft Windows or Linux operating system). When asked if you want erase everything on the disk, specify **yes** to proceed:

```
ContentEngine# disk delete-partitions disk03
This will erase everything on disk. Are you sure? [no] yes
```

# Stopping Applications from Using a Disk Drive

In the ACNS 5.3.1 software and later releases, the **disk unuse** EXEC command allows you to stop applications from using a specific disk drive (for example, disk01) without having to reboot the drive:

```
ContentEngine# disk unuse disk01
```

The disk unuse feature cannot be used with disk00 (the first disk drive) or with the drive that contains the /local/local1 directory (for example, if disk01 contains the /local/local1 directory then you cannot use the disk unuse command with disk01). For more information, see the *Cisco ACNS Software Upgrade and Maintenance Guide, Release 5.x.*

# Displaying the Current Disk Configuration

Use the **show disks** EXEC commands to display the current disk configuration of a standalone Content Engine.

```
ContentEngine# show disks ?
 configured        Show configured configurations (effective after reboot)
 current           Show current configurations
 details           Show current configurations with more details
 failed-sectors    Show the list of failed sectors on the disks
 network-attached  Show Network Attached Storage on this device
 SMART-info        Show hard drive diagnostic and failure prediction info
                   (provided by SMART disk monitor)
 storage-array     Show Disk Information on storage array (if any)
```

For example, enter the **show disks details** EXEC command to display detailed information about a Content Engine's current disk configuration.

```
ContentEngine# show disks details
disk00: Normal          (IDE disk)                 38160MB( 37.3GB)
        disk00/04: PHYS-FS       15137MB( 14.8GB) mounted internally
        disk00/04: MEDIAFS         532MB(  0.5GB) mounted internally
        disk00/05: SYSFS          1023MB(  1.0GB) mounted at /local1
        disk00/06: CFS           15359MB( 15.0GB)
        System use:               6130MB(  6.0GB)
        FREE:                       16MB(  0.0GB)
disk01: Not present
No NAS share is attached to this device.
```

Disk drives that are currently marked as "bad" are shown as "Not used" in the output of the **show disks details** EXEC command. Future "bad" disk drives (drives that will not be used after the next time the Content Engine is reloaded) are shown with an asterisk (*). In the following case, disk03 is a future bad disk drive that will not be used after the Content Engine is reloaded.

```
ContentEngine# show disks details
(*) Disk drive won't be used after reload.
......
disk03: Normal        (h00 c00 i03 l00 - Int DAS)        70001MB( 68.4GB) (*)
    FREE:                   70001MB( 68.4GB)
......
```

For information about disk monitoring, see the "Monitoring Critical Disk Drives on Standalone Content Engines" section on page 21-17.

# Mounting to a Network Attached Storage Device

The ACNS 5.x software provides a Common Internet File System (CIFS) client and a Network File System (NFS) client for Content Engines to communicate with network-attached storage (NAS) devices. Content Engines can be attached to NAS devices to increase their storage space. These Content Engines function as NFS or CIFS clients while accessing the NAS servers. NAS servers include UNIX-mode NFS servers or Microsoft Windows systems for CIFS sharing.

NAS servers support the cdnfs and mediafs for Content Engines. You can choose the type of file system to be attached to the NAS depending on whether you need to store cached WMT, RealMedia, or other streaming content.

NFS and CIFS servers export either an entire file system to a Content Engine or a specified directory on a file system. In both cases, you need to specify the amount of disk space to be assigned to the Content Engine. Different Content Engines attach different directories on an NFS or CIFS server, and it is not possible to share the same directory among multiple Content Engines. NFS servers support host-based authentication and UNIX file system access control. You need to specify the client IP address that matches the list of hosts that an NFS server trusts. The client is then allowed to mount and access files based on the permissions assigned to it. On the other hand, CIFS servers share files and authenticate users on the server itself, instead of exporting data to clients for authentication. CIFS servers support NTLM, plain text password, and LDAP authentication.

Mounting NAS shares to a Content Engine can be performed in these ways:

- Through the CLI for standalone Content Engines
- Through the Content Distribution Manager GUI or the CLI for centrally managed Content Engines

In the ACNS 5.3.1 software and later releases, you can configure a CIFS share name by using the **share-name** option of the **network-filesystem server cifs share-web-site** global configuration command. For more information about how to mount a Content Engine to a NAS device and configure a CIFS share name through the CLI, see the *Cisco ACNS Software Upgrade and Maintenance Guide, Release 5.x.*

# Saving the Current Configuration on Standalone Content Engines

A standalone Content Engine has two types of system configuration:

- Startup system configuration that is stored in nonvolatile memory
- Running system configuration

To use the Content Engine CLI to save the current running configuration as the startup configuration, use the **copy running-config** global configuration command. The running system configuration can be saved to the sysfs partition, flash memory, or TFTP server. For example, enter the following to save the running configuration to flash memory:

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

**Note** The **copy running-config startup-config** command is equivalent to the **write memory** command.

To save the current configuration during a Setup utility session, enter **y** when asked if you want to save the current configuration.

# Disabling Transparent Caching Services on Standalone Content Engines

To disable transparent caching on a standalone Content Engine in a WCCP environment without powering down the Content Engine, disable the running version of WCCP on the Content Engine by entering the **no wccp version** global configuration command (for example, use the **no wccp version 2** command to disable WCCP Version 2). The Content Engine will still service proxy-style requests, if so configured, and preserve its configuration settings.

# Creating Custom Message Pages for Standalone Content Engines

In the ACNS 5.4.1 software and later releases, you can create the following types of customized error pages on a standalone Content Engine:

- HTTP custom error message pages, as described in the next section, "Creating HTTP Custom Error Pages for Standalone Content Engines"
- FTP native custom message pages, as described in the "Creating Custom Messages for FTP Proxy Responses for FTP Native Requests" section on page 5-19.

## Creating HTTP Custom Error Pages for Standalone Content Engines

In the ACNS 5.1 software and later releases, you can create HTTP customized error pages. If you create these customized pages, then the standalone Content Engine displays the appropriate customized error page instead of the default error message when proxy errors occur for HTTP (HTTP, HTTPS, and FTP-over-HTTP) requests.

Table 5-3 describes the custom HTTP error messages and their usage for standalone Content Engines running the ACNS 5.3.1 software and later releases.

***Table 5-3 Custom Error Page Messages for Standalone Content Engines***

| Message Identifier | Usage |
|---|---|
| blocked-dueto-filter-error | Error response when a request is blocked because of a filter. |
| cache-read-error | Error response when a cache files system (cfs) read fails. |
| cache-write-error | Error response when a cfs write fails. |
| client-access-denied-msg | Error response when a client access is denied. |
| client-connection-broken-error | Error response when a client connection is lost. |
| dns-not-available-error | Error response when DNS is unavailable for resolution. |
| error-signature | Signature message that is appended to the final error page that is displayed to the end user (added in the ACNS 5.3.1 software release). If you do not create a customized error-signature message, then the default error-signature message is used. For more information about creating a customized error-signature message, see the "Creating a Custom Error-Signature Message for HTTP Custom Error Pages" section on page 5-17. |

*Table 5-3 Custom Error Page Messages for Standalone Content Engines (continued)*

| Message Identifier | Usage |
|---|---|
| expect-failed-error | Error response when Expect specifier in the HTTP request header cannot be met. |
| ftp-bad-login-error | Error response when the FTP login fails. |
| ftp-bad-url-error | Error response when the FTP request receives a bad URL. |
| ftp-disabled-error | Error response when the FTP is disabled. |
| ftp-failure-error | Error response when FTP fails. |
| ftp-internal-error | Error response when the FTP interval is exceeded. |
| ftp-not-found-error | Error response when the FTP file not found. |
| ftp-put-created-msg | Error response when the FTP PUT is successful. |
| ftp-put-error | Error response when the FTP PUT fails. |
| ftp-put-modified-msg | Response when the FTP update is successful. |
| ftp-unavailable-msg | Error response when the FTP file is unavailable. |
| http-blocked-port-msg | Error response when an HTTP request comes through a blocked port. |
| https-blocked-port-msg | Error response when an HTTPS request comes through a blocked port. |
| icap-processing-error | Error response when an error occurred in ICAP processing. |
| invalid-port-error | Error response when an invalid port is accessed. |
| looped-req-error | Error response when a looped request is unsuccessful. |
| not-enough-resources-error | Error response when enough resources are not available for the request process. |
| not-in-cache | Error response when the object is not found in the cache. |
| offline-miss-error | Error response when an off-line Content Engine finds a cache miss. |
| outgoing-proxy-fail-error | Error response when all outgoing proxy fails. |
| proxy-allow-domain-error | Error response when the domain is not allowed to authenticate in proxy mode (added in the ACNS 5.3.1 software release). |
| proxy-no-default-domain-error | Error response when there is no default domain available to authenticate in proxy mode (added in the ACNS 5.3.1 software release). |
| proxy-unauthenticated-error | Error response when the proxy authentication fails. |
| radius-redirect-error | Error response for a RADIUS redirect message. |
| request-blocked-msg | Error response when the request is blocked. |
| request-malformed-error | Error response when the request headers are malformed. |
| rev-dns-not-available-msg | Error response when DNS is not available. |
| server-connection-broken-error | Error response when the server connection is lost. |
| www-allow-domain-error | Error response when the domain is not allowed to authenticate (added in the ACNS 5.3.1 software release). |

*Table 5-3 Custom Error Page Messages for Standalone Content Engines (continued)*

| Message Identifier | Usage |
|---|---|
| www-no-default-domain-error | Error response when there is no default domain available to authenticate (added in the ACNS 5.3.1 software release). |
| www-unauthenticated-error | Error response when the server authentication fails |

You can use the Content Engine GUI or the CLI to create HTTP customized error pages for any of the error messages listed in Table 5-3.

- From the Content Engine GUI, choose **Caching > Customized Error Page**. Use the displayed Custom Error Page Configuration window. For more information about how to use the Custom Error Page Configuration window, click the **HELP** button in the window.
- To use the Content Engine CLI to configure HTTP custom error pages, use the **http custom-error-page** EXEC command.

  **http custom-error-page download** *message url* | **reset** {**all** | *message*} | **upload** {*ip-address* | *hostname*} *dirname filename message*

Table 5-4 describes the parameters for the **http custom-error-page** command.

*Table 5-4 Parameters for the http custom-error-page CLI Command*

| Parameter | Description |
|---|---|
| **download** | Copies the custom error message file to the Content Engine from the specified URL. To change the text for a specific message, use this option to identify the message you want to change, and specify the URL that is the source for the custom message file. The custom message file can be up to 16 KB in size and is used instead of the standard message page for the specified message. |
| *url* | Specifies the source of the custom error file. The file size cannot exceed 16 KB. |
| *message* | Specifies the type of custom error message (for example, ftp-put-error). See Table 5-3 for a list of these custom error messages. |
| **reset** | Reverts to the default error page. |
| *message* | Specifies the message that you want to return to the default page. |
| **upload** | Uploads the custom error message file to the specified host, directory, and file. |
| *ip-address* | Specifies the IP address of the host to which to copy the error page. |
| *hostname* | Specifies the hostname to which to copy the error page. The host should be reachable and allow copying a file to the specified directory. |
| *dirname* | Specifies the directory name to which to copy the error page. |
| *filename* | Specifies the filename to which to copy the error page. |

To display a list of the custom error messages:

```
ContentEngine# http custom-error-page download ?
```

To display a list of all the configured custom error pages:

```
ContentEngine# show http custom-error-page configured
```

To display the contents of the specified custom error page:

```
ContentEngine# show http custom-error-page custom-error page-filename
```

To copy a custom error message page to the Content Engine for the cache-read-error message:

```
ContentEngine# http custom-error-page download
http://www.myserver.com/errors/cache-read-error.txt cache-read-error
```

To copy the current contents of the cache-read-error message to a file in the errors directory on the host with the IP address 192.168.1.1:

```
ContentEngine# http custom-error-page upload 192.168.1.1 /errors
cache-read-error.txt cache-read-error
```

When the authentication method is NTLM, the Content Engine sends the proxy-allow-domain-error/www-allow-domain-error message to the end user if the user-supplied domain does not match the configured domains. You can customize the error that is sent to the end user as follows:

```
ContentEngine# http custom-error-page download proxy-allow-domain-error ?
WORD  URL from where the file will be fetched, File size cannot exceed 16K
```

With certain browsers (for example, with the Netscape browser) the end user must specify the domain if the Content Engine does not have an NTLM default domain configured. If there is no default domain available for authentication, the proxy-no-default-domain-error/www-no-default-domain-error message is sent to the end user. You can customize the error message that is sent to the end user as follows:

```
ContentEngine# http custom-error-page download proxy-no-default-domain-error ?
WORD  URL from where the file will be fetched, File size cannot exceed 16K

ContentEngine# http custom-error-page download www-no-default-domain-error ?
WORD  URL from where the file will be fetched, File size cannot exceed 16K
```

To reset the cache-read-error message to the default text:

```
ContentEngine# http custom-error-page reset cache-read-error
```

To reset all of the configured custom error pages to the default text:

```
ContentEngine# http custom-error-page reset all
```

# Creating a Custom Error-Signature Message for HTTP Custom Error Pages

In the ACNS 5.3.1 software and later releases, you can include an error signature on the HTTP custom error pages. To support this feature, the **error-signature** message identifier was added to the list of custom error messages (listed in Table 5-3).

The following is the default error signature that will be appended to the final error message page sent to the end user:

```
<br clear="all">
<hr noshade size=1>
Generated %t by
(<a href="http://www.cisco.com/">Application and Content Networking System Software
5.3.0</a>)
</BODY></HTML>
```

If desired, you can create a custom error signature that will be appended to the final error message page instead of the default error signature. The following example shows an example of a custom error signature:

```
The request %R , from the Client %i does not conform to the HTTP Request.
Please refer RFC-XXX, for further information on request format.
</body>
</html>
```

After creating a custom error signature, you use the **http custom-error-page download error-signature** EXEC command to copy the file that contains the custom error signature to the Content Engine. For example:

```
ContentEngine# http custom-error-page download error-signature ftp://somewhere/sig.htm
```

A custom error signature message supports dynamic data. This dynamic data is specific to the particular request and response that is to be shown with the error messages.

Table 5-5 lists the argument specifiers for the error signature. These argument specifiers are replaced with the corresponding value when the error signature is appended to the final error message page that is sent to the end user.

***Table 5-5 Argument Specifiers for Error Signatures***

| Argument Specifier | Description |
|---|---|
| c | Squid error code |
| d | Seconds elapsed since the request was received |
| e | Error number |
| E | Error string |
| h | Cache hostname (hostname of the Content Engine) |
| H | Server hostname |
| i | Client IP address |
| I | Server IP address |
| M | Request method |
| p | URL port # (number) |
| P | Protocol |
| R | Full HTTP request |
| t | Local time |
| T | Coordinated Universal Time (UTC) |
| U | URL without password |

The following example shows how the template will be transformed into the final error message page that is sent to the end user. The template contains the following content for a request malformed error:

```
<html>
<body>
<h1> Malformed request </h1>
The request %R is not based on the Standard
</body>
</html>
```

You create a custom error signature that uses the “R” and “i” argument specifiers:

```
<html>
<body>
<h1> Malformed request </h1>
The request %R , from the Client %i does not conform to the HTTP Request.
Please refer RFC-XXX, for further information on request format.
</body>
</html>
```

When a request malformed error occurs for the client with the IP address of 192.168.192.161, the following custom error signature is appended to the final error message page that is sent to the end user (client 192.168.192.161):

```
<html>
<body>
<h1> Malformed request </h1>
The request  GET http://www.abccorp.com HTT/P10, from the client 192.168.192.161
 does not conform to the HTTP Request.
Please refer RFC-XXX, for further information on request format.
</body>
</html>
```

Because the “R” and “i” argument specifiers were specified as part of the custom error signature, the actual full HTTP request (http://www.abccorp.com HTT/P10) and the IP address of the client (192.168.192.161) are included in the error signature of the final error message page that is sent to the end user.

# Creating Custom Messages for FTP Proxy Responses for FTP Native Requests

In the ACNS 5.4.1 software and later releases, you can use the **ftp-native custom-message** EXEC command to create, upload, and download files that contain the following custom messages:

- A custom FTP proxy-mode welcome message. The message is a custom welcome message that the Content Engine displays to the FTP client to welcome the FTP proxy-mode connection from the client. FTP clients include Reflection X clients, WS-FTP clients, or UNIX or DOS command line FTP programs, which are requesting a proxy-mode connection with the Content Engine in order to send nontransparent native FTP requests directly to the Content Engine.
- A custom error message used for when an FTP client is denied access based on the IP access control lists (ACLs) that have been defined for the native FTP proxy service.

If you intend to enable proxy authentication, then the welcome message should inform the user to authenticate with the proxy before logging in to the origin server, as shown in the following example:

```
Welcome to ce-Boxman.  BigCorp’s Content Engine for Native FTP Proxy. Please login to the
proxy with your username and password.
```

Alternatively, if you do not intend to enable proxy authentication, then the welcome message should inform the user to log in to the origin server using either the USER or SITE method, as shown in the following examples:

```
Welcome to ce-Boxman. BigCorp’s Content Engine for Native FTP Proxy. Please login to the
origin server using the 'username@server-hostname' format.

Welcome to ce-Boxman. BigCorp’s Content Engine for Native FTP Proxy. Please login to the
origin server using the 'SITE server-hostname' command followed by the 'USER username'
command.
```

To use the Content Engine CLI to configure FTP custom message pages, use the **ftp-native custom-message** EXEC command.

```
ContentEngine# ftp-native custom-message ?
  download  Download the custom message file specified by the URL to the CE
  reset     Revert to default message and delete the local file on the CE
  upload    Upload the custom message file to the specified host, directory and
            filename using the FTP protocol
```

**ftp-native custom-message download** {**welcome** *welcome-message url* | **acl-denied** *acl-denied-message url*}| **reset** {**acl-denied** | **welcome** | **all** } | **upload** {*ip-address* | *hostname*} *dirname filename message*

Table 5-6 describes the parameters for the **ftp-native custom-message** command.

***Table 5-6 Parameters for the ftp-native custom-message CLI Command***

| Parameter | Description |
|---|---|
| **download** | Copies the custom message file to the Content Engine from the specified URL. To change the text for a specific message, use this option to identify the message you want to change, and specify the URL that is the source for the custom message file. The custom message file can be up to 16 KB in size and is used instead of the standard message for the specified message. |
| **welcome** | Indicates that you want to download the custom welcome message for the FTP proxy-mode welcome message. |
| **acl-denied** | Indicates that you want to download the custom error message that the Content Engine is to display to the FTP client because the client is being denied access based on the IP ACLs that have been defined for the native FTP proxy service. |
| *message url* | Specifies the URL from which the custom message file (the file that contains the FTP proxy-mode welcome message or the ACL access-denied error message) should be retrieved. The file size cannot exceed 16 KB. |
| **reset** | Specifies that the Content Engine (the FTP proxy) is to revert to the default message. Also deletes the local message files on the Content Engine. |
| **welcome** | Specifies that the Content Engine is to revert to the default FTP proxy-mode welcome message. |
| **acl-denied** | Specifies that the Content Engine is to revert to the default ACL access-denied error message. |
| **all** | Specifies that the Content Engine is to revert to all default FTP proxy messages. |
| **upload** | Uploads the custom message file from the Content Engine to the specified host, directory, and file. |
| *ip-address* | Specifies the IP address of the host to which the Content Engine is to copy the custom message file. |
| *hostname* | Specifies the hostname to which the Content Engine is to copy the file that contains the custom message. The host should be reachable and allow a file to be copied to the specified directory. |
| *dirname* | Specifies the directory name to which to copy the custom message file. |
| *filename* | Specifies the filename to which to copy the custom message file. |

To display a list of the names of the configured FTP native custom messages:

```
ContentEngine# show ftp-native custom-message
```

To display the contents of the local copy of the specified custom message (for example, the ACL-denied message or the welcome message that has been downloaded to the Content Engine) on the CLI display screen:

```
ContentEngine# show ftp-native custom-message {acl-denied acl-denied-filename | welcome
welcome-filename}
```

To copy the FTP native custom welcome message to the Content Engine:

```
ContentEngine# ftp-native custom-message download welcome
http://www.myserver.com/errors/ftp-native-welcome.txt
```

# Shutting Down Standalone Content Engines

A controlled shutdown refers to the process of properly shutting down a Content Engine without turning off the power on the device. With a controlled shutdown, all of the application activities and the operating system are properly stopped on a standalone Content Engine but the power is still on. Controlled shutdowns of a standalone Content Engine can help you minimize the downtime when the Content Engine is being serviced.

**Caution** If a controlled shutdown is not performed, the Content Engine file system can be corrupted. It also takes longer to reboot the Content Engine if the Content Engine is not properly shut down.

The **shutdown** EXEC command enables you to shut down and optionally power off a Content Engine.

- *Shutdown* means that all application activities (applications and operating system) are stopped, but the power is still on. This is a shutdown only, and is similar to the Linux **halt** command.
- *Shutdown poweroff* means that the Content Engine is powered down by the ACNS software after being shut down. This operation is also referred to as a *software poweroff*. The implementation of the shutdown poweroff feature uses the advanced configuration and power interface (ACPI) power management interface.

  When a shutdown poweroff is performed on a standalone Content Engine, these conditions result:

  - All application activities are stopped on the Content Engine, and the Content Engine is shut down through the ACNS software.
  - Power is turned off through a software power off.
  - The fan is not running; however, the power LED is flashing on the Content Engine.

**Note** Content Engines cannot be powered on again through software after a software poweroff. You must press the power button once on these Content Engines to bring these Content Engines back online.

Table 5-7 describes the shutdown-only operation and the shutdown poweroff operation for standalone Content Engines.

*Table 5-7 Shutting Down Standalone Content Engines Through CLI Commands*

| Activity | All Content Engine Models Supported in the ACNS 5.3.1 Software or Later | Content Engines with Power Management Capability in ACNS 5.3.1 Software or Later |
|---|---|---|
| User performs a shutdown operation on the Content Engine | Shutdown only<br>`ContentEngine# shutdown` | Shutdown poweroff<br>`ContentEngine# shutdown poweroff` |
| User intervention to bring Content Engine back online | To bring a Content Engine that has an on/off switch on the back (for example, the CE-507, CE-507AV, CE-560, CE-560AV, or the CE-590) back online after a shutdown operation, turn the on/off switch twice.<br>To bring a Content Engine that has a power button (instead of an on/off switch on the back) back online after a shutdown operation: first press and hold the power button for several seconds to power off these models, and then press the power button once again. | After a shutdown poweroff, you must press the power button once to bring the Content Engine back online. |
| File system check | Will not be performed after you turn the power on again and reboot the Content Engine. | Will not be performed after you turn the power on again and reboot the Content Engine. |

All of the Content Engine models that are supported in the ACNS 5.3 release support the shutdown feature (the **shutdown** EXEC command), which was added in the ACNS 5.3.1 software release. However, the shutdown poweroff feature (the **shutdown poweroff** EXEC command) is only supported on the newer Content Engine models that support the power management capability (for example, the CE-510, CE-510A, CE-511, CE-511A, CE-565, CE-565A, CE-566, CE-566A, CE-7305, CE-7305A, CE-7320, CE-7325, and CE-7326).

The **shutdown** EXEC command is supported in all device modes (Content Distribution Manager, Content Engine, Content Router, and IP/TV Program Manager). For a description of how to use the **shutdown** EXEC command with a standalone Content Engine (a Content Engine that is not registered with a Content Distribution Manager), see the next section, "Shutting Down Standalone Content Engines from the Command Line." For information about how to use the **shutdown** EXEC command with other types of devices (for example, a Content Distribution Manager, a Content Engine that is registered with a Content Distribution Manager, or a Content Router), see the *Cisco ACNS Software Command Reference, Release 5.5* publication.

# Shutting Down Standalone Content Engines from the Command Line

You can enter the **shutdown** EXEC command from a console session or from a remote session (Telnet or SSH Version 1 or Version 2) to shut down a standalone Content Engine.

To perform a shutdown on a standalone Content Engine, follow these steps:

**Step 1** Enter the **shutdown** EXEC command.

```
ContentEngine# shutdown
```

**Step 2** When you are asked if you want to save the system configuration, enter **yes**.

```
System configuration has been modified. Save?[yes]:yes
```

**Step 3** When you are asked if you want to proceed with the shutdown, press **Enter** to proceed with the shutdown operation.

```
Device can not be powered on again through software after shutdown.
Proceed with shutdown?[confirm]
```

A message appears, reporting that all services are being shut down on this Content Engine.

```
Shutting down all services, will timeout in 15 minutes.
shutdown in progress ..System halted.
```

**Step 4** After the system is shut down (the system has halted), an ACNS software shutdown shell displays the current state of the system (for example, "System has been shut down") on the console. You are asked whether you want to perform a software power off (the **Power down system by software** option), or if you want to reload the system through the software.

```
================= SHUTDOWN SHELL =================
System has been shut down.
You can either
   Power down system by pressing and holding power button
or
1. Reload system through software
2. Power down system through software
```

**Step 5** To power down the Content Engine, press and hold the power button on the Content Engine, or use one of the following methods to perform a shutdown poweroff:

- From the console command line, enter **2** when prompted, as follows:

  ```
  ================= SHUTDOWN SHELL =================
  System has been shut down.
  You can either
     Power down system by pressing and holding power button
  or
  1. Reload system through software
  2. Power down system through software
  ```

- From the Content Engine CLI, follow these steps:

  a. Enter the **shutdown poweroff** EXEC command.

  ```
  ContentEngine# shutdown poweroff
  ```

  b. When you are asked if you want to save the system configuration, enter **yes**.

  ```
  System configuration has been modified. Save?[yes]:yes
  ```

  c. When you are asked to confirm your decision, press **Enter**.

  ```
  Device can not be powered on again through software after poweroff.
  Proceed with poweroff?[confirm]
  Shutting down all services, will timeout in 15 minutes.
  poweroff in progress ..Power down.
  ```

## Shutting Down and Rebooting Standalone Content Engines from the Content Engine GUI

You can use the Content Engine GUI to perform a controlled shutdown on a standalone Content Engine. The Content Engine performs a controlled shutdown and then restarts the operating system on the Content Engine.

The Content Engine releases all WCCP connections to a router during the reboot process if these conditions exist:

- The Clean WCCP shutdown check box is checked in the main window of the Content Engine GUI.
- WCCP Version 2 is enabled on the Content Engine.

To use the Content Engine GUI to perform a controlled shutdown on a standalone Content Engine, follow these steps:

**Step 1** From the Content Engine GUI, click the **Reboot** button in the Content Engine main window (Figure 4-17).

The Content Engine performs the controlled shutdown and then restarts the operating system on the Content Engine.

**Step 2** To reboot this standalone Content Engine, click the **Reboot** button in the Content Engine main window.

**Tip** If the Content Engine main window (Figure 4-17) is not currently displayed in your browser, click the words "Content Engine" in the upper-left corner any Content Engine GUI window to return to the Content Engine main window.

**Step 3** When you are prompted to confirm your decision, click **OK** to begin rebooting this standalone Content Engine.

# Removing or Replacing a Content Engine

See the Content Engine hardware documentation for instructions on physically removing a Content Engine from an active network.

The router and the Content Engine are in constant communication when WCCP is enabled; thus, when the router notices that the Content Engine is no longer responding to it, the router stops sending requests to the Content Engine. This is transparent to users. If other Content Engines are attached to the router, the router continues sending requests to the other Content Engines.

# Remotely Upgrading the BIOS

In the ACNS 5.3.1 software and later releases, you can perform a BIOS upgrade remotely through the CLI. This feature is currently only supported for the Content Engine model CE-7326. For more information on this topic, see the *Cisco ACNS Software Upgrade and Maintenance Guide, Release 5.x.*