



Configuring Additional Network Interfaces and Bandwidth on Standalone Content Engines

This chapter describes how to set up additional network interfaces and configure bandwidth for these interfaces and content services in a locally managed deployment. It contains the following sections:

- [Configuring Additional Network Interfaces, page 15-1](#)
- [Configuring Bandwidth for Interfaces and Content Services, page 15-5](#)
- [Configuring Disk Space, page 15-7](#)
- [Mounting to a Network Attached Storage Device, page 15-8](#)



Note

For complete syntax and usage information for the CLI commands used in this chapter, refer to the *Cisco ACNS Software Command Reference, Release 5.2* publication.

Configuring Additional Network Interfaces

When you initially configured a standalone Content Engine, you chose an initial interface and either configured it for DHCP, or gave it a static IP address. You can use the Content Engine CLI to configure additional network interfaces on the Content Engine for redundancy, load balancing, and performance optimization.

This section describes how to configure additional interfaces on standalone Content Engines.

Configuring Multiple Network Interfaces

You can configure multiple network interfaces as either active-active interfaces or as active-standby interfaces. You configure multiple interfaces as active-active by using the **interface** global configuration command and by assigning an IP address to each interface. When multiple interfaces are configured, they are active simultaneously. This configuration is used to achieve better performance. For example:

```
ContentEngine# configure
ContentEngine(config)# interface FastEthernet 0/0
ContentEngine(config-if)# ip address 10.10.10.10 255.0.0.0
```

With an active-standby configuration, the interfaces are configured using the **standby** configuration interface command, and they remain inactive unless an active interface fails. When an active network interface fails (because of cable trouble, Layer 2 switch failure, high error count, or other failure), and that interface is part of a standby group, a standby interface can become active and take the load off the failed interface. With active-standby interface configuration, only one interface is active at a given time. Active-standby is used mainly for fault tolerance purposes. The performance of the device is limited by the single active interface.

```
ContentEngine# configure
ContentEngine(config)# interface FastEthernet 0/1
ContentEngine(config-if)# standby ?
    <1-4> Standby group number
ContentEngine(config-if)# standby 3 ?
    errors    Set the maximum number of errors allowed on this interface
    ip        Set the IP address of a standby group
    priority  Set the priority of an interface for the standby group
ContentEngine(config-if)# standby 3 errors ?
    <0-4294967295> Max. no. of errors allowed on this interface for the standby
                  group
ContentEngine(config-if)# standby 3 ip ?
    A.B.C.D IP address of the standby group
ContentEngine(config-if)# standby 3 priority ?
    <0-4294967295> Priority of this interface for the standby group
```

To configure standby interfaces, interfaces are logically assigned to standby groups. The following rules define the standby group relationships:

- A standby group comprises two or more interfaces.
- The maximum number of standby groups on a Content Engine is four.
- Each interface is assigned a unique IP address, and each standby group is assigned a unique standby IP address, shared by all members of the group.
- Configure the duplex and speed settings of the standby group member interfaces for better reliability.
- Each interface in a standby group is assigned a priority. The operational interface with the highest priority in a standby group is the active interface. Only the active interface uses the group IP address.
- If the active interface fails, the operational interface in its standby group that is assigned the next highest priority becomes active.
- If all the members of a standby group fail and then one recovers, the ACNS software brings up the standby group on the operational interface.
- The priority of an interface in a standby group can be changed at runtime. The interface that has the highest priority after this change becomes the new active interface (the default action is to preempt the currently active interface if an interface with higher priority exists).
- The maximum number of errors allowed on the active interface before the interface is shut down and the standby is brought up is configured with the **errors** option of the standby command; this option is disabled by default.



Note

Interface IP addresses and standby group IP addresses must be on different subnets to ensure reliable operation. You can use dummy IP addresses in the private address space to serve as interface primary IP addresses, and use the real Content Engine IP address to serve as the standby group IP address in a different subnet to satisfy this requirement. Make sure to configure the interface default gateway using the **ip default-gateway** global configuration command instead of the **ip route** command.

This example configures three interfaces to be part of the same standby group (standby group 1), with interface 3/0 as the active interface.

```
Console(config)# interface fastEthernet 3/0 standby 1 ip 172.16.10.10 255.255.254.0
Console(config)# interface fastEthernet 3/1 standby 1 ip 172.16.10.10 255.255.254.0
Console(config)# interface fastEthernet 3/2 standby 1 ip 172.16.10.10 255.255.254.0
Console(config)# interface fastEthernet 3/0 standby 1 priority 300
Console(config)# interface fastEthernet 3/1 standby 1 priority 200
Console(config)# interface fastEthernet 3/2 standby 1 priority 100
Console(config)# interface fastEthernet 3/0 standby 1 errors 10000
Console(config)# interface fastEthernet 3/1 standby 1 errors 10000
Console(config)# interface fastEthernet 3/2 standby 1 errors 10000
```

Use the **show standby EXEC** command to view your standby interface configuration on a Content Engine.

```
Console# show standby
Standby Group:1
IP address: 172.16.10.10, netmask: 255.255.254.0
Maximum errors allowed on the active interface: 10000
  Member interfaces:
    FastEthernet 3/0      priority: 300
    FastEthernet 3/1      priority: 200
    FastEthernet 3/2      priority: 100
Active interface: FastEthernet 3/0
```

Configuring Multiple IP Addresses on a Single Interface

Use the **interface secondary** global configuration command to configure more than one IP address on the same interface. By configuring multiple IP addresses on a single interface, the Content Engine can be present in more than one subnet. This allows you to optimize response time because the content goes directly from the Content Engine to the requesting client without being redirected through a router. The Content Engine becomes visible to the client because both are configured on the same subnet.

Up to four secondary addresses can be assigned to an interface on a Content Engine. These addresses become active only after the primary address is configured. No two interfaces can have the same IP address in the same subnetwork. To set these secondary IP addresses, use the **ip address** command.

For example:

```
ContentEngine# configure
ContentEngine(config)# interface FastEthernet 0/0
ContentEngine(config-if)# ip address 10.10.10.10 255.0.0.0 secondary
```

Configuring the Fibre Channel Interface

ACNS 5.x software supports Fibre Channel interfaces. Fibre Channel is the chosen technology for interconnecting storage devices and servers in a storage area network (SAN). In a SAN, the storage need not be directly attached to the server, and data transfer happens over a high-throughput, high-availability network. Fibre Channel is capable of operating at speeds of 1 gigabit per second (Gbps) and 2 Gbps.

To detect the presence of Fibre Channel storage, the storage array must be configured to assign storage space for the Content Engine, and the Content Engine must be reloaded before it can detect the storage assignment. To confirm whether the Content Engine has detected the storage assignment, use the **show disks** and the **show disks details EXEC** commands.

To configure the Fibre Channel interface on the Content Engine, use the **interface FibreChannel slot/port** command in interface configuration mode. For example:

```
ContentEngine# configure
ContentEngine(config)# interface FibreChannel 0/0
ContentEngine(config-if)#?
    exit      Exit from this submode
    mode      Change the fibre channel interface operating mode
    no        Negate a command or set its defaults
    speed     Change the fibre channel interface speed
ContentEngine(config-if)# mode ?
    autosense      Use this mode to have the CE autosense
    direct-attached Use this mode when the CE is directly connected to storage array
    switched       Use this mode when the CE is connected to a switch
ContentEngine(config-if)# speed ?
    1              1Gbps
    2              2Gbps
    autosense     autosense
```


Note

For a complete description of the **interface FibreChannel** command syntax and usage, refer to the *Cisco ACNS Software Command Reference, Release 5.2*. For information regarding which Fibre Channel storage arrays are supported by Cisco Systems, refer to the *Release Notes for Cisco ACNS Software, Release 5.2*.

Configuring EtherChannel

EtherChannel for ACNS 5.x software supports the grouping of up to four same-speed network interfaces into one virtual interface. This grouping capability allows the setting or removing of a virtual interface that consists of two, three, or four Fast Ethernet interfaces or two Gigabit Ethernet interfaces. EtherChannel also provides interoperability with Cisco routers, switches, and other networking devices or hosts supporting EtherChannel; load balancing; and automatic failure detection and recovery based on each interface's current link status.

To create an EtherChannel, use the **interface PortChannel number** command in interface configuration mode. For example:

```
ContentEngine# configure
ContentEngine(config)# interface PortChannel 2
ContentEngine(config-if)# ip address 10.10.10.10 255.0.0.0
ContentEngine(config-if)# exit
```

To remove an EtherChannel, use the **no** form of the command:

```
ContentEngine(config)# interface PortChannel 2
ContentEngine(config-if)# no ip address 10.10.10.10 255.0.0.0
ContentEngine(config-if)# exit
ContentEngine(config)# no interface PortChannel 2
```

To add or remove ports from an EtherChannel, use the commands in the following examples. These commands add a physical Fast Ethernet port to a previously created Fast EtherChannel. The channel number is the same as the channel number specified when the port channel interface was created. You can use either the Fast Ethernet or the Gigabit Ethernet ports to form an EtherChannel; however, an EtherChannel cannot contain both Fast Ethernet and Gigabit Ethernet interfaces. Note that a physical interface can be added to an EtherChannel subject to the device configuration.

To add an interface to a channel group:

```
ContentEngine# configure
ContentEngine(config)# interface FastEthernet 1/1
ContentEngine(config-if)# channel-group 2
ContentEngine(config-if)# exit
```

To remove an interface from a channel group:

```
ContentEngine(config)# interface FastEthernet 1/1
ContentEngine(config-if)# no channel-group 2
ContentEngine(config-if)# exit
```

To configure for load balancing, use the **port-channel load-balance** global configuration command.

```
CE(config)# port-channel load-balance
```

The following **load-balance** options are available:

```
dst-ip          Destination IP Address
dst-mac         Destination MAC Address
round-robin     Round robin each interface (default)
```

Round robin allows traffic to be distributed evenly between all interfaces in the channel group. The other balancing options give you flexibility in choosing interfaces when sending an Ethernet frame. The **load-balance** command is effective globally. If two channel groups are configured, they have to use the same load-balancing option.

Configuring Bandwidth for Interfaces and Content Services

With the various types of traffic originating from a device, every type of traffic, such as streaming media, HTTP, and metadata, consumes network resources.

Configuring Interface Bandwidth

To configure an interface bandwidth on a standalone Content Engine, use the **bandwidth** interface configuration command. Bandwidth is specified in megabits per second (Mbps). The **1000 Mbps** option is not available on all ports and is the same as autosense.

```
bandwidth {10 | 100 | 1000}
```

```
no bandwidth {10 | 100 | 1000}
```

To restore default values, use the **no** form of this command.

For a Content Engine CE-7320 model that has an optical Gigabit Ethernet interface; the speed of this interface cannot be changed. Therefore, Gigabit Ethernet interfaces only run at 1000 Mbps for a CE-7320. For newer models of the Content Engine (for example, the CE-510, CE-565, CE-7305, and CE-7325) that have a Gigabit Ethernet interface over copper, this restriction does not apply; you can configure these Gigabit Ethernet interfaces to run at 10, 100, or 1000 Mbps. Note that on these newer Content Engine models, the 1000 Mbps setting implies autosense (for example, you cannot configure the Gigabit Ethernet interface to run at 1000 Mbps and half duplex). The ACNS 5.x software automatically enables autosense if the speed is set to 1000 Mbps.

With ACNS 5.x software, you can also configure a maximum bandwidth for the preloading process using the **pre-load max-bandwidth** global configuration command.

Configuring Bandwidth for Content Services

On a standalone Content Engine, use the **bandwidth** global configuration command to specify bandwidth limits for WMT and RealProxy live content or the streaming media that is being cached on the Content Engine.

For each type of content service (WMT and RealProxy), you can specify the maximum amount of bandwidth on the Content Engine that should be allocated to that service during a specified period. This is called “scheduled bandwidth.” For example, you can limit the RealProxy bandwidth to 1000 kbps from Monday at 8:00 a.m. to Friday at 6:00 p.m.

```
ContentEngine(config)# bandwidth allow 1000 real-proxy start-time monday 8:00
end-time friday 18:00
```

Although there are no default values for any of the bandwidth configuration fields, the values that you enter depend upon the bandwidth license in effect for your specific system. If you enter a value that is beyond the allowable bandwidth based on your system’s bandwidth capacity, the value is accepted but a warning message is displayed. Internally, your system bandwidth is limited to the maximum value granted by the license. All values entered are in kilobits per second (kbps).

Configuring Bandwidth for a WMT Proxy Upstream

ACNS 5.x software includes the WMT proxy, which has the ability to cache on-demand media files when the user requests these files for the first time. All subsequent requests for the same file are served by the WMT proxy using the MMS protocol. The WMT proxy can also live-split a broadcast, which causes a single unicast stream to be requested from the origin server every time a broadcast is made.

The bit rate between the proxy and the origin server is called the incoming bit rate. Use the **bitrate** global configuration command to limit the maximum bit rate per session for large files delivered using either the HTTP or the MMS protocol.

```
bitrate {http default bitrate | wmt {incoming bitrate | outgoing bitrate}}
```

Table 15-1 describes the parameters for the **bitrate** command.

Table 15-1 Parameters of the **bitrate** Command

Parameter	Description
http	Configures the maximum pacing bit rate for large files sent using the HTTP protocol in kilobits per second (kbps).
default	Sets the default bit rate in kbps for large files.
<i>bitrate</i>	Bit rate in kbps (0–2000000).
wmt	Configures the bit rate, in kbps, for large files sent using the WMT protocol.
incoming	Sets the incoming bit rate settings.
<i>bitrate</i>	Incoming bit rate in kbps (0–2147483647).
outgoing	Sets the outgoing bit rate settings.
<i>bitrate</i>	Outgoing bit rate in kbps (0–2147483647).



Note The aggregate bandwidth used by all concurrent users is still limited by the default device bandwidth, or by the limit configured using the **bandwidth** command.

The following example shows how to configure an incoming bit rate for a file sent over HTTP.

```
ContentEngine(config)# bitrate http default 100
```

The following example shows how to configure an incoming bit rate for a file sent over MMS.

```
ContentEngine(config)# bitrate wmt incoming 300000
ContentEngine(config)# exit
```

Use the **show wmt EXEC** command to check the currently configured bit rate.

```
ContentEngine# show wmt
```

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 15-2](#).

Table 15-2 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.

Use the **disk add EXEC** command to add a single disk with the specified partitions. Use the **disk config sysfs EXEC** command to configure disk resources for standalone Content Engine. Use the **show disks EXEC** command to view information about the disk configurations for standalone Content Engines.

In ACNS 5.2 software, the ability to monitor Content Engine disk drives was added. 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 20-22. For more information about monitoring critical disks, see the [“Monitoring Critical Disk Drives on Standalone Content Engines”](#) section on page 20-21.

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** command in EXEC mode.

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** Enter the **disk config sysfs** EXEC command. 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.

---

## Mounting to a Network Attached Storage Device

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

For more information about mounting a Content Engine to a NAS device, refer to the *Cisco ACNS Software Configuration Guide for Centrally Managed Deployments, Release 5.2*.

■ Mounting to a Network Attached Storage Device