CacheOS™ 3.1 - Management and Configuration Guide

Note For optimal graphics quality when displaying this file in Acrobat™, set your display to greater than 256 colors, and set the Acrobat Viewer zoom factor to 134% or 160%.

CacheFlow Inc. (408) 220-2200 Voice

650 Almanor Avenue (408) 220-2250 FAX

Sunnyvale, California 94086 (888) 702-3569 Technical Support

info@cacheflow.com www.cacheflow.com

Printed in U.S.A.

Document Number: 231-00407-00

Document Revision: 31C 10/2000

Contents

This page intentionally blank.

Document Conventions

The information in this manual conforms to several typographic conventions to make reading the material easier. The following table lists and explains the typographic styles used in this manual.

Typographic convention	Description
`Monospace text`	Monospace text indicates a command prompt or console output.
Bold text *	Bold text indicates a command the reader should type exactly as shown.
Bold and italic text*	Bold and italic text indicates a command variable. The reader should substitute information appropriate to their installation.
Text separated by vertical bars (e.g., 2 | 3 | 4)	Vertical bars indicate explicit command options.
Blue Text (PDF files only)	Blue text is a hyper jump to another location in this document or to an internet location.
Italic text	Italic text is a reference to another publication.
Bold sans serif text	Bold sans serif text indicates that the paragraph is especially important and should be read.

* For readability, commands inside of command tables are not bolded, but italics are preserved.

Graphics Quality Viewing .PDF Files

For optimal graphics quality when displaying this file in Acrobat, set your display to greater than 256 colors, and set the Acrobat Viewer zoom factor to 134% or 160%.

This page intentionally blank.

Chapter 1 - CacheOS Features

CacheFlow Content Acelerators provide the ultimate in Web caching performance. A Content Accelerator is a single-purpose device, based on the patent-pending CacheOS™ operating system. It is designed specifically for extremely efficient caching of Web objects.

CacheOS is a new generation of Web caching technology invented by CacheFlow Inc. It combines features like Transparent and Active Caching, Object Pipelining, DNS Caching and Advanced Filtering and Forwarding, and Denial of Service attack resilience, along with an ultra-efficient storage system.

Transparent Caching

Transparent caching allows you to deploy CacheOS without requiring users to configure their Web browsers. This simplifies installation and ensures users actually use the cache. CacheOS is configured for transparency by default.

Active Caching

Active Caching is the method CacheOS uses to store and refresh objects in the cache. When a Web page is requested by a client, CacheOS tracks a variety of information for every object on the page, including the frequency of requests, frequency of object modifications, and time to retrieve the object. CacheOS then uses this information to determine the refresh pattern for the object.

Rather than waiting to refresh objects as clients request them, forcing the clients to wait while the objects are verified, CacheOS constantly analyzes the cache and refreshes objects according to each object's refresh pattern. This drastically reduces the amount of time required to deliver the objects to the clients.

Object Pipelining

Each Web page can be composed of dozens of objects, such as images, sounds, Java applets, etc. Under normal browser operations, the client requests a Web page, and the HTML document is retrieved and delivered to the client. As the client reads the HTML document, it begins requesting the rest of the objects that make up the page.

CacheOS accelerates this process. When CacheOS retrieves the HTML document, it reads the document and begins requesting the objects that make up the page. CacheOS can parse the document and request the objects much faster than the client can request them. By the time the client begins requesting objects on the page, CacheOS has already started loading them in the cache. Object pipelining delivers complete pages to the client faster, even when the page has to be retrieved from the server.

DNS Caching

CacheOS also maintains a large DNS cache using active caching techniques similar to those used for Web objects. The shared DNS cache boosts overall Web cache performance by eliminating latencies incurred by contacting a DNS server to resolve an address.

Rules-Based Filtering and Forwarding

CacheOS provides a rich expression library making it possible to create sophisticated filtering and forwarding rules ranging in applicability from an individual client to an entire organization.

Content Filtering

Content filtering allows you to control the type of content served to clients. CacheOS provides support for scheduled updates of filter lists from Websense™ and SmartFilter™, two leading providers in the industry.

Security

CacheOS security provides external CacheOS Administrator and proxy user authentication using either Lightweight Directory Access Protocol (LDAP) or Remote Authentication Dial-In User Service (RADIUS).

Server-Side Transparency

Sometimes, tracing a client address is important. Server-side transparency provides this capability. When server-side transparency is enabled, CacheOS retains client IP addresses for all port 80 traffic to and from the Content Accelerator. In this scheme, the client IP address is always revealed to the server, allowing the server to keep accurate records of what client accessed the server on a given date at a given time.

Multiprocessor Support

On 5000 series systems, CacheOS seamlessly supports a dual processors. Performance of a two-processor system is dramatically better than the same system with a single processor. CacheOS intelligently balances processor loads without requiring any related configuration or tuning. The status and utilization of both processors can be viewed in the CacheOS Statistics applets.

Gigabit Ethernet Support

On the 3000 and 5000 series systems, CacheOS supports gigabit Ethernet adapters, which can be included in a Content Accelerator as an option. When used in combination with a gigabit switch or router, data transfers between the Content Accelerator and switch or router occur up to ten times faster than with 100-Base T adapters.

Dynamic Bypass

Dynamic bypass provides a maintenance-free method for improving performance of the Content Accelerator. CacheOS does this by adding dynamic bypass entries, containing the URLs of sites that have returned an error, to the Content Accelerator's local bypass list. The performance gains realized with this feature are substantial because redundant attempts to contact the origin server are minimized.

Multiple Default Gateways (load balancing)

CacheOS supports multiple default gateways. This capability allows CacheOS to distribute traffic originating at the cache through multiple default gateways. Further, you can fine-tune how the traffic is distributed.

Enhanced Real Networks Streaming Media Performance

CacheOS supports Real Networks streaming media and provides a configuration mechanism through which bandwidth use, stream splitting, and other options can be customized to fit the bandwidth considerations of any network. CacheOS 3.1 includes significant RealProxy performance enhancements.

Environment Subsystem

On systems with hardware monitoring, such as the models 3000 and 5000, CacheOS can report real-time information about many components of the system such as CPU temperature, power supply and cooling fan status, and network card or disk failures.

Configuration Save and Restore

Archiving a CacheFlow device's system configuration on a regular basis is a prudent measure. In the rare case of a complete system failure, restoring a Content Accelerator to its previous state is simplified by loading an archived system configuration from an HTTP, FTP, or TFTP server. The archive contains all system settings differing from system defaults, along with any forwarding, filtering, and access lists installed on the Content Accelerator.

Denial of Service (Dos) Attack Resilience

CacheOS includes internal security measures to detect and stop denial of service attacks.

This page intentionally blank.

Chapter 2 - Working with CacheOS

CacheOS has two user interfaces, a Graphical User Interface (GUI) and a Command Line Interface (CLI). Either of this user interfaces can be used to perform setup, management, and maintenance tasks. Certain tasks can be performed using only one interface or the other. Most tasks, however, can be performed using either the GUI or CLI. In this reference, tasks are described using both interfaces, unless only one interface is available.

First-Time Setup of a CacheFlow System

Because a new Content Accelerator does not yet have a network or administrator configuration, this initial configuration must be performed. The quickest and simplest way to set up the Content Accelerator's initial network configuration and connect it to the network, because no additional equipment is required, is to use the front panel LCD and joystick, if your system is equipped with them. You can also perform this task by using a direct connection between the Content Accelerator and one of the following:

- Stand-alone serial terminal
- PC with an available serial (COM) port

After the initial network configuration has been accomplished, the Content Accelerator can be connected to the network.

Note Whether you connect the Content Accelerator to a stand-alone serial terminal or PC serial port, use the serial cable supplied with the system.

Using the Front Panel LCD and Joystick

Using the front panel LCD and joystick is the quickest method for setting the Content Accelerator's initial network configuration and connecting it to the network. This section details operation of the joystick for inputting configuration settings.

Note the following when using the joystick

- Pressing the joystick inward for one second switches the system LCD between Edit Mode and Review Mode. This is called "Joystick Enter".
- Pressing the joystick up or down in Edit Mode increments or decrements the numeric value.
- Pressing the joystick to the left or to the right in Edit Mode moves the cursor between fields.
- Pressing the joystick up or down in Review Mode cycles the display through the list of current network settings.

Note When the system is in Edit Mode, the cursor blinks. When the system is in Review Mode, the cursor does not blink.

Once the machine completes its powering-on sequence, you'll see a series of screens displaying various network system statistics.

Connecting to the Content Accelerator Using a Serial Terminal

This method for performing initial system setup is straightforward, as no PC serial port issues are involved.

To connect using a stand-alone serial terminal

1. Connect the serial cable between the serial terminal and the Content Accelerator's serial port.
2. Turn on the serial terminal and verify the terminal is set as follows
   - Baud rate: 9600 bps
   - Data bits: 8
   - Parity: none
   - Stop bits: 1
   - Flow control: none
   - Smooth-scroll: disabled
3. Turn on the Content Accelerator
4. Once the system has finished booting, a configuration alert is displayed.

```
******************** CONFIGURATION ALERT ********************
        One or more IP addresses have not been configured.
        System startup cannot continue until the missing IP
        addresses are provided.
********** SYSTEM STARTUP TEMPORARILY SUSPENDED  **********
Press "enter" three times to activate the setup console
```

5. Press Enter three times to activate the setup console.
   Once this prompt appears, the system is ready for initial configuration.

Connecting to the Content Accelerator Using a PC

If the PC is using standard serial port settings, the connection should be problem free. Be aware that this method for performing initial system setup can be complicated by non-standard PC serial port settings.

To connect using a PC

1. Shut down the PC.
2. Connect the serial cable between an available serial port on the PC and the Content Accelerator's serial port.
3. Boot the PC, and start a terminal emulator such as HyperTerminal and connect to an available serial port. Verify that the serial port is set as follows
   - Baud rate: 9600 bps
   - Data bits: 8

- Parity: none
- Stop bits: 1
- Flow control: none
- Smooth-scroll: disabled

4. Turn on the Content Accelerator
5. Once the system has finished booting, a configuration alert is displayed.

```
****************** CONFIGURATION ALERT ******************
        One or more IP addresses have not been configured.
        System startup cannot continue until the missing IP
        addresses are provided.
********** SYSTEM STARTUP TEMPORARILY SUSPENDED  **********
Press "enter" three times to activate the setup console
```

6. Press Enter three times to activate the setup console.

   Once this prompt appears, the system is ready for initial configuration.

Initial Network Configuration Using the Front Panel LCD and Joystick

Using the front panel LCD and joystick is the quickest method for setting the Content Accelerator's initial network configuration and connecting it to the network. Note that front panel setup only allows configuration of basic network IP addresses.

To perform initial network configuration using the front panel LCD and joystick

1. Power the system on.
2. When the Push to configure prompt appears, press Joystick Enter.
3. At the PIN prompt, press Joystick Enter.

   A PIN is not entered at this point because one has not yet been assigned to the Content Accelerator.

   A help screen appears.

   Note When the system is in Edit Mode, the cursor blinks. When the system is in Review Mode, the cursor does not blink.

   Note The PIN is set using the **security front-panel-pin** command in the CLI. Once set, the PIN must be provided before any changes to the system configuration are allowed through the joystick interface. This command is accepted, but has no effect on systems without a front panel LCD and joystick.

4. Press the Joystick down to begin defining network settings.
5. Switch to Edit Mode by pressing Joystick Enter. Enter the requested information by pressing the joystick up, down or sideways, switching back to Review Mode when done.
6. Press the joystick down to display the next setting. Continue entering requested information, switching from Review Mode to Edit Mode and back as needed.

7. After all of your changes are made, confirm your entries by pressing the joystick as the system requests. If you choose to review your entries before confirming them, press the joystick in the direction the system indicates.

Initial Network Configuration Using a Direct Connection

Once you have used one of the aforementioned methods to establish direct communication with the Content Accelerator, you can configure network and user account settings. After the initial configuration is completed, the Content Accelerator can be connected to the network.

To perform an initial network configuration

1. Establish communications with the Content Accelerator as detailed in the previous section.
2. If you have more than one network adapter installed, a `Configuring adapter [0]` prompt is displayed. Type the number of the adapter to configure.
3. At the IP address prompt, type the IP address reserved for the Content Accelerator.
4. At the IP subnet mask prompt type the subnet mask for the IP address.
5. At the IP gateway prompt type the address of the gateway on the network.
6. At the DNS server prompt type the address of the DNS server on the network.

   CacheOS displays a summary of the IP address information that has been entered.
7. At the prompt to change the Content Accelerator's IP addresses, type **y** or **n** as appropriate.

   If you type **n**, you are prompted to enter these values again.
8. At the prompt to create a console user account type **y**.
9. Enter a console username and password, and the enable password as prompted.

   Important To prevent unauthorized access to the Content Accelerator, the console username and passwords should be given only to those who will administer the Content Accelerator.
10. At the prompt to restrict access to an authorized workstation type **y** or **n** as appropriate.

    If you type **y**, you are prompted to enter information about the workstation(s) to which access should be restricted.

    The Initial Network Configuration is now complete.

Session Example

```
******************** CONFIGURATION ALERT ********************
        One or more IP addresses have not been configured.
        System startup cannot continue until the missing IP
        addresses are provided.
********** SYSTEM STARTUP TEMPORARILY SUSPENDED  **********
Press "enter" three times to activate the setup console

Welcome to the CacheOS Setup Console
---------------------- (page 1 of 4) ----------------------
```

```
DIRECTIONS:
        This setup console is used to assign IP addresses to
        the CacheOS device.  After assigning the IP addresses
        you can connect to the command line interface or
        Web interface to perform additional management tasks.
IP address [0.0.0.0]: 10.25.36.47
IP subnet mask [0.0.0.0]: 255.255.0.0
IP gateway [0.0.0.0]: 10.25.0.1
DNS server [0.0.0.0]: 10.25.0.2
You have entered the following IP addresses:
IP address: 10.25.36.47
IP subnet mask: 255.255.0.0
IP gateway: 10.25.0.1
DNS server: 10.25.0.2
Would you like to change any of them? Y/N [No]n
---------------------- (page 2 of 4) ---------------------
DIRECTIONS:
        You can connect to the command line interface or
        Web interface to perform additional management tasks.
        WARNING - access to the CacheOS device is currently
        unrestricted and it may be configured by unauthorized
        persons.
Would you like to create a console user account now? Y/N [Yes]y
Enter console username: Admin
Enter console password: *****
Verify password: *****
Enter enable password: *****
Verify password: *****
---------------------- (page 3 of 4) ---------------------
DIRECTIONS:
        Access to the command line interface and Web interface
        can be restricted to specific workstations, identified
        by their IP address.
        This setup console allows you to add one IP address to the
        list of authorized workstations (additional workstations
        may be configured later from either the command line
        interface or Web interface).
        The CacheOS device can currently be accessed from
        any workstation.
Would you like to restrict access to an authorized workstation? Y/N [No]n
---------------------- (page 4 of 4) ---------------------
        The CacheOS device has been successfully configured
        to use IP address: "10.25.36.47"
```

```
        You can connect to the command line interface or
        Web interface to perform additional management tasks.
        To connect to the command line interface, open the
        following location from your Telnet application:
        10.25.36.47
        To connect to the Web interface, go to the following
        location with your Internet browser:
        http://10.25.36.47:8081/
        The CacheOS device is password protected and may be
        accessed from any workstation.
---------------- CONFIGURATION COMPLETE ------------------
Press Enter three times to activate the serial console
```

Logging on to the Content Accelerator

After the Initial Network Configuration is complete, you can use a telnet application or a serial terminal emulator such as HyperTerminal to access the command line interface. If you connect using telnet, you are prompted for the console username and password configured when you initially configured the Content Accelerator. If you connect using a serial terminal emulator, you are prompted for a password only after you issue the **enable** command.

The command line interface uses two passwords: the CacheOS Server Edition console password is required to establish a connection to the interface, and a priveliged mode password can be set to restrict access to the privileged mode configuration options. If you have forgotten the username or password, you can reset them by restoring factory defaults and performing an initial network configuration.

Chapter 3 - Configuring Network Settings

Configuring a Network Adapter

You can use either the Web or Command Line interface to configure the Ethernet adapter(s) in your Content Accelerator. In this example, a single adapter is configured. Repeat the configuration process if the system is equipped with additional adapters.

To configure a network adapter

1. Select Management from the CacheOS home page.
2. Select an adapter from the adapter drop-down list.
3. Enter the IP address and subnet mask for the adapter.
4. Select the Gateways tab and add a default IP gateway address for the adapter.

   For information adding a gateway, refer to the Using Multiple IP Gateways section in this chapter.

   Important The IP gateway specified applies to all network adapters in the system.
5. To configure link settings or restrict inbound connections for the adapter, click Advanced Settings. Enter your changes and click OK to close the Advanced Settings dialog.
6. Click Apply to save changes.

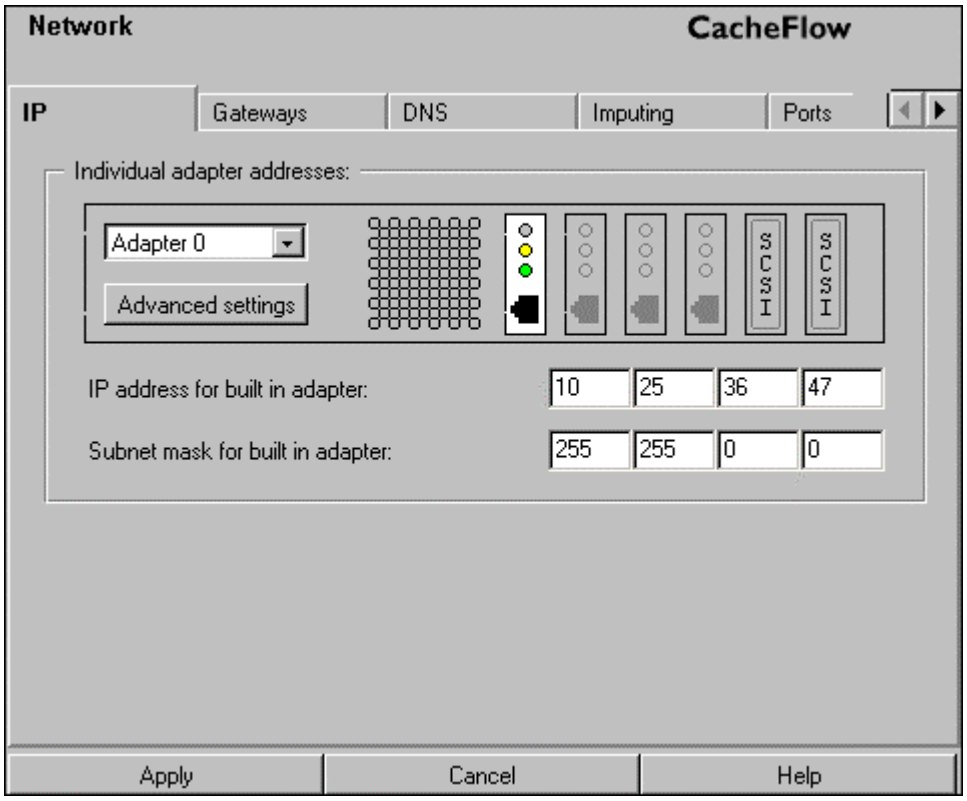


Figure 3-1 Configuring a network adapter

To configure a network adapter using the CLI

1. Open a terminal session with the Content Accelerator and type your Username and Password when prompted.
2. At the command prompt, type **enable** and type your Password when prompted.
3. At the command prompt, type **configure terminal**.
4. At the command prompt, type **interface *0***.
   The prompt changes to `(config interface 0)`.
5. Type **ip-address *10.25.36.47*** to set the adapter IP address.
6. Type **subnet-mask *255.255.0.0*** to set the subnet in which the Content Accelerator's IP address is located.
7. Type **exit** to return to the `(config)` prompt.

Advanced Network Adapter Configuration

The Advanced Settings button allows you to restrict inbound connections on the selected adapter, and to choose manual or automatic configuration of the adapter link settings.

Important Keep in mind that rejecting inbound connections improperly, or manually configuring link settings improperly, can cause the Content Accelerator to malfunction. Make sure that you know the correct settings before attempting either of these. If the Content Accelerator fails to operate properly after changing these settings, contact CacheFlow Support.

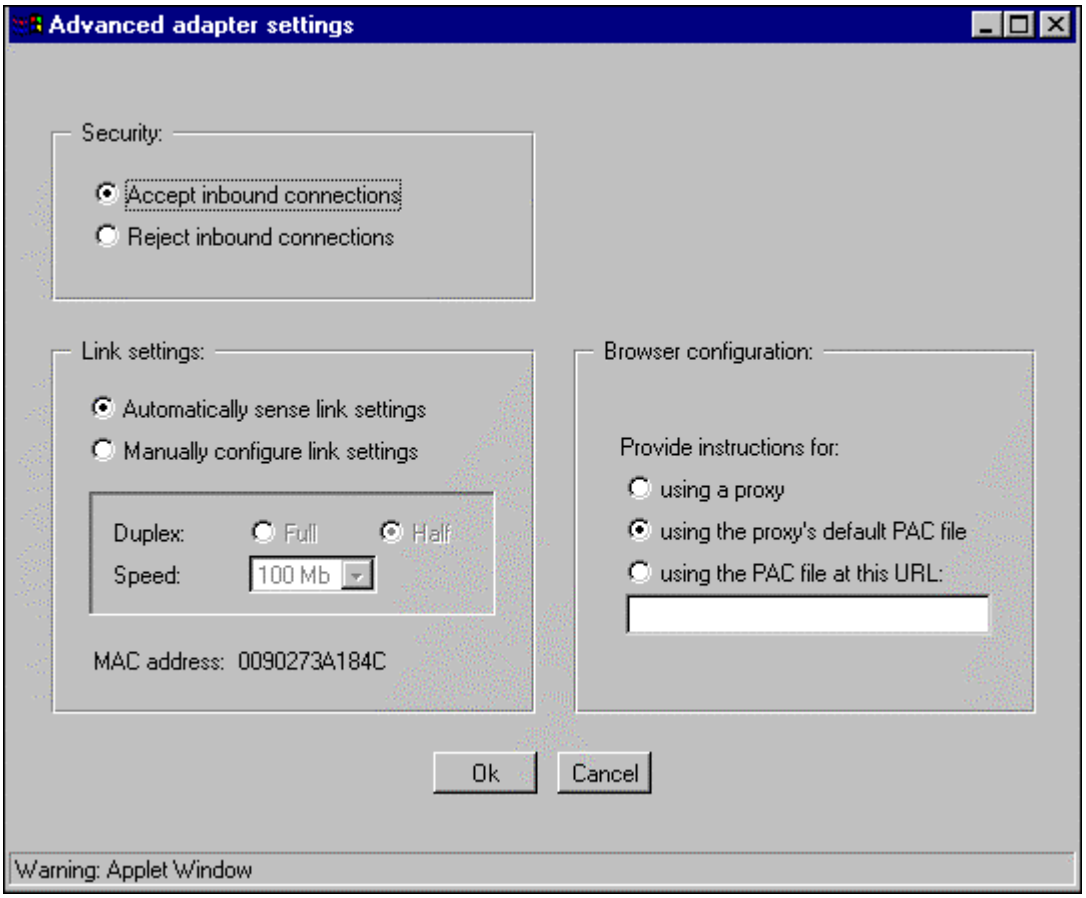


Figure 3-2 Advanced network adapter configuration

Rejecting Inbound Connections

By default, inbound connections are allowed on all network adapters.

To restrict inbound connections on a network adapter

1. Select Management from the CacheOS home page.
2. Select an adapter from the drop-down list.
3. Click Advanced Settings.
4. To allow inbound connections, click the Accept inbound connections radio button. To reject inbound connections, click the Reject inbound connections radio button.
5. Click OK to close the Advanced Settings dialog box.
6. To save changes, click Apply.

To restrict inbound connections on a network adapter using the CLI

1. Open a terminal session with the Content Accelerator and type your Username and Password when prompted.
2. At the command prompt, type **enable** and type your Password when prompted.
3. At the command prompt, type **configure terminal**.
4. At the command prompt, type **interface *0***.
   The prompt changes to `(config interface 0)`.
5. Type **no accept inbound** to reject inbound connections.
6. Type exit to return to the `(config)` prompt.

Manually Configuring Link Settings

By default, the CacheFlow device automatically determines the link settings for all network adapters. If your network adapter is incorrectly identified by the device, you can manually configure the link settings.

To manually configure link settings on a network adapter

1. Select Management from the CacheOS home page.
2. Select an adapter from the drop down list.
3. Click Advanced Settings.
4. Select Manually configure link settings.
5. Select Half or Full duplex.
6. Select the correct network speed.
7. Click OK to close the Advanced Settings dialog box.
8. To save changes, click Apply.

To manually configure link settings on a network adapter using the CLI

1. Open a terminal session with the Content Accelerator and type your Username and Password when prompted.
2. At the command prompt, type **enable** and type your Password when prompted.
3. At the command prompt, type **configure terminal**.
4. At the command prompt, type **interface *0***.
   The prompt changes to `(config interface 0)`.
5. At the command prompt, type **no link-autosense**.
6. Type **full-duplex** or **half-duplex** as applicable.
7. Type **speed 10** or **speed 100** as applicable.
8. Type **exit** to return to the `(config)` prompt.

Generating Browser Configuration Instructions for Clients

If your network does not use CacheOS's transparency feature, clients must configure their browsers to use either an explicit proxy server or a PAC (Proxy Auto-Configuration) file. CacheOS generates client instructions that describe

how to configure Internet Explorer, Netscape Communicator, and other browsers based upon instructions selected by the Content Accelerator Administrator. Client instructions can be configured for each network adapter in the Content Accelerator.

After client instructions have been selected, the Content Accelerator's administrator notifies clients to go to the Content Accelerator's home page and follow the instructions in the Browser Configuration section. CacheOS detects the browser installed on the client and displays the appropriate instructions.

Three options for client instructions are available

- Instructions to configure the client browser to use the Content Accelerator as a proxy server.
- Instructions to configure the client browser to use the default PAC (Proxy Auto-Configuration) file located on the Content Accelerator.
- Instructions to configure the client browser to use a custom PAC file located on a Web server.

Using the Content Accelerator as a Proxy

To use the Content Accelerator as a proxy when transparent caching is disabled you must provide customized instructions to clients informing them to configure their browser to use the Content Accelerator as a proxy server.

To provide instructions for configuring the client browser to use the Content Accelerator as a proxy server

1. Select Management from the CacheOS home page.
2. Select an adapter from the drop down list.
3. Click Advanced Settings.
4. Enable the using a proxy radio button.
5. Click OK to close the Advanced Settings dialog box.
6. To save changes, click Apply.
7. Inform clients to read the browser configuration section of the Content Accelerator's homepage.

To provide instructions for configuring the client browser to use the Content Accelerator as a proxy server using the CLI

1. Open a terminal session with the Content Accelerator and type your Username and Password when prompted.
2. At the command prompt, type **enable** and type your Password when prompted.
3. At the command prompt, type **configure terminal**.
4. At the command prompt, type **interface *0***.

   The prompt changes to `(config interface 0)`.
5. At the command prompt, type **instructions proxy**.
6. Type **exit** to return to the `(config)` prompt.

Configuring the Browser to Use the Default PAC File

The Content Accelerator includes a default PAC file that can be used to auto-configure clients. The default PAC file contains the following instructions:

Note xxx.xxx.xxx.xxx is the device's IP address, and yyy is the device proxy port. You can easily change from the default PAC file to a custom PAC file without requiring the clients to change their configuration. See Switching PAC Files without Client Reconfiguration for additional information.

```
function FindProxyForURL(url, host) {
    if (url.substring(0, 5) == "http:") {
      return "PROXY xxx.xxx.xxx.xxx:yyy; DIRECT"; }
    else if (url.substring(0, 6) == "https:") {
      return "PROXY xxx.xxx.xxx.xxx:yyy; DIRECT";}
    else {
      return "DIRECT"; }
    }
```

To provide instructions for configuring the browser to use the default PAC file

1. Select Management from the CacheOS home page.
2. Select an adapter from the drop down list.
3. Click Advanced Settings.
4. Enable the using the proxy's default PAC file radio button.
5. Click OK to close the Advanced Settings dialog box.
6. To save changes, click Apply.
7. Inform clients to read the browser configuration section of the Content Accelerator's homepage.

To provide instructions for configuring the browser to use the default PAC file using the CLI

1. Open a terminal session with the Content Accelerator and type your Username and Password when prompted.
2. At the command prompt, type **enable** and type your Password when prompted.
3. At the command prompt, type **configure terminal**.
4. At the command prompt, type **interface *0***.
   The prompt changes to `(config interface 0)`.
5. At the command prompt, type **instructions default-pac**.
6. Type **exit** to return to the `(config)` prompt.

Configuring the Browser to Use a Custom PAC File

There are two ways to create a custom PAC file. The default PAC file can be customized and saved as a new file, or you can create a new custom PAC file. In either case it is important that the client instructions for configuring proxy settings contain the URL of the custom PAC file.

To provide instructions for configuring the browser to use a custom PAC file

1. Select Management from the CacheOS home page.
2. Select an adapter from the drop down list.
3. Click Advanced Settings.
4. Enable the using the PAC file at this URL radio button.
5. Type the fully qualified path, including filename, to the PAC file you want to use.
6. Click OK to close the Advanced Settings dialog box.
7. To save changes, click Apply.
8. Inform clients to read the browser configuration section of the Content Accelerator's homepage.

To provide instructions for configuring the browser to use a custom PAC file using the CLI

1. Open a terminal session with the Content Accelerator and type your Username and Password when prompted.
2. At the command prompt, type **enable** and type your Password when prompted.
3. At the command prompt, type **configure terminal**.
4. At the command prompt, type **interface *0***.
   The prompt changes to `(config interface 0)`.
5. At the command prompt, type **instructions central-pac *URL***.
   The URL is a fully qualified path, including filename, to the PAC file you want to use.
6. Type **exit** to return to the `(config)` prompt.

For more information on creating a custom PAC file, go to the Advanced Features section in the online documentation.

Switching PAC Files without Client Reconfiguration

If your clients are using the default PAC file, and it becomes necessary to use a custom PAC file, you can accomplish this transparently without any client notification or browser reconfiguration.

Switching PAC files without client reconfiguration

1. Select Management from the CacheOS home page.
2. Select an adapter from the drop down list.
3. Click Advanced Settings.
4. Enable the using the PAC file at this URL radio button.
5. Type the fully qualified path, including filename, to the PAC file you want to use.
6. Click OK to close the Advanced Settings dialog box.
7. To save changes, click Apply.

Switching PAC files without client reconfiguration using the CLI

1. Open a terminal session with the Content Accelerator and type your Username and Password when prompted.
2. At the command prompt, type **enable** and type your Password when prompted.
3. At the command prompt, type **configure terminal**.
4. At the command prompt, type **interface *0***.
   The prompt changes to `(config interface 0)`.
5. At the command prompt, type **instructions central-pac *URL***.
   The URL is a fully qualified path, including filename, to the PAC file you want to use.
6. Type **exit** to return to the `(config)` prompt.

When a browser requests the default PAC file, and the Content Accelerator is configured to use a PAC file at another location, the Content Accelerator transparently returns the PAC file at the specified URL to the browser.

Automatic Detection of Network Adapter Faults

CacheOS uses a set of heuristics to detect whether the network adapter(s) in a Content Accelerator is/are functioning properly. If an adapter is found to be faulty, CacheOS refrains from using it. When the cause for the fault is remedied, CacheOS detects the functioning adapter and uses it normally.

The heuristics used to determine whether an adapter is functioning properly are

1. Checking whether the link is active (i.e. a cable is connected and both sides are up).
2. Checking the ratio of error packets to good packets – both sent and received.
3. Checking if packets have been sent without any packets received.

If an adapter fault is detected, and it has an IP address assigned to it, a severe event is logged. The log entry is not made when an adapter does not have an IP address.

Using Multiple Default IP Gateways for Load Balancing

A key feature of CacheOS is the ability to distribute traffic originating at the cache through multiple IP gateways. Further, you can fine-tune how the traffic is distributed among gateways. This feature works with any routing protocol (e.g. static routes, RIP).

Note Load balancing through multiple IP gateways is independent from the per-interface load balancing CacheOS automatically does when more than one network interface is installed.

Using Multiple Default Gateways

CacheOS's choice of which gateway to use at a given time is determined by how the Administrator configures the assignment of preference groups to default gateways. Multiple gateways can be defined within the same preference group. A Content Accelerator can have from 1 to 10 preference groups.

Initially, all gateways in the lowest preference group are considered as the active gateways. If a gateway becomes unreachable, it is dropped from the active gateway list, but the remaining gateways within the group continue to be used until they all become unreachable, or until an unreachable gateway in a lower preference group becomes reachable again. If all gateways in the lowest preference group become unreachable, the gateways in the next lowest preference group become the active gateways.

In addition to a preference group, each gateway within a group can be assigned a relative weight value from 1 - 100. The weight value is used to determine how much bandwidth a gateway is given relative to the other gateways in the same group. For example, in a group with 2 gateways, assigning both gateways the same weight value, whether one or one hundred, results in the same traffic distribution pattern. In a group with 2 gateways, assigning one gateway a value of 10, and the other gateway a value of 20, results in the Content Accelerator sending approximately twice the traffic to the gateway with a weight value of 20.

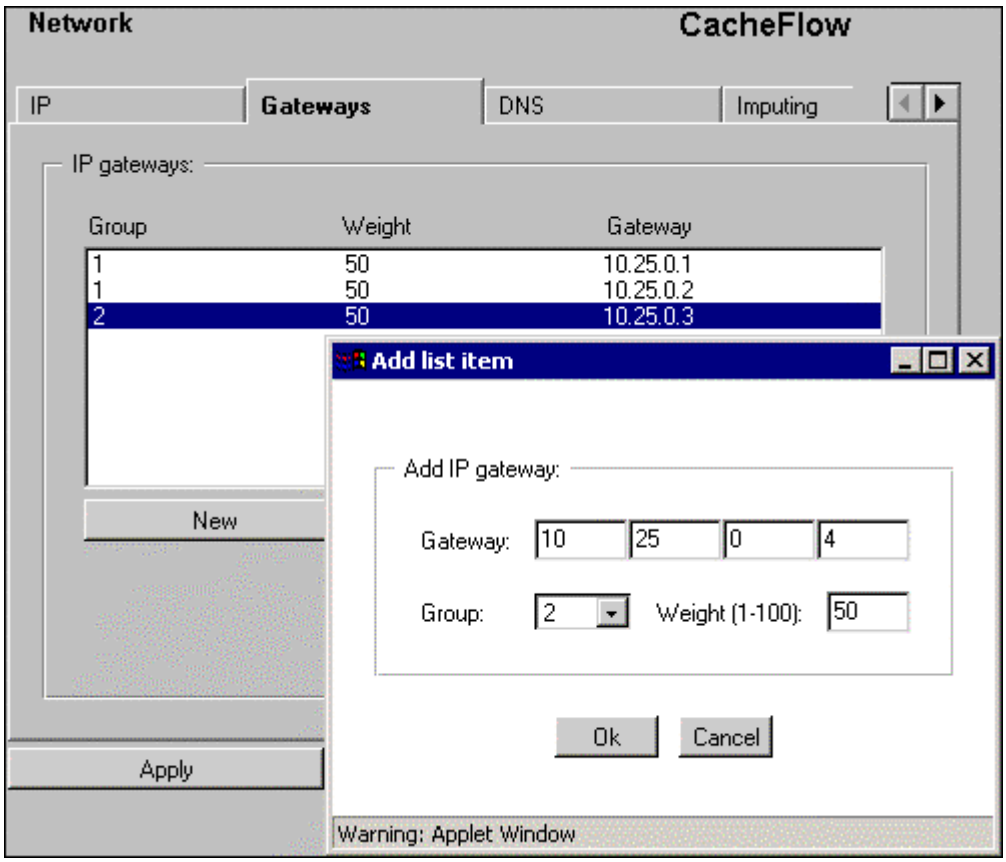


Figure 3-3 Adding an IP gateway

To configure load balancing

1. Select Management from the CacheOS home page.
2. Select the Gateways tab.
3. Click New.
4. Enter the IP address, group, and weight for the gateway.
5. Click OK.
6. Repeat steps 3 to 5 until IP addresses, groups, and weights have been defined for all of your IP gateways.
7. Click Apply to save changes.

To configure load balancing using the CLI

1. Open a terminal session with the Content Accelerator and type your Username and Password when prompted.
2. At the command prompt, type **enable** and type your Password when prompted.
3. At the command prompt, type **configure terminal**.
4. At the command prompt, type **ip-default-gateway** ***255.255.0.0 1 50***.

The first value is the IP address of the gateway, the second value is the preference group, and the third value is the relative weighting for this gateway.

5. Repeat step 4 until all IP addresses, groups, and weights of your IP gateways have been defined.

Specifying DNS Servers

A single primary DNS server is entered using the Setup console when initial system setup is performed. You can also define single or multiple additional primary and alternate DNS servers after initial setup is complete.

If you have more than 1 DNS server defined, the method CacheOS uses to determine when to failover to the next DNS server, or return an error to the client, is as follows:

The first primary DNS server is always tried first. If CacheOS receives a response and answer, no attempt is made to contact any other DNS server for the request.

If the response from the first primary DNS server is "Non existent domain", CacheOS attempts to connect through the first alternate DNS server, if it is defined. If no alternate DNS server has been defined, an error "UNRESOLVED_HOSTNAME" is returned. If the first alternate DNS server also responds with "Non existent domain", an error "UNRESOLVED_HOSTNAME" is returned to the client, and no other connections are attempted for the request.

If the response from the first primary DNS server is no error and no answer, CacheOS attempts to connect through the first alternate DNS server, if it is defined. If no alternate DNS server has been defined, an error "UNRESOLVED_HOSTNAME" is returned. If the first alternate DNS server also responds with no error and no answer,, an error "UNRESOLVED_HOSTNAME" is returned to the client, and no other connections are attempted for the request.

If the response from the first primary DNS server is anything other than case2 and case3, CacheOS attempts to connect through the list of primary in the order defined in the Web or CLI Management console until a connection is made, or the end of the DNS server list is reached. If the end of the DNS server list is reached without a connection, an error is returned to the client, and no other connections are attempted for the request.

Split DNS Support

Customers with split DNS server configuration may choose to populate an Alternate DNS server list as well as the Primary DNS server list. This scheme is typically employed in environments that maintain private internal DNS servers containing organizational DNS naming information for intranet communications, and external DNS servers containing DNS naming information for the Internet. In CacheOS, the internal DNS servers would normally be placed in the Primary list, while external DNS servers would populate the Alternate list.

To enter an additional primary DNS server

1. Select Management from the CacheOS home page.
2. Select the DNS tab.
3. Click New.
4. Enter the IP address of the DNS server and click OK.
5. Click Apply to save changes.

To enter an additional primary DNS server using the CLI

1. Open a terminal session with the Content Accelerator and type your Username and Password when prompted.
2. At the command prompt, type **enable** and type your Password when prompted.
3. At the command prompt, type **configure terminal**.
4. At the command prompt, type **dns server *10.25.0.1***.
5. Type **exit** to return to the `(config)` prompt.

To enter an alternate DNS server

1. Select Management from the CacheOS home page.
2. Select the DNS tab.
3. Select Alternate DNS in the drop down list.
4. Click New.
5. Enter the IP address of the DNS server and click OK.
6. Click Apply to save changes.

To enter an alternate DNS server using the CLI

1. Open a terminal session with the Content Accelerator and type your Username and Password when prompted.
2. At the command prompt, type **enable** and type your Password when prompted.
3. At the command prompt, type **configure terminal**.
4. At the command prompt, type **dns alternate *10.25.0.2***.
5. Repeat step 4 until alternate DNS servers have been defined.
5. Type **exit** to return to the `(config)` prompt.

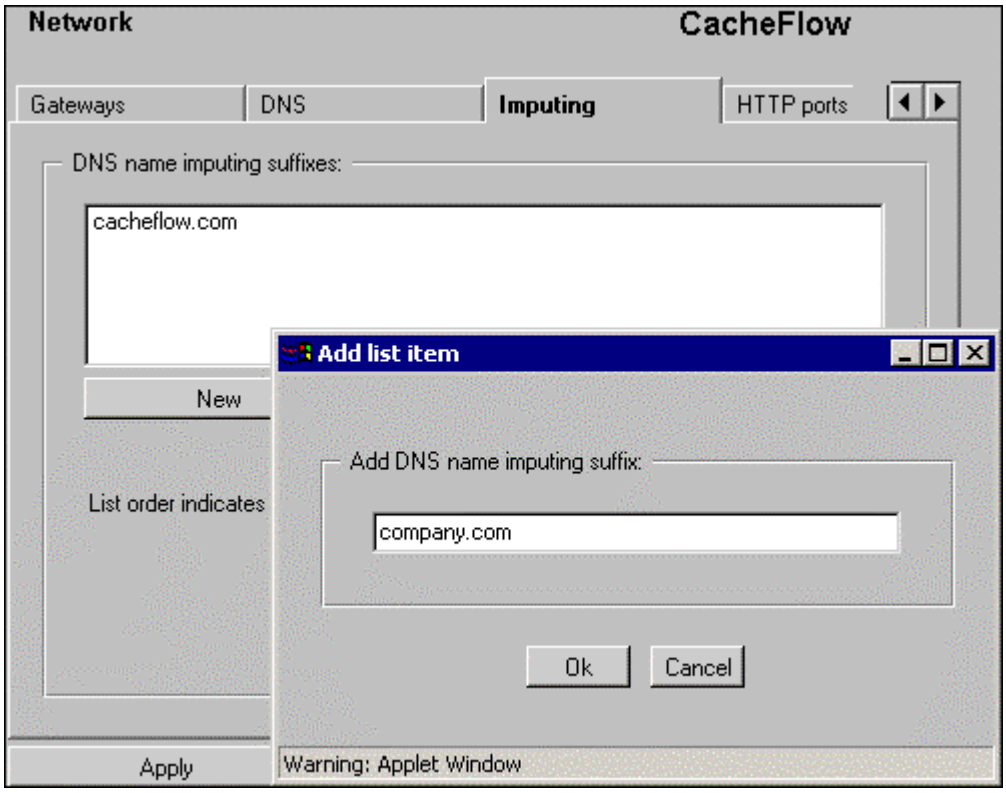


Figure 3-4 Entering DNS Servers

Changing the Order of DNS Servers

DNS servers are used in the order displayed. You can organize the list of servers so the preferred servers appear at the top of the list.

To change the order in which DNS servers are accessed

1. Select Management from the CacheOS home page.
2. Select the DNS server to promote or demote.
3. Click Promote or Demote as appropriate.
4. Click Apply to save changes.

To change the order in which DNS servers are accessed using the CLI

Not available in the CLI. Use the Web interface.

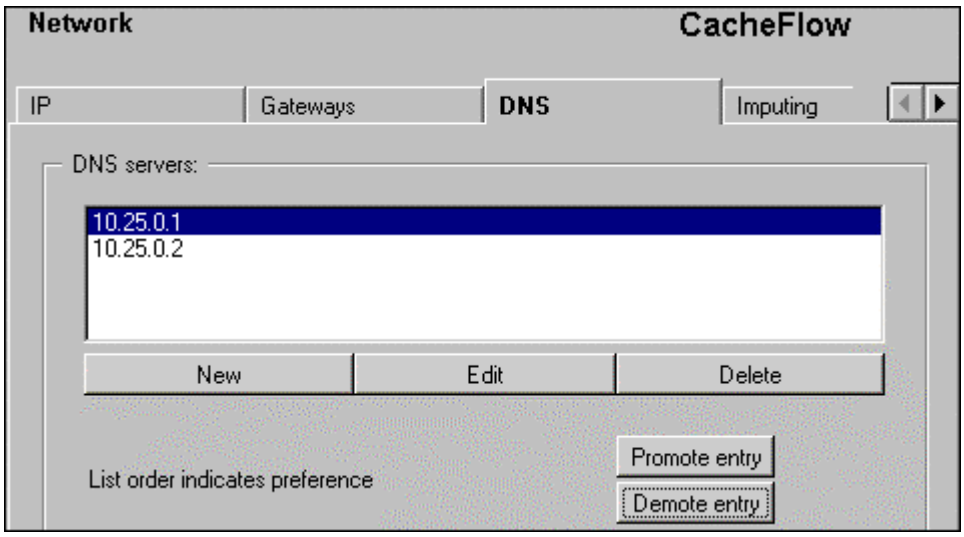


Figure 3-5 Changing the order in which DNS servers are accessed

Using Name Imputing

Name imputing allows CacheOS to resolve host names based on a partial name specification. When CacheOS submits a host name to the DNS server, the DNS server resolves the name to an IP address. If the host name cannot be resolved, CacheOS adds the first entry in the name-imputing list to the end of the host name and resubmits it to the DNS server. CacheOS tries each entry in the name-imputing list until the name is resolved, or the end of the list is reached. If the end of the list is reached and the name is not resolved, CacheOS returns a DNS failure.

For example, if the name imputing list contains the entries `company.com` and `com`, and a user submits a host name of `eedept`, CacheOS will resolve the host names in the following order.

```
eedept
eedept.company.com
eedept.com
```

To add names to the imputing list

1. Select Management from the CacheOS home page.
2. Select the Imputing tab.
3. Click New to add a new name to the imputing list.
4. Enter the name and click OK.
5. Click Apply to save changes.

To add names to the imputing list using the CLI

1. Open a terminal session with the Content Accelerator and type your Username and Password when prompted.
2. At the command prompt, type **enable** and type your Password when prompted.
3. At the command prompt, type **configure terminal**.

4. At the command prompt, type **dns imputing *company.com*** to set an imputing suffix.
5. Repeat step 4 until all imputing suffixes have been entered.

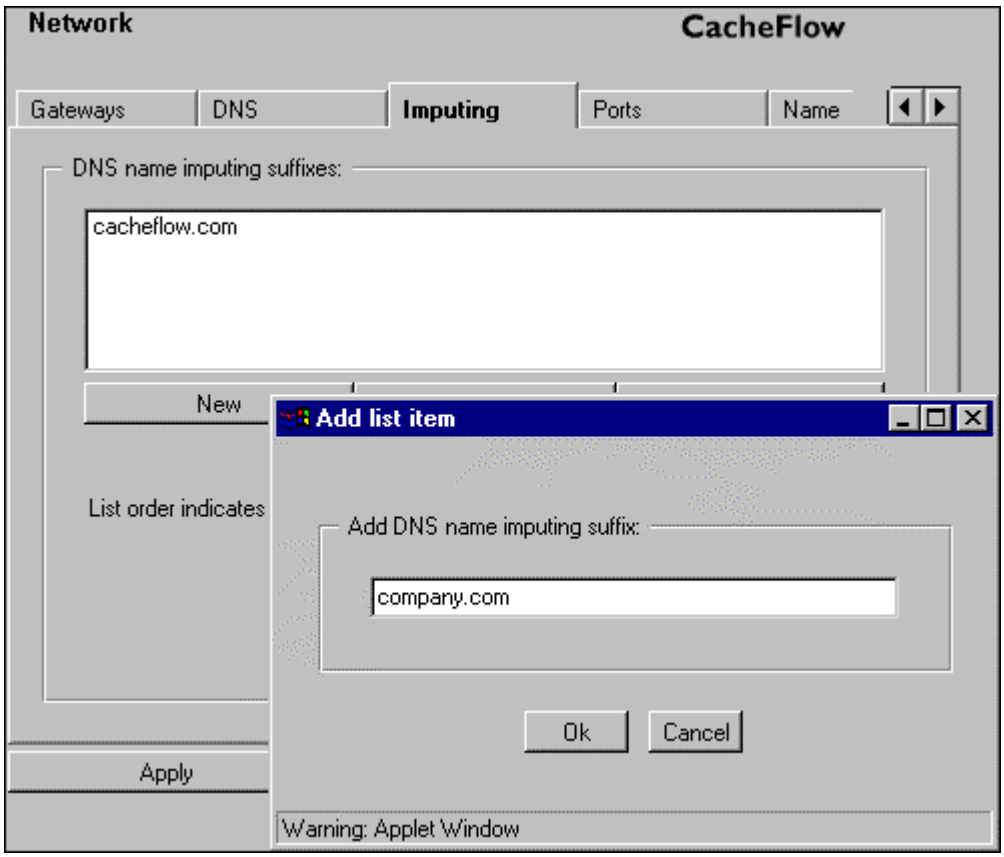


Figure 3-6 Adding a DNS name imputing suffix

Changing the Order of DNS Name Imputing Suffixes

Imputing suffixes are used in the order displayed. You can organize the list of suffixes so the preferred suffix appears at the top of the list.

To change the order in which imputing suffixes are accessed

1. Select Management from the CacheOS home page.
2. Select the Imputing tab.
3. Select the imputing suffix to promote or demote.
4. Click Promote or Demote as appropriate.
5. Click Apply to save changes.

To change the order in which imputing suffixes are accessed using the CLI

Not available in the CLI. Use the Web interface.

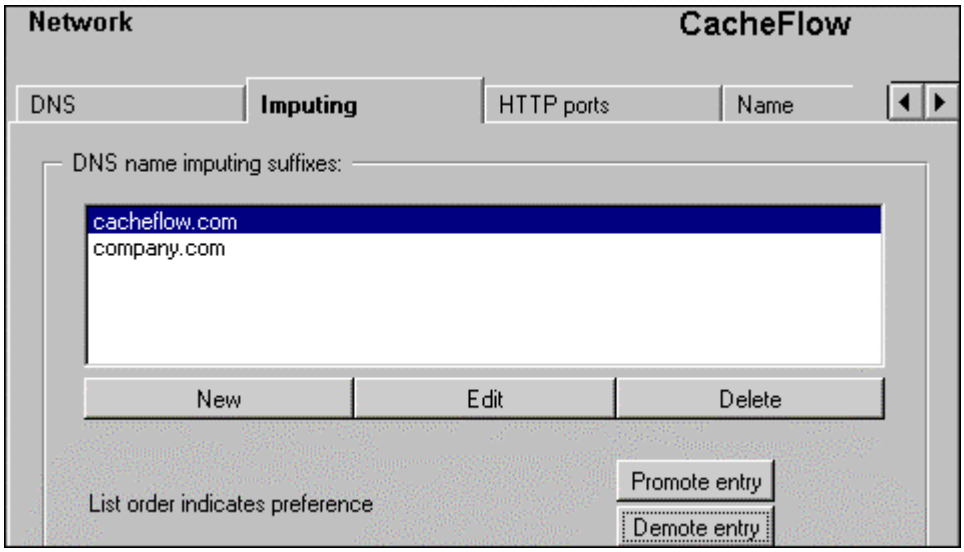


Figure 3-7 Changing the order in which DNS servers are accessed

Configuring HTTP Ports

You can set the IP ports CacheOS uses to listen for HTTP requests and for accessing the Content Accelerator Web interface. The default port for HTTP requests is 8080. The default port for the Web interface is 8081.

To change IP ports

1. Select Management from the CacheOS home page.
2. Select the HTTP ports tab.
3. Enter the port to use for HTTP requests.
4. Enter the port to use for the Management console.
5. Click Apply to save changes.

To change IP ports using the CLI

1. Open a terminal session with the Content Accelerator and type your Username and Password when prompted.
2. At the command prompt, type **enable** and type your Password when prompted.
3. At the command prompt, type **configure terminal**.
4. At the command prompt, type **http-proxy-port *8080*** to set the HTTP proxy port.
5. At the command prompt, type **management-port *8081*** to set the management port.
6. Type **exit** to return to the `(config)` prompt.

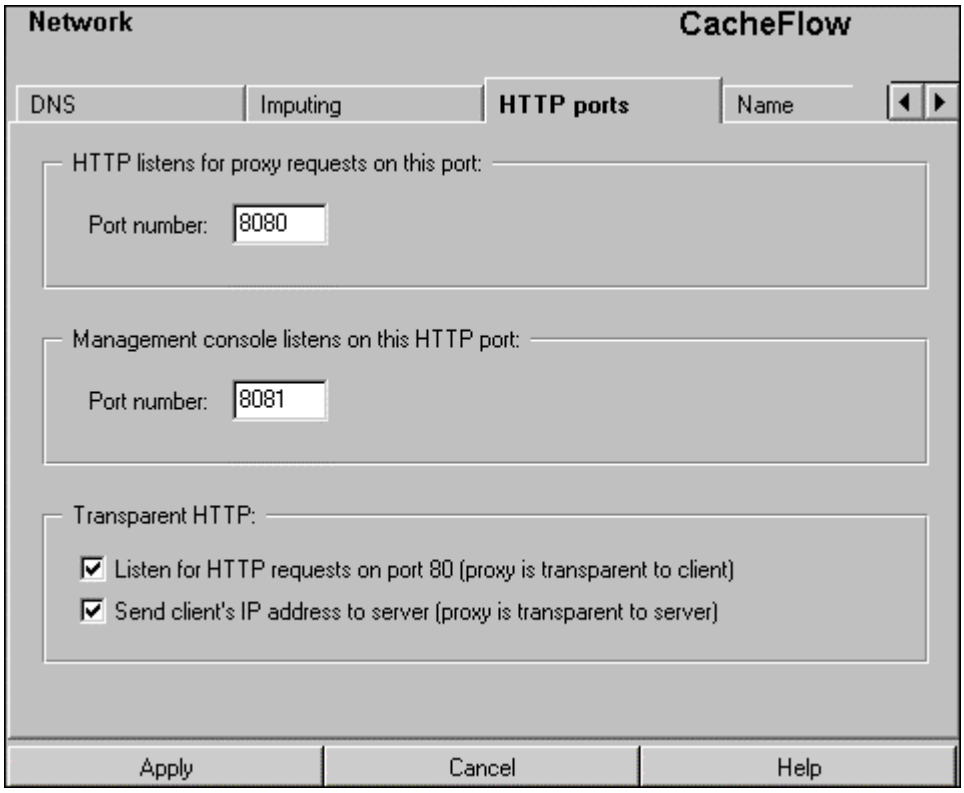


Figure 3-8 Configuring IP Ports

See the section *Tracking Client IP Adresses Using Server-Side Tranparency* for detailed information about the Send client's IP address to server setting.

Relationship Between Proxy Port Number and Transparent vs. Explicit Proxying

The relationship between the HTTP proxy port number (0 or 8080), and whether transparent proxying on port 80 is enabled or disabled, determines how explicit and transparent proxy requests are handled by CacheOS. The following table delineates CacheOS's behavior in these cases.

	Transparent Proxying Enabled	Transparent Proxying Disabled
HTTP Proxy Port Set to 0	Only transparent proxy requests to port 80 are accepted.	Both transparent proxy requests to port 80 and explicit proxy requests to the proxy port are rejected.
HTTP Proxy Port Set to 8080	Transparent and explicit-proxy requests to port 80, and explicit proxy requests to the proxy port are accepted.	Only explicit proxy requests to the proxy port and explicit requests to port 80 are accepted.

Setting the Content Accelerator Name

You can assign a name to a Content Accelerator. Any descriptive name that helps identify the system will do.

To set the Content Accelerator name

1. Select Management from the CacheOS home page.
2. Select the Name tab.
3. Enter the Content Accelerator name in the name field.
4. Click Apply to save changes.

To set the Content Accelerator name using the CLI

1. Open a terminal session with the Content Accelerator and type your Username and Password when prompted.
2. At the command prompt, type **enable** and type your Password when prompted.
3. At the command prompt, type **configure terminal**.
4. At the command prompt, type **hostname** ***name*** to set the Content Accelerator name.
5. Type **exit** to return to the `(config)` prompt.

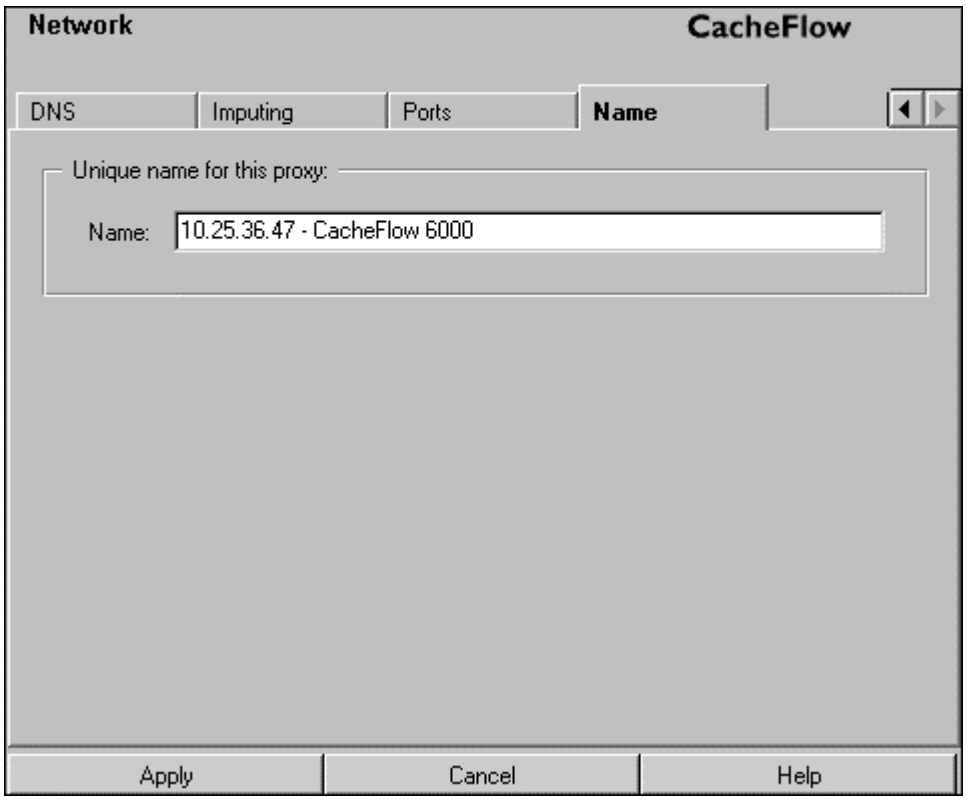


Figure 3-9 Setting the Content Accelerator Name

Chapter 4 - Content Filtering

Content Filtering gives you the option of controlling the type of content retrieved by the cache. CacheOS supports content filtering lists from SmartFilter™ and Websense™.

SmartFilter lists are provided by Secure Computing, Inc.

Websense lists are provided by Net Partners, Inc.

There are four main steps to implement content filtering

1. Open an account with the content filtering vendor you selected.
2. Enable content filtering in CacheOS, entering the authorization information provided by your content filtering vendor.
3. Configure an automatic filter update schedule.
4. Select the categories of information to be blocked.

Enabling Content Filtering

CacheOS ships with content filtering disabled. To start using content filtering, the Cache administrator must decide which content filtering service to use and contact the related vendor for license and authorization information.

Once the required license and related information is obtained from the content filtering vendor, you can use the CacheOS CLI or Web Interface to activate and customize content filtering.

Warning Once you have enabled CacheFlow content filter services, or after using the category block or category unblock commands, clear the CacheFlow system cache. This eliminates the possibility that a blocked URL which was previously accessed and cached, might be served from cache.

Enabling content filtering using SmartFilter

1. Select Management from the CacheOS home page.
2. Select the Filtering applet.
3. Select the Vendor tab.
4. In the Filter vendor box, enable the SmartFilter radio button.
5. Click OK in the configuration alert dialog.

   The tabs appropriate to SmartFilter settings are displayed.
6. To enable automatic downloading, in the Automatic Download box check Download new filter, select a download time in the drop-down list, and check the days on which you want the filter list updated.

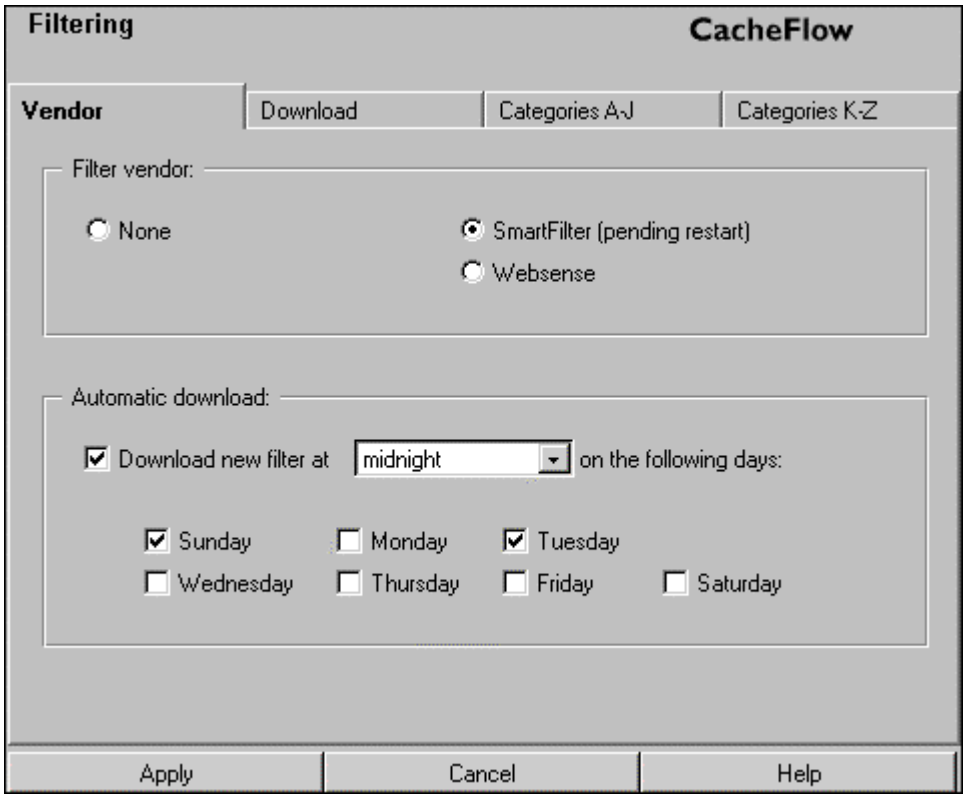


Figure 4-1 Selecting a vendor and setting a schedule

7. Select the Download Tab and fill in the fields with the information provided by SmartFilter.

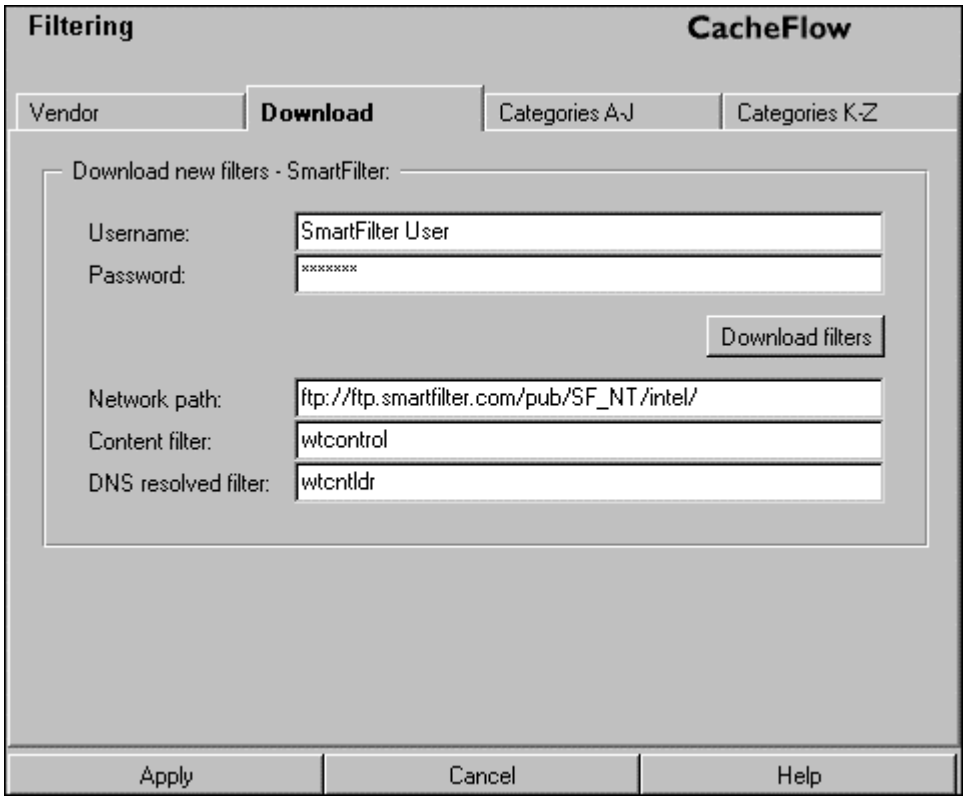


Figure 4-2 Configuring User ID and filter download settings

8. Click Download filters.
9. Select the Categories tabs and check the categories of information to be filtered.

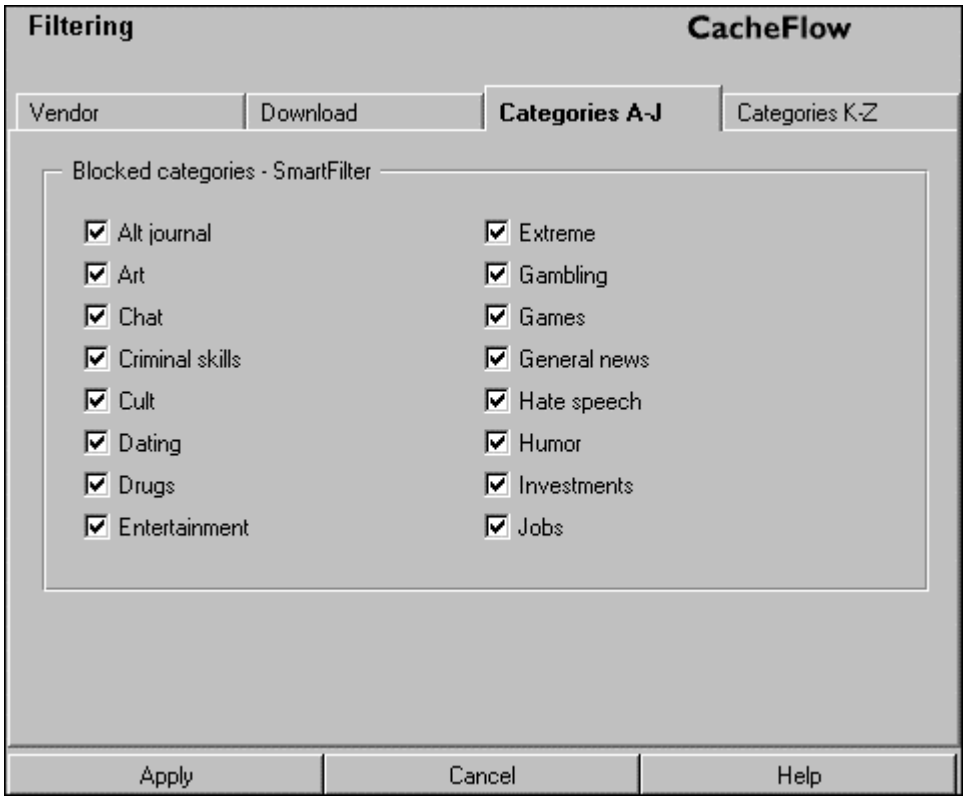


Figure 4-3 Selecting categories of information to block

10. Click Apply.
11. Restart the Content Accelerator.

Enabling SmartFilter using the CLI

1. Open a terminal session with the CacheFlow device and type your Username and Password when prompted.
2. At the Command prompt, type **Enable** and type your Password when prompted.
3. At the Command prompt, type **configure terminal** to enter terminal configuration mode.
4. At the `(config)` prompt, type **content-filter** to enter content-filter configuration mode.
   The prompt changes to `(config content-filter)`.
5. Type **disable** to turn off content filtering.
   Content filtering must be disabled before selecting a provider.
6. Type **select-provider smartfilter** to select SmartFilter as your provider.
7. Type **enable** to re-enable content filtering.
8. Type **smartfilter** to enter SmartFilter configuration mode.
   The prompt changes to `(config smartfilter)`
9. Type the following commands to configure SmartFilter
   download control-file wtcontrol

download DNR-control-file wtcntldr

download username *username*

download password *password*

download path ftp://ftp.smartfilter.com/pub/SF_NT/intel

SmartFilter is supplied with default names for the Control-file, DNR-Control-file and Path. You do not need to configure these parameters unless you are instructed to do so by SmartFilter.

10. Type **download get-now** to get the current filter list.
11. Restart the Content Accelerator.

Enabling content filtering using Websense

1. Select Management from the CacheOS home page.
2. Select the Filtering applet.
3. Select the Vendor tab.
4. In the Filter vendor box, enable the Websense radio button.
5. Click OK in the configuration alert dialog.

   The tabs appropriate to Websense settings are displayed.
6. To enable automatic downloading, in the Automatic Download box check Download new filter, select a download time in the drop-down list, and check the days on which you want the filter list updated.

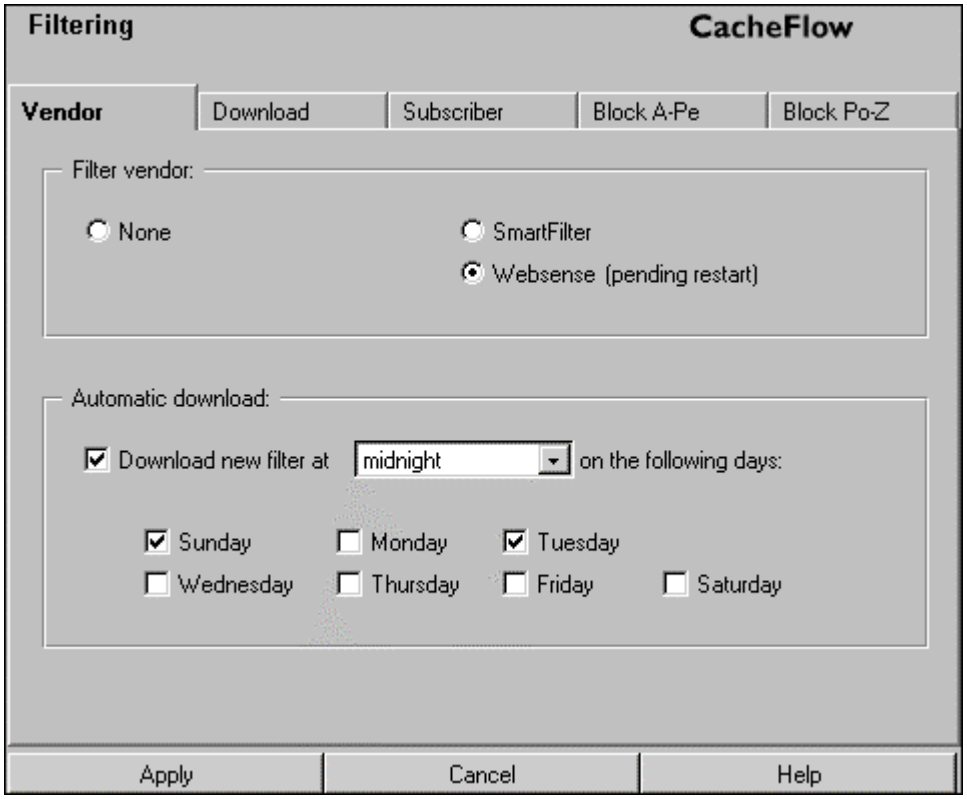


Figure 4-4 Selecting a vendor and setting a schedule

7. Select the Subscriber tab and fill in the fields with the information provided by Websense.

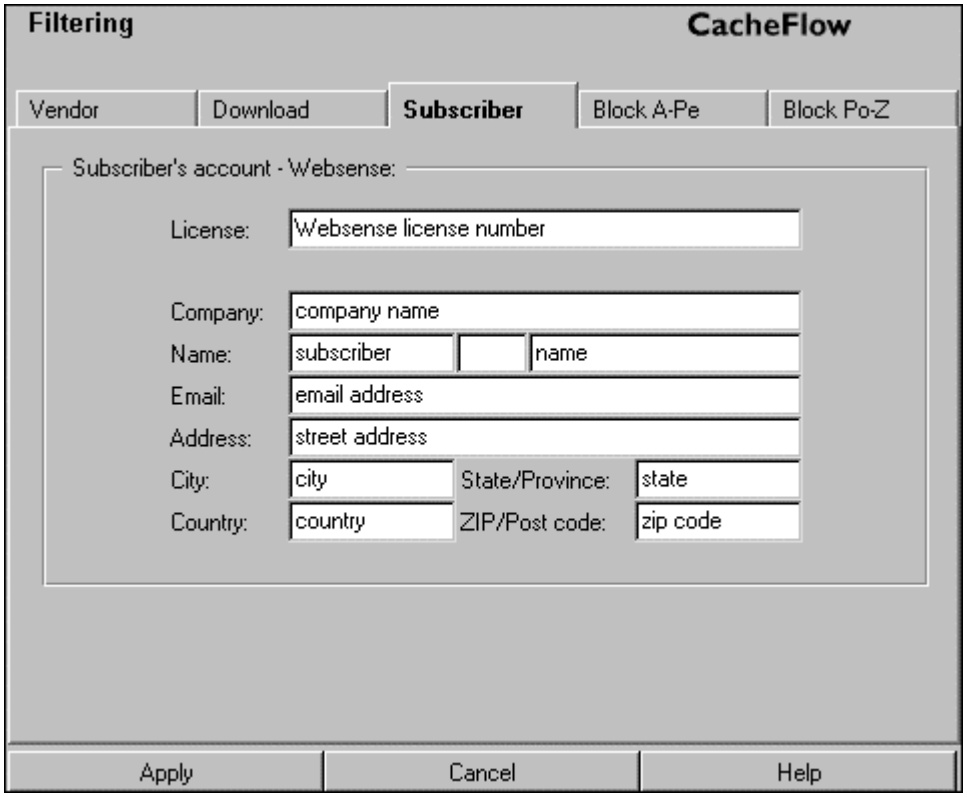


Figure 4-5 Entering subscriber information

8. Select the Download tab, type the Username and Password provided by Websense, and click Download filter.

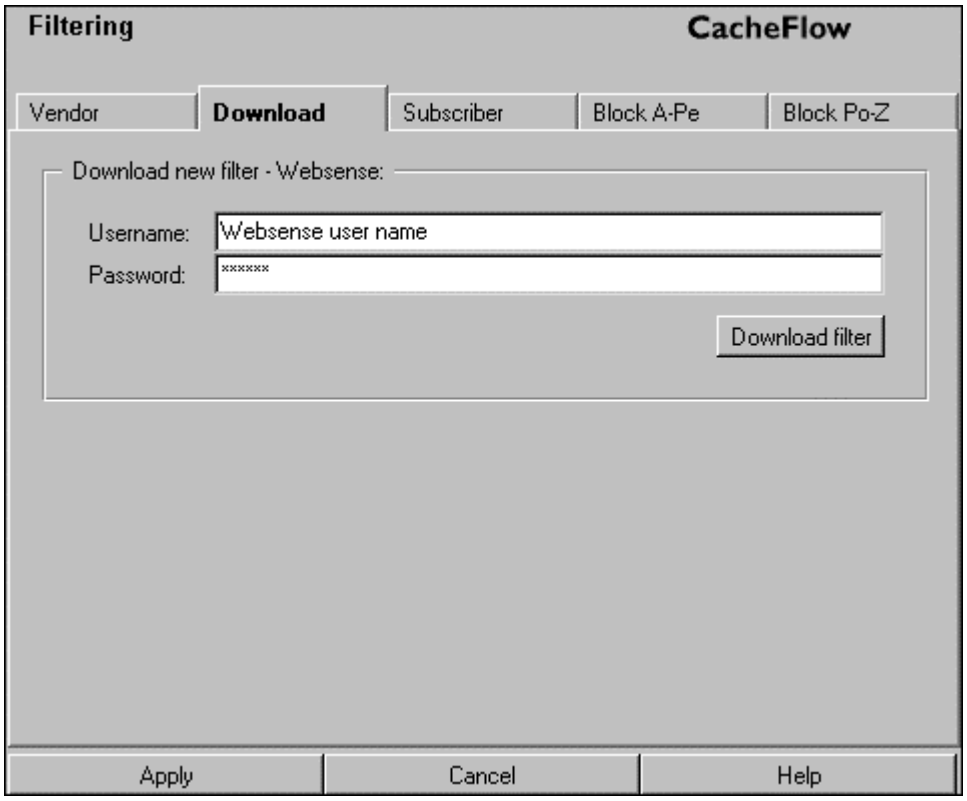


Figure 4-6 Downloading a filter list

9. Select the Block tabs and check the categories of information to be filtered.
10. Restart the Content Accelerator.

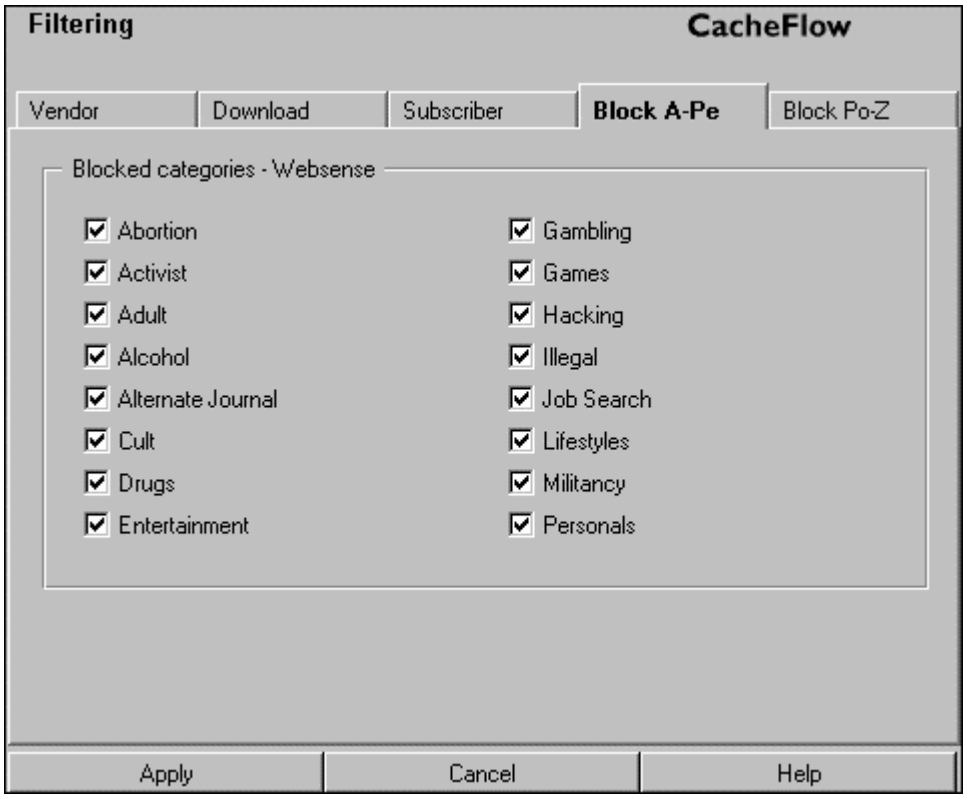


Figure 4-7 Selecting categories of information to block

Enabling Websense using the CLI

1. Open a terminal session with the CacheFlow device and type your Username and Password when prompted.
2. At the Command prompt, type **enable** and type your Password when prompted.
3. At the Command prompt, type **configure terminal** to enter terminal configuration mode.
4. At the `(config)` prompt, type **content-filter** to enter content-filter configuration mode.
   The prompt changes to `(config content-filter)`.
5. Type **disable** to turn off content filtering.
   Content filtering must be disabled before selecting a provider.
6. Type **select-provider websense** to select Websense as your provider.
7. Type **enable** to re-enable content filtering.
8. Type **websense** to enter Websense configuration mode.
   The prompt changes to `(config websense)`
9. Type **download license-key *1234567890***, substituting your key for the example key shown here.
10. Type **download get-now** to get the current filter list.
11. Restart the Content Accelerator.

Changing the WebSense Server Address

By default, CacheOS automatically uses the WebSense Server closest to the Content Accelerator. In some situations, CacheFlow or WebSense support might direct you to use a specific server. This capability is available using the CLI, but not the Web interface.

To change the WebSense server address

1. Open a terminal session with the CacheFlow device and type your Username and Password when prompted.
2. At the Command prompt, type **enable** and type your Password when prompted.
3. At the Command prompt, type **configure terminal** to enter terminal configuration mode.
4. At the `(config)` prompt, type **content-filter** to enter content-filter configuration mode.

   The prompt changes to `(config content-filter)`.
5. Type **websense** to enter Websense configuration mode.

   The prompt changes to `(config websense)`
6. Type **download server** ***IP address or name of server***.
7. Type **exit** to return to return to the `(config)` prompt.

Working with Content Filtering

Once you have enabled and configured the content filtering services, you can use a variety of CLI commands to selectively or globally control the content served to your CacheFlow user base. A few examples are presented here. Refer to the Command Reference chapter for a complete list of commands and their functions.

Blocking and Unblocking Categories

You can use the `category` command to block and unblock categories. To block all content groups, use the `category block all` command. The `category unblock all` command will unblock everything. For example:

```
(config smartfilter)category block all
(config smartfilter)category unblock all
```

You can also block or unblock individual content categories by replacing all, in the `category block` or `category unblock` commands with the specific category name.

```
(config smartfilter)category block gambling
(config smartfilter)category unblock gambling
```

Note SmartFilter and Websense use different category names. Use the category names applicable to your vendor.

Viewing Content Filter Status

You can use the **show content-filter status** command to show current filter settings and content categories currently blocked.

Overriding Blocked Categories

You can use the CacheFlow filter file to override content filtering for a specific URL. This enables the Cache to serve the specified URL even if it is disallowed by the content filtering file. For example, including the following line in your CacheFlow filter file overrides content filtering for the specified URL.

http://www.company.com **content_filter_override=yes**

Scheduling Automatic Downloads

You can configure and enable periodic automatic downloads for content filter lists. In this example the `download day-of-week all` command specifies automatic downloads for every day of the week, the `download time-of-day 20` sets the download time for 8:00pm, and the `download enable-auto` enables automatic downloads.

```
(config smartfilter)download day-of-week all
(config smartfilter)download time-of-day 20
(config smartfilter)download enable-auto
```

Chapter 5 - Setting the System Time

To manage objects in the cache, a Content Accelerator must know the current UTC (Universal Time Coordinates) time. By default, the CacheOS attempts to connect to an NTP (Network Time Protocol) server to acquire the UTC time. CacheOS includes a list of NTP servers available on the Internet, and attempts to connect to them in the order they appear in the NTP server list on the NTP tab. If the Content Accelerator cannot access any of the listed NTP servers, the UTC time must be set manually.

To acquire UTC time from an NTP server

1. Select Management from the CacheOS home page.
2. Select the Time applet.
3. Verify that the Enable NTP checkbox is enabled.
4. To set your local time, select a time zone from the Timezone drop-down list.
   Once the local time zone is selected, event and access logs record the local time instead of GMT.
5. Click Acquire UTC time.
8. Click Apply to save changes.

To acquire UTC time from an NTP server using the CLI

1. Open a terminal session with the CacheFlow device and type your Username and Password when prompted.
2. At the command prompt, type **enable** and type your Password when prompted.
3. At the command prompt, type **acquire-utc**.
   If NTP is disabled, an error is displayed.

To set UTC time manually

1. Select Management from the CacheOS home page.
2. Select the Time applet.
3. Clear the Enable NTP checkbox.
   The UTC time and date fields become editable when NTP is disabled.
4. To set your local time, select a time zone from the Timezone drop-down list.
   Once the local time zone is selected, event and access logs record the local time instead of GMT.
5. Click Pause to stop the system clock.
6. Enter the current UTC time and date in the UTC time and date fields.
7. Click Resume to start the system clock.
8. Click Apply to save changes.

To set UTC time manually using the CLI

Not available in the CLI. Use the Web interface.

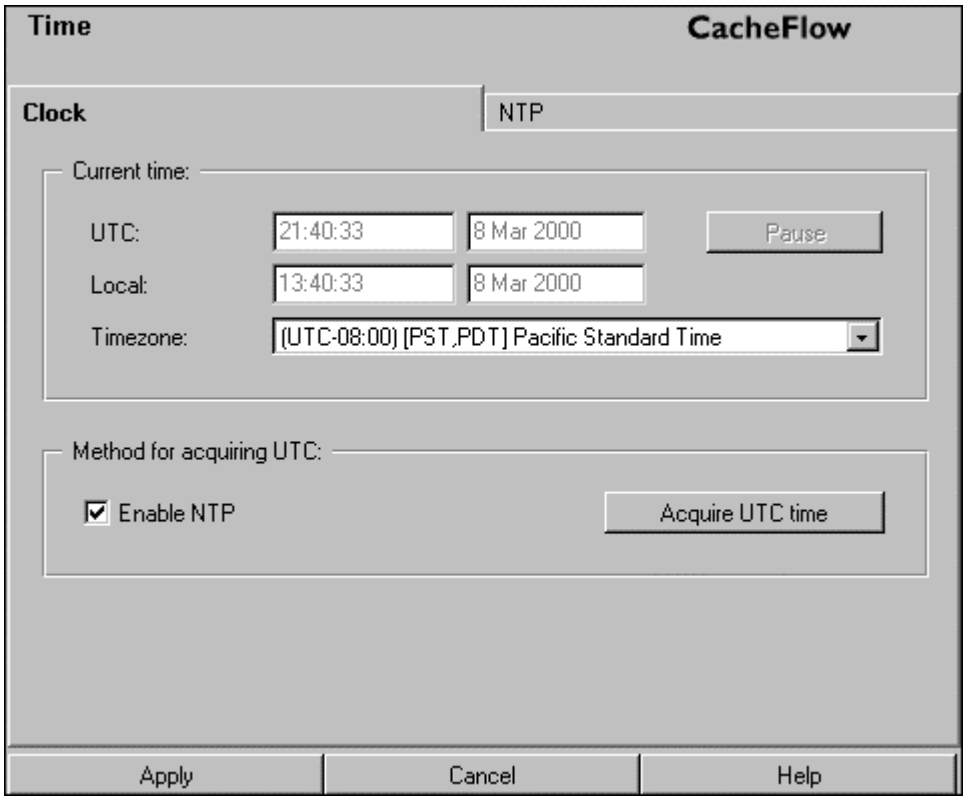


Figure 5-1 Setting the system time

Configuring the NTP Server List

You can add, delete, edit, and reorder the list of NTP servers CacheOS uses for acquiring the time.

To add an NTP server

1. Select Management from the CacheOS home page.
2. Select the Time applet.
2. Select the NTP tab.
3. Click New to add a new server to the list.
4. Enter either the domain name or IP address of the NTP server and click OK.
5. Click Apply to save changes.

To add an NTP server using the CLI

Not available in the CLI. Use the Web interface.

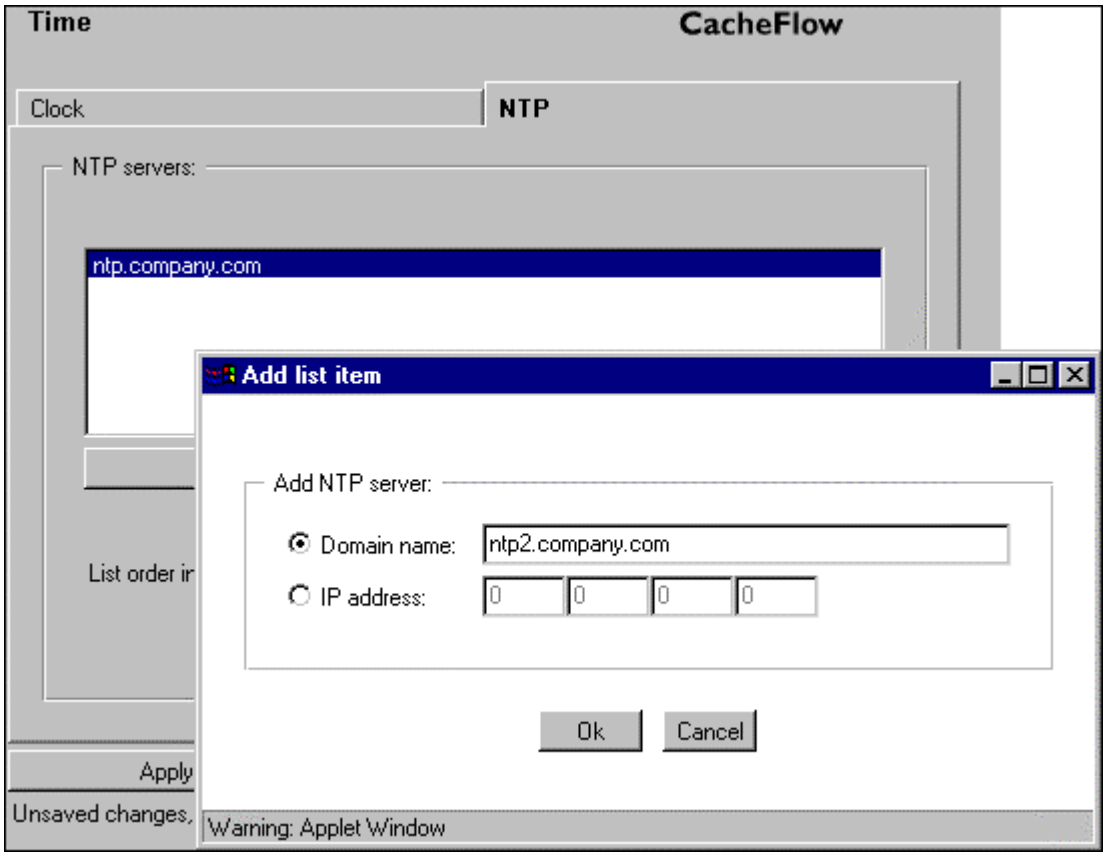


Figure 5-2 Adding an NTP server

Changing the Order of NTP Server Access

NTP servers are accessed in the order displayed. You can organize the list of servers so the preferred server appears at the top of the list.

To change the order in which NTP servers are accessed

1. Select Management from the CacheOS home page.
2. Select the Time applet.
3. Select the NTP tab.
4. Select the NTP server to promote or demote.
5. Click Promote or Demote as appropriate.
6. Click Apply to save changes.

To change the order in which NTP servers are accessed using the CLI

Not available in the CLI. Use the Web interface.

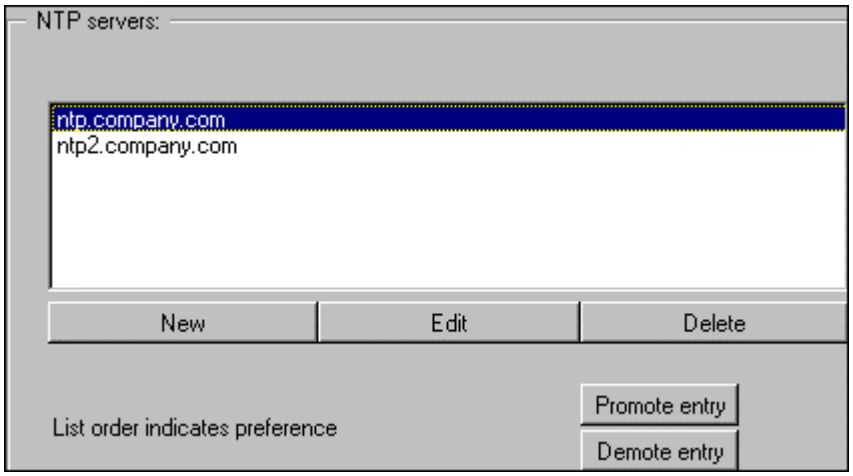


Figure 5-3 Promoting and demoting NTP servers

Chapter 6 - Configuring Caching Options

When the CacheFlow device retrieves an object from the Web and returns it to the client, the object is considered fresh: The CacheOS knows it is fresh because it just retrieved the object from the source. The goal of the Web cache is to keep fresh as many of the objects in the cache as possible, so that when the objects are requested, CacheOS can deliver them to the client without having to retrieve them from the source.

To keep objects fresh, CacheOS uses a variety of techniques to learn the update patterns of the objects. It then refreshes the objects in the background before a client requests them. You can define how hard the Web cache works to ensure freshness. If you set the desired freshness to 99%, the Web cache will work harder verifying objects are fresh than if you set the desired freshness to 95%. The higher you set the desired freshness, the higher the percentage of objects will be fresh in the cache upon request. Object freshness must be considered along with bandwidth usage. Generally, as the desired freshness setting is increased, so does use of network bandwidth by the Content Accelerator.

To set the desired freshness

1. Select Management from the CacheOS home page.
2. Select the Caching applet.
3. Enter the desired freshness in the Desired freshness field, up to 100%.
4. If you want to limit the network bandwidth the Content Accelerator can use to maintain freshness, select the radio button next to the Limit refresh bandwidth field and enter the bandwidth limit.

   You should let CacheOS manage refresh bandwidth unless you are experiencing problems on your network.
5. Click Apply to save changes.

To set the desired freshness and bandwidth utilization using the CLI

1. Open a terminal session with the CacheFlow device and type your Username and Password when prompted.
2. At the command prompt, type **enable** and type your Password when prompted.
3. At the command prompt, type **configure terminal** to enter terminal configuration mode.
4. At the `(config)` prompt, type **caching** to enter cache configuration mode.

   The prompt changes to `(config caching)`.
5. At the `(config caching)` prompt, type **refresh automatic** to let CacheOS manage cache bandwidth usage, or type **refresh no automatic** to specify a custom bandwidth usage value.

   If you typed **refresh no automatic** go on to step 6. Otherwise skip to step 7.
6. At the `(config caching)` prompt, type **refresh bandwidth *200*** to specify bandwidth usage in Kbps.
7. At the `(config caching)` prompt, type **refresh desired-freshness *95*** to specify the desired freshness percentage of HTTP objects.
8. Type **exit** to return to configuration mode.

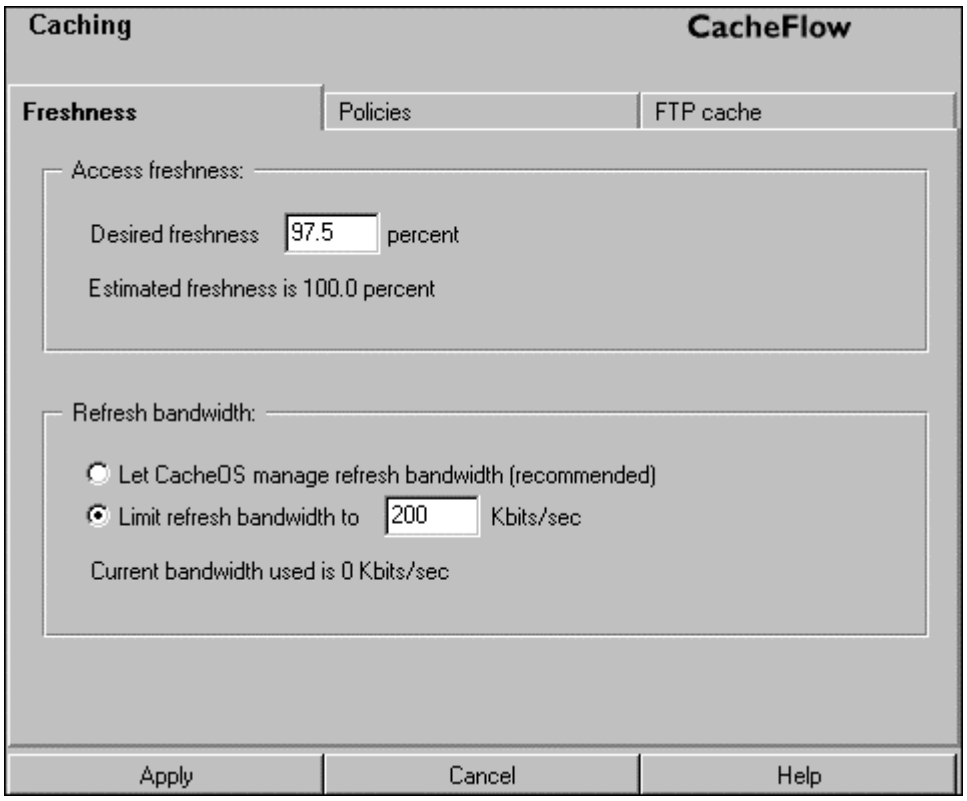


Figure 6-1 Setting cache freshness options

Setting Network Bandwidth Utilization

You can configure the amount of bandwidth CacheOS uses for refresh operations. CacheOS tracks the amount of time it takes to retrieve an object and calculates the bandwidth utilization based on past performance. You set the bandwidth utilization in Kbits per second. For example, if your network bandwidth to the outside world is 256 Kbits per second, setting the bandwidth utilization to 25 Kbits per second will restrict CacheOS to approximately 10% of the network bandwidth for background refresh operations. If you have multiple Web caches, you must take into account each cache when setting the bandwidth utilization. In the example above, if you have two CacheFlow Web caches you can set the bandwidth utilization to 12 Kbits for each cache to maintain 10% utilization across both caches.

Setting HTTP Cache Refresh Policies

When an HTTP object in the cache expires, it is placed in a refresh list. CacheOS processes the refresh list in the background, when it is not serving requests. Refresh policies define how CacheOS handles the refresh process.

To set cache refresh policies

1. Select Management from the CacheOS home page.
2. Select the Caching applet.
3. Select the Policies tab.
4. Select the refresh policies you want to use for HTTP objects.
5. Click Apply to save changes.

To set cache refresh policies using the CLI

1. Open a terminal session with the CacheFlow device and type your Username and Password when prompted.
2. At the command prompt, type **enable** and type your Password when prompted.
3. At the command prompt, type **configure terminal** to enter terminal configuration mode.
4. At the `(config)` prompt, type **caching** to enter cache configuration mode.

   The prompt changes to `(config caching)`.
5. At the `(config caching)` prompt, type **always-verify-source** or **no always-verify-source** depending on whether you want CacheOS to always check objects on the source before serving them to the client.
6. At the `(config caching)` prompt, type **max-cache-size *50*** to set the largest HTTP object to cache in megabytes.
7. At the `(config caching)` prompt, type **negative-response *10*** to set the amount of time negative responses are cached in minutes.
8. Type **exit** to return to configuration mode.

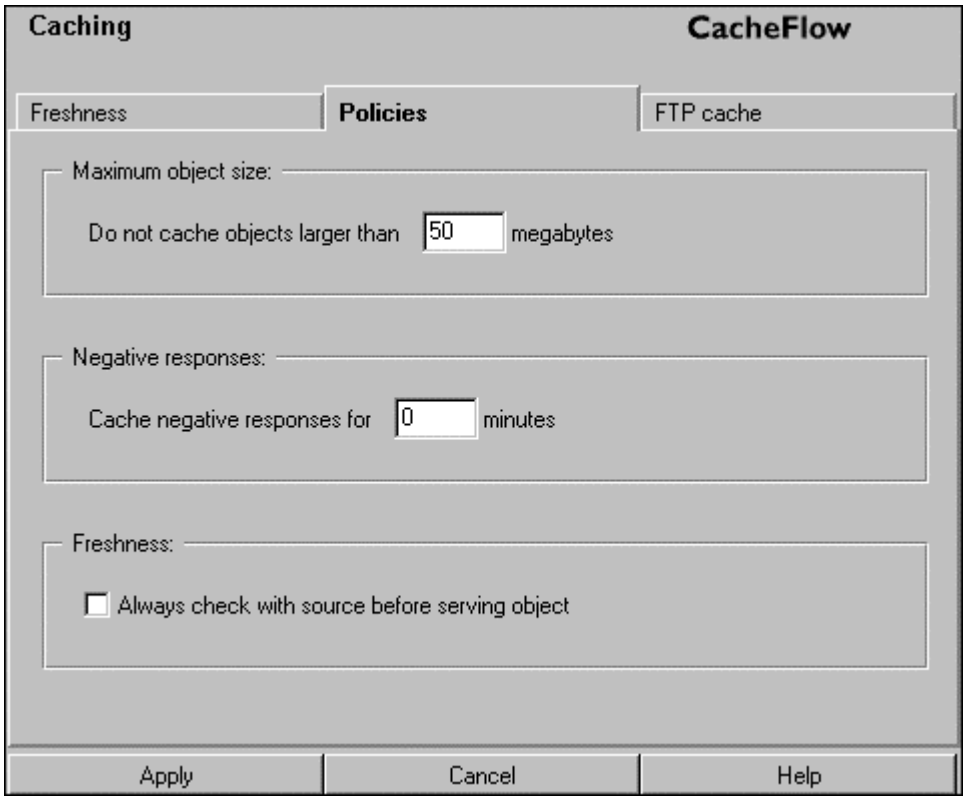


Figure 6-2 Setting freshness policies

Setting the Maximum Object Size

You can set the maximum object size to store in the cache. All objects retrieved that are greater than the maximum object size will be delivered to the client, but they will not be stored in the Cache.

Caching Negative Responses

If CacheOS receives a non-200 HTTP origin-server error response code when attempting to retrieve an object, it can cache the negative response. If CacheOS caches the response, it will return a failure for additional requests for the specified number of minutes. If CacheOS does not cache the response, it will continue to attempt each failed request.

Guaranteed Freshness

The Freshness option allows you to guarantee that all objects served from the cache are current. When this option is checked, CacheOS always checks the object on the source before serving it to the client. This option guarantees that all objects served to the client are fresh, but it's use must be weighed against performance, as enabling this option causes CacheOS to verify every object before it is served to the client.

Setting FTP Caching Options

In addition to HTTP objects, CacheOS can cache objects requested using FTP. When CacheOS retrieves an FTP object and places it in the cache, it uses two methods to determine how long the object should stay in the cache.

- If the object has a last modified date, CacheOS will assign a refresh date to the object that is a percentage of the last modified date.
- If the object does not have a last modified date, CacheOS assigns a refresh date to the object based on a fixed period of time.

The FTP caching options also allows you to specify the maximum size of FTP objects to cache.

To configure FTP cache options

1. Select Management from the CacheOS home page.
2. Select the Caching applet.
3. Select the FTP cache tab.
4. Enter FTP cache size and refresh values.
5. Click Apply to save changes.

To configure FTP cache options using the CLI

1. Open a terminal session with the CacheFlow device and type your Username and Password when prompted.
2. At the command prompt, type **enable** and type your Password when prompted.
3. At the command prompt, type **configure terminal** to enter terminal configuration mode.
4. At the `(config)` prompt, type **caching** to enter cache configuration mode.

   The prompt changes to `(config caching)`.
5. At the `(config caching)` prompt, type **ftp** to enter FTP cache configuration mode.

   The prompt changes to `(config caching ftp)`.
6. At the `(config caching ftp)` prompt, type **disable** or **enable** to disable or enable caching of FTP objects.

   If you typed enable, continue on to step 7, otherwise proceed to step 10.
7. At the `(config caching ftp)` prompt, type **max-cache-size *50*** to set the largest FTP object to cache in megabytes.
8. At the `(config caching ftp)` prompt, type **m-percent *10*** to set the time to live for objects with a last modified time in percent since the object was last modified.
9. At the `(config caching ftp)` prompt, type **type-n-initial *24*** to set the time to live in hours for objects without a last modified time.
10. Type **exit** to return to the `(config caching)` prompt.

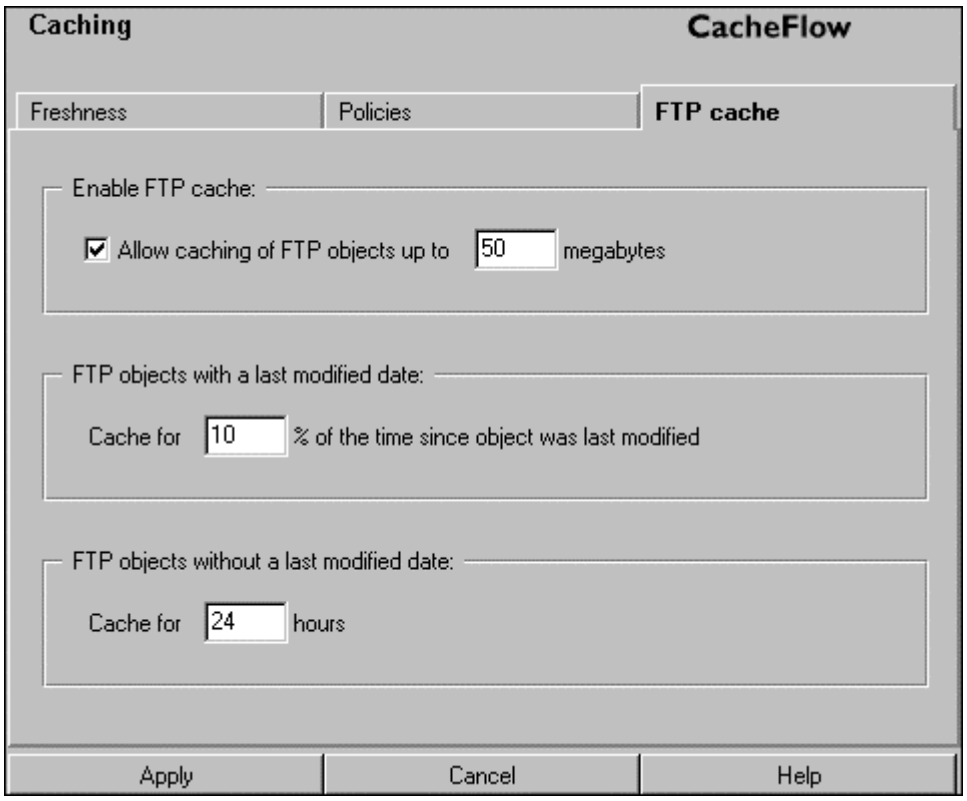


Figure 6-3 Setting FTP caching options

Chapter 7 - Configuring Forwarding Options

The forwarding options available through CacheOS allow you to define how Content Accelerators interact with each other. This type of control is often needed in distributed and hierarchical cache environments. In both cases, a number of Content Accelerators interact to serve up complete content to the client.

To control interaction between caches, CacheOS supports a number of options such as ICP (Internet Caching Protocol), simple gateway forwarding, a direct or deny list, and WCCP (Web Cache Control Protocol). This chapter addresses basic forwarding configuration tasks. For detailed information on creating ICP and advanced forwarding configurations, see the Configuring Hierarchical Caches chapter.

Internet Caching Protocol (ICP)

ICP is a communication protocol for caches. It allows a cache to query other caches for an object, without actually requesting the object. By using ICP, the cache can determine if the object is available from a neighboring cache, and which cache will provide the fastest response.

Installing an ICP or Advanced Forwarding Configuration

Once you have created the ICP or advanced forwarding configuration file, place the file on an FTP or HTTP server so it can be downloaded to the Content Accelerator.

To install an ICP or advanced forwarding settings

1. Select Management from the CacheOS home page.
2. Select the Forwarding applet.
3. Select the ICP tab
4. Enter the fully qualified URL, including the filename, where the configuration file is located.
   You can click View to display the configuration file before installing it.
5. Click Install to download the configuration file.
6. Click Apply to save changes.

To install an ICP or advanced forwarding settings using the CLI

1. Open a terminal session with the CacheFlow device and type your Username and Password when prompted.
2. At the command prompt, type **enable** and type your Password when prompted.
3. At the command prompt, type **configure terminal** to enter terminal configuration mode.
4. At the `(config)` prompt, type **icp path *URL*** to install the ICP or advanced forwarding settings.
   Enter a fully qualified URL, including the filename, where the configuration file is located.
5. At the `(config)` prompt, type **load icp-settings**.

5. Type **exit** to leave configuration mode.

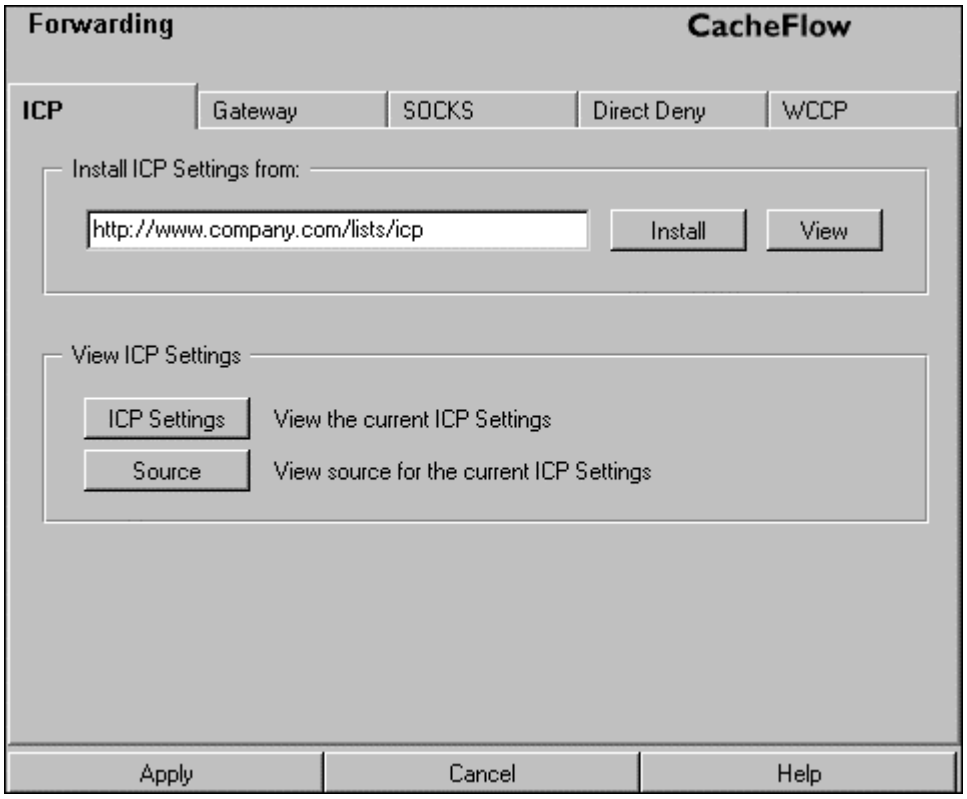


Figure 7-1 Installing ICP or advanced forwarding settings

Using Simple Gateway Forwarding

If you use simple forwarding, all requests for objects not found in the cache are forwarded to a single cache host gateway, or an alternate gateway if the primary gateway is unavailable. If a gateway is specified, when an object is requested that is not in the cache, the Web cache will forward the request to the gateway rather than retrieve the object from the network.

To configure a forwarding gateway

1. Select Management from the Content Accelerator home page.
2. Select the Forwarding applet.
3. Select the Gateway tab.
4. Select the gateway to configure (primary or alternate) in the Settings for drop-down list.
5. Enter the domain name or IP address and port of the gateway.

   If the gateway is a SOCKS server, enable the Use SOCKS when forwarding HTTP requests to this gateway checkbox.
6. Click Apply to save changes.

To configure a forwarding gateway using the CLI

1. Open a terminal session with the CacheFlow device and type your Username and Password when prompted.
2. At the command prompt, type **enable** and type your Password when prompted.
3. At the command prompt, type **configure terminal** to enter terminal configuration mode.
4. At the `(config)` prompt, type **forwarding primary gateway** to enter primary gateway configuration mode.
   The prompt changes to `(config forwarding primary)`.
5. Type **address** ***10.25.36.0*** to specify the IP address of the gateway, or type **address name** ***DomainName*** to specify the domain name of the gateway.
6. Type **port** ***8082*** to set the port number to use on the gateway.
7. Type **socks** or **no socks** as appropriate to define whether the gateway is a SOCKS server.
8. Type **exit** to return to the `(config)` prompt.

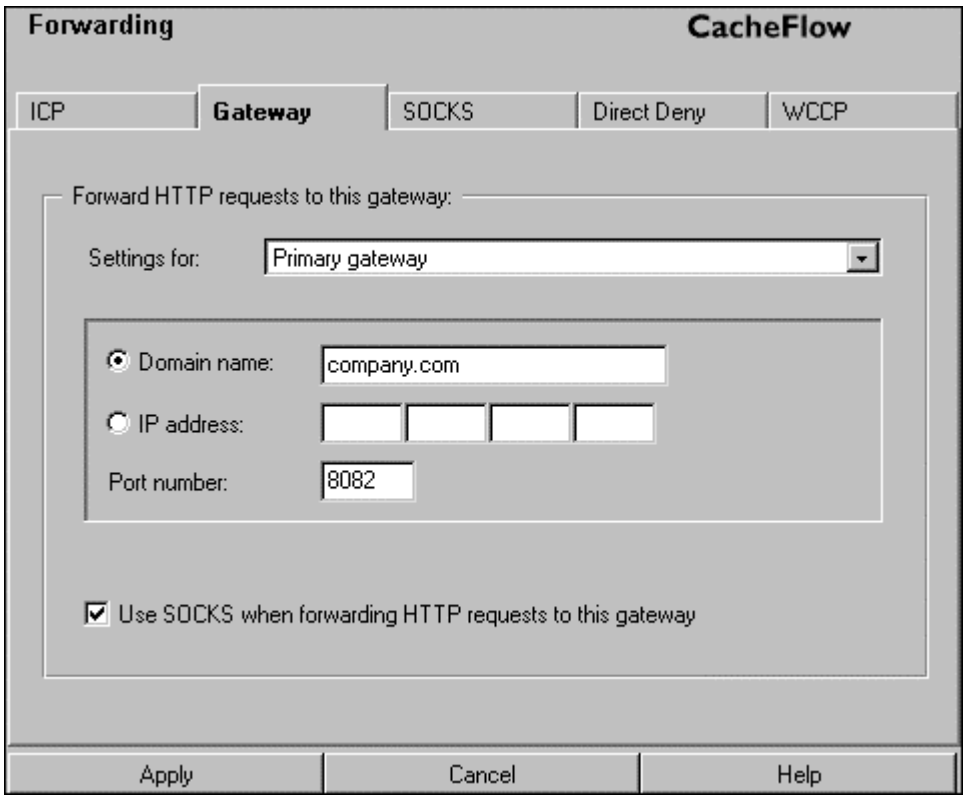


Figure 7-2 Configuring an HTTP gateway

Using a SOCKS Server

If you are using a SOCKS server for the primary or alternate forwarding gateway, you must specify the Content Accelerator's ID for the Identification (Ident) protocol used by the SOCKS gateway. The machine ID should be configured to be the same as the Content Accelerator's name.

To specify the SOCKS machine ID

1. Select Management from the CacheOS home page.
2. Select the Forwarding applet.
3. Select the SOCKS tab.
4. Enter the SOCKS machine ID in the Machine ID field.
5. Click Apply to save changes.

To specify the SOCKS machine ID using the CLI

1. Open a terminal session with the CacheFlow device and type your Username and Password when prompted.
2. At the command prompt, type **enable** and type your Password when prompted.
3. At the command prompt, type **configure terminal** to enter terminal configuration mode.
4. At the `(config)` prompt, type **socks-machine-id *machineID*** to set the SOCKS machine ID.
5. Type **exit** to leave the configuration mode.

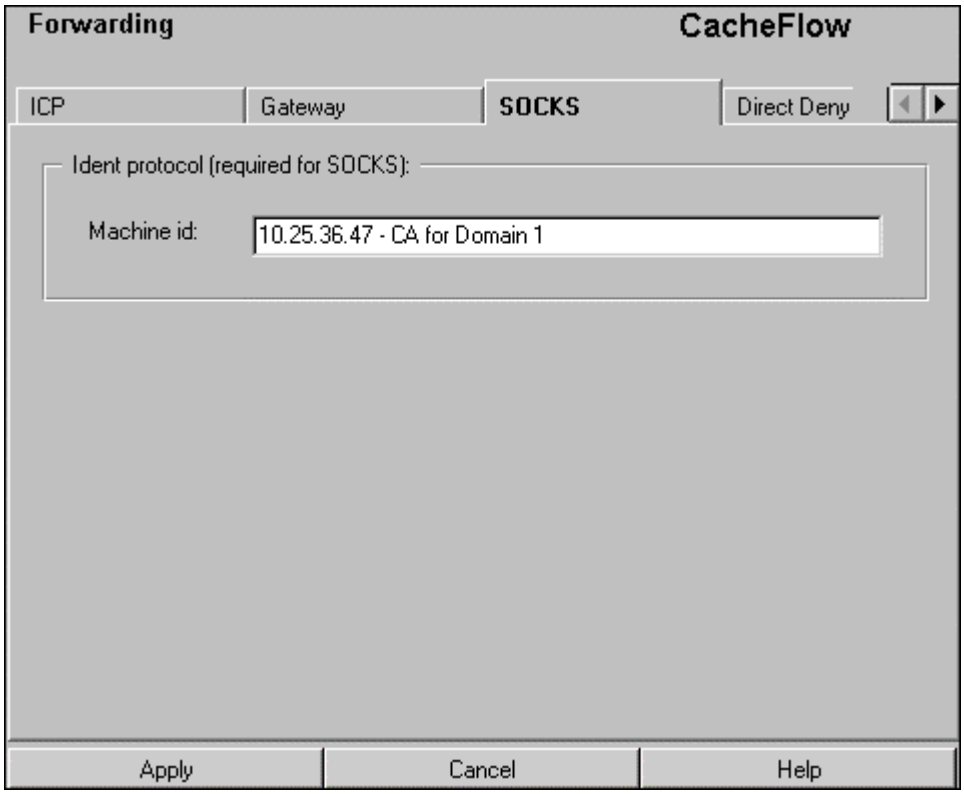


Figure 7-3 Entering a SOCKS Machine ID

Installing Direct or Deny Settings

When using a proxy gateway, CacheOS forwards requests to the gateway server. The gateway server then determines what to do with the request. The gateway can be used to forward requests to an external network.

Direct addresses are addresses CacheOS should send out on the network rather than forward to the gateway. Deny addresses are addresses to which CacheOS should deny access. The direct and deny address specifications are made up of a subnet and mask. Requested addresses are compared to the subnet and mask to determine a match. If the request does not match an address in the direct or deny list, CacheOS sends the request to the gateway.

The direct or deny list is a simple text file containing a list of IP addresses, subnet masks, and commands. A sample direct or deny list is illustrated below:

```
10.25.36.47   255.255.0.0   DENY
10.25.36.48   255.255.0.0   DENY
10.25.36.49   255.255.0.0   DIRECT
```

To enter a direct or deny list, create a text file with the direct or deny commands then place the file on an HTTP or FTP server so it can be downloaded to the Content Accelerator.

To install direct or deny settings

1. Select Management from the CacheOS home page.
2. Select the Forwarding applet.
3. Select the Direct Deny tab.
4. Enter the fully qualified URL, including the filename, where the configuration file is located.
   You can click View to display the list before installing it.
5. Click Install to download the list.

To install direct or deny settings using the CLI

1. Open a terminal session with the CacheFlow device and type your Username and Password when prompted.
2. At the command prompt, type **enable** and type your Password when prompted.
3. At the command prompt, type **configure terminal** to enter terminal configuration mode.
4. At the `(config)` prompt, type **direct-deny-list path *URL*** to install the direct-or-deny settings.
   Enter a fully qualified URL, including the filename, where the configuration file is located.
5. Type **exit** to leave configuration mode.

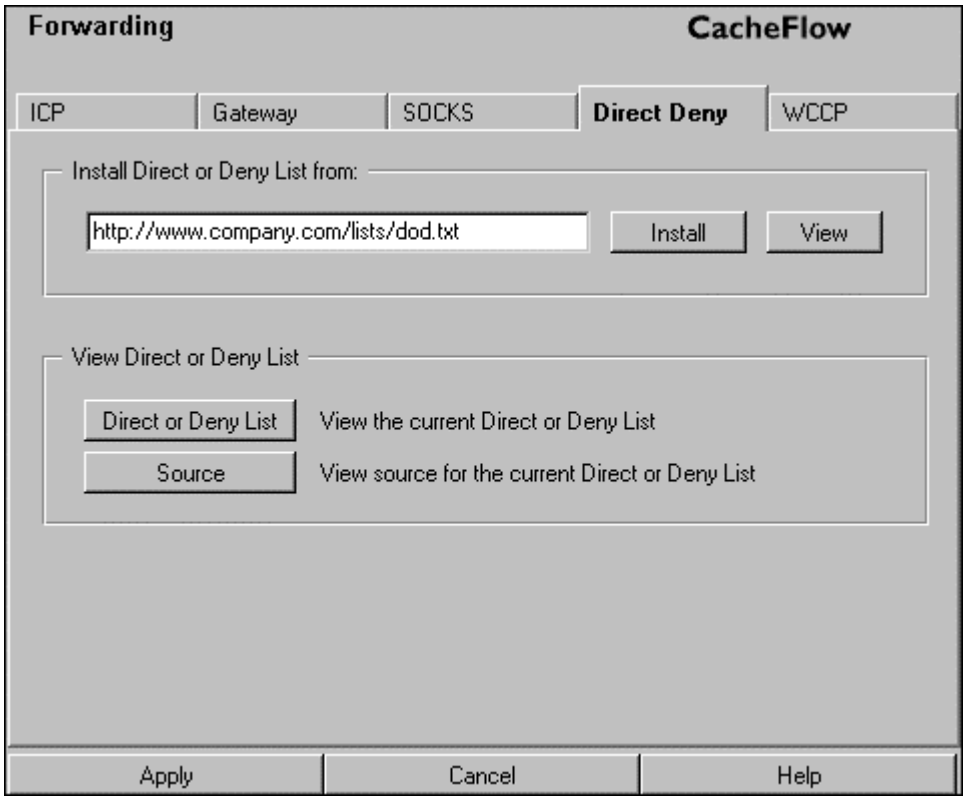


Figure 7-4 Installing direct or deny settings

Installing WCCP Settings

The Content Accelerator can be configured to participate in a WCCP (Web Cache Control Protocol) scheme, where a WCCP-capable router collaborates with a set of WCCP-configured Content Accelerators to service requests. WCCP is a Cisco-developed protocol. For more information about WCCP, refer to the *Appendix B: WCCP (Web Cache Control Protocol)*.

To install WCCP settings

1. Select Management from the CacheOS home page.
2. Select the Forwarding applet.
3. Select the WCCP tab.
4. Enter the fully qualified URL, including the filename, where the configuration file is located.
   You can click View to display the configuration file before installing it.
5. Click Install to download the configuration file.

To install WCCP settings using the CLI

1. Open a terminal session with the CacheFlow device and type your Username and Password when prompted.
2. At the command prompt, type **enable** and type your Password when prompted.

3. At the command prompt, type **configure terminal** to enter terminal configuration mode.
4. At the `(config)` prompt, type **wccp path *URL*** to install the WCCP settings.
   Enter a fully qualified URL, including the filename, where the configuration file is located.
5. At the `(config)` prompt, type load **wccp-settings**.
6. At the `(config)` prompt, type **wccp enable** to enable WCCP-based forwarding.
7. Type **exit** to leave configuration mode.

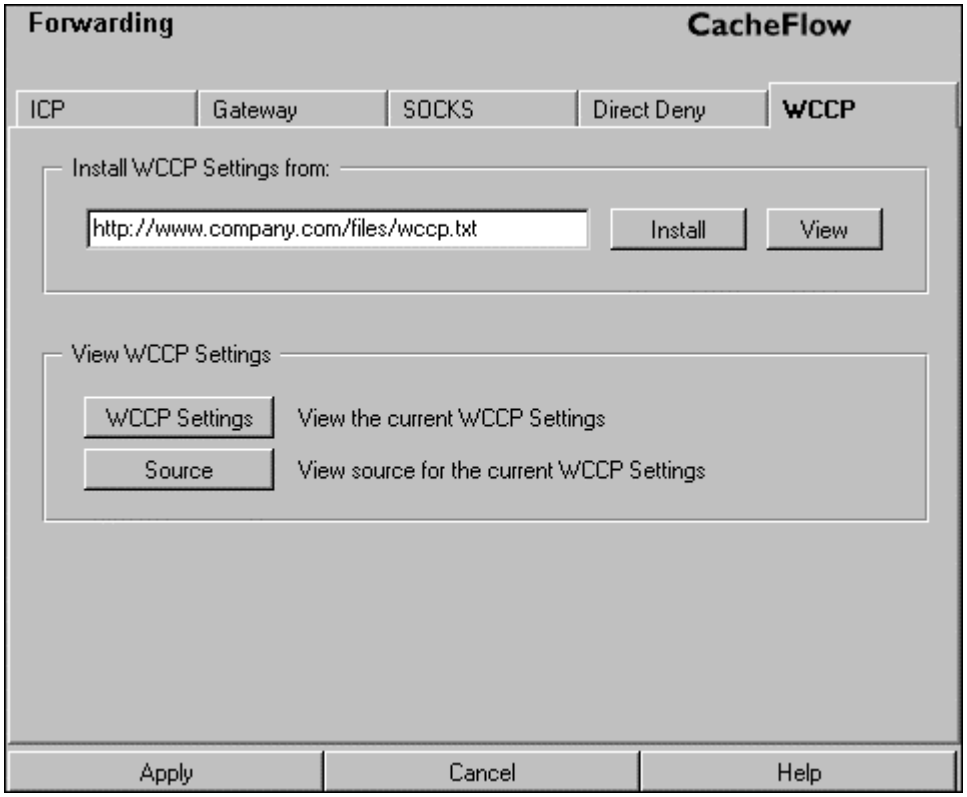


Figure 7-5 Installing WCCP settings

This page intentionally blank.

Chapter 8 - Configuring Hierarchical Caches

When you have one Web cache, objects are either served from the cache or retrieved from the source. When you have multiple Web caches, you can create a cache hierarchy to manage requests. If one cache doesn't have the requested object, it can pass the request to other caches rather than going to the source. This chapter describes the options for creating and managing cache hierarchies.

When CacheOS receives a request for an object, it must determine how to process the request. If the object is in the cache, the CacheOS returns it. If the object is not in the cache, CacheOS can get the object from the source or request it from another cache in the cache hierarchy.

You create a cache hierarchy by defining forwarding or ICP (Internet Caching Protocol) hosts the CacheOS can use to forward requests. A forwarding host can be any HTTP cache. An ICP host is a cache that supports the ICP protocol.

Forwarding Options

There are three forwarding options: simple forwarding, advanced forwarding, and ICP.

Simple Forwarding

Simple forwarding means the CacheOS is configured to forward all requests not found in the cache to a single cache host gateway, or an alternate gateway if the primary gateway is unavailable. You should use simple forwarding only when you want all requests to go through the same gateway. Simple forwarding forwards all requests for objects not found in the cache to the same gateway. For example, if you have one departmental cache and only one border cache that provides access outside your network, you can use simple to forward requests to the border cache. This is illustrated in the following figure.

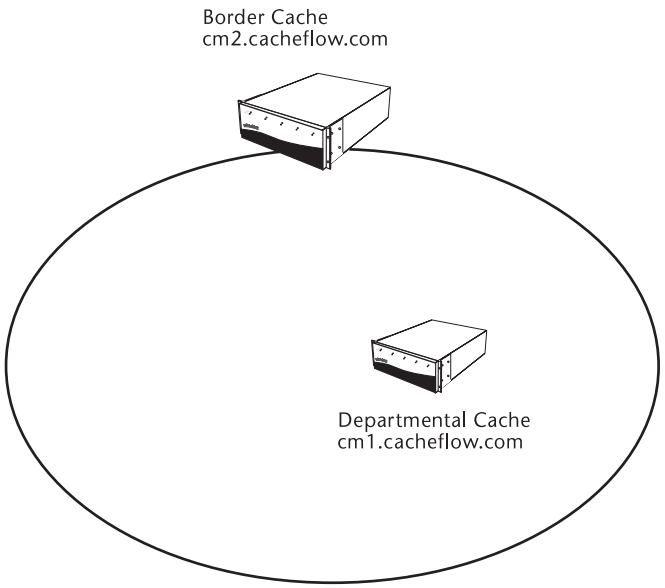


Figure 8-1 Simple Forwarding Configuration

Advanced Forwarding

Advanced forwarding allows you to forward requests based on the URL or IP address of the requested object, and can be used to selectively forward requests to different caches, or even groups of caches. When a group of caches is defined, the CacheOS balances requests across the group, using a hash of the destination server's domain name to select the cache to use. This ensures an object from a specific server is always requested from the same cache in the hierarchy, but spreads the requests across all members of the group.

You should use advanced forwarding when there are multiple caches to which you want to forward requests, or when you want to create complex forwarding rules based on the location or type of object requested. For example, you might want to forward requests for objects on the departmental networks directly to the departmental caches, but forward requests for objects on external networks to a group of border caches. This example is illustrated in the following figure.

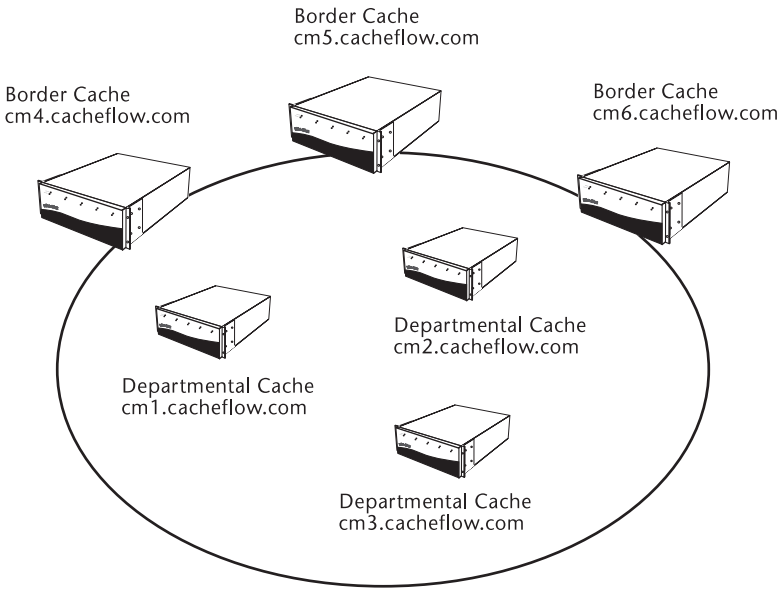


Figure 8-2 Advanced Forwarding Configuration

ICP (Internet Caching Protocol)

ICP is a communication protocol for caches. It allows a cache to query other caches for an object, without actually requesting the object. By using ICP, the cache can determine if the object is available from a neighboring cache, and which cache will provide the fastest response. You should use ICP only if you have ICP hosts available for you to use, or you want the CacheOS to support ICP requests from other ICP hosts.

By design, the ICP protocol requires the requesting host to wait up to two seconds for all ICP hosts to respond to the request for an object. That means the cost of using ICP (the possibly delay just to see if the object is available) must be less than the cost of retrieving the object directly from the source. ICP is commonly used in Europe, where the cost of retrieving the data from the United States (or other countries) might be considerably higher than the potential delay of searching for the object in local ICP caches.

Configuring Simple Forwarding

To configure simple forwarding, all you have to do is define the address of a primary and alternate forwarding gateway. To define the forwarding gateways using the command line interface, see the **forwarding** command.

Configuring Advanced Forwarding

To configure advanced forwarding you must create a configuration file and load it on the CacheOS. The advanced forwarding configuration is defined in the shared ICP/Forwarding configuration file. For information on loading the configuration using the command line interface, see the **icp** command.

Advanced Forwarding Configuration Commands

The advanced forwarding configuration directives are listed below:

- fwd_host
- fwd_host_url_regex
- fwd_host_domain
- fwd_host_ip
- domain_alias

The fwd_host directive is used to specify forwarding hosts, and forwarding groups. The fwd_host_url_regex, fwd_host_domain, and fwd_host_ip directives define the requests to forward.

Defining a Forwarding Host

To define a single host to use as the forwarding host, use the fwd_host command, then enter a fwd_host_domain, fwd_host_ip, or fwd_host_url_regex directive to forward requests to the host:

```
; directive  hostName               httpPort  attributes
fwd_host     cm1.cacheflow.com      8080

; directive      hostname            domain
fwd_host_domain  cm1.cacheflow.com   eng.cacheflow.com
```

In this example, all requests that match the eng.cacheflow.com domain will be forwarded to cm1.cacheflow.com. Enter additional fwd_host commands to specify additional forwarding hosts. You can include the **default** attribute to define the default parent to use for requests that do not match a fwd_host_domain, fwd_host_ip, fwd_host_url_regex directive.

```
; directive  hostName               httpPort  attributes
fwd_host     cm1.cacheflow.com      8080
fwd_host     cm2.cacheflow.com      8080
fwd_host     cm3.cacheflow.com      8080      default

; directive         hostname            url
fwd_host_url_regex  cm1.cacheflow.com   http://.*\.cacheflow\.com.*
fwd_host_url_regex  cm2.cacheflow.com   http://.*\.acc\.cacheflow\.com.*
fwd_host_url_regex  cm3.cacheflow.com   http://.*\.ec\.cacheflow\.com.*
```

In this example, requests are forwarded as follows:

- all requests for the acc.cacheflow.com domain are forwarded to cm2.cacheflow.com
- all requests for the ec.cacheflow.com domain are forwarded to cm3.cacheflow.com
- all other requests to the cacheflow.com domain are forwarded to cm1.cacheflow.com
- all requests that do not match a fwd_host_domain, fwd_host_url_regex, or fwd_host_ip specification are forwarded to cm3.cacheflow.com

Defining a Forwarding Group

If you want to balance requests over a group of forwarding hosts, you can define forwarding groups. To define a forwarding group, specify the group attribute with the fwd_host directive, then specify the group name with the fwd_host_domain, fwd_host_ip, or fwd_host_url_regex directive:

```
; directive   hostName            httpPort  attributes
fwd_host      cm4.cacheflow.com   8080      group=border1
fwd_host      cm5.cacheflow.com   8080      group=border1
fwd_host      cm6.cacheflow.com   8080      group=border1

; directive      groupname domain
fwd_host_domain  border1   company.com

; directive      groupname IP address    subnet mask
fwd_host_ip      border1   10.25.0.1     255.255.255.0
```

In this example, all requests for the company.com domain, or an IP address that matches the subnet specification, are sent to the border1 group.

The fwd_host Directive

The fwd_host directive defines forwarding hosts to use with the fwd_host_domain, fwd_host_ip, and fwd_host_url_regex directives. The parameters for the fwd_host directive are described below:

```
fwd_host hostname httpPort [default | backup | group=groupname] [deferred | socks]
```

Parameters:	Value	Description
hostname		The host name of the cache.
HTTPport		TCP port where the cache accepts HTTP requests. The common HTTP port is 8080.
default		If specified, designates a cache host to be the default forwarding host. All requests that do not match a fwd_host_domain, fwd_host_ip, or fwd_host_url_regex specification will be forwarded to the default host.
backup		A backup default cache host if the default host is not available.
group	groupname	The forwarding group to which this cache host belongs. When you define a forwarding group, the CacheOS will balance requests across members of the group. The first instance of the group name creates the group.

Parameters:	Value	Description
deferred		This specifies a deferred request in which a Web-server style request format is used for this host.
socks		Socks must always appear at the end of the fwd_host command. Socks supports the default, backup and group options. The deferred option is not valid with socks because requests are deferred by the socks gateway. Socks uses port 1080 by default.

The fwd_host_domain Directive

The fwd_host_domain directive defines which requests are sent to which cache hosts or cache host groups. The parameters for the fwd_host_domain directive are described below:

```
fwd_host_domain hostname|groupname domain
```

Parameters:	Description
hostname | groupname	Either the host name or group name defined by a fwd_host command. You can also handle requests locally by specifying “direct” for the host name, or deny requests by specifying “deny” for the host name.
domain	The domain to match. All requests that match the specified domain will be forwarded to the cache host or group. If you specify an asterisk (*) for the domain, the host or group will be used for all requests that fail to match a domain specification.

The fwd_host_ip Directive

The fwd_host_ip directive works like the fwd_host_domain directive, except you can specify an IP address and subnet mask rather than a domain. If you use the fwd_host_ip directive, the CacheOS will look up the IP address for requests that specify a domain name, using DNS. The DNS requests are cached, but the lookups might still affect performance.

The parameters for the fwd_host_ip directive are described below:

```
fwd_host_ip hostname|groupname IP address/subnet mask
```

Parameters:	Description
hostname | groupname	Either the host name or group name defined by a fwd_host command. You can also handle requests locally by specifying “direct” for the host name, or deny requests by specifying “deny”

Parameters:	Description
	for the host name.
IP address/subnet mask	The IP address to match. All requests that match the specified address and subnet mask will be forwarded to the cache host. If you specify 0.0.0.0 for the address, the host or group will be used for all requests that fail to match an address specification.

The fwd_host_url_regex Directive

The fwd_host_url_regex directive works like the fwd_host_domain directive, except you can specify a regular expression to match the URL. For information on using regular expressions, see "Regular Expressions." The parameters for the fwd_host_url_regex directive are described below:

```
fwd_host_url_regex hostname|groupname URL
```

Parameters:	Description
hostname | groupname	Either the host name or group name defined by a fwd_host command. You can also handle requests locally by specifying "direct" for the host name, or deny requests by specifying "deny" for the host name.
URL	The URL specification to match. The URL specification will be treated as a regular expression string. All requests that match the URL will be forwarded to the cache host.

The domain_alias Directive

The domain_alias directive prevents duplication of content in the cache when different host names point to the same objects. Domain_alias converts all URLs with one domain name into another domain name. The directive parameters are:

Parameter	Description
original	This is the original name of the domain.
alias	This is an alias the user chooses. Content from the original parameter is converted into the alias.

Configuring ICP

An ICP cache hierarchy is comprised of a group of caches with defined parent and sibling relationships. A relationship exists between two caches. A cache parent is a cache that can return the object if it is in the cache, or

request the object from the source on behalf of the requester if the object is not in the cache. A cache sibling is a cache that can only return the object if it is in the cache. One cache acting as a parent can also act as a sibling to other caches.

1. The ICP conversation between caches is simple:
2. When an object is not in the cache, the cache sends an ICP query to its neighbors (parents and siblings) to see if any of its peers holds the object.
3. Each neighbor that holds the requested object returns an ICP_HIT reply.
4. Each neighbor that does not hold the object returns an ICP_MISS reply.

Based on the responses, the cache can determine where to request the object: from one of its neighbors, or from the source. If an ICP_HIT reply is received, the request is sent to the cache host that returned the first reply. If no ICP_HIT reply is received, the request is forwarded to the first parent that replied. If no parents respond or are configured, the request is made directly to the source.

ICP Configuration Directives

To configure ICP you must create a configuration file and load it on the CacheOS. The ICP configuration is defined in the shared ICP/Forwarding configuration file. For information on loading the configuration using the command line interface, see the **icp** command. For information on loading the configuration using the Web interface, see page 140.

The ICP directives are

```
icp_host hostname peertype HTTPport ICPport default | backup
icp_access_domain allow | deny domain
icp_access_ip allow | deny IP address/subnet mask
```

icp_host

The icp_host directive describes cache peers in the hierarchy. There should be one entry for each cache you want to use. The parameters for the icp_host command are described below:

```
icp_host hostname peertype HTTPport ICPport default | backup | feeder
```

Parameters:	Value	Description
hostname		The host name of the cache.
peertype	parent sibling	Relationship of the cache to the cache you are configuring.
HTTPport		TCP port where the cache accepts HTTP requests. The common HTTP port is 8080.
ICPport		UDP port where the cache accepts ICP requests. The common ICP port is 3130.
default		If specified, designates a cache host parent to be the default ICP parent. If no ICP reply is received, all requests will be forwarded to the default parent.

Parameters:	Value	Description
backup		If specified, designates a cache host parent to be the backup default ICP parent. If the default parent is not available, the CacheOS will use the backup default parent.
feeder		If specified, designates the cache host sibling as a feeder-type host, using ICP request loops to populate the cache.

Sample icp_host directives are listed below:

```
; Define ICP parent and sibling caches.
icp_host cm1.cacheflow.com parent 8080 3130  default
icp_host cm2.cacheflow.com sibling 8080 3130
icp_host cm3.cacheflow.com sibling 8080 3130
icp_host cm4.cacheflow.com sibling 8080 3130
icp_host cm5.cacheflow.com parent 8080 3130
```

Restricting Access

You can restrict access to the CacheOS Web cache by other ICP hosts using the icp_access_domain and icp_access_ip directives. By default, when ICP is configured all ICP hosts are allowed access. You should deny access to all domains other than the ICP hosts you want to use.

icp_access_domain

The icp_access_domain directive requires a reverse DNS lookup of each ICP query to validate the IP address. Whenever possible, you should use the icp_access_ip directive to avoid potential problems.

The icp_access_domain directive defines which hosts can request objects from the Web cache using ICP. The default action is to allow all requests. When you use icp_access_domain, each ICP query requires a reverse DNS lookup to validate the IP address. Depending on the number of ICP requests, these lookups can consume CacheOS resources.

The parameters for the icp_access_domain directive are described below:

```
icp_access_domain allow|deny domain
```

Parameters:	Description
allow|deny	Allow or deny ICP queries from neighbors that match the domain specification.
domain	The domain to match. All ICP queries from neighbors that match the specified domain will be handled by the cache host. The special domain of “all” defines the default action when there is no domain match.

Sample icp_access _domain directives are listed below:

```
; allow ICP access to this CacheFlow Web cache from the cacheflow.com domain
icp_access_domain allow cacheflow.com
icp_access_domain deny all
; the deny all option should always be specified to deny all other domains
```

icp_access_ip

The icp_access_ip directive works like the icp_access_domain command, except you can specify an IP address and subnet mask rather than a domain. The parameters for the icp_access_ip command are described below:

```
icp_access_ip allow|deny address/subnet
```

Parameters:	Description
allow|deny	Allow or deny ICP queries from neighbors that match the address specification.
address/subnet mask	The address and subnet mask to match. All ICP queries that match the specified address will be handled by the cache host. The special address of 0.0.0.0 defines the default action when there is no address match.

Sample icp_access_ip directives are listed below:

```
; allow ICP access to this CacheFlow Web cache from the local subnet
icp_access_ip allow 192.168.10.0/255.255.255.0
icp_access_ip deny 0.0.0.0
; the deny all option should always be specified to deny all other domains
```

Other Advanced Forwarding Options

In addition to the ICP and advanced forwarding directives described in the sections above, you can specify the following commands in the ICP/Forwarding configuration file:

```
icp_port 0
neighbor_timeout 2
icp_failcount attempts
http_failcount attempts
host_fail_notify on|off
host_recover_notify on|off
```

icp_port

The icp_port directive sets the port the CacheOS will use to listen for ICP requests. The default port is 3130. If you set the port to 0, ICP will be disabled.

neighbor_timeout

The neighbor_timeout directive sets the number of seconds CacheOS waits for ICP replies. When CacheOS sends an ICP request, it waits for all hosts to reply or for the neighbor_timeout to expire. The default timeout is 2 seconds.

icp_failcount

The icp_failcount directive sets the number of consecutive failures the CacheOS can receive before considering the ICP host as failed. By default, the ICP failure count is set to 20. Each time a request fails, the failure count is incremented. When a request succeeds, the failure count is reset to zero.

http_failcount

The http_failcount directive sets the number of consecutive failures the CacheOS can receive before considering the HTTP host as failed. By default, the HTTP failure count is set to 5. Each time a request fails, the failure count is incremented. When a request succeeds, the failure count is reset to zero. When an HTTP host fails, the CacheOS will wait five minutes before attempting to use it again as a forwarding target. If the next request fails, the CacheOS will continue to wait five minutes between attempts until the cache becomes available.

host_fail_notify

The host_fail_notify directive tells CacheOS to send event notification email when a connect fails persistently.

host_recover_notify

The host_recover_notify directive tells CacheOS to send event notification email when a failed host recovers.

Forwarding Order

To determine how a request should be handled, the CacheOS checks for a forwarding match in the following order:

1. Check for an advanced forwarding regular expression match on the URL (fwd_host_url_regex). If there are multiple regular expressions that match the URL, the longest expression is used.
2. Check for an advanced forwarding domain name match on the request (fwd_host_domain).
3. Check for an advanced forwarding IP address match on the request (fwd_host_ip).
4. Check for an advanced forwarding default domain match (fwd_host_domain with a domain of "*").
5. If there are ICP peers, send out ICP queries and wait for a response or timeout.
6. Check for a healthy advanced forwarding default host (fwd_host).
7. Check for a healthy advanced forwarding backup host (fwd_host).
8. Check for a simple forwarding direct or deny list match.
9. Check for a healthy primary gateway for simple forwarding.
10. Check for a healthy alternate gateway for simple forwarding.
11. Do not forward the request, retrieve the object from the source.

    The CacheOS uses the first match to process the request.

This page intentionally blank.

Chapter 9 - Configuring Security

CacheOS provides the following security features:

- Console account username and password
- Access control lists which allow you to restrict access to the CacheOS Management console by station address or subnet
- Proxy user authentication through a Unix password file
- CacheOS administrator and proxy user authentication using either Lightweight Directory Access Protocol (LDAP) or Remote Authentication Dial-In User Service (RADIUS).

Important To prevent unauthorized access to the Content Accelerator, the console username and password should be given only to those who will administer the Content Accelerator.

Setting the Console Username and Password

The command line interface and Web interface should be protected with a username and password. The username and password are set during Initial Network Configuration. If you forget the username or password, they can be reset using the Setup console as described in Initial Network Configuration.

Important Always define a console account user name and password. If left blank, the Management console can be accessed without specifying a username and password.

To set the username and password from the Management console

1. Select Management from the CacheOS home page.
2. Select the Security applet.
3. Enter the username in the User name field.
4. Enter the password in the New password field.
5. Re-enter the password in the Verify password field.
6. Click Apply to save changes.

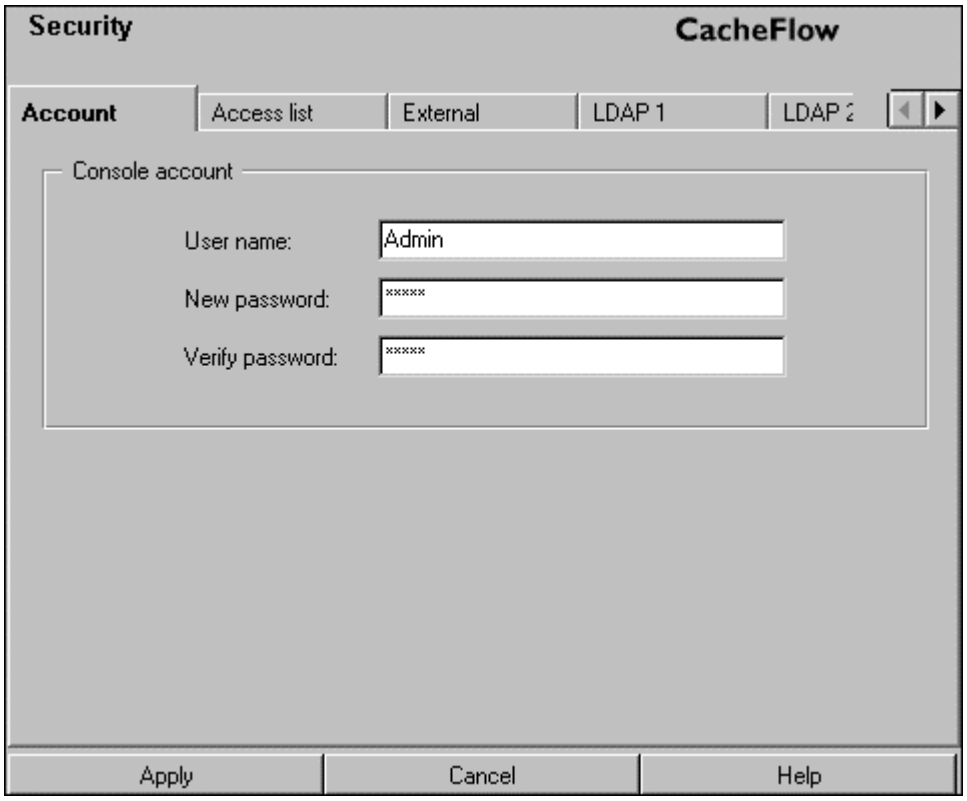


Figure 9-1 Setting the management console username and password

To set the username and password using the CLI

1. Open a terminal session with the Content Accelerator and type your Username and Password when prompted.
2. At the command prompt, type **enable** and type your Password when prompted.
3. At the command prompt, type **configure terminal**.
4. At the command prompt, type **security username** ***username***
5. At the command prompt, type **security password** ***12345***

Setting Access Restrictions

CacheOS allows you to restrict access to the Management console by source address. You can restrict access to individual IP addresses or subnets by creating an access control list.

To create an access control list

1. Select Management from the CacheOS home page.
2. Select the Security applet.
3. Select the Access list tab.
4. Click New and enter a static IP address specification in the Subnet (Source address) field.

5. Enter the subnet mask in the Mask (Subnet mask) field. To restrict access to an individual workstation, enter 255.255.255.255.
6. Click OK.
7. Enable the Engage console access control list checkbox.
8. Click Apply to save changes.

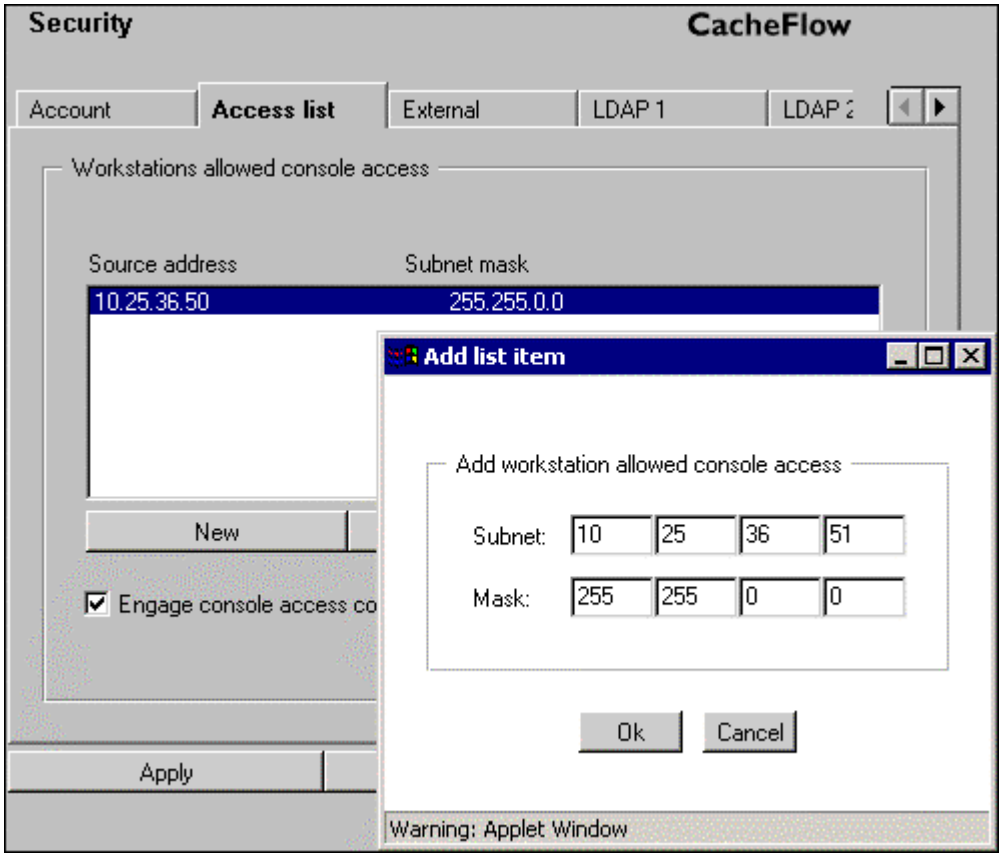


Figure 9-2 Setting access restrictions

Important! Before you enable the access control list, verify the station restrictions entered will not restrict your current workstation from accessing the Management console. If you set the access restrictions incorrectly, you can correct the problem using the Setup console as described in Initial Network Configuration.

To create an access control list using the CLI

1. Open a terminal session with the Content Accelerator and type your Username and Password when prompted.
2. At the command prompt, type **enable** and type your Password when prompted.
3. At the command prompt, type **configure terminal**.
   The prompt changes to (config).
4. At the command prompt, type **security allowed-access *10.25.36.50 255.255.0.0***
5. Repeat step 4 for each workstation that you need to add to the console access list.

External User Authentication

CacheOS supports external authentication of administrators and users through the use of a UNIX password file, Lightweight Directory Access Protocol (LDAP) and Remote Authentication Dial-In User Service (RADIUS). LDAP and RADIUS are communication mechanisms for accessing an external database to perform administrator and user authentication.

General Authentication Notes

- Transparent mode caching should not be used when external user authentication is enabled. The related procedure is covered in the next section of this document.
- CacheOS can be configured to log Administrator and User accesses and changes. There are three different types of logs:
  - Event Log shows all administrator accesses and changes they have made to the Content Accelerator. Any Severe error messages, such as: the server not responding or invalid accesses are logged in the Event log.
  - Syslog provides the same information as the event log.
  - Access log can be used to log each user's HTTP accesses by selecting Common log format to display usernames. You can also configure a custom log format with the %u option to display usernames.

    Note For further details on configuring access logging, see the related section of this manual.

Bypassing External Authentication for Certain URLs

When you enable external authentication, it is enabled for all URLs by default. You can bypass external authentication by defining the URLs in either a central or local filter list and adding the **proxy_authentication=no** directive. A typical entry might appear as **www.company.com proxy_authentication=no**. You can also use wildcards to define groups of URLs for which external authentication should be bypassed, for example, www.*.company.com. See the Using a Filter List section for information about creating and installing filter lists.

Disabling Transparent Mode Caching

Transparent mode caching should be disabled before selecting and configuring external authentication.

To disable transparent mode caching from the Management console

1. Select Management from the CacheOS home page.
2. Select the Network applet.
3. Select the Ports tab.
4. Clear the Enable transparent proxying on port 80 check box.
5. Click Apply to save changes.

To disable transparent mode caching using the CLI

1. Open a terminal session with the Content Accelerator and type your Username and Password when prompted.
2. At the command prompt, type **enable** and type your Password when prompted.

3. At the command prompt, type **configure terminal**.
   The prompt changes to (config).
4. At the command prompt, type **transparent-proxy** ***disable***

Configuring Authentication Using a Unix Password File

The CacheFlow device can utilize a Unix Password file to limit proxy services to specific users.

For details on creating a Unix password file and loading it to the CacheFlow device, see the related Technical Note available at http://download.cacheflow.com/support/common/docs/v2200/Technical_Notes/PUATN.pdf

Note Transparent mode caching should be disabled before selecting and configuring external authentication.

Once the Unix password file is loaded into the Cache Machine, you can enable user authentication.

To enable user authentication using a Unix password file

1. Select Management from the CacheOS home page.
2. Select the Security applet.
3. Select the External tab.
4. Select the Unix password file radio button.
5. Select the Authenticate users check box.
6. Click Apply to save changes.

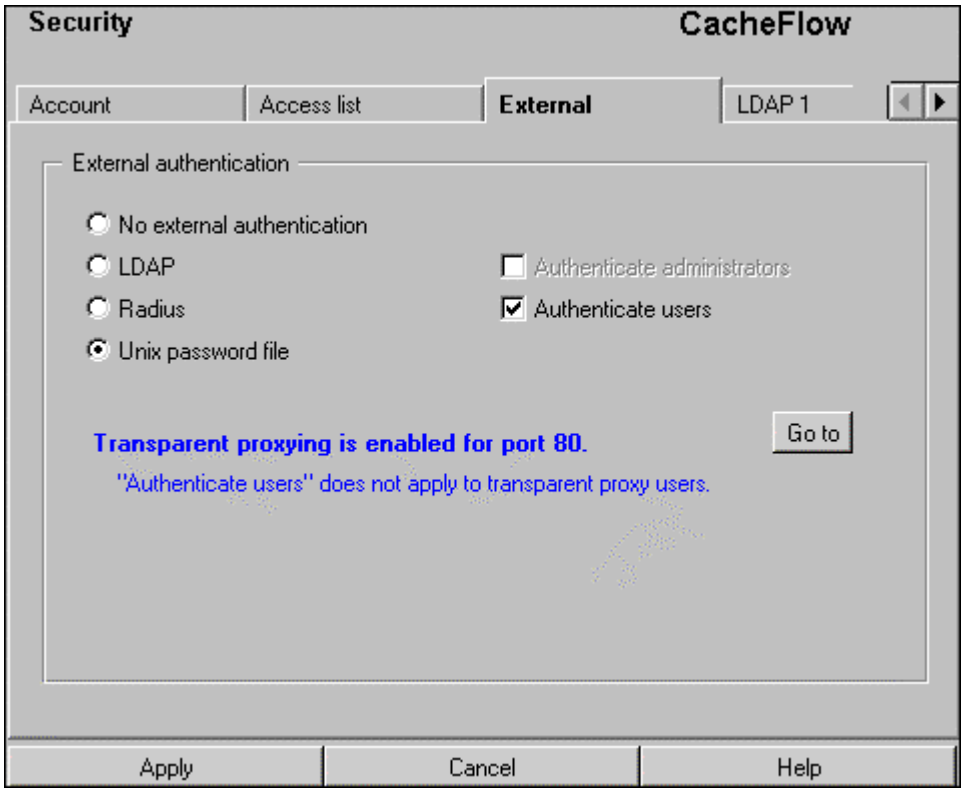


Figure 9-3 Setting user authentication with a Unix password file

To enable user authentication using a Unix password file using the CLI

1. Open a terminal session with the Content Accelerator and type your Username and Password when prompted.
2. At the command prompt, type **enable** and type your Password when prompted.
3. At the command prompt, type **configure terminal**.
   The prompt changes to (config).
4. At the command prompt, type **authentication protocol *unix-password-file***
5. At the command prompt, type **authentication user-verification *enable***

Configuring Authentication with LDAP

Cache supports the use of LDAP as a communications mechanism for accessing an external database to authenticate administrators and HTTP proxy users.

Note Transparent mode caching should be disabled before selecting and configuring external authentication.

LDAP Authentication Notes

- The Console account user name and password remain valid, and the console account user always has administrator read/write privileges, even when using LDAP authentication. If the LDAP server fails, the Console account username and password can still be used to access the Management console.

Important! You should always define a Console account user name and password. If the user name and password fields are left blank, the Management Console can be accessed without specifying a username and password.

- LDAP authentication provides the ability to define:
  - Additional administrators who have Read/Write and Read only access authorization to the Content Accelerator
  - Proxy users
- LDAP authentication applies to the Management console and CLI commands.
- Read Write access allows administrators to access the CacheFlow Management console and all of the CLI commands.
- CacheOS caches proxy user credentials but does not cache administrator credentials.
- The enable password is still required for the CLI.
- Read only access lets administrators view the Management console but not make any changes. It also allows access to any unprivileged CLI commands. Unprivileged CLI commands are those that do not require the enable password.
- The LDAP server schema must have valid admin and user attribute types.

  Detailed LDAP information is available in the RFC/STD/FYI/BCP archives at http://www.faqs.org

To configure authentication with LDAP

1. Select Management from the CacheOS home page.
2. Select the Security applet.
3. Select the LDAP1 tab.

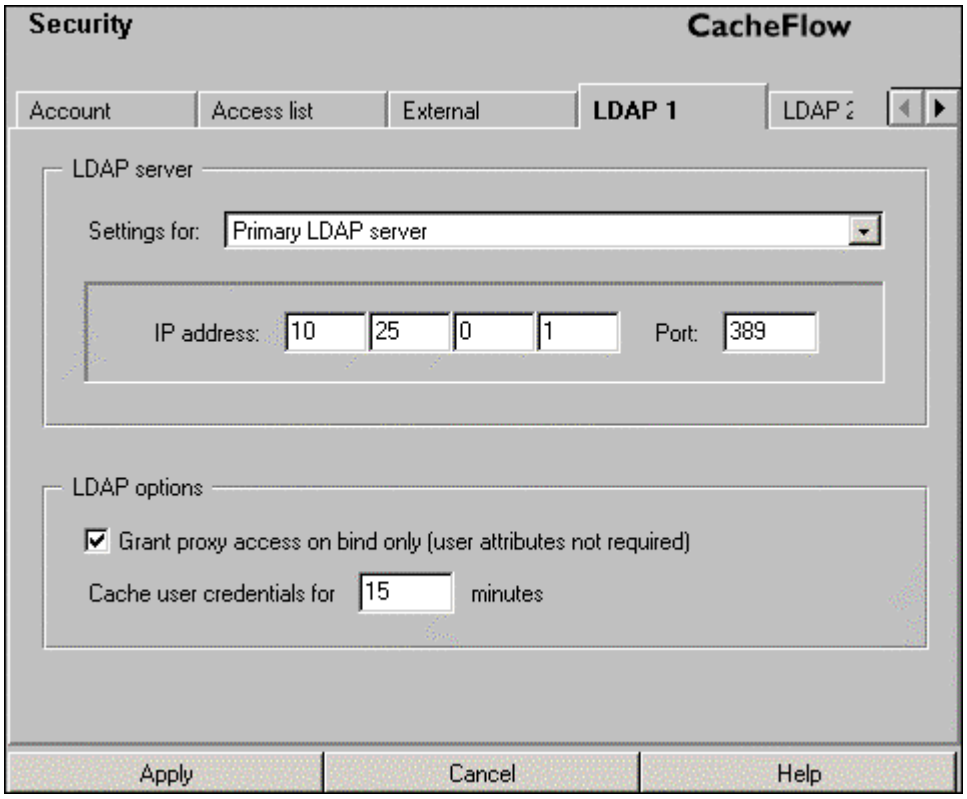


Figure 9-4 Setting Primary LDAP server address and options

4. Specify the IP address and Port for the Primary LDAP server. The default port is 389.
5. Optional: Specify the IP address and Port for the Alternate LDAP server. The default port is 389.

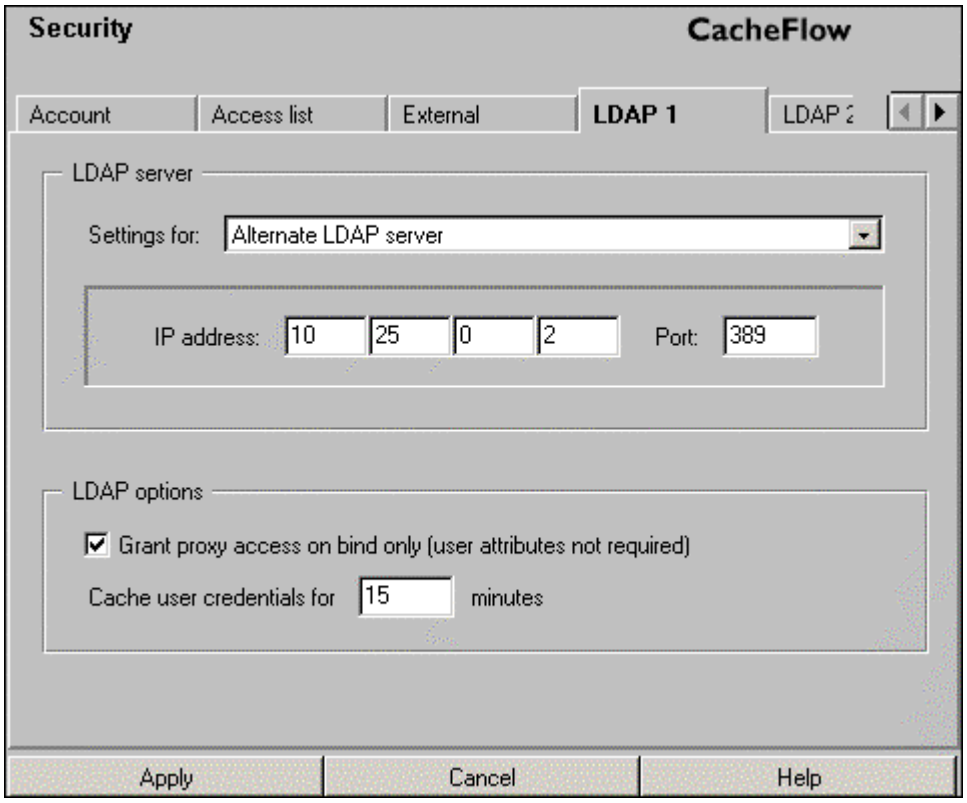


Figure 9-5 Setting Alternate LDAP server address and options

6. Specify the length of time in minutes that user credentials received from the LDAP server are cached by CacheOS. User credentials can be cached for up to 65535 minutes. This parameter applies to user entries only. Administrator credentials are not cached.

   Warning If you specify 0, user credentials are not cached. This increases traffic to the LDAP server because each HTTP get request results in CacheOS generating an authorization request to the server.

7. Optional: Select Grant proxy access on bind only (user attributes not required). The default setting is disabled. When enabled, Proxy users only need to specify their username and password in order to be granted proxy privileges to the Content Accelerator. Following a successful bind to the LDAP server, the Content Accelerator does not query the LDAP server to verify User Attributes.
8. Select the LDAP 2 tab.

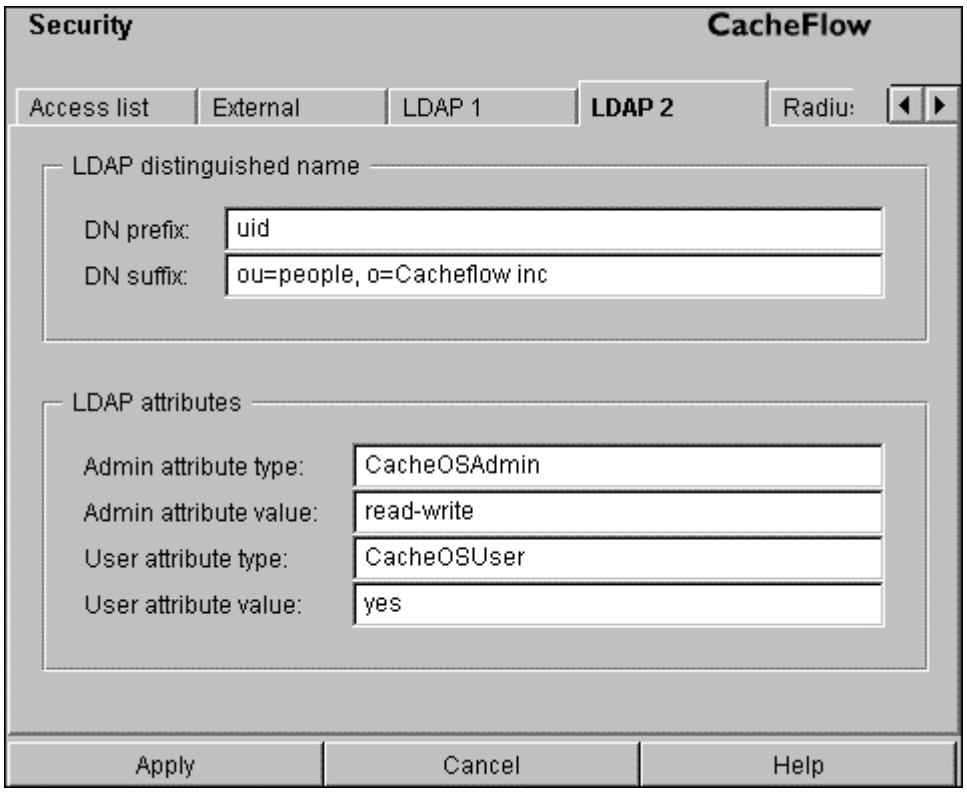


Figure 9-6 Configuring LDAP distinguished name and user attributes

9. Specify the LDAP distinguished name

   A distinguished name consists of a DN prefix and DN suffix.

   - DN prefix: Use an alphanumeric string of up to 32 characters to specify one unique key field in the LDAP user database. The default for DN prefix is uid.
   - DN suffix: Use an alphanumeric string of up to 64 characters to define an LDAP distinguished name. All users must be in the same location in the LDAP schema.

     Note Using the values specified in the preceding figure for DN prefix and DN suffix, the distinguished name would be DN = "uid=[some user id], ou=people, o=Cacheflow Inc"

     You must use a valid DN suffix. See the LDAP RFC at the RFC/STD/FYI/BCP archives for additional details. The archives are available at http://www.faqs.org.

10. Specify LDAP Admin attributes

    - Admin attribute type: Use this parameter to distinguish administrators from CacheOS proxy users. Use a case-sensitive string to specify the LDAP attribute type that identifies a given LDAP entry as one belonging to a CacheOS administrator. This attribute is checked by the LDAP database to determine whether or not an administrator associated with a given entry has read-write or read-only administrator privileges on the Content Accelerator. The default value is CacheOSAdmin. The maximum length of this string is 32 characters.

- Admin attribute value: Use a case-sensitive string to define the LDAP attribute value to be recognized as Boolean-True for the Admin attribute type read-write capability. The default value is read-write. The maximum length of this string is 32 characters.

11. Specify LDAP User attributes:
    - User attribute type: Use this parameter to distinguish users who have CacheOS user rights. Use a case sensitive string to specify the LDAP attribute type that identifies a given LDAP entry as one belonging to a CacheOS user. This attribute is checked by the LDAP CacheOS database to determine whether or not the user associated with a given entry has user privileges on the Content Accelerator. The default value is CacheOSUser. The maximum length of this string is 32 characters.
    - User attribute value: Use a case sensitive string to define the LDAP attribute value to be recognized as Boolean-True for the User attribute type. The default value is yes. The maximum length of this string is 32 characters.
12. Click Apply to save changes.
13. Select the External tab and enable the LDAP radio button, the Authenticate administrators, and Authenticate users check boxes.

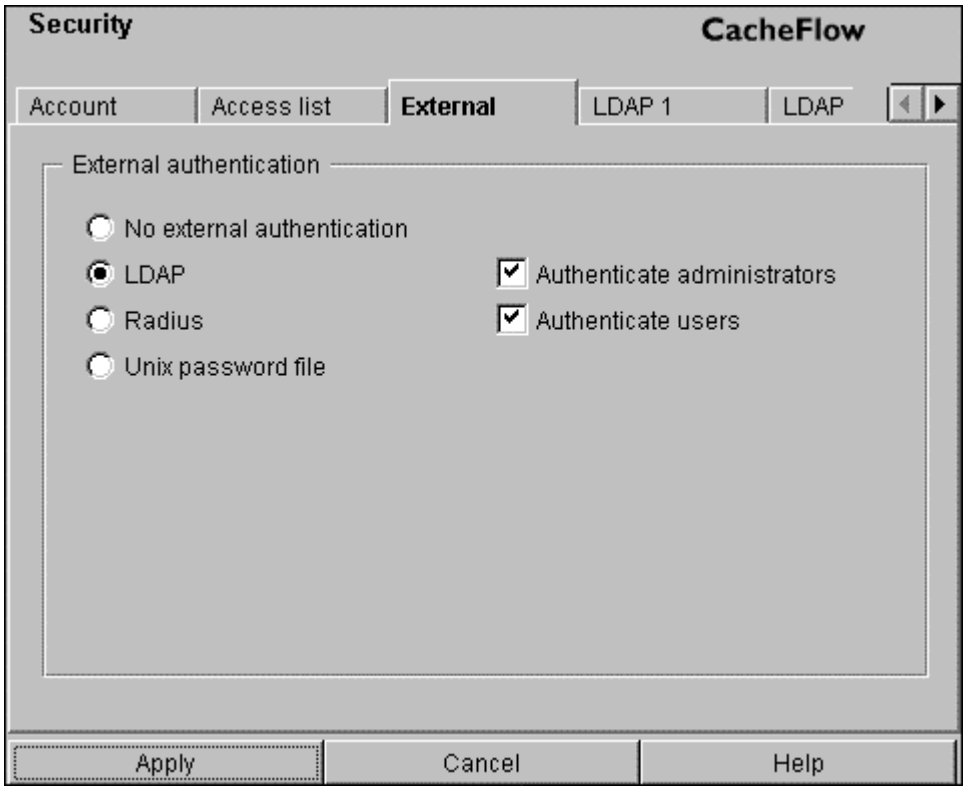


Figure 9-6 Enabling LDAP security

To configure authentication with LDAP using the CLI

1. Open a terminal session with the Content Accelerator and type your Username and Password when prompted.
2. At the command prompt, type **enable** and type your enable Password when prompted.

3. At the command prompt, type **configure terminal**.
4. At the command prompt, type **authentication protocol *ldap***
5. At the command prompt, type **authentication admin-verification *enable***
6. At the command prompt, type **authentication user-verification *enable***
7. At the command prompt, type **authentication *ldap***

   The prompt changes to (config ldap).
8. At the command prompt, type **primary-server ip *10.25.36.46***
9. At the command prompt, type **primary-server port *389***
10. At the command prompt, type **alternate-server ip *10.25.36.47***
11. At the command prompt, type **alternate-server port *389***
12. At the command prompt, type **cache-duration *15***

    Cache-duration specifies the length of time in minutes that user credentials received from the LDAP server are cached by CacheOS. User credentials can be cached for up to 65535 minutes. This parameter applies to user entries only. Administrator credentials are not cached.

    Warning If you specify 0, user credentials are not cached. This increases traffic to the LDAP server because each HTTP get request results in CacheOS generating an authorization request to the server.
13. Optional: At the command prompt, type **grant-access-on-bind**.
14. At the command prompt, type **distinguished-name prefix *uid***

    Use an alphanumeric string of up to 32 characters to specify one unique key field in the LDAP user database. The default for DN prefix is uid.
15. At the command prompt, type **distinguished-name suffix *ou=people, o=Cacheflow Inc***

    Use an alphanumeric string of up to 64 characters to define an LDAP distinguished name. All users must be in the same location in the LDAP schema.
16. At the command prompt, type **admin-attribute type *CacheOSAdmin***

    Use this parameter to distinguish administrators from CacheOS proxy users. Use a case-sensitive string to specify the LDAP attribute type that identifies a given LDAP entry as one belonging to a CacheOS administrator. This attribute is checked by the LDAP database to determine whether or not an administrator associated with a given entry has read-write or read-only administrator privileges on the Content Accelerator. The default value is CacheOSAdmin. The maximum length of this string is 32 characters.
17. At the command prompt, type **admin-attribute value *read-write***

    Use a case-sensitive string to define the LDAP attribute value to be recognized as Boolean-True for the Admin attribute type read-write capability. The default value is read-write. The maximum length of this string is 32 characters.
18. At the command prompt, type **user-attribute type *CacheOSUser***

    Use this parameter to distinguish users who have CacheOS user rights. Use a case sensitive string to specify the LDAP attribute type that identifies a given LDAP entry as one belonging to a CacheOS user. This attribute is checked by the LDAP CacheOS database to determine whether or not the user associated with a given entry has user privileges on the Content Accelerator. The default value is CacheOSUser. The maximum length of this string is 32 characters.
19. At the command prompt, type **user-attribute value *yes***

    Use a case sensitive string to define the LDAP attribute value to be recognized as Boolean-True for the User attribute type. The default value is yes. The maximum length of this string is 32 characters.

Configuring Authentication with RADIUS

CacheOS supports the use of RADIUS as a communications mechanism for accessing an external database to authenticate administrators and HTTP proxy users. The CacheOS Authentication software uses RADIUS to authenticate CacheOS Administrators or HTTP Proxy Users with the related RADIUS server.

Note Transparent mode caching should be disabled before selecting and configuring external authentication.

RADIUS Authentication Notes

- The Console account user name and password remain valid, and the specified user always has administrator read/write privileges, even when using RADIUS authentication. If the RADIUS server fails, the Console account username and password can still be used to access the Management console.

  Important! You should always define a Console account user name and password. If the user name and password fields are left blank, the Management Console can be accessed without specifying a username and password.

- RADIUS authentication provides the ability to define:
  - Additional administrators who have Read/Write and Read only access authorization to the Content Accelerator
  - Proxy users
- RADIUS authentication applies to the Management console and CLI commands.
- Read Write access allows administrators to access the CacheFlow Management console and all of the CLI commands.
- CacheOS caches proxy user credentials but does not cache administrator credentials.
- The enable password is still required for the CLI.
- Read only access lets administrators view the Management console but not make any changes. It also allows access to any unprivileged CLI commands. Unprivileged CLI commands are those that do not require the enable password.
- The RADIUS server must know the IP address and the shared secret of the related RADIUS client, in this case, the Content Accelerator.

Detailed RADIUS information is available in the RFC/STD/FYI/BCP archives at http://www.faqs.org.

To configure authentication with RADIUS

1. Select Management from the CacheOS home page.
2. Select the Security applet.
3. Select the Radius tab.
4. Specify the IP address and Port for the Primary RADIUS server. The default port is 1812.

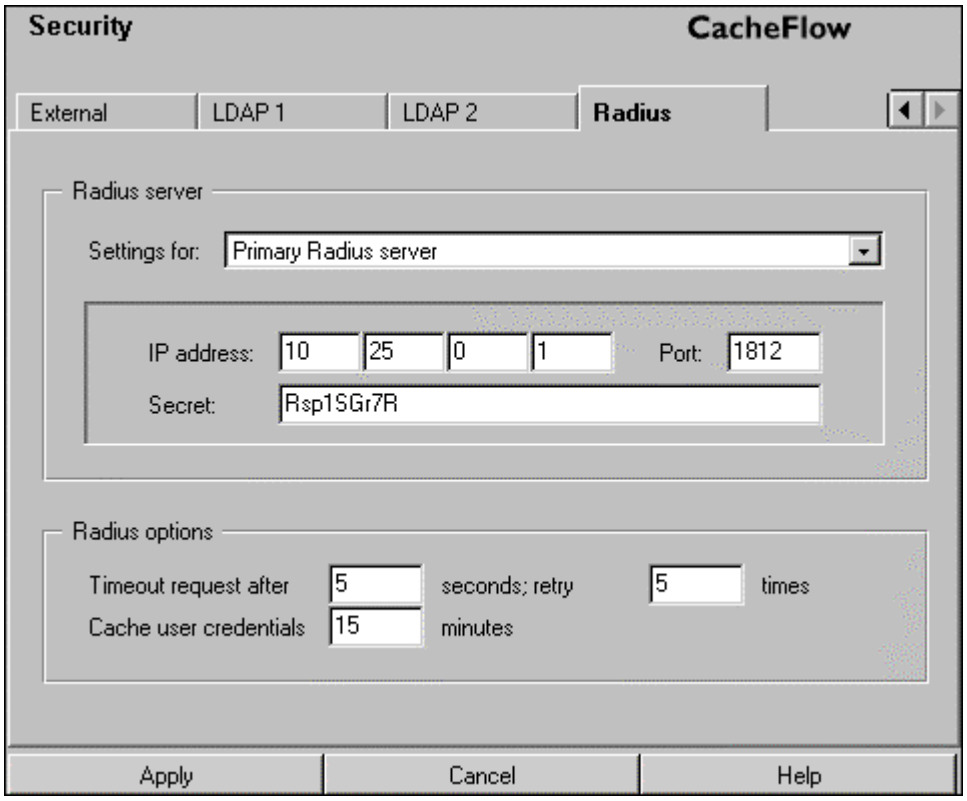


Figure 9-7 Setting primary RADIUS server and options

5. Optional: Specify the IP address and Port for the Alternate RADIUS server. The default port is 1812.

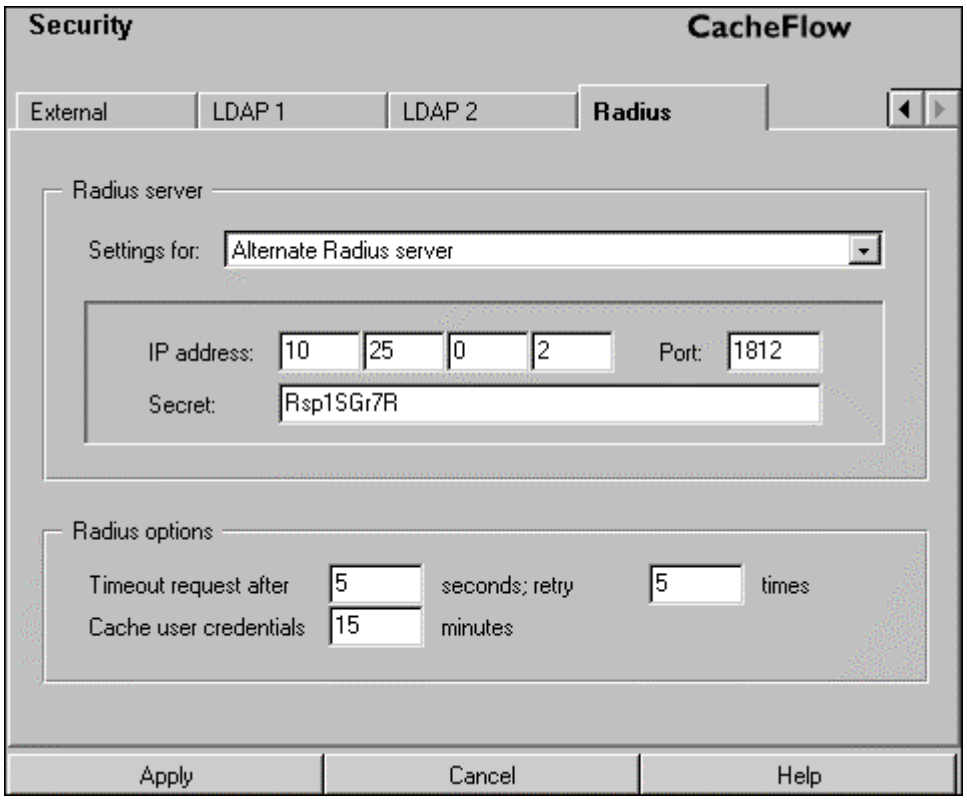


Figure 9-8 Setting alternate RADIUS server and options

6. Specify a shared secret in the Secret field. This is a case-sensitive alphanumeric string, up to 32 characters long, which is known only to the RADIUS server and the Content Accelerator.
7. In the Timeout request after field, enter a value in seconds. Possible values are 0 – 65535 minutes.
8. In the Retry field, enter a value for the number of retries to attempt. Possible values for retry are 0 – 65535.
9. In the Cache user credentials field, enter in the number of minutes that user credentials received from the RADIUS server are cached.

   Possible values for caching user credentials are 0 – 65535 minutes. User credentials are not cached if a value 0 is entered. This parameter applies to user entries only. Administrator credentials are not cached.
10. Click Apply to save changes.
11. Select the External tab and enable the Radius radio button, the Authenticate administrators, and Authenticate users check boxes.

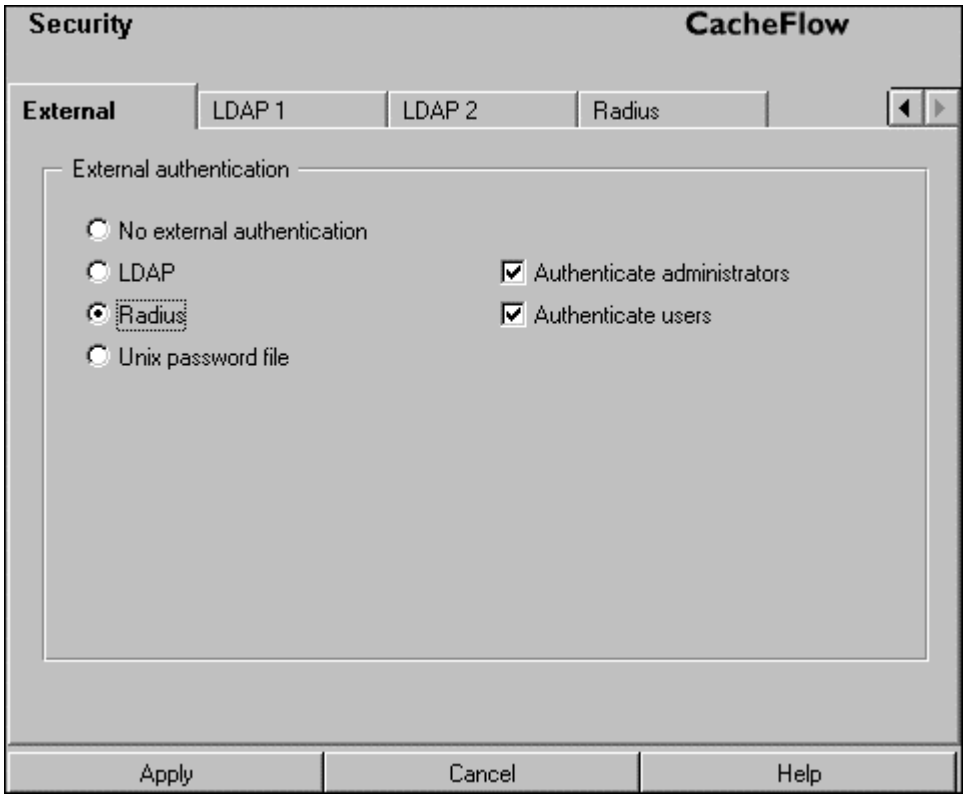


Figure 9-9 Enabling Radius security

To configure authentication with RADIUS using the CLI

1. Open a terminal session with the Content Accelerator and type your Username and Password when prompted.
2. At the command prompt, type **enable** and type your Password when prompted.
3. At the command prompt, type **configure terminal**.
4. At the command prompt, type **authentication protocol *radius***
5. At the command prompt, type **authentication admin-verification *enable***
6. At the command prompt, type **authentication user-verification *enable***
7. At the command prompt, type **authentication *radius***

   The prompt changes to `(config radius)`.
8. At the command prompt, type **primary-server ip *10.25.36.46***
9. At the command prompt, type **primary-server port *1812***
10. At the command prompt, type **primary-server secret *12aAbB759ABCGFgf***

    Specify a shared secret in the Secret field. This is a case-sensitive alphanumeric string, up to 32 characters long, which is known only to the RADIUS server and the Content Accelerator.
11. At the command prompt, type **alternate-server ip *10.25.36.47***
12. At the command prompt, type **alternate-server port *1812***

13. At the command prompt, type **alternate-server secret *12aAbB759ABCGFgf***

    Specify a shared secret in the Secret field. This is a case-sensitive alphanumeric string, up to 32 characters long, which is known only to the RADIUS server and the Content Accelerator.

14. At the command prompt, type **query-timeout *10***

    Specify a value in seconds. Possible values for timeout are 0 – 65535.

15. At the command prompt, type **server-retry *15***

    Specify a value for the number of retries to attempt. Possible values for retry are 0 – 65535.

16. At the command prompt, type **cache-duration *15***

    Specify the length of time in minutes that user credentials received from the RADIUS server are cached by CacheOS. User credentials can be cached for up to 65535 minutes. This parameter applies to user entries only. Administrator credentials are not cached.

    Warning If you specify 0, user credentials are not cached. This increases traffic to the RADIUS server because each HTTP get request results in CacheOS generating an authorization request to the server.

RADIUS Server Configuration

RADIUS Attributes

RADIUS Attributes carry specific information related to authentication and authorization, plus information and configuration details for the Access-Request and Access-Reply packets. Following is a list of the attributes supported from CacheOS and their intended use. These attributes are configured on the RADIUS server.

- User-Name: Indicates the name of the user to be authenticated. It is only used in Access-Request packets.
- User-Password: Indicates the password of the user to be authenticated. This attribute is used in Access-Request packets.
- NAS-IP-Address: Indicates the identifying IP Address of the NAS, i.e. the CacheOS, which is requesting authentication of the user. It is only used in Access-Request packets. CacheOS will always include the NAS-IP-Address in an Access-Request packet.
- Service-Type: Indicates the type of service the user has requested, or the type of service to be provided. It is used in both Access-Request and Access-Accept packets. Valid types included:
  - Administrative: Used for read-write admin access
  - NAS Prompt: Used for read-only admin access
  - Authenticate Only: Used for proxy access
- Reply-Message: Indicates text, which can be recorded along with transaction information. When used in an Access-Accept, it is the success message. When used in an Access-Reject, it is the failure message.

Radius Server Settings

- CacheOS users in the Radius Server database should be configured with the following attributes:
  - Service-Type Check List
  - The Service Type attribute depends on the type of user. Use one of the values described in the previous section.

  - Return List attribute.
- The Cache Machine should be configured as a Standard Radius client.
- The RADIUS server must know the IP address and the shared secret of the related RADIUS client, in this case, the Content Accelerator.

Tracking Client IP Adresses Using Server-Side Tranparency

CacheOS supports client-side transparency and server-side transparency.

Client-side transparency masks the Content Accelerator IP address with the Web server IP address for all port 80 traffic destined to the client. This effectively conceals the Content Accelerator address from the client, making the Content Accelerator transparent to the client. It also effectively conceals the identity of the client from the Web server.

Under some circumstances, however, tracing a client is very important. Server-side transparency provides this capability. When server-side transparency is enabled, CacheOS retains client IP addresses for all port 80 traffic to and from the Content Accelerator. In this scheme, the client IP address is always revealed to the server, allowing the server to keep accurate records of what client accessed the server on a given date at a given time.

When client-side transparency and server-side transparency are both enabled, the routing device and the Content Accelerator are transparent to the client and to the server. To the client and to the server it appears as if they are communicating directly with each other.

The following diagram details IP addresses used, and the route a request takes when server-side transparency is enabled in a scenario using a Layer 4 switch.

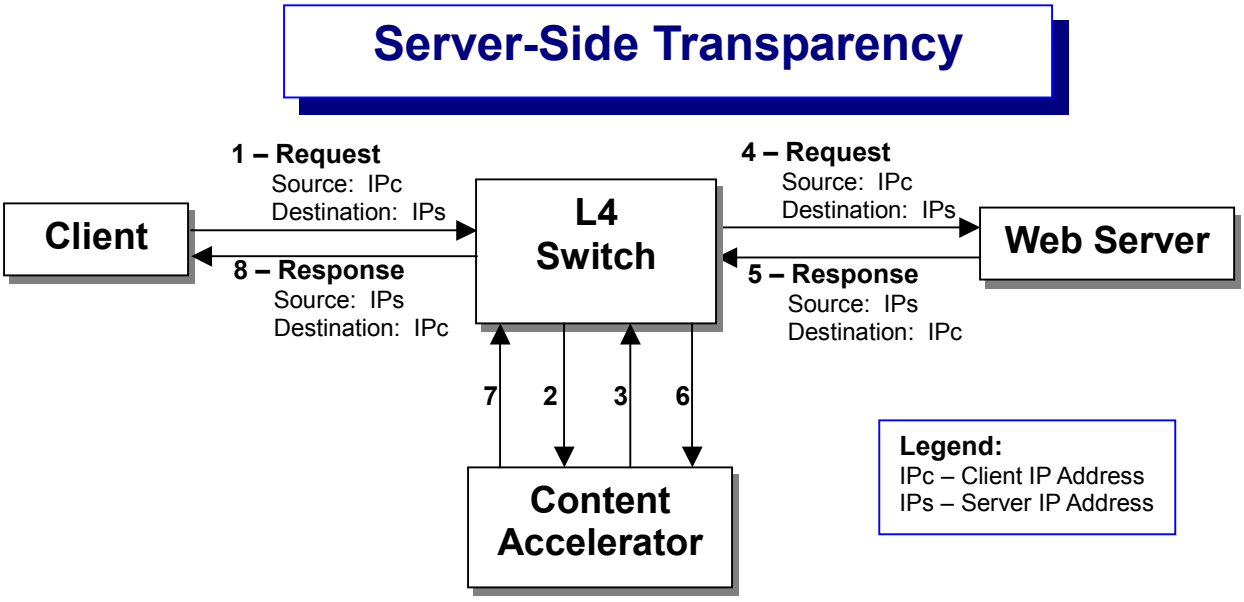


Figure 9-10 Information flow in server-side transparency

Note The routing device must be configured for client-side transparency, server-side transparency or both to work properly. Client-side transparency and server-side transparency can operate with multiple Content Accelerators or with groups of routing devices. The network, however, must be configured so that the same route is used between a client and a server for sending and receiving data because of port-recognition protocols.

Configuring Server-Side Transparency

Server-side transparency is configurable in the Management Console through the HTTP ports tab by clicking the Send client's IP address to server (proxy is transparent to server) option:

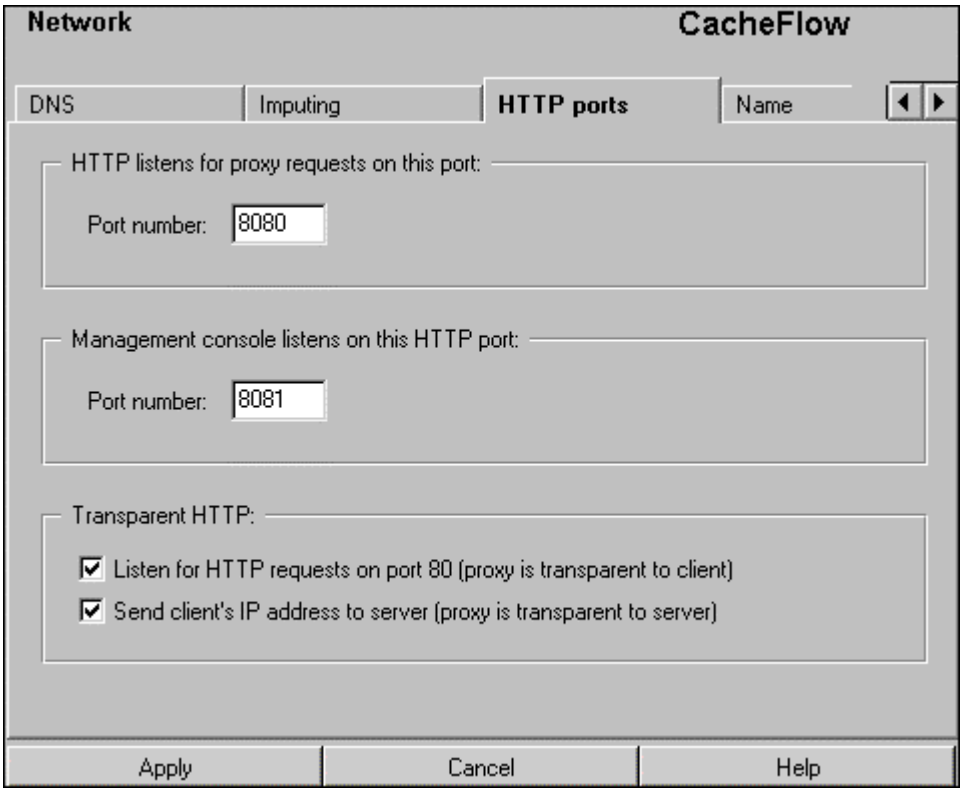


Figure 9-11 Configuring server-side transparency

After clicking this option, a confirmation prompt appears:

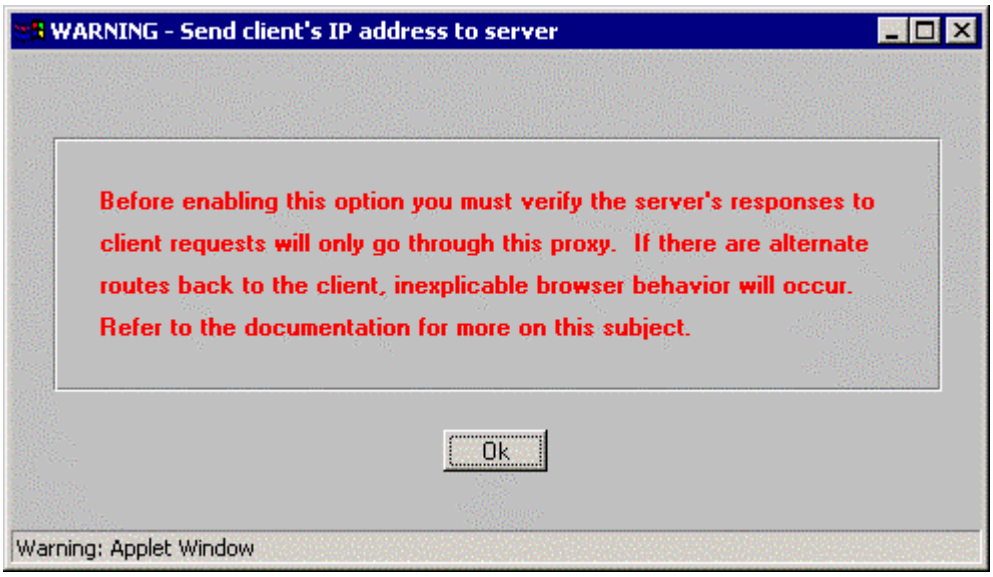


Figure 9-12 Confirmation prompt

Complete the verification if such verification needs to be done. Click Ok, and then click Apply on the HTTP ports tab.

To disable this feature, click the Send client's IP address to server (proxy is transparent to server) option so that the check mark does not appear and then click Apply on the HTTP ports tab.

Configuring Server-Side Transparency in the CLI

Server-side transparency is configurable in the CLI through the **transparent-proxy** command and the subcommand **send-client-ip**.

Example: Enabling Server-Side Transparency

```
CacheFlow#(config)transparent-proxy send-client-ip
  ok
```

Other subcommands under the command **transparent–proxy** are **disable**, **enable** and **no**. The **disable** and **enable** subcommands pertain to client-side transparency only. The **no** subcommand works for either client-side transparency or server-side transparency.

Example: Disabling Server-Side Transparency

```
CacheFlow#(config)transparent-proxy no send-client-ip
  ok
```

Object Pipelining and Object Refreshing in Server-Side Transparency

This table details what IP addresses are logged to the Web server when object pipelining and object refreshing occur:

Function	IP Address Logged to the Web Server
Object Pipelining	IP address of the first client that opened the TCP/IP connection is used.
Object Refreshing	**Refresh initiated by Content Accelerator** The Content Accelerator IP address is used. **Refresh initiated by client** The client IP address is used.

This page intentionally blank.

Chapter 10 - Configuring SNMP

A Content Accelerator can be viewed using a Simple Network Management Protocol (SNMP) management station. CacheOS supports MIB-2 (RFC 1213), Proxy MIB, and the RFC2594 MIB.

Enabling SNMP

To view a Content Accelerator from an SNMP management station, SNMP support must first be enabled and configured on the Content Accelerator.

To enable and configure SNMP

1. Select Management from the CacheOS home page.
2. Select the SNMP applet.
3. Check the Enable SNMP checkbox.
4. In the sysLocation field, enter a string that describes the Content Accelerator's physical location.
5. In the sysContact field, enter a string that identifies the person responsible for administering the Content Accelerator.
6. Click Apply to save changes.

To enable and configure SNMP using the CLI

1. Open a terminal session with the Content Accelerator and type your Username and Password when prompted.
2. At the command prompt, type **enable** and type your Password when prompted.
3. At the command prompt, type **configure terminal**.
4. At the `(config)` prompt type **snmp** to enter snmp configuration mode.

   The command prompt changes to `(config snmp)`.
5. At the `(config snmp)` prompt, type **enable** to enable SNMP.
6. At the `(config snmp)` prompt, type **sys-location** ***location*** to specify the Content Accelerator's physical location.
7. At the `(config snmp)` prompt, type **sys-contact** ***contact*** to identify the person responsible for administering the Content Accelerator.
8. Type **exit** to return to the `(config)` prompt.

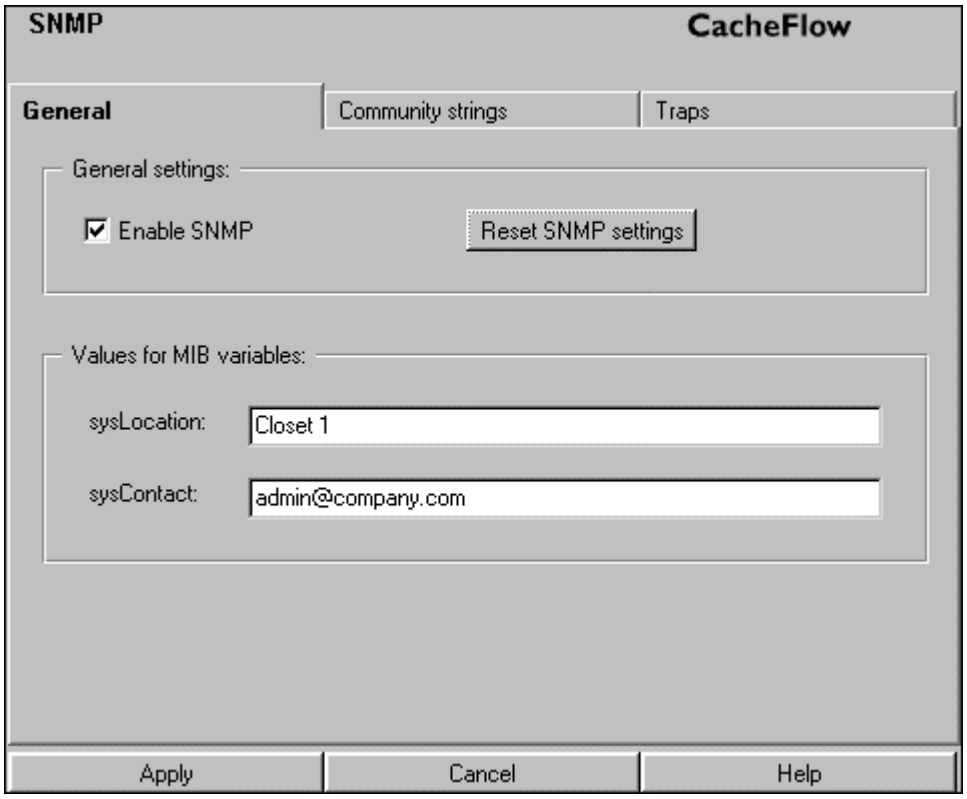


Figure 10-1 Enabling SNMP

Configuring SNMP Community Strings

Community strings are used to restrict access to SNMP data. To read SNMP data on the Content Accelerator, specify a read community string. To write SNMP data to the Content Accelerator, specify a write community string. To receive traps, specify a trap community string. By default, all community string passwords are set to public.

Security Caution If you enable SNMP, make sure to change all 3 community string passwords to values that are hard to guess. Use a combination of uppercase, lowercase, and numeric characters. An easily guessed community string password makes it easier to gain unauthorized access to the Content Accelerator and network.

To configure community strings

1. Select Management from the CacheOS home page.
2. Select the SNMP applet.
3. Select the Community strings tab.
4. Enter a read community password in the Read community field.
5. Enter a write community password in the Write community field.

6. Enter a trap community password in the Trap community field.
7. To save changes, click Apply.

To configure community strings using the CLI

1. Open a terminal session with the Content Accelerator and type your Username and Password when prompted.
2. At the command prompt, type **enable** and type your Password when prompted.
3. At the command prompt, type **configure terminal**.
4. At the `(config)` prompt type **snmp** to enter snmp configuration mode.
   The command prompt changes to `(config snmp)`.
5. At the `(config snmp)` prompt, type **enable** to enable SNMP.
6. At the `(config snmp)` prompt, type **read-community** ***password*** to specify the password for read privileges.
7. At the `(config snmp)` prompt, type **write-community** ***password*** to specify the password for write privileges.
8. At the `(config snmp)` prompt, type **trap-community** ***password*** to specify the password to receive traps.
9. Type **exit** to return to the `(config)` prompt.

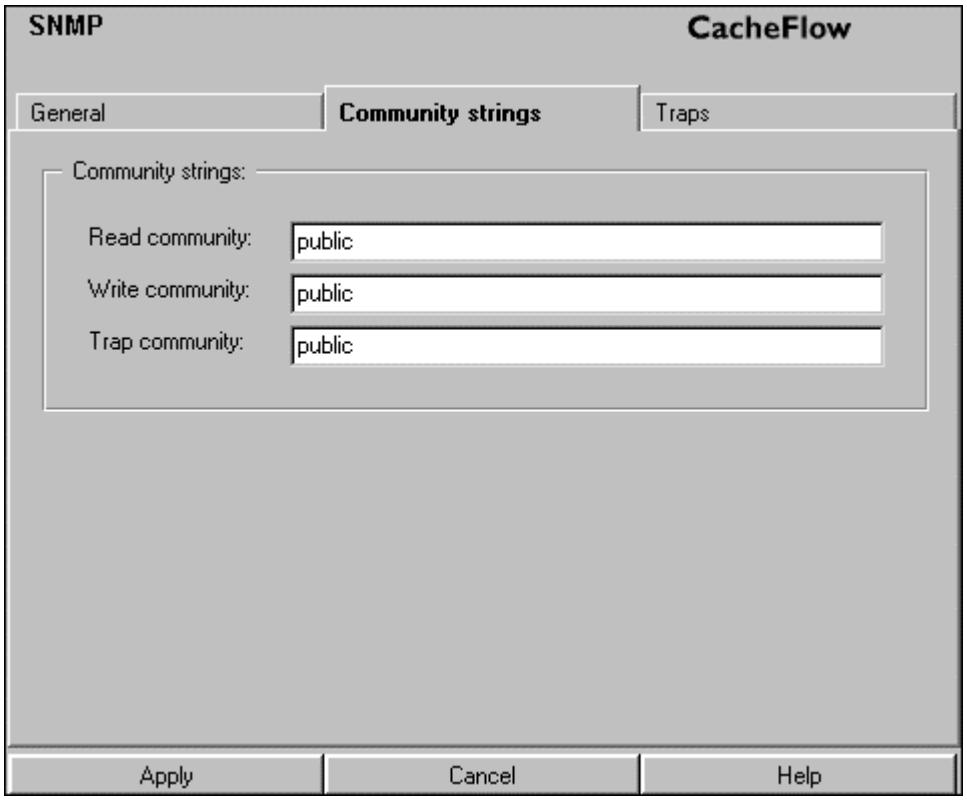


Figure 10-2 Configuring SNMP community strings

Configuring SNMP Traps

CacheOS can send SNMP traps to a management station as they occur. By default, all system-level traps are sent to the address specified. You can also enable authorization traps to send notification of attempts to access the CacheOS Management console.

To enable SNMP traps

1. Select Management from the CacheOS home page.
2. Select the SNMP applet.
3. Select the Traps tab.
4. In the Send traps to fields, Enter the IP address(es) of the workstation(s) where traps are to be sent.
5. To receive authorization traps, activate the Enable authorization traps checkbox.
6. To save changes, click Apply.

To enable SNMP traps using the CLI

1. Open a terminal session with the Content Accelerator and type your Username and Password when prompted.
2. At the command prompt, type **enable** and type your Password when prompted.
3. At the command prompt, type **configure terminal**.
4. At the `(config)` prompt type **snmp** to enter snmp configuration mode.

   The command prompt changes to `(config snmp)`.
5. At the `(config snmp)` prompt, type **enable** to enable SNMP.
6. At the `(config snmp)` prompt, type **trap-address 1** ***10.25.36.47*** to specify the IP address for trap 1.

   To add additional trap addresses, repeat step 6 substituting **trap-address 2** or **trap-address 3** to specify the IP address for traps 2 and 3.
7. Optional: At the `(config snmp)` prompt, type **authorize-traps** to enable authorization traps.
8. Type **exit** to return to the `(config)` prompt.

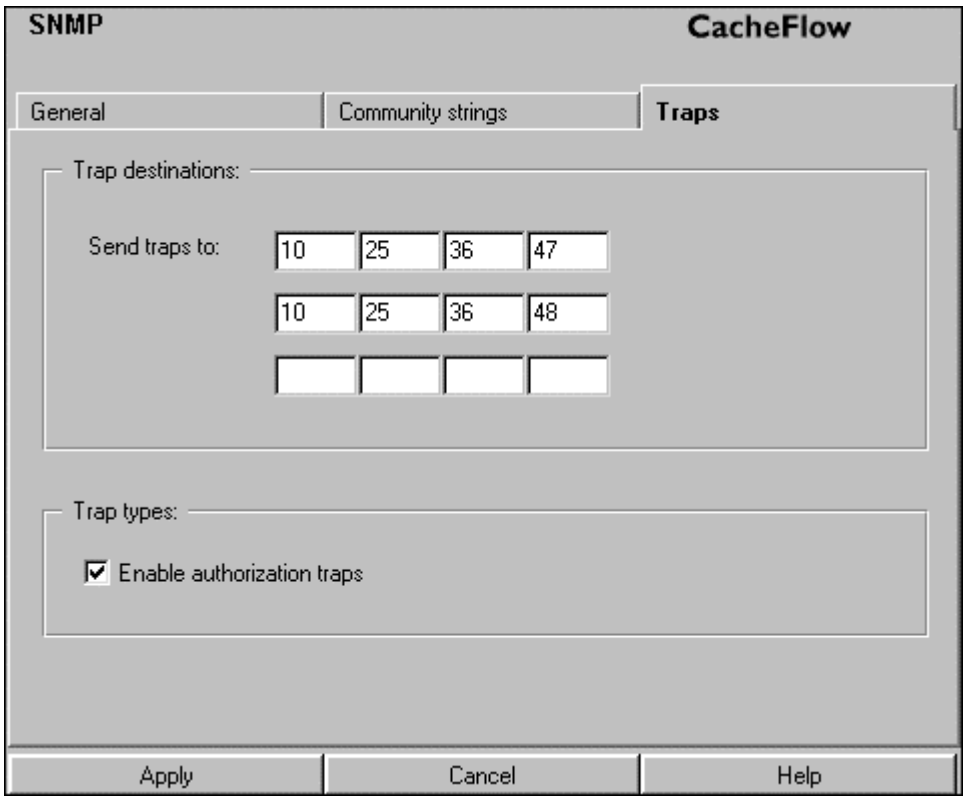


Figure 10-3 Configuring SNMP traps

This page intentionally blank.

Chapter 11 - Configuring Access Logging

CacheOS can maintain an access log for each HTTP request made. The access log can be stored in one of two formats, which can be read by a variety of reporting utilities. Refer to the Setting the Access Log Format section for additional information.

When you enable URL access logging, the CacheOS logs every client HTTP request. The access log is uploaded to a FTP host, based upon the default upload schedule, or a schedule you specify. The access log can be interpreted using popular HTTP log reporting programs.

To enable access logging

1. Select Management from the CacheOS home page.
2. Select the Logging applet.
3. On the General tab, select the Enable URL access logging checkbox.
4. Click Apply to save changes.

To enable access logging using the CLI

1. Open a terminal session with the Content Accelerator and type your Username and Password when prompted.
2. At the command prompt, type **enable** and type your Password when prompted.
3. At the command prompt, type **configure terminal**.
4. At the `(config)` prompt type **access-log** to enter access log configuration mode.
   The command prompt changes to `(config access-log)`.
5. At the `(config access-log)` prompt, type **enable** to enable access logging.
6. Type **exit** to return to the `(config)` prompt.

Note In addition to enabling access logging, you must also configure the related upload site, upload schedule, and log format.

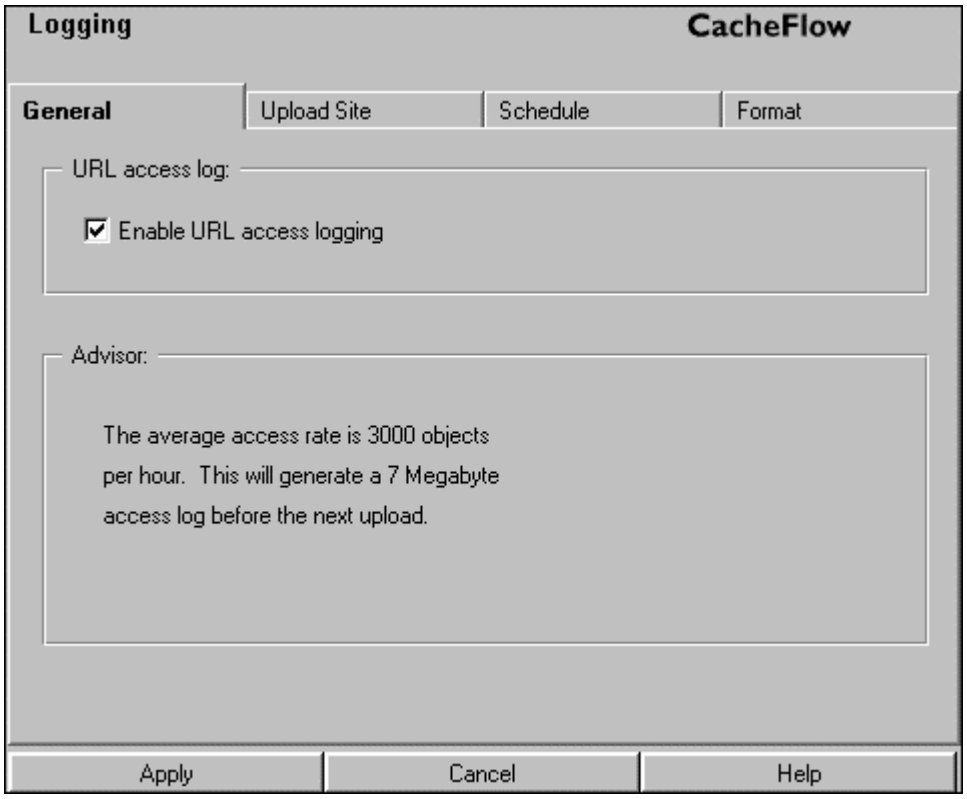


Figure 11-1 Enabling access logging

Setting the Access Log Upload Site

CacheOS uploads the access log to an FTP server based on an upload schedule. You must specify the FTP server address, directory, and login information.

To configure the access log upload site

1. Select Management from the CacheOS home page.
2. Select the Logging applet.
3. Select the Upload Site tab.
4. Enter the Filename prefix if needed.
5. Enter the FTP server address in the Host field.
6. Enter the directory path on the FTP server in the Path field.
7. Enter the username to log into the FTP server in the username field.
8. Enter the password for the username in the password field.
9. Click Apply to save changes.

To configure the access log upload site using the CLI

1. Open a terminal session with the Content Accelerator and type your Username and Password when prompted.
2. At the command prompt, type **enable** and type your Password when prompted.
3. At the command prompt, type **configure terminal**.
4. At the `(config)` prompt type **access-log** to enter access log configuration mode.

   The command prompt changes to `(config access-log)`.
5. At the `(config access-log)` prompt, type **filename-prefix** ***prefix*** to specify the log filename prefix.
6. At the `(config access-log)` prompt, type **primary host** ***FTP URL*** to specify the primary FTP server to which logs should be uploaded.
7. At the `(config access-log)` prompt, type **primary path** ***path*** to specify the directory on the primary FTP server to which logs should be uploaded.
8. At the `(config access-log)` prompt, type **primary username** ***username*** to specify the username on the primary FTP server to which logs should be uploaded. The username must have write privileges in the access log upload directory.
9. At the `(config access-log)` prompt, type **primary password** ***password*** to specify the password for the username specified in step 8.
10. Type **exit** to return to the `(config)` prompt.

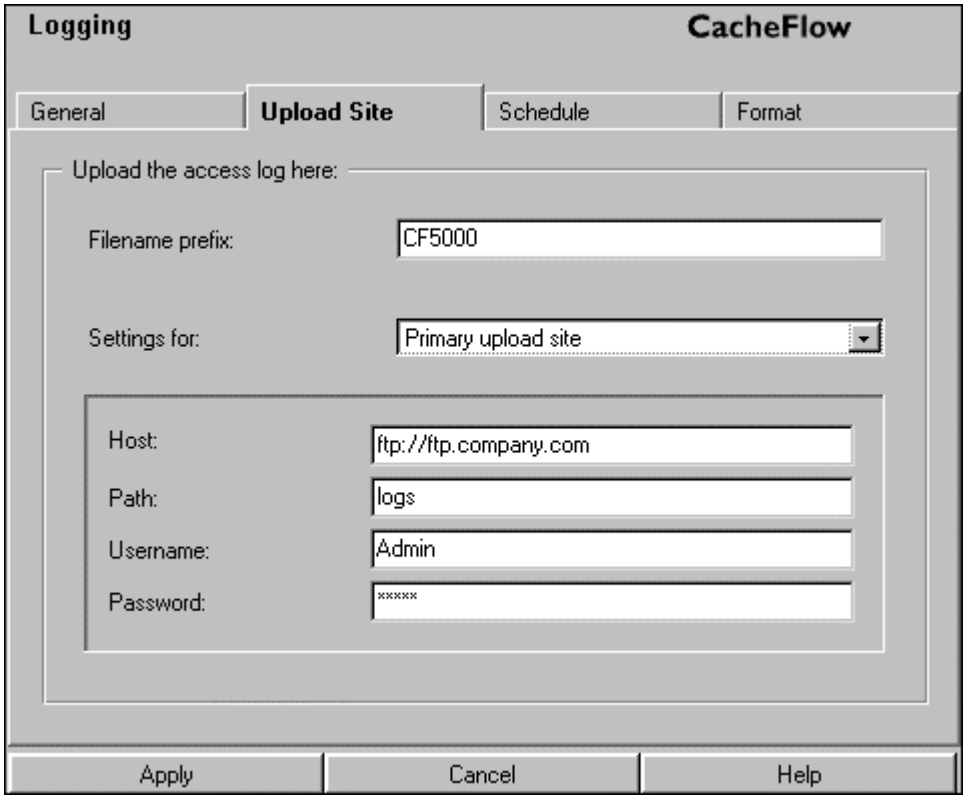


Figure 11-2 Setting the access log upload site

Specifying an Alternate Upload Site

If the primary FTP server is not available, CacheOS uploads the access log to an alternate FTP server if one is specified. To specify settings for the alternate FTP server, repeat the previous Web or CLI procedure, making sure to specify alternate upload site information.

Setting the Access Log Upload Schedule

CacheOS uploads the access log to an FTP server based on the upload schedule configured.

To set the upload schedule

1. Select Management from the CacheOS home page.
2. Select the Logging applet.
3. Select the Schedule tab.
4. Set the schedule.
5. Set contingency options.
6. Click Apply to save changes.
7. Select the General tab to view an estimate of how large the log will grow based on the current schedule.

   The Advisor notice on the General tab describes how large the access log will grow based on the current schedule and usage. You can use this information to determine an appropriate schedule based on your usage.

To set the upload schedule using the CLI

These settings apply to both the primary and alternate FTP servers.

1. Open a terminal session with the Content Accelerator and type your Username and Password when prompted.
2. At the command prompt, type **enable** and type your Password when prompted.
3. At the command prompt, type **configure terminal**.
4. At the `(config)` prompt type **access-log** to enter access log configuration mode.

   The command prompt changes to `(config access-log)`.
5. At the `(config access-log)` prompt, type **upload hourly *0-65535* or upload daily *0-23*** to specify how often the access log is to be uploaded.
6. At the `(config access-log)` prompt, type **primary host *FTP URL*** to specify the primary FTP server to which logs should be uploaded.
7. At the `(config access-log)` prompt, type **action upload** or **action stop** to specify what to do if access log exceeds its allotted size
8. At the `(config access-log)` prompt, type **threshold *1-100*** to specify the percentage of disk access log can consume.
9. Type **exit** to return to the `(config)` prompt.

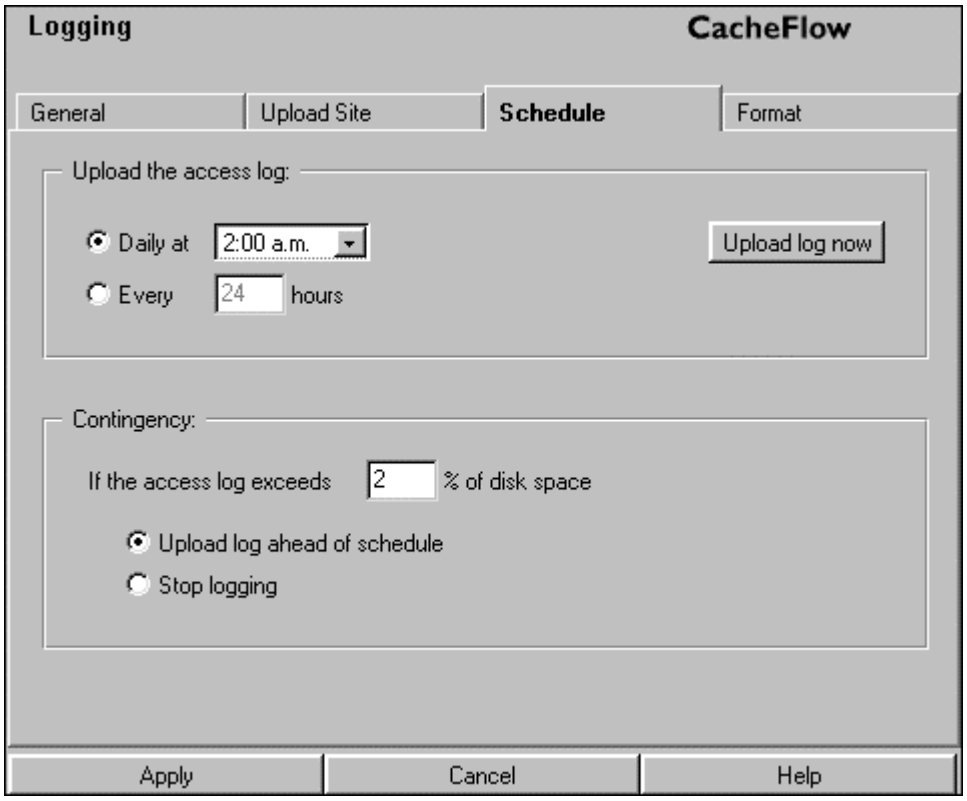


Figure 11-3 Setting the access log upload schedule

Setting the Access Log Format

CacheOS can save the access log in common log format, Squid-compatible format, or a custom format. Refer to Appendix A for detailed information on log formats.

To set the access log format

1. Select Management from the CacheOS home page.
2. Select the Logging applet.
3. Select the Format tab.
4. Select the format you want to use.
5. Click Apply to save changes.

To set the access log format using the CLI

1. Open a terminal session with the Content Accelerator and type your Username and Password when prompted.
2. At the command prompt, type **enable** and type your Password when prompted.
3. At the command prompt, type **configure terminal**.
4. At the `(config)` prompt, type **access-log** to enter access log configuration mode.

The command prompt changes to `(config access-log)`.

5. At the `(config access-log)` prompt, type **format common** or **format squid-compatible** to specify the log format. To use a custom format, type **format custom** ***format string***. Refer to the Log Formats section for information about creating a custom log format.
6. Type **exit** to return to the `(config)` prompt.

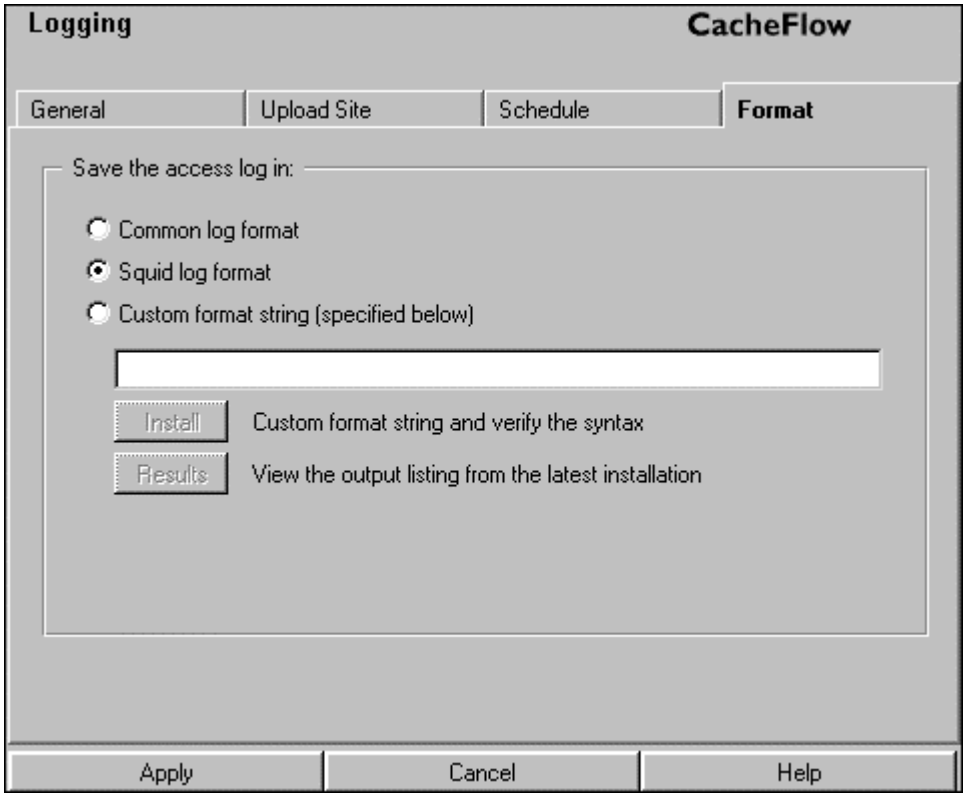


Figure 11-4 Setting the access log format

Uploading the Access Log on Demand

You can force an upload of the access log at any time by clicking Upload log now.

To upload the access log on demand using the CLI

1. Open a terminal session with the Content Accelerator and type your Username and Password when prompted.
2. At the command prompt, type **enable** and type your Password when prompted.
3. At the command prompt, type **upload access-log**.

Chapter 12 - Event Logging and Notification

The Content Accelerator can be configured to log system events as they occur. Event logging allows you to specify the types of system events logged, the size of the event log, and to configure Syslog monitoring. CacheOS can also notify you by email if an event is logged.

Configuring Which Events to Log

The event level options are listed from the most to least important events. Because each event requires some disk space, setting the event logging to log all events will fill the event log more quickly. The event log size is set on the Size tab of the Events applet.

To set the event logging level

1. Select Management from the CacheOS home page.
2. Select the Event applet.
3. Select the Level tab.
4. Select the events you want to log. When you select an event level, all levels above the selection are included. For example, if you select verbose, all event levels will be included.
5. Click Apply to save changes.

To set the event logging level using the CLI

1. Open a terminal session with the Content Accelerator and type your Username and Password when prompted.
2. At the command prompt, type **enable** and type your Password when prompted.
3. At the command prompt, type **configure terminal**.
4. At the `(config)` prompt type **event-log** to enter event log configuration mode.

   The command prompt changes to `(config event-log)`.
5. At the `(config event-log)` prompt, type **level severe**, **level resource**, **level informational**, or **level verbose** to specify what level events are to be logged.
6. Type **exit** to return to the `(config)` prompt.

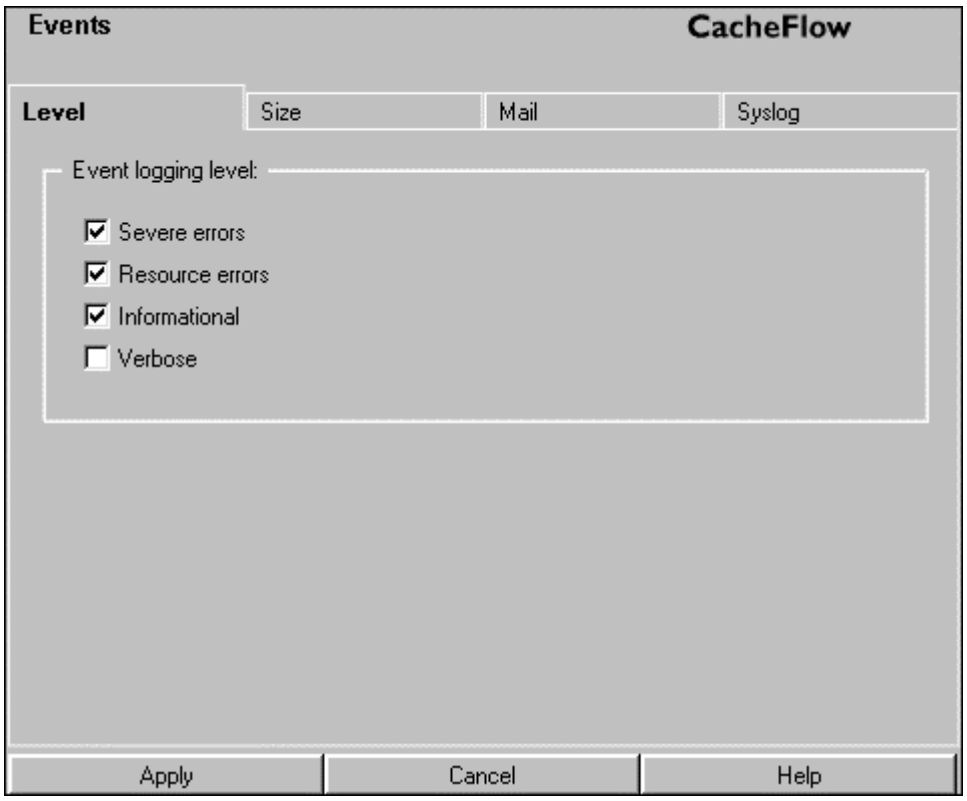


Figure 12-1 Selecting which events are logged

Setting Event Log Size

You can limit the size of the Content Accelerator's event log and specify what CacheOS should do if the log size limit is reached.

To set the event log size

1. Select Management from the CacheOS home page.
2. Select the Event applet.
3. Select the Size tab.
4. In the Event log size box, enter the maximum size of the event log in megabytes.
5. Select either Overwrite earlier events or Stop logging new events to specify the desired behavior when the event log reaches maximum size
6. Click Apply to save changes.

To set the event log size using the CLI

1. Open a terminal session with the Content Accelerator and type your Username and Password when prompted.
2. At the command prompt, type **enable** and type your Password when prompted.
3. At the command prompt, type **configure terminal**.

4. At the `(config)` prompt, type **event-log** to access event log configuration mode.
   The command prompt changes to `(config event-log)`.
5. At the `(config event-log)` prompt, type **log-size *size*** to specify that maximum event log size in megabytes.
6. At the `(config event-log)` prompt, type **when-full overwrite**, or **when-full stop** to specify event logging behavior should the event log become full.
7. Type **exit** to return to the `(config)` prompt.

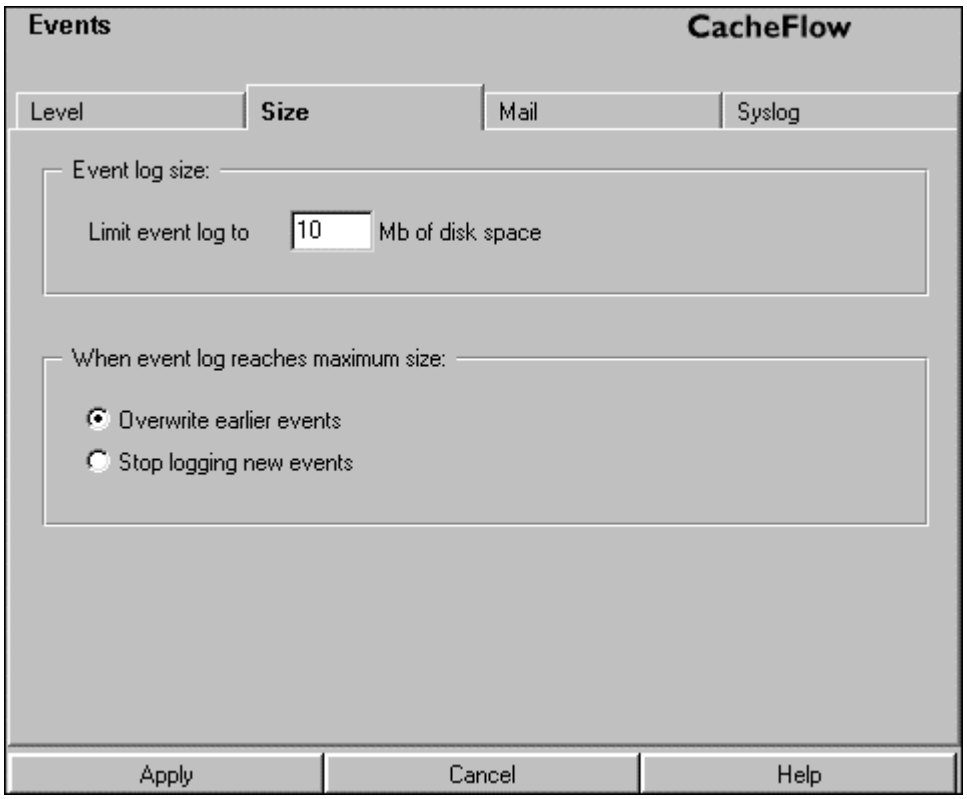


Figure 12-2 Configuring event log size

Enabling Event Notification

CacheOS can send event notifications to Internet email addresses using SMTP. You can also send event notifications directly to CacheFlow for support purposes. See the Configuring Diagnostic Reporting section for related information.

To enable event notifications

1. Select Management from the CacheOS home page.
2. Select the Events applet.
3. Select the Mail tab.
4. Click New to add a new email address.

5. Enter the host name of your mail server in the SMTP gateway name field, or enter the IP address of your mail server in the SMTP gateway IP field.

   CacheOS must know the host name or IP address of your SMTP mail gateway to mail event messages to the email address(es) you have entered. If you do not have access to an SMTP gateway, you can use the CacheFlow default gateway to send event messages directly to CacheFlow. Note that the CacheFlow SMTP gateway will only send mail to CacheFlow, it will not forward mail to other domains.
6. Click Apply to save changes.

To enable event notifications using the CLI

1. Open a terminal session with the Content Accelerator and type your Username and Password when prompted.
2. At the command prompt, type **enable** and type your Password when prompted.
3. At the command prompt, type **configure terminal**.
4. At the `(config)` prompt, type **event-log** to access event log configuration mode.

   The command prompt changes to `(config event-log)`.
5. At the `(config event-log)` prompt, type **mail smtp-gateway *gateway***. The gateway can be a domain name or IP address.

   If you do not have an SMTP gateway, you can use the CacheFlow default gateway to send event messages directly to CacheFlow. Note that the CacheFlow SMTP gateway will only send mail to CacheFlow, it will not forward mail to other domains.
6. At the `(config event-log)` prompt, type **mail add *cacheadmin@company.com*** to add an event recipient.
7. At the `(config event-log)` prompt, type **mail cacheflow-notify** to send event notifications directly to CacheFlow for support purposes.
8. Type **exit** to return to the `(config)` prompt.

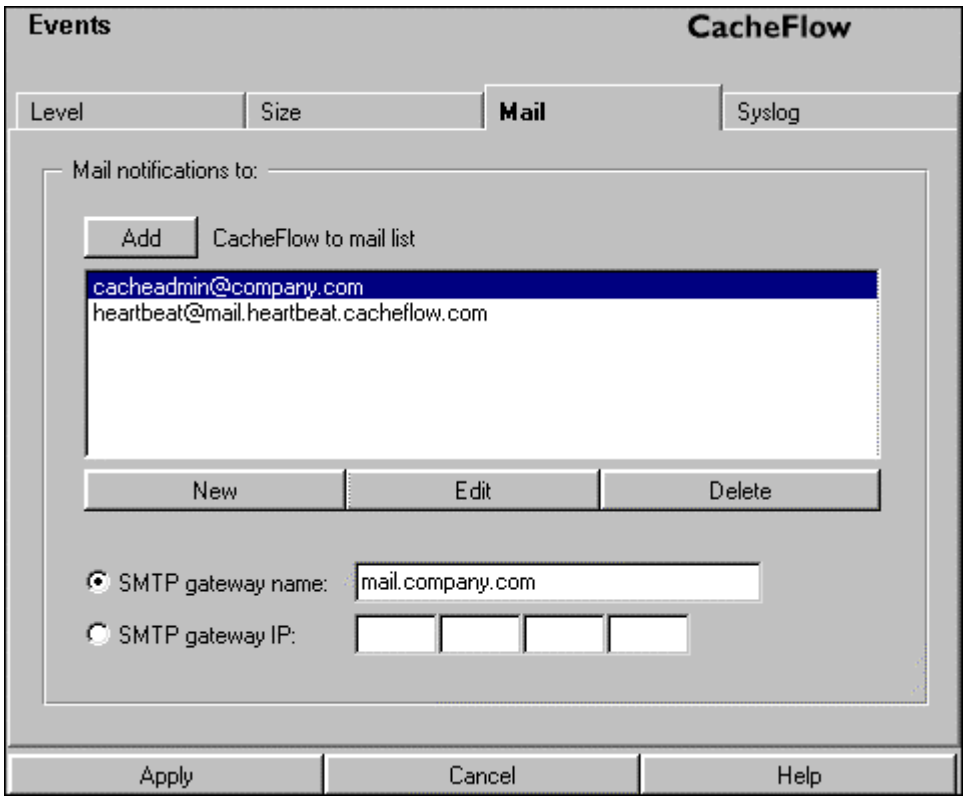


Figure 12-3 Enabling event notifications

Syslog Event Monitoring

Syslog is an event monitoring scheme that is especially popular in Unix environments. In sites where Syslog is used, there is typically a log host node, which acts as a sink for several devices on the network. You must have a Syslog daemon operating in your network to use Syslog monitoring. The Syslog format is: "Date Time Hostname Event."

Most clients using Syslog have multiple devices sending messages to a single Syslog daemon. This allows viewing a single chronological event log of all of the devices assigned to the Syslog daemon. An event on one network device might trigger an event on other network devices, which, on occasion, can point out faulty equipment.

To enable Syslog monitoring

1. Select Management from the CacheOS home page.
2. Select the Events applet.
3. Click the Syslog tab.
4. In the Loghost field, type the domain name or IP address of your loghost server.
5. Activate the Enable Syslog checkbox.
6. Click Apply to save changes.

To enable Syslog monitoring using the CLI

1. Open a terminal session with the Content Accelerator and type your Username and Password when prompted.
2. At the command prompt, type **enable** and type your Password when prompted.
3. At the command prompt, type **configure terminal**.
4. At the `(config)` prompt, type **event-log** to access event log configuration mode.
   The command prompt changes to `(config event-log)`.
5. At the `(config event-log)` prompt, type **syslog loghost** ***loghost***. The log host can be a domain name or IP address.
6. At the `(config event-log)` prompt, type syslog enable to activate logging.
7. Type **exit** to return to the `(config)` prompt.

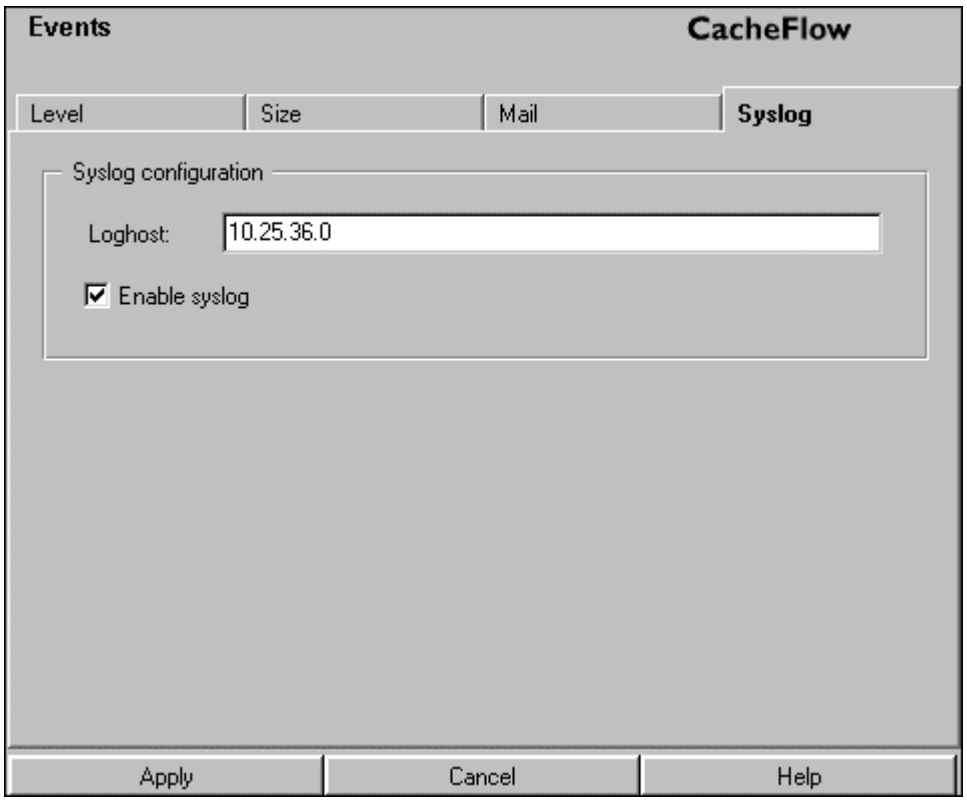


Figure 12-4 Setting up Syslog monitoring

Chapter 13 - Maintenance

The maintenance applet provides a set of tools that are used for managing and configuring an array of system-wide parameters such as restarting the Content Accelerator, upgrading the OS, configuring RealProxy and maintaining the cache. Also included are tools for configuring a variety of filtering and routing parameters.

Restoring System Defaults

When you restore system defaults, the Content Accelerator's IP address, default gateway, and the DNS server addresses are cleared. In addition, any lists (e.g., filtering, forwarding, bypass) are cleared. After restoring system defaults, it is necessary to restore the Content Accelerator's basic network settings as described in First time Setup of a CacheFlow System and reset any customizations.

To restore system defaults

1. Select Management from the CacheOS home page.
2. Select the Maintenance applet.
3. Click Restore to restore system defaults.
4. Click OK to confirm.

To restore system defaults using the CLI

1. Open a terminal session with the Content Accelerator and type your Username and Password when prompted.
2. At the command prompt, type **enable** and type your Password when prompted.
3. At the command prompt, type **restore-defaults**.

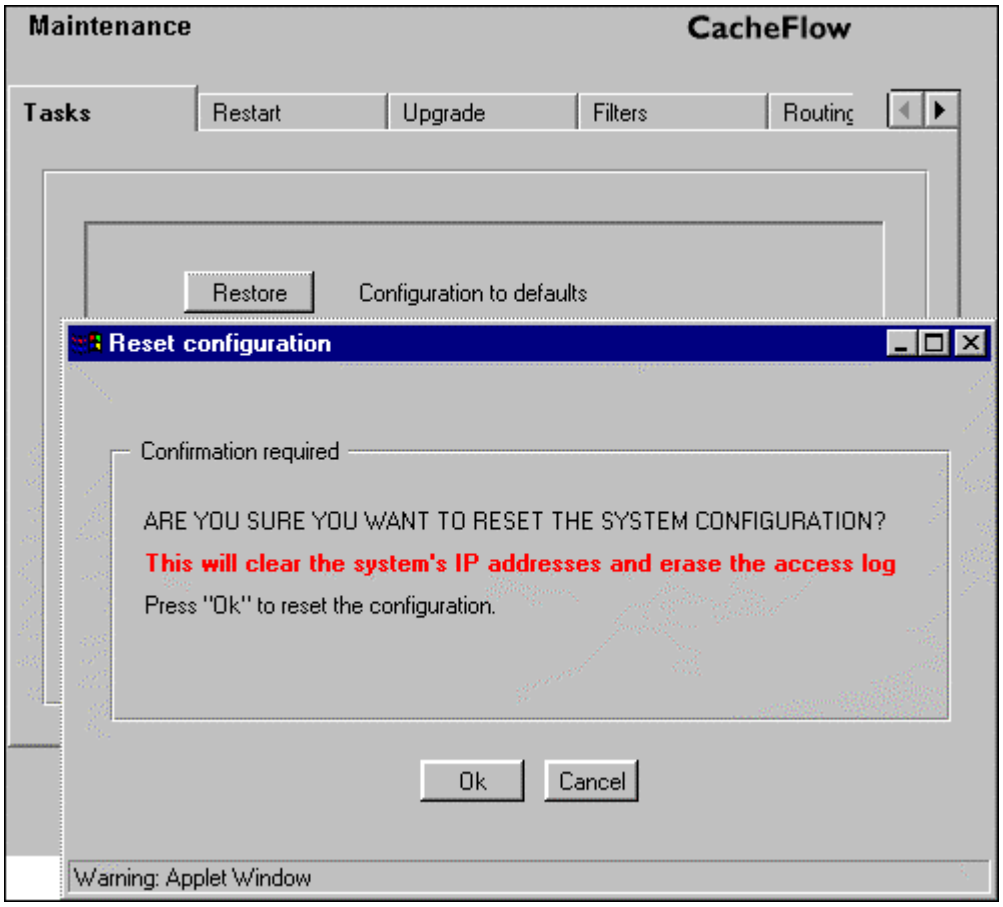


Figure 13-1 Restoring system defaults

Purging the DNS Cache

You can purge the DNS cache at any time. You might need to purge the DNS cache if you have experienced a problem with your DNS server, or if you have changed your DNS configuration.

To purge the DNS cache

1. Select Management from the CacheOS home page.
2. Select the Maintenance applet.
3. Click Purge to purge the DNS cache.
4. Click OK to confirm.

To purge the DNS cache using the CLI

1. Open a terminal session with the Content Accelerator and type your Username and Password when prompted.
2. At the command prompt, type **enable** and type your Password when prompted.
3. At the command prompt, type **purge-dns-cache**.

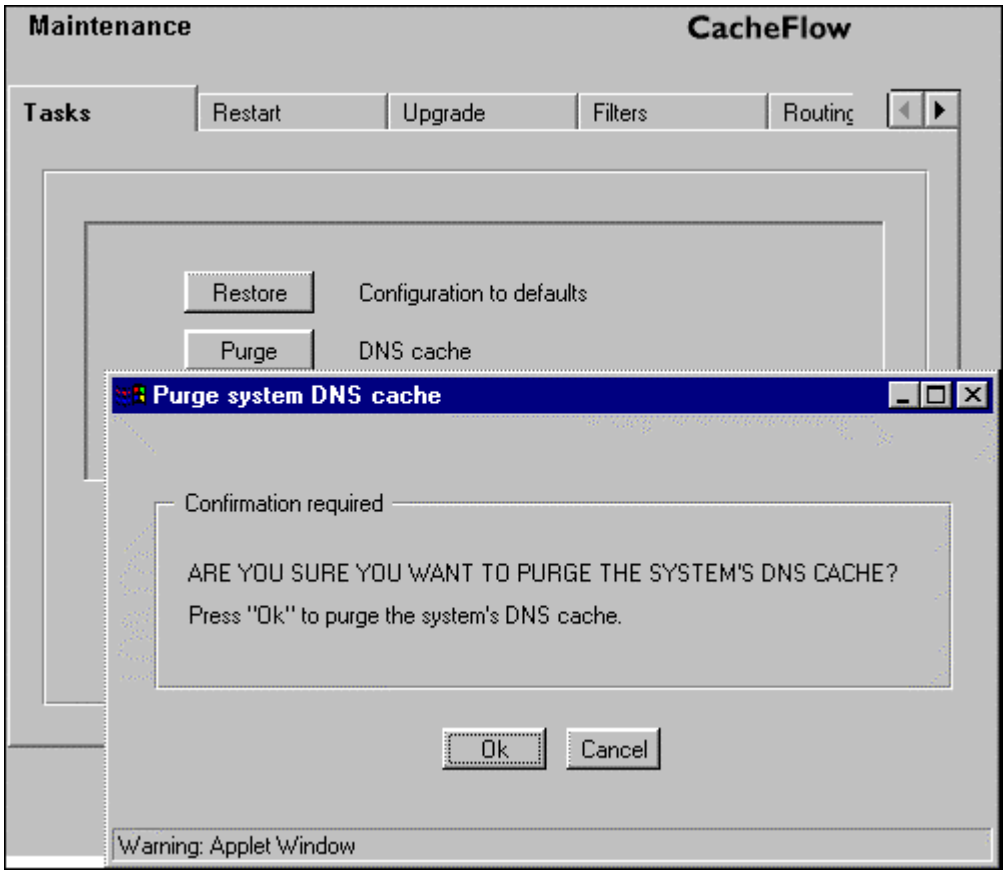


Figure 13-2 Purging the DNS cache

Clearing the System Cache

You can clear the system cache at any time. When you clear the system cache, all objects in the cache are set to expired. The objects are not immediately removed from memory or disk, but a subsequent request for any object requested is retrieved from the source before it is served.

To clear the system cache

1. Select Management from the CacheOS home page.
2. Select the Maintenance applet.
3. Click Clear to clear the system cache.
4. Click OK to confirm.

To clear the system cache using the CLI

1. Open a terminal session with the Content Accelerator and type your Username and Password when prompted.
2. At the command prompt, type **enable** and type your Password when prompted.

3. At the command prompt, type **clear-cache**.

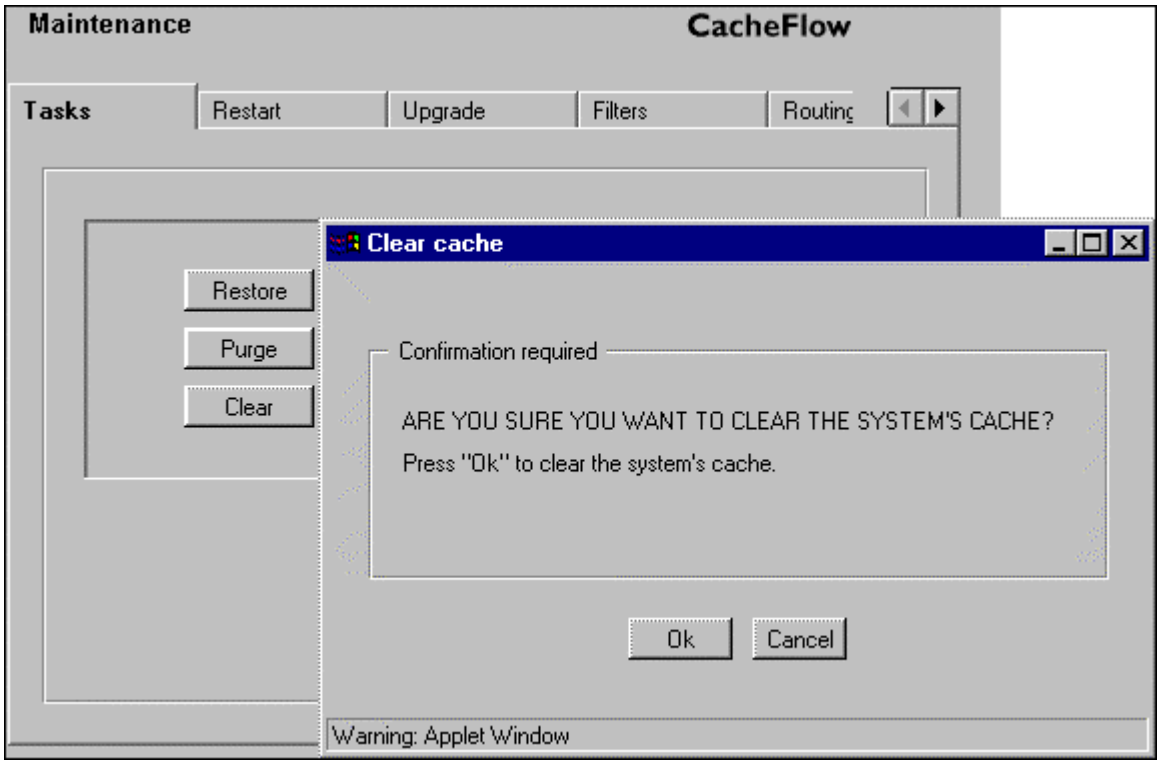


Figure 13-3 Clearing the system cache

Restarting the Content Accelerator

The restart options control the restart attributes of the Content Accelerator in case a restart is needed due to a system fault.

Important The default settings of the restart tab will suit most systems. Changing them without assistance from CacheFlow Technical Support is not recommended.

Core Image Restart Options

The core image options specify how much detail about the system state is logged to disk when a system is restarted. Although this information is not accessible to the Content Accelerator user, it is used by CacheFlow Support in resolving system problems. The more detail logged, the longer it takes the Content Accelerator to restart.

There are three options:

- None. No system state information is logged. Not recommended.
- Context only. The state of active processes are logged to disk. This is the default.
- Full. The full contents of memory are logged to disk. Only use when asked to do so by CacheFlow support.

The default setting of Context only, either with or without compression (depending on your system model), is the optimum balance between restart speed and the information needs of CacheFlow Support in helping to resolve a system problem. The option to compress the logged information might speed or slow the restart depending on which model Content Accelerator is being restarted. In most situations, the compression default for a specific system model provides an optimum balance between image size and speed.

Hardware and Software Restart Options

The Restart settings determine if the Content Accelerator performs a faster software only restart, or a more comprehensive, but slower, hardware and software restart. The latter can take up to several minutes longer, depending upon the amount of memory and number of disk drives in the Content Accelerator configuration.

The default setting of Software only will suit most situations. Restarting both the hardware and software is recommended in situations where a hardware fault is suspected.

Note Be sure to Click Apply if changes are made to restart settings and you want them to apply to the next Content Accelerator restart.

To restart the Content Accelerator

1. Select Management from the CacheOS home page.
2. Select the Maintenance applet.
3. Select the Restart tab.
4. Select your preferred restart options.
5. Click Apply to save changes.
6. Click Restart now
7. Click OK to confirm and restart the Content Accelerator.

To restart the Content Accelerator using the CLI

1. Open a terminal session with the Content Accelerator and type your Username and Password when prompted.
2. At the command prompt, type **enable** and type your Password when prompted.
3. At the command prompt, type **restart regular**.

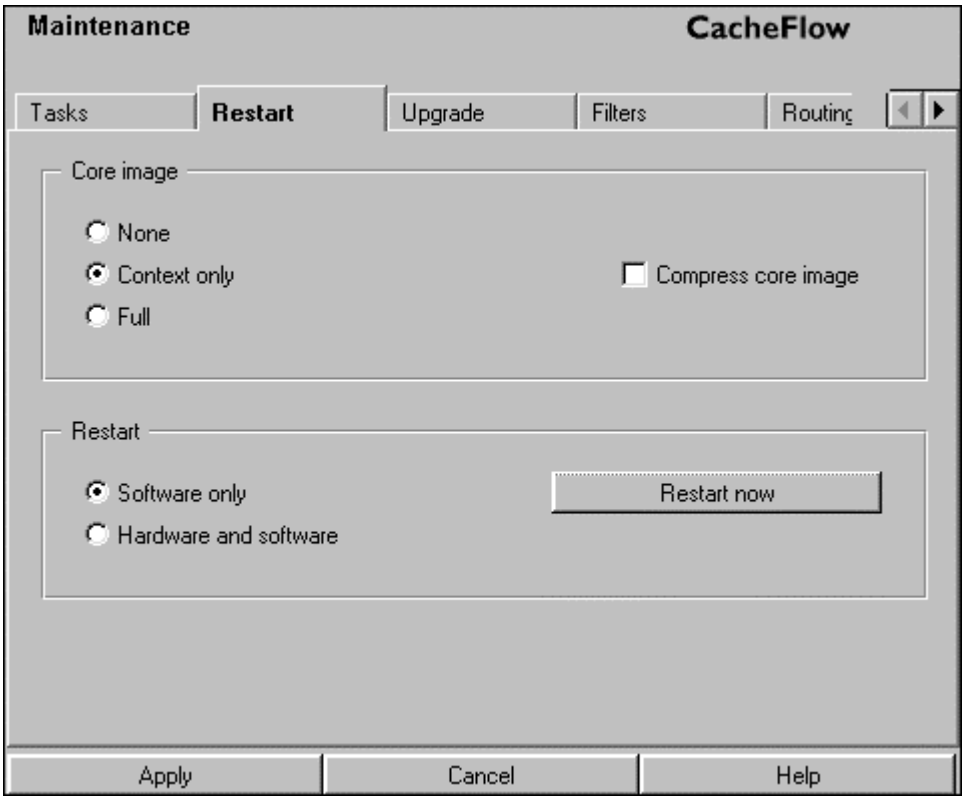


Figure 13-4 Restarting the Content Accelerator

Upgrading CacheOS

When an upgrade to CacheOS becomes available, you can download and install the upgrade over the internet.

To upgrade CacheOS

1. Select Management from the CacheOS home page.
2. Select the Maintenance applet.
3. Select the Upgrade tab.
4. Click the Show me button to connect to CacheFlow's download page, follow the links, and note the URL of the CacheOS upgrade for your system model.
5. Enter the URL in the Download new system software from this URL box and click Download.
6. When the download is complete, click Restart, then click OK, to start running the new version.

To upgrade CacheOS using the CLI

1. Open a terminal session with the Content Accelerator and type your Username and Password when prompted.
2. At the command prompt, type **enable** and type your Password when prompted.
3. At the command prompt, type **configure terminal**.

4. At the `(config)` prompt type **upgrade path *URL*** to set the location of the OS upgrade image.
5. At the `(config)` prompt, type **exit** to return to privileged mode.
6. At the command prompt, type **load upgrade** to copy the upgrade image to the Content Accelerator.
7. At the command prompt, type **restart upgrade** to restart the system and run the upgraded version of CacheOS.

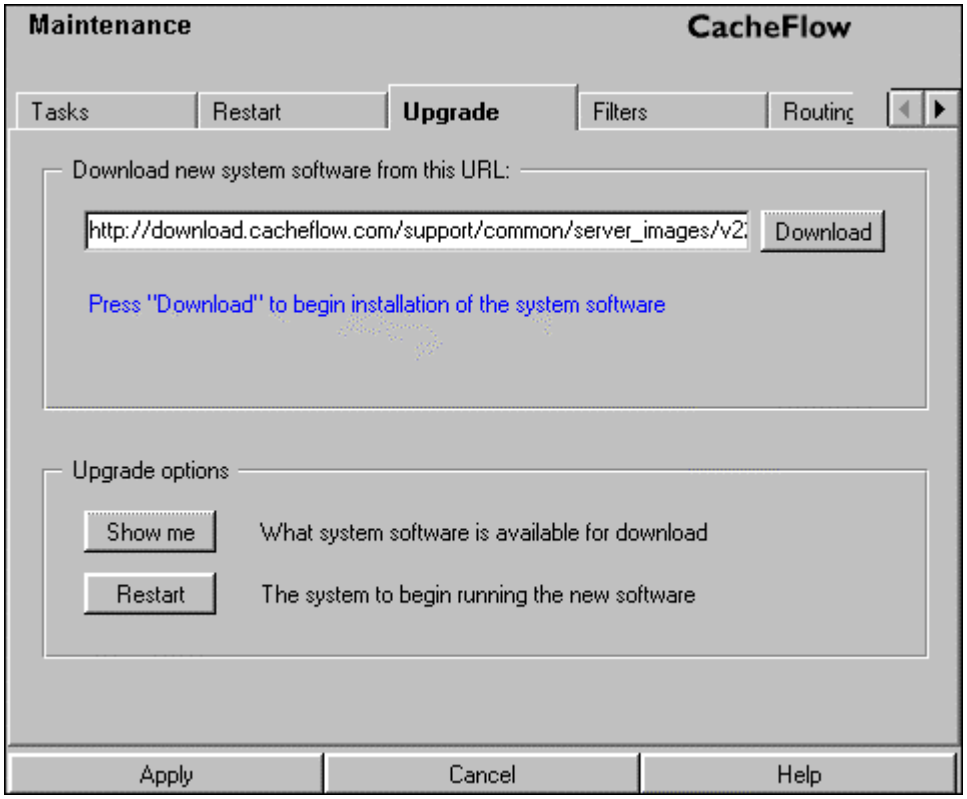


Figure 13-5 Upgrading CacheOS

Using a Filter List

CacheOS can filter requests made by clients using a filter list. When a filter list is loaded, all requested URLs are compared to the list and processed based on the results.

A filter list can be used to assign the following actions for a URL:

- access direct
- bypass LDAP authentication
- cache advertising objects
- case-insensitive matching
- content-filter override
- deny service

- do not cache
- do not refresh
- time to live (TTL)
- version control

Order of Evaluation

It is important to note that CacheOS does not evaluate items in a filter list by the order in which they appear, rather, prefix filters are evaluated first, then domain suffix filters, and lastly, regular expression filters.

Installing a Local Filter List

The local filter list is a list you create and maintain on your network. You can use a local filter list alone, or in conjunction with a central list. If you decide to use a local filter list, create the filter list and place it on an HTTP or FTP server so it can be downloaded to the Content Accelerator.

To install a local filter list

1. Select Management from the CacheOS home page.
2. Select the Maintenance applet.
3. Select the Filters tab.
4. In the Local file field, enter the URL where the local filter list is located.
5. Click Install to download and install the list.
   You can click View to display the list before installing it.
6. Click Apply to save changes.

To install a local filter list using the CLI

1. Open a terminal session with the Content Accelerator and type your Username and Password when prompted.
2. At the command prompt, type **enable** and type your Password when prompted.
3. At the command prompt, type **configure terminal**.
4. At the `(config)` prompt type **filter-list local-path *URL*** to set the location of the filter list file.
5. At the `(config)` prompt, type **load filter-list local** to install the filter list.
6. At the `(config)` prompt, type **exit** to leave configuration mode.

Installing a Central Filter List

The central filter list is a shared list of addresses that is used by multiple Content Accelerators. You can create your own central filter list to manage multiple Content Accelerators, or you can use the central filter list maintained by CacheFlow Technical Support at http://www.cacheflow.com/support/subscriptions/CentralFilterList.txt

If you decide to use a custom central filter list, create the filter list and place it on an HTTP or FTP server so it can be downloaded.

To install a central filter list

1. Select Management from the CacheOS home page.
2. Select the Maintenance applet.

3. Select the Filters tab.
4. In the Central file field, enter the URL where the central filter list is located, or use the default URL pointing the to list maintained by CacheFlow.
5. Click Install to download and install the list.

   You can click the View button to display the list before installing it.
6. Click Apply to save changes.

To install a central filter list using the CLI

1. Open a terminal session with the Content Accelerator and type your Username and Password when prompted.
2. At the command prompt, type **enable** and type your Password when prompted.
3. At the command prompt, type **configure terminal**.
4. At the `(config)` prompt type **filter-list central-path *URL*** to set the location of the filter list file.
5. At the `(config)` prompt, type **load filter-list central** to install the filter list.
6. At the `(config)` prompt, type **exit** to leave configuration mode.

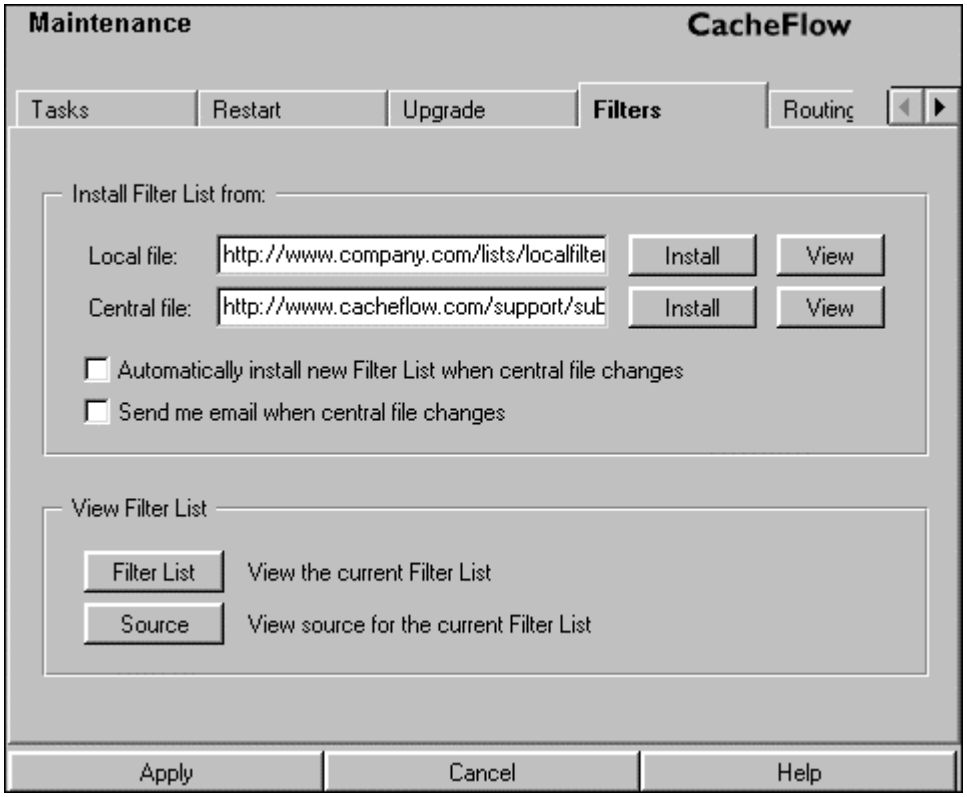


Figure 13-6 Using filter lists

Creating a Filter List

There are two types of filter lists: the local filter list and the central filter list.

To use the filter list, create a text file that contains a filter statement for each URL you want to filter. The syntax of the filter statement is shown below:

url `filter=value filter=value filter=value ……`

The URL can be specified as a full URL, or a regular expression. For information on using regular expressions, see the *Regular Expressions* appendix. The URL can be a host name or directory path. Some sample URLs are shown below:

http://www.company.com

http://www.company.com/quotes

http://www.company.com/news

If you specify a directory path, CacheOS will treat the specification as a full directory name. For example, the URL http://www.company.com/news will not match the URL http://www.company.com/newspaper. CacheOS filtering parameters are described in the following table.

Tip A time-saving method for maintaining a filter list in which most users conform to the same settings is to use the default_filter_properties directive at the beginning of your filter file. By including this directive setting to accommodate most users, you have only to override the settings for exception cases.

Parameters:	Value	Description
advertisement	yes	Cache objects at this URL, and request the objects in the background to maintain the hit count.
cache	no	Do not cache the object.
case_insensitive	yes	Match URLs using case-insensitivity. By default all URLs are matched in a case-sensitive manner. This filter should be set to match URLs served by Operating Systems such as Windows NT, which is case insensitive.
default_filter_properties	protocol = ftp | http | https acl = *user* service = yes | no	Defines basic settings that apply to the entire filter list. Can be overridden by individual filter statements.
direct	yes	Do not forward requests to a parent proxy or SOCKS server. This filter only applies when the device is configured to forward requests.
proxy_authentication	no	Bypasses LDAP authentication for the URLs specified.

Parameters:	Value	Description
refresh	no	Do not refresh the object if it is cached.
service	no	Deny service to the URL.
ttl	number of seconds	Sets the expiration time of a url/object. Notes: • The Advertisment filter option overrides the TTL. • The HTTP command line option "Force explicit expirations: Never serve after" must be enabled. If disabled, CacheOS's Probabalistic Refresh overrides the TTL value.
version_control	yes | no	Specifies whether the first version string appearing in a URL should be stripped. Setting version_control to yes followed by a URL means to strip the version control string and, on a cache miss, forward the resulting URL to the server. Setting version_control to no means that the version string is used by CacheOS in determining which object to serve. The default version_control setting is no.

Some sample filter statements follow. Companies 2 and 5 illustrate the use of multiple filter statements for a single entry.

```
Default_filter_properties service==no refresh=yes
ftp://.* direct=yes
http://www.company.com service=no
http://www.company2.com cache=no refresh=no
http://www.company3.com direct=yes
http://.*\.company4.* direct=yes
http://www.company5.com case_insensitive=yes cache=no
http://www.company6.com content_filter_override=yes
http://www.company7.com proxy_authentication=no
http://www.company8.com advertisement=yes
http://www.company8.com service=yes
http://www.company10.com ttl=60
http://www.company11.* version_control=yes
```

If you want to enable multiple filter directives for a single entry, you must specify all the filter directives on the same line as the URL to which the directives apply. In the above list, the parameter `service=yes` for www.company8.com would activate, while the parameter `advertisement=yes` for www.company8.com would not activate. In all lists, CacheOS always operates on the last parameter option for a URL and disregards previous options, which is the case with www.company8.com.

The converse is also true. The parameters `case_insensitive=yes` and `cache=no` for www.company5.com both activate because they are both placed on the same line.

Filter statement rules

- Type each statement on a separate line.
- Place all filter options for a URL on the the same line.
- Begin each comment line with a semicolon.
- The maximum length of a line is 4096 bytes.

Domain Suffix Filtering

Domain suffix filters can be used in place of certain regular expression filters and provide better performance than the equivalent regular expression filters. Domain suffix filters are intended to replace regular expression filters of the form: http://.*\.?domain/ and match all objects from the domain and its sub-domains. CacheOS supports a filter list containing many domain-suffix filters with minimal system overhead.

Using Domain Suffix Filters

To accommodate domain suffix filtering, filter lists now use filter section headers. The use of section headers is not required for filter list text files that do not contain domain suffix filters. This file format will continue to be interpreted correctly. Formal filter section headers are required, however, if domain-suffix filters are specified within a filter file. Regardless of whether section headers are present in a filter file, The CacheOS Web and CLI interfaces display filter files with section headers. Filter section headers must be located on a single line.

The following filter file section headers are supported

- [prefix] - for prefix filter entries.
- [domain-suffix] - for domain suffix filter entries.
- [regular-expression] - for regular expression filter entries.

The appearance of a section header within a filter file indicates that all subsequent filter entries are to be interpreted as specified within the section header. If section headers are used, CacheOS automatically checks to ensure that regular expression filter entries only appear within the [regular-expression] filter section.

Filter entries within the [domain-suffix] section look very similar to entries within the [prefix] filter section. The only difference between them is the manner in which domains are matched. For example, the “last-one-wins” rule for prefix filters also applies for domain-suffix filters.

Note that in the absence of filter section headers, filters are considered to be prefix filters unless they contain one or more regular expression meta-characters. If a filter entry does contain regular expression meta-characters, it is considered to be a regular expression.

The order in which CacheOS evaluates filters is as follows

1. Filter entries including the keyword “All” (if ACL enabled)
2. Prefix filters
3. Domain-suffix filters
4. Regular-expression filters
5. Default-filter properties

A filter entry match in any section as the filter evaluation moves from 1 to 5 in the search hierarchy prevents any further match attempts.

Domain Suffix Filter Example

The following example shows a filter list containing domain suffix filters. Filter lists that include domain suffix filters must follow a structure that explicitly identifies the type of filter.

This example illustrates the the three possible filter types.

```
[Prefix]
http://www.confidential.com/ service=no
[Domain-suffix]
http://company.com/ service=no
[Regular-expression]
http://.*xyz.com/ service=no
```

The above 3 filters all result in denial of service to a group of distinct URLs:

- The prefix filter http://www.confidential.com/ will deny service to all URLs exactly matching the domain www.confidential.com and any path relative to the aforementioned domain, including the null path.
- The domain suffix filter http://company.com/ will deny service to all URLs where company.com is a proper super-domain and any path relative to the matched domain, including the null path. For example, service will be denied to the URL http://www.intranet.company.com/, but not http://mycompany.com/ since mycompany.com is not a proper sub-domain of company.com.
- The regular expression filter entry http://.*xyz.com/ will deny service to any URL containing a domain ending in the string xyz.com. This regular expression filter is included only in order to be complete. Regular expression filters should only be used when prefix or domain suffix filters are insufficient since processing of regular expression filters requires more system resources.

Important If you include a period at the beginning of the domain name in a filter, it might not produce the expected match, for example, .company.com will not match company.com. This also holds true for filters which specify only the ending part of the domain name, for example, org works as expected, but .org does not work as you might expect.

Using a Filter List to Restrict Cache Access

A useful strategy for controlling user access is to use a filter list in combination with a filter list Access Control List (ACL). The following example illustrates this strategy.

Note Do not to confuse the filter list ACL with the management console ACL. They are separate lists.

1. First, create a filter list ACL where the group `myusers` has full access to the internet, while the `restricteduser` has access to just a few sites.

```
define acl myusers
10.1.1.20.0/24
end acl myusers

define acl restricteduser
10.1.211.212
end acl restricteduser
```

2. Next, add the following ALL directive to the ACL, allowing the set of `myusers` and `restricteduser` to proceed in the evaluation. The ALL directive is evaluated at the start of a filter query. Note that the ALL directive must appear in uppercase. The line below will block access to all users not part of `myusers` or `restricteduser`.

```
ALL acl=!(myusers || restricteduser) service=no
```

3. Next, create a filter list that defines the subset of URLs to which `restricteduser` has access.

```
http://www.company.com/ acl=restricteduser service=yes
http://www.intranet.company.com/ acl=restricteduser service=yes
http://*.company2.com/ acl=restricteduser service=yes
```

4. Lastly, to prevent members of `restricteduser` having access to other URLs, the default_filter_properties directive can be added to the end of the filter file to prevent further access based on protocol.

```
default_filter_properties protocol=http acl=restricteduser service=no
default_filter_properties protocol=https acl=restricteduser service=no
default_filter_properties protocol=ftp acl=restricteduser service=no
```

Defining Static Routes

CacheOS can be configured to use static routes. To use static routes you must create a routing table and place it on an HTTP or FTP server so it can be downloaded and installed on the Content Accelerator. The routing table is a text file containing a list of IP addresses, subnet masks, and gateways. A sample routing table is shown below:

```
10.25.36.0   255.255.255.0   10.25.46.57
10.25.37.0   255.255.255.0   10.25.46.58
10.25.38.0   255.255.255.0   10.25.46.59
```

When a routing table is loaded, all requested URLs are compared to the list, and routed based on the best match.

To install a routing table

1. Select Management from the CacheOS home page.
2. Select the Maintenance applet.
3. Select the Routing tab.
4. Enter the URL where the routing table is located.
5. Click Install to download and install the table.
   You can click View to display the routing table before installing it.
6. Click Apply to save changes.

To install a routing table using the CLI

1. Open a terminal session with the Content Accelerator and type your Username and Password when prompted.
2. At the command prompt, type **enable** and type your Password when prompted.
3. At the command prompt, type **configure terminal**.
4. At the `(config)` prompt type **static-routes path *URL*** to set the location of the routing table file.
5. At the `(config)` prompt, type **load static-route-table** to install the routing table.
6. At the `(config)` prompt, type **exit** to leave configuration mode.

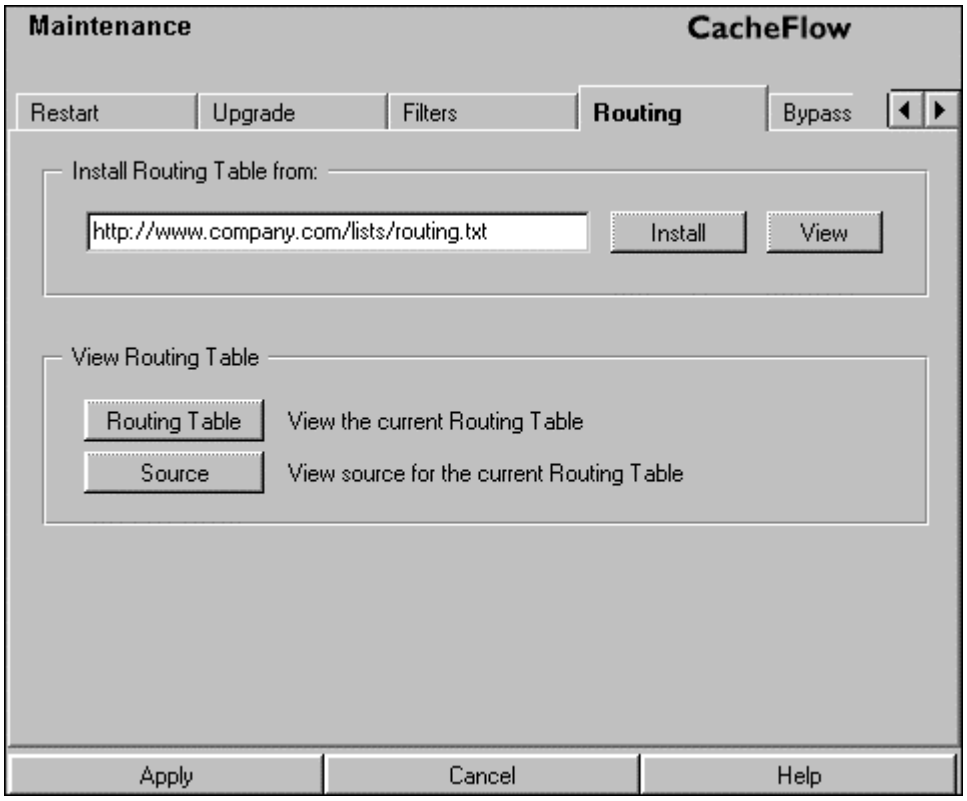


Figure 13-7 Installing a static route table

Using a Bypass List

A bypass list is used to prevent CacheOS from transparently proxying requests to servers that perform IP authentication with clients. The bypass list contains IP addresses, subnet masks and gateways. When a request matches an IP address and subnet mask specification in the bypass list, the request is sent to the designated gateway. A bypass list is only used for transparent caching.

There are two types of bypass lists: local and central.

Local Bypass List

The local bypass list is a list you create and maintain on your network. You can use a local bypass list alone, or in conjunction with a central list.

The gateways specified in the bypass list must be on the same subnet as the CacheOS device.

The local bypass list contains a list of IP addresses, subnet masks, and gateways. It can also define the default bypass gateway to be used by both the local bypass list and central bypass list. The gateways specified in the bypass list must be on the same subnet as the Content Accelerator. When you download a bypass list, the list is stored in the Content Accelerator until it is replaced by downloading a new list. A sample local bypass list is shown below:

```
;define the default gateway for the local and central bypass list
```

```
BYPASS_GATEWAY 10.25.46.57
;define addresses to bypass
;IP address      subnet              gateway (or use default gateway)
10.25.36.47      255.255.255.255
10.25.36.48      255.255.255.255
10.25.0.0        255.255.255.0       10.25.46.58
```

Note The BYPASS_GATEWAY and default gateway must be on a different subnet from the IP addresses.

If you do not specify the BYPASS_GATEWAY, and you do not designate the gateway in the address specification, CacheOS forwards the request to the default gateway defined in the network configuration.

Once the bypass list is created, place it on an HTTP or FTP server so it can be installed onto the Content Accelerator.

To install a local bypass list

1. Select Management from the CacheOS home page.
2. Select the Maintenance applet.
3. Select the Bypass tab.
4. Enter the URL where the bypass list is located in the Local file field.
5. Click Install to download and install the list.
   You can click View to display the list before installing it.
6. Click Apply to save changes.

To install a local bypass list using the CLI

1. Open a terminal session with the Content Accelerator and type your Username and Password when prompted.
2. At the command prompt, type **enable** and type your Password when prompted.
3. At the command prompt, type **configure terminal**.
4. At the `(config)` prompt type **bypass-list local-path *URL*** to set the location of the bypass list file.
5. At the `(config)` prompt, type **load bypass-list local** to install the bypass list.
6. At the `(config)` prompt, type **exit** to leave configuration mode.

Central Bypass List

The central bypass list is a shared list of addresses that is used by multiple CacheFlow devices. The central list contains addresses to bypass, but does not specify gateways (because the CacheFlow devices will be located on different subnets, using different gateways). The gateway used for matches in the central bypass list is defined using the BYPASS_GATEWAY command in the local bypass list. If there is no BYPASS_GATEWAY command, the CacheFlow device will use the default gateway defined by the network configuration.

You can create your own central bypass list to manage multiple CacheFlow devices, or you can use the central bypass list maintained by CacheFlow Technical Support at http://www.cacheflow.com/support/subscriptions/CentralBypassList.txt

The central bypass list maintained by CacheFlow contains addresses CacheFlow has identified as using client authentication.

To install a central bypass list

1. Select Management from the CacheOS home page.
2. Select the Maintenance applet.
3. Select the Bypass tab.
4. Enter the URL where the bypass list is located in the Central file field.
5. Click Install to download and install the list.
   You can click the View button to display the list before installing it.
6. Click Apply to save changes.

To install a central bypass list using the CLI

1. Open a terminal session with the Content Accelerator and type your Username and Password when prompted.
2. At the command prompt, type **enable** and type your Password when prompted.
3. At the command prompt, type **configure terminal**.
4. At the `(config)` prompt type **bypass-list central-path *URL*** to set the location of the bypass list file.
5. At the `(config)` prompt, type **load bypass-list central** to install the bypass list.
6. At the `(config)` prompt, type **exit** to leave configuration mode.

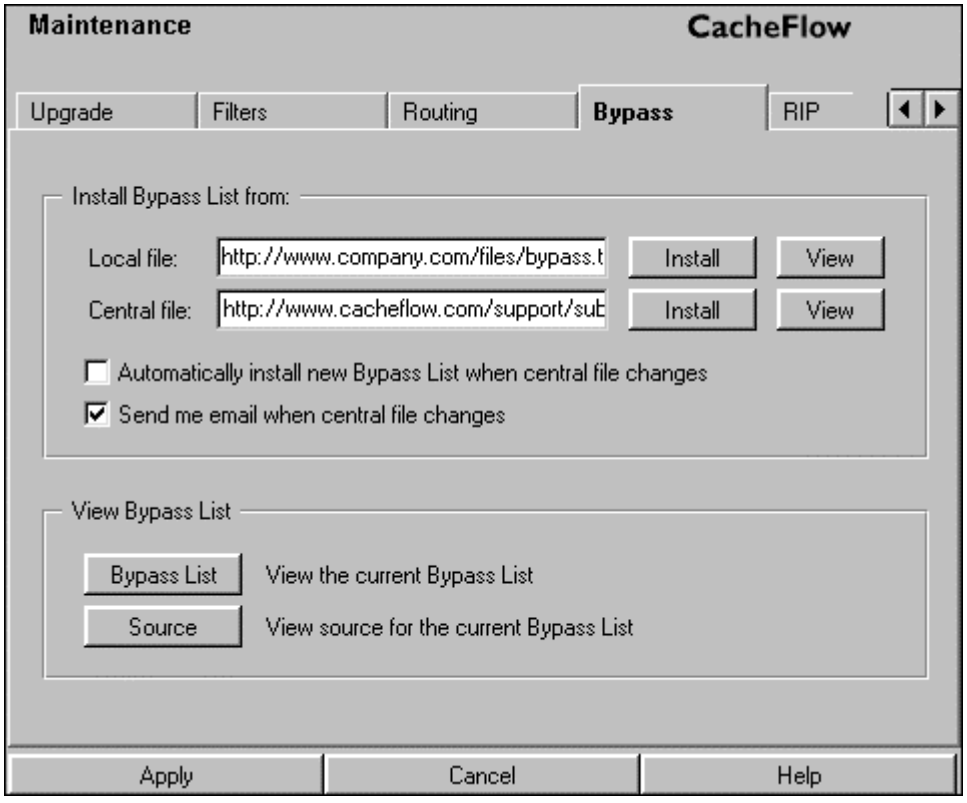


Figure 13-8 Using a bypass list

Using Dynamic Bypass

Dynamic bypass provides a maintenance-free method for improving performance of the Content Accelerator. CacheOS does this by adding dynamic bypass entries, containing the server IP address of sites that have returned an error, to the Content Accelerator's local bypass list. For a configured period of time further requests for the error-causing URL are sent immediately to the origin server, saving CacheOS processing time. The amount of time a dynamic bypass entry stays in the list, and the types of errors that cause CacheOS to add a site to the list, along with several other settings, is configurable from the CLI.

A common scenario where dynamic bypass is useful is as follows. If a client requests a URL and receives an error in response, CacheOS adds a dynamic bypass entry, containing the error-causing URL, to the local bypass list for a pre-configured period of time, known as the dynamic bypass timeout. If the client requests the same URL again during the dynamic bypass period, the Content Accelerator routes the request directly to the origin server without servicing the request. Once the dynamic bypass timeout for a URL has ended, CacheOS removes the URL from the bypass list. On the next client request for the URL, the Content Accelerator attempts to contact the origin server. If the origin server still returns an error, the URL is once again added to the local bypass list for the configured dynamic bypass timeout. If the URL does not return an error, the request is handled in the normal manner.

The performance gains realized with this feature are substantial if the client base is large, and clients are requesting many error-causing URLs in a short period of time (for example, many users clicking a browser's refresh button over and over to get an overloaded origin server to load a URL). Dynamic bypass increases CacheOS efficiency because redundant attempts to contact the origin server are minimized.

Errors that can trigger dynamic bypass are

- Non-HTTP traffic at the HTTP port
- 400 bad request
- 401 unauthorized
- 403 forbidden
- 405 method not allowed
- 406 not acceptable
- 500 internal server error

Configuring Dynamic Bypass

Dynamic bypass is disabled by default. Enabling and fine-tuning dynamic bypass is accomplished in two steps:

1. Edit or create a local bypass list, adding the desired dynamic bypass timeout and threshold parameters.
2. Use the CLI to enable dynamic bypass and set the types of errors that will cause dynamic bypass to add a dynamic bypass entry to the bypass list. Dynamic Bypass is not configurable from the Web interface.

Adding Dynamic Bypass Parameters to the Local Bypass List

The first step in configuring dynamic bypass is to edit the local bypass list to set the SERVER_BYPASS_THRESHOLD and/or DYNAMIC_TIMEOUT values. This step is optional, as CacheOS will use default settings if you do not specify them in the local bypass list. Use the default values unless you have

specific reasons for changing them. Contact CacheFlow technical support for detailed advice on customizing these settings.

The SERVER_BYPASS_THRESHOLD value defines the maximum number of entries in the dynamically generated portion of the local bypass list before CacheOS consolidates client–server pair entries into a single server entry. The range is 1 – 256. The default is 16. When a consolidation occurs, the lifetime of the consolidated entry is set to the value of DYNAMIC_TIMEOUT.

The DYNAMIC_TIMEOUT value defines the number of minutes a dynamic bypass entry can remain unreferenced before it is deleted from the bypass list. The range is 1–6000. The default value is 60.

See the previous section, Using Bypass Lists for specific information on how to edit and apply a filter list.

Important Points Regarding Dynamic Bypass

- Dynamic bypass applies to transparent mode connections only.
- Dynamic bypass entries are lost when the Content Accelerator is restarted or the static bypass file is reinstalled.
- No filtering checks are performed on client requests that match entries in the dynamic bypass list.
- Sites requiring client accesses to always be subjected to CacheOS filtering considerations must use CacheOS in explicit proxy mode, or leave dynamic bypass functionality disabled.

Enabling Dynamic Bypass and Specifying Triggers

Enabling dynamic bypass and specifying the types of errors that will cause a URL to be added to the local bypass list is accomplished at the CLI.

Enabling dynamic bypass and trigger events

1. Open a terminal session with the CacheFlow device and type your Username and Password when prompted.
2. At the command prompt, type **enable** and type your Password when prompted.
3. At the command prompt, type **configure terminal** to enter terminal configuration mode.
4. At the `(config)` prompt, type **dynamic-bypass enable** to enable dynamic bypass.
5. At the `(config)` prompt, type **dynamic-bypass trigger *event***.

   The value for ***event*** can be any item in the following table

Event	Description
all	Enable all dynamic bypass triggers
non-http	Enable dynamic bypass for non-HTTP responses
400	Enable dynamic bypass for HTTP 400 responses
401	Enable dynamic bypass for HTTP 401 responses
403	Enable dynamic bypass for HTTP 403 responses
405	Enable dynamic bypass for HTTP 405 responses
406	Enable dynamic bypass for HTTP 406 responses
500	Enable dynamic bypass for HTTP 500 responses

Negating Dynamic Bypass Triggers

Negating one or more specific dynamic bypass triggers is an easy way to customize which errors cause a dynamic bypass entry to be created. For example, if you want all error events except 401 responses to create a dynamic bypass entry, you can enable all triggers and then negate only the 401event trigger.

To negate one or more dynamic bypass triggers

1. Open a terminal session with the CacheFlow device and type your Username and Password when prompted.
2. At the command prompt, type **enable** and type your Password when prompted.
3. At the command prompt, type **configure terminal** to enter terminal configuration mode.
4. At the `(config)` prompt, type **dynamic-bypass no trigger** ***event***.

   The value for ***event*** can be any item in the following table

Event	Description
all	Disable all dynamic bypass triggers
non-http	Disable dynamic bypass for non-HTTP responses
400	Disable dynamic bypass for HTTP 400 responses
401	Disable dynamic bypass for HTTP 401 responses
403	Disable dynamic bypass for HTTP 403 responses
405	Disable dynamic bypass for HTTP 405 responses
406	Disable dynamic bypass for HTTP 406 responses
500	Disable dynamic bypass for HTTP 500 responses

Clearing the Dynamic Bypass List

To clear the dynamic bypass list

1. Open a terminal session with the CacheFlow device and type your Username and Password when prompted.
2. At the command prompt, type **enable** and type your Password when prompted.
3. At the command prompt, type **configure terminal** to enter terminal configuration mode.
4. At the `(config)` prompt, type **dynamic-bypass clear**.

Displaying the Dynamic Bypass List

To display the dynamic bypass list

1. Open a terminal session with the CacheFlow device and type your Username and Password when prompted.
2. At the command prompt, type **enable** and type your Password when prompted.
3. At the command prompt, type **show bypass-list**.

   The bypass list can also be displayed by using a web browser and going to the URL **http://Content Accelerator:8081/TCP/IP-bypass**

Viewing the Current Dynamic Bypass Configuration

To view the current dynamic bypass configuration

1. Open a terminal session with the CacheFlow device and type your Username and Password when prompted.
2. At the command prompt, type **enable** and type your Password when prompted.
3. At the command prompt, type **show dynamic-bypass**.

Disabling Dynamic Bypass

To disable dynamic bypass

1. Open a terminal session with the CacheFlow device and type your Username and Password when prompted.
2. At the command prompt, type **enable** and type your Password when prompted.
3. At the command prompt, type **configure terminal** to enter terminal configuration mode.
4. At the `(config)` prompt, type **dynamic-bypass disable**.

Using RIP

The Routing Information Protocol (RIP) is designed to select the fastest route to a destination. RIP support is built into CacheOS, and is configured by installing a RIP configuration file onto the Content Accelerator. Refer to Appendix D for the RIP command reference.

Configuring RIP

Once a RIP configuration file is created, place it on an HTTP or FTP server so it can be downloaded and installed on the Content Accelerator.

To install a RIP configuration

1. Select Management from the CacheOS home page.
2. Select the Maintenance applet.
3. Select the Rip tab.
4. In the Install RIP Setting from box, enter the URL where the RIP configuration file is located.
5. Click Install to install the configuration file.
   You can click View to display the configuration file before installing it.
6. Activate the Enable RIP check box.
7. Click Apply to save changes.

To install a RIP configuration using the CLI

1. Open a terminal session with the Content Accelerator and type your Username and Password when prompted.
2. At the command prompt, type **enable** and type your Password when prompted.
3. At the command prompt, type **configure terminal**.
4. At the `(config)` prompt type **rip path *URL*** to set the location of the RIP command file.
5. At the `(config)` prompt, type **load rip-settings** to install the RIP command file.

6. At the `(config)` prompt, type **exit** to leave configuration mode.

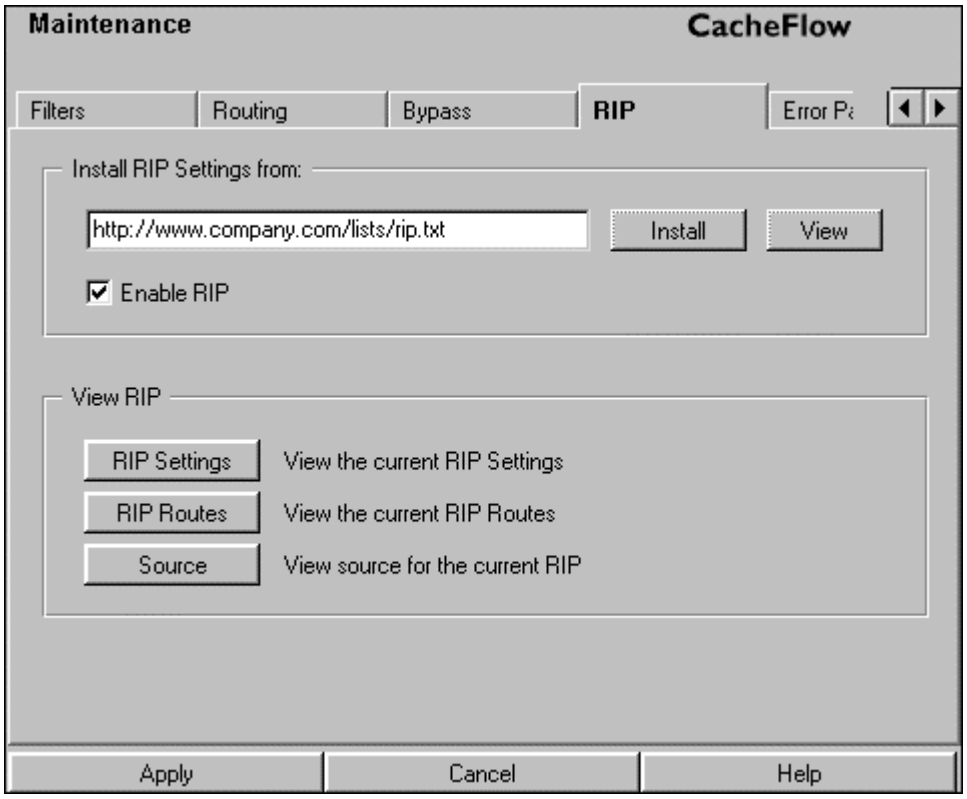


Figure 13-9 Installing RIP settings

Using Customized Error Messages

Error pages are generated by CacheOS under certain exception conditions. For example, when the user enters a URL with a typographic error in the domain portion, HTTP cannot successfully resolve the DNS domain and presents the user with an error page. The error message page is created by CacheOS on demand.

CacheOS provides the ability to install customized error messages on your Content Accelerator. Note that you cannot create new categories of error messages, you can only customize existing messages.

User configurable error messages allow you to perform customizations including, but not limited to

- Placing your company logo on each error page.
- Displaying a very minimalist error page.
- Localizing the language of an error page.
- Redirecting the user to another URL.
- Customizing error messages related to content filtering services. You can state the specific reason for denying access to a particular URL, such as an adult site.

Installing an Error Page

Once a custom error page is created, place it on an HTTP or FTP server so it can be downloaded and installed on the Content Accelerator.

To install an error page

1. Select Management from the CacheOS home page.
2. Select the Maintenance applet.
3. Select the Error Pages tab.
4. In the Install Error Pages from box, enter the URL where the message file is located.
5. Click Install to install the error message file.
   You can click View to display the configuration file before installing it.
6. Click Apply to save changes.

To install an error page using the CLI

1. Open a terminal session with the Content Accelerator and type your Username and Password when prompted.
2. At the command prompt, type **enable** and type your Password when prompted.
3. At the command prompt, type **configure terminal**.
4. At the `(config)` prompt type **error-pages path *URL*** to set the location of the error pages file.
5. At the `(config)` prompt, type **load error-pages** to install the error pages file.
6. At the `(config)` prompt, type **exit** to leave configuration mode.

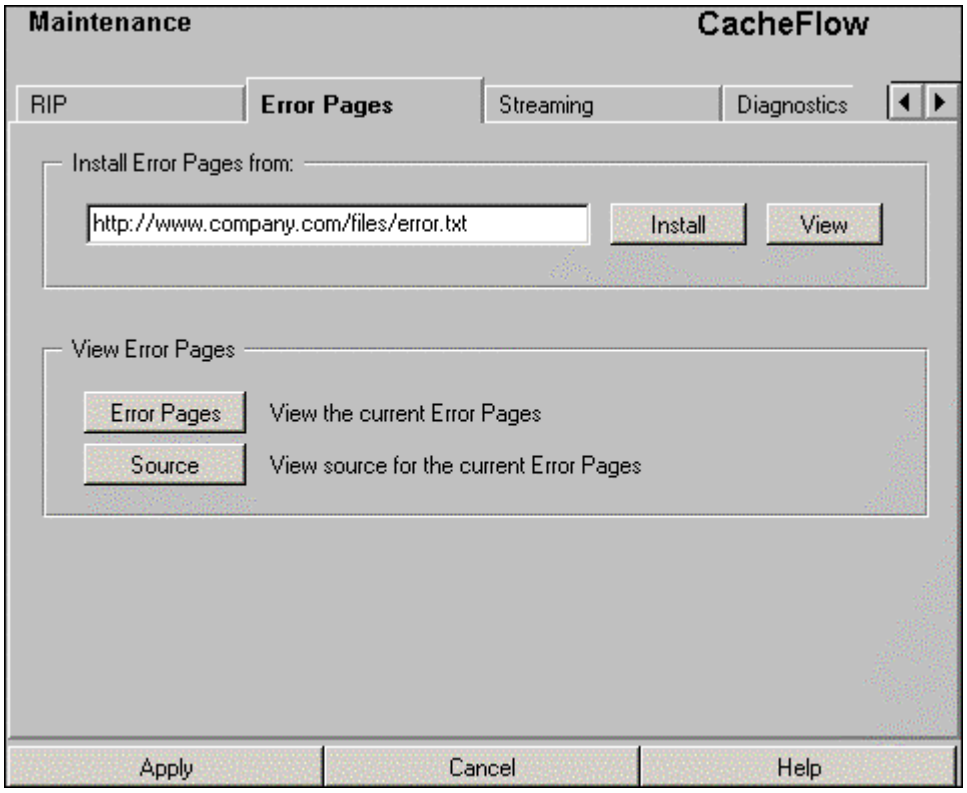


Figure 13-10 Installing a custom error page

Customizing Error Messages

Customizing error messages is a three-step process

1. Copy the source code for the existing error message file, supplied with CacheOS, to a new file.
2. Edit the code in the new file to create your customized error messages, and save the file on an HTTP server visible to the Content Accelerator.
3. Follow the procedure for installing an error page to install the customized error message file on the Content Accelerator.

Caution

- To modify the content section of an error message, we recommend that you have a working knowledge of HTML coding.
- To modify the HTTP section of an error message, we recommend that you have a basic understanding of HTTP.

Message Tokens and Descriptions

Following is a listing of the errors for which you can configure the related messages. Each message configuration begins with one of the following tokens. Message types that are not defined in the supplied configuration file will use the default message for that type.

Message Tokens	Description
REQUEST_INVALID	The request was not parseable by CacheOS. For example if a browser initiated a TCP connection but did not send an HTTP request.
UNSUPPORTED_PROTOCOL	The requested protocol is not supported.
SOCKS_DENIED	The request was denied as specified in the SOCKS direct-deny configuration.
FILTER_DENIED	The request was denied as specified in the URL Filter configuration.
UNRESOLVED_HOSTNAME	DNS could not resolve the hostname specified in the URL.
DNS_SERVER_FAILURE	The DNS server(s) could not be contacted to resolve the hostname.
CONTENT ACCELERATOR_INTERNAL_ERROR	A CacheOS internal error occurred.
TCP_ERROR	A TCP/IP error occurred while attempting to obtain the requested URL.
GATEWAY_ERROR	An error occurred while attempting to obtain the requested URL via the configured gateway (HTTP or SOCKS gateway).
UFS_WEBSENSE_DENIED	Access to the requested URL was denied by the WebSense filtering module.
UFS_SMARTFILTER_DENIED	Access to the requested URL was denied by the SmartFilter filtering module.

Return Token Names and Codes

The first entry in an HTTP section must be an HTTP response line. Use one of the Token Names shown below. This is the bare minimum required for an HTTP section. In the example provided in the previous section of this document, the HTTP response code is 400 – Bad Request.

Token Name	HTTP Return code	Reason Phrase
HTTP_RSP_200	200	OK
HTTP_RSP_203	203	Non-Authoritative Information
HTTP_RSP_204	204	No Content
HTTP_RSP_205	205	Reset Content
HTTP_RSP_206	206	Partial Content
HTTP_RSP_301	301	Moved Permanently

Token Name	HTTP Return code	Reason Phrase
HTTP_RSP_302	302	Found
HTTP_RSP_303	303	See Other
HTTP_RSP_304	304	Not Modified
HTTP_RSP_305	305	Use Proxy
HTTP_RSP_307	307	Temporary Redirect
HTTP_RSP_400	400	Bad Request
HTTP_RSP_401	401	Unauthorized
HTTP_RSP_403	403	Forbidden
HTTP_RSP_404	404	Not Found
HTTP_RSP_405	405	Method Not Allowed
HTTP_RSP_406	406	Not Acceptable
HTTP_RSP_407	407	Proxy Authentication Required
HTTP_RSP_408	408	Request Timeout
HTTP_RSP_409	409	Conflict
HTTP_RSP_410	410	Gone
HTTP_RSP_413	413	Request Entity Too Large
HTTP_RSP_414	414	Request-URI Too Long
HTTP_RSP_415	415	Unsupported Media Type
HTTP_RSP_500	500	Internal Server Error
HTTP_RSP_501	501	Not Implemented
HTTP_RSP_502	502	Bad Gateway
HTTP_RSP_503	503	Service Unavailable
HTTP_RSP_504	504	Gateway Timeout
HTTP_RSP_505	505	HTTP Version Not Supported

Multiple HTTP headers may be added by using the HTTP_HDR_ tokens. For each of these, parameters should follow the token they apply to, separated by a space.

Header Identifiers

Header identifiers are only used in the HTTP sections of the error message source code:

Token Name	Description	Parameters
HTTP_HDR_DATE	Date	Integer32 - Offset in seconds.
HTTP_HDR_EXPIRES	Expires	Unsigned32 - Offset in seconds.
HTTP_HDR_LOCATION	Location	String - Redirection URL (should be a full URL).

Token Name	Description	Parameters
HTTP_HDR_NOCACHE	Pragma: no-cache Cache-Control: no-cache	
HTTP_HDR_PRIVATE	Cache-Control: private	
HTTP_HDR_MAXAGE	Cache-Control: max-age	Unsigned32 - Maximum age in seconds.
HTTP_HDR_CONTENT_LENGTH	Content-Length	
HTTP_HDR_CONTENT_TYPE	Content-Type	String - Type of content (example : text/html)
HTTP_HDR_RETRY	Retry-After	Unsigned32 - Delta seconds.
HTTP_HDR_WARNING	Warning	String - Text to display in the warning.

Substitute Identifiers (Message Tokens)

The following tokens fill in specific information when an error message is generated for a related request. These identifiers are only used in the content section of the error message source code.

Identifier	Description
$(CLIENT:IP)	Requesting Client's IP Address
$(CLIENT:User)	Requesting Client's user name—only available with user authentication
$(CLIENT:Browser)	Requesting Client's browser type—from the request headers
$(CACHEOS:Version)	The Cache's currently running OS version
$(CACHEOS:IP)	The Cache's current IP Address
$(CACHEOS:Name)	The Cache's current name—Configured in Management →Network→Name
$(CACHEOS:Time)	The current time used by CacheOS
$(CACHEOS:Date)	The current date used by CacheOS
$(CACHEOS:HTTP-Version)	The HTTP version currently supported by CacheOS
$(URL:Full)	The full URL requested by the Client
$(URL:Host)	The Hostname of the URL requested by the Client
$(URL:Path)	The path of the URL requested by the Client
$(URL:Protocol)	The protocol of the URL requested by the Client
$(URL:Port)	The destination port of the URL requested by the Client
$(UFS:Category)	The user filtering category of this URL

Default Substitute Identifiers

In CacheOS's default error messages, "test-browser-type" is used as the URL corresponding to $ (Client : Browser), and http://www.test-url.com/main.html is used as the URL corresponding to $ (URL : Full).

Coding Rules for Error Message Files

- CacheOS's internal compiler's internal input buffer is 256 bytes long. Any line exceeding 254 bytes is flagged as an error.
- If the error message file is created on a UNIX system, use the *unix2dos* tool to convert the file to DOS format prior to loading it on the Content Accelerator.

Archiving and Restoring a System Configuration

Archiving a CacheFlow device's system configuration on a regular basis is a prudent measure. In the rare case of a complete system failure, restoring a Content Accelerator to its previous state is simplified by loading an archived system configuration from an FTP, HTTP, or TFTP server. The archive contains all system settings differing from system defaults, along with any forwarding, filtering, and security lists installed on the Content Accelerator.

Archive and restore operations must be performed from the CLI. There is no Web interface for archive and restore.

Important You can archive a system configuration to an FTP or TFTP server that allows either anonymous login, or requires a specific Username and Password. Likewise, to restore a system configuration, the server storing the archive can be configured either to allow anonymous login, or require a Username and Password.

Before archiving a system configuration

1. Obtain write permission to a directory on an FTP or TFTP server. This is where the archive will be stored.

   The system configuration must be stored using FTP or TFTP.
2. Open a terminal session with the CacheFlow device and type your Username and Password when prompted.
3. At the command prompt, type **enable** and type your Password when prompted.
4. At the command prompt, type **configure terminal** to enter terminal configuration mode.
5. At the `(config)` prompt, type **archive-configuration protocol *protocol*** to set the upload protocol to either FTP or TFTP.
6. At the `(config)` prompt, type **archive-configuration host *hostname*** to set the address of the server on which the archive will be stored.

   Hostname is the IP address of the server.
7. At the `(config)` prompt, type **archive-configuration password *password*** to set the password used to access the server.
8. At the `(config)` prompt, type **archive-configuration path *path*** to set the directory on the server where the archive is to be stored relative to the preset FTP directory.
9. At the `(config)` prompt, type **archive-configuration username *username*** to set the username used to access the server.

Example session

```
CacheFlow 5000>enable
Password: *****
```

```
CacheFlow 5000#configure terminal
Enter configuration commands, one per line.  End with CTRL-Z.
(config)archive-configuration host 10.25.36.47
  ok
(config)archive-configuration password password
  ok
(config)archive-configuration username username
  ok
(config)archive-configuration path path
  ok
(config)archive-configuration protocol protocol
  ok
```

Note To clear the host, password, or path, type the above commands using empty double-quotes instead of the variable. For example, to clear the path, type **archive-configuration path “”**

To archive a system configuration

1. Open a terminal session with the CacheFlow device and type your Username and Password when prompted.
2. At the command prompt, type **enable** and type your Password when prompted.
3. At the command prompt, type **upload configuration**.

Example session

```
CacheFlow 5000>enable
Password: *****
CacheFlow 5000#upload configuration
  ok
```

To restore an archived system configuration

1. Using an FTP browser, locate the archived configuration to be restored and note the URL.
2. Open a terminal session with the CacheFlow device and type your Username and Password when prompted.
3. At the command prompt, type **enable** and type your Password when prompted.
4. At the command prompt, type **configure network “*URL*”**.

   The URL must be in quotes and fully qualified (including the protocol, server name or IP address, path, and filename of the archive). The configuration archive is downloaded from the server, and the Content Accelerator settings are updated.

   Note If you rename the archived configuration file so that it does not contain any spaces, the quotes surrounding the URL are unnecessary when restoring the file.
5. At the command prompt, type **restart mode software** to restart the Content Accelerator with the restored settings.

Example session

```
CacheFlow 5000>enable
Password: *****
```

```
CacheFlow 5000#configure network "ftp://10.25.36.46/path/10.25.36.47
 - CacheFlow 5000 0216214521.config"
% Configuring from ftp://10.25.36.46/path/10.25.36.47 - CacheFlow 5000
0216214521.config
  .
  .
  .
  ok
CacheFlow 5000#restart mode software
```

Real Networks Streaming Media Support

CacheOS Version 3.1 provides caching and proxying functionality for Real Networks' RealMedia streams. CacheOS manages client requests for RealServer content. Streaming support provided in CacheOS makes it operationally fully compatible with Real Networks' RealProxy product This functionality is available in both explicit and transparent proxy modes.

Proxy Modes Supported

CacheOS supports three RealProxy modes:

1. Passthrough
2. Splitting
3. Caching

Passthrough

Passthrough mode handles both live and on-demand streams. In passthrough mode all streaming media traffic passes through one point. CacheOS creates a data connection to the RealServer for each client. No bandwidth saving are realized when data is handled in passthrough mode.

Splitting

Pull splitting mode is for use with live material only. When a client first requests a particular stream, CacheOS contacts the source RealServer and then sends the stream to the client. If a second client requests the same live stream they receive it directly from CacheOS. This provides faster service to the client since the content is now being delivered from a local source.

Note The streaming server must be running RealServer Version 7.0. CacheOS Real Networks streaming configuration defaults for Splitting are set to work with the RealServer 7.0 Pull Splitter installation defaults. These defaults are:

- mount point: /split/
- port: 3030
- Protocol: UDP

Caching

Caching is supported for on-demand streams from source RealServers. Streamed data is cached when requested by the first client. When a second client requests the same streamed data, CacheOS checks the cache to see if a stored version is present. To ensure that the stored version is up-to-date, CacheOS checks with the source RealServer to see if a newer version exists. If the stored copy is the latest version, CacheOS streams it to the second client.

CacheOS ensures high-quality data at all times by monitoring the quality of both the cached data it is streaming and the connection between the source RealServer and the client. If cached data becomes impaired in some way, the stream halts and clients receive an error message. If the accounting connection between the client and the source RealServer is interrupted, CacheOS terminates the stream, and the client receives an error message.

Note When RealServer is installed, all its streams are cacheable by default. If a client requests streams from a source RealServer that is configured to prevent caching, they will still receive the streams, but the streams will not be cached. Even if a RealServer manager decides to prevent caching of some content (such as advertisements), they will probably permit it for most items. Since RealServers can reach more clients when caching is allowed, managers are encouraged to leave all content cacheable.

Configuring Caching and Proxying for Real Networks' RealMedia Streams

CacheOS default streaming settings can be changed by using any text editor to create a configuration file. When you create a configuration file you must use correct syntax. See the following *Default Streaming Configuration* section for proper syntax. CacheOS does not display messages related to syntax errors; instead, it will simply ignore settings it does not understand.

Default Streaming Configuration

Following are the default streaming configuration settings provided with CacheOS 3.0. See the *Streaming Configuration Variables* section of this document for details on these configuration settings.

```
; Product Authorization Key (PAK) parameter. Input license key obtained
; from CacheFlow in the text field.
License = text

; Maximum bandwidth allowed between proxy and gateway in kilobits per second (Kbps).
; If this variable is set to 0 or ; left blank, the maximum available bandwidth is
; used.
Bandwidth MaxGateway = 0

; Maximum bandwidth allowed for all connected clients in kilobits per second (Kbps).
; If this variable is set to 0 or left blank, the maximum available bandwidth is used.
Bandwidth MaxProxy = 0
```

```
; Maximum number of concurrent client connections. Connection MaxProxyConnections can
; be set from 1 to 32767.
Connection MaxProxyConnections = 0

; PNA Proxy Port
Port PNA Proxy = 1090

; RTSP Proxy Port
Port RTSP Proxy = 1091

; Enable/disable logging (0=disabled, 1=enabled)
Logging Enabled = 1

; RealProxy Logging Style (0-5)
Logging Style = 3

; RealProxy Logging stats (0-7)
Logging Stats = 0

; Splitting protocol (udp or tcp)
PullSplitting Protocol = udp

; Parent RTSP proxy (address port)
Upstream Proxy RTSP = [none] 0

; Parent PNA proxy (address port)
Upstream proxy PNA = [none] 0

; Enable/disable multicasting (0=disable;1=enable)
Multicast Enable = 0

; Multicast address range (must be between 224.0.0.255-239.255.255.255)
Multicast AddressRange = no default

; Enable/disable announcement of broadcast (1=enable;0=disable)
Multicast SAP = 0

; Router hops allowed (0-255)
Multicast TTL= 16

; In order for a client to connect to the stream they must be setup for multicast.
(0=disable,1=enable)
```

```
Multicast DeliveryOnly = 0

; Multicast RTSP port number
Multicast Port RTSP= 554

; Multicast PNA port number
Multicast Port PNA= 7070

; Access control (RuleNumber IP Subnet) (Any can replace IP to allow anyone (100 Any))
Multicast Accept= 100 Any
```

Streaming Configuration Variables

Bandwidth Management

Bandwidth management is controlled by setting the variables listed below. Once you set values for these variables, CacheOS limits access when the lower threshold is reached. If a client tries to make a request after a limit has been reached, the client receives an error message.

- Bandwidth MaxGateway: this setting is used to set the maximum limit, in kilobits per second (Kbps), for the amount of bandwidth RealProxy uses to send requests to its gateway. The gateway could be a RealProxy server, RealServer, or the Internet. If this variable is set to 0 or left blank, the maximum available bandwidth is used. The default value is 0. Limiting gateway bandwidth limits the following streaming related functions:
  - passthrough data connections
  - pull splitter data connections
  - initial cache requests
- Bandwidth MaxProxy: determines the total bits per second that CacheOS will generate at any given time for all streaming client connections. This setting sets the maximum limit, in kilobits per second (Kbps). If this variable is set to 0 or left blank, the maximum available bandwidth is used. The default value is 0.
- Connection MaxProxyConnections: determines the total number of RealPlayers (clients) that can connect concurrently. Once this limit is reached, clients that attempt to connect receive an error message, and are not allowed to connect until other clients disconnect.

  Connection MaxProxyConnections can be set from 1 to 32767. The default value is 0.

Setting Proxy Ports

- PNA Proxy Port. The default setting is as follows:

  ```
  Port PNA Proxy = 1090
  ```
- RTSP Proxy Port: The default setting is as follows:

  ```
  Port RTSP Proxy = 1091
  ```

Streaming Media Logging Settings

CacheOS can log streaming activity in the Content Accelerator's access log. RealMedia log entries record the IP addresses of the clients that have connected, the clips they listened to, the times of day they connected, and potentially much more depending on the log format and log style selected.

The settings for logging format and logging style are customizable through the following configuration file variables:

- Logging Enabled: this setting is used to enable RealMedia logging (0=disabled and 1=enabled). Default = 1.

  Note Real Networks streaming activity is logged to the Content Accelerator's HTTP access log. To enable Real proxy logging, you must enable both HTTP access logging (Enable URL access logging) on the Content Accelerator and RealMedia logging. See *RealMedia Log Format* for a sample Content Accelerator access log.

- Logging Style: this setting determines the fields that appear in the Content Accelerator's access log for each RealMedia record. Possible settings range from 0-5 and the Default = 3.
- Logging Stats : this setting lets you select the extra statistics appearing in each RealProxy record in the access log. See the *RealMedia Log Format* section of this document for details related to [Stat 1-3]. Logging Stats can be set to 0-7. Default = 0.

For details related to using and interpreting Logging Style and Logging Stats setting see the *Customizing Information Reported by the Proxy Log* and *RealMedia Log Format* sections of this document.

Splitting Protocol Settings

You can choose one of the following splitting protocols (udp or tcp). The default is udp. See the Splitting section of this document for additional details related to other, non-configurable, splitting defaults for CacheOS.

Configuring Upstream Proxy Settings (Chaining)

To enable chaining you must specify IP addesses for the Upstream Proxy PNA and Upstream Proxy RTSP. For details on chaining see the *Configuring Chaining* section of this document.

The first setting is Upstream Proxy RTSP. This sets the proxy to query if an RTSP request cannot be satisfied locally. The second setting is Upstream Proxy PNA This sets the proxy to query if a PNA request cannot be satisfied locally.

When creating or editing the configuration file, these settings appear as follows. Substitute the IP address and port number of the upstream Proxy server.

Upstream Proxy PNA = *IP address port*

Upstream Proxy RTSP = *IP address port*

The following settings are used to clear the above chaining related parameters:

Upstream Proxy PNA = *none 0*

Upstream Proxy RTSP = *none 0*

Multicast Settings

Multicasting helps you conserve bandwidth. It requires a specially configured network. Multicast setting are customizable through the following configuration file variables. For additional detail on multicasting, refer to the RealServer documentation.

- Multicast Enable: this setting is used to enable or disable multicasting (0=disabled and 1=enabled). Default = 0.
- Multicast SAP: this setting is used to enable or disable announcement of broadcast (1=enable;0=disable). Default = 0

- Multicast AddressRange: is used to specify a multicast address range. The multicast address range must be between 224.0.0.255-239.255.255.255. This parameter does not have a default setting. A multicast address range must for supplied in order for multicasting to work.
- Multicast TTL: sets the number of Router hops allowed. This value has a range of (0-255). Default = 16
- Multicast DeliveryOnly: when enabled, a client must be setup for multicast in order to connect to the stream (1=enable;0=disable). Default = 0
- Multicast Port RTSP: specifies the Multicast RTSP port number. Default = 554
- Multicast Port PNA: specifies the Multicast PNA port number. Default = 7070
- Multicast Accept : Access control (RuleNumber IP Subnet) (Any can replace IP Subnet address to allow anyone access (100 Any)). Default = 100 Any

RealMedia Log Format

CacheOS stores information about each streaming media clip it serves in a separate record. Each record is delimited by a new line. Fields within each record are separated by spaces. One record is created for every clip served. If a client requests a presentation that includes several clips, one record is created for each clip in the presentation. The fields that appear within each record depend on the logging format and logging style.

Assuming all possible streaming related information is being gathered, (Logging style is set to 5), the information logged is shown below:

```
<RealMedia>client_IP_address - - [timestamp] "GET filename protocol/version"
HTTP_error_code bytes_sent [client_info] [client_GUID] file_size file_time sent_time
resends failed_resends [stream_components] start_time server_address average_bitrate
packets_sent presentation_id [proxy_info]
```

Sample Content Accelerator Access Log

```
Access Log Tail
The current time is Fri Mar 24, 2000 16:15:03 GMT
http://www.f5.com/images/gsite_sm.gif - DIRECT/- image/gif
953913510.314       0 10.253.221.37 TCP_HIT/200 4923 GET
http://www.f5.com/images/iw_bos_small.jpg - DIRECT/- image/jpeg
953913510.326       0 10.253.221.37 TCP_HIT/200 8577 GET
http://www.f5.com/images/globalsitebox.jpg - DIRECT/- image/jpeg
.
.
<RealMedia> 10.253.221.18 - - [24/Mar/2000:16:13:56 +0000]  "GET rtsp://ticonderoga-
1.real.com:554/showcase/channels/screeningroom/sponsor.rp RTSP/1.0" 200 3077
[WinNT_4.0_6.0.7.380_plus32_SP61_en-US_686] [00000000-0000-0000-0000-000000000000]
[UNKNOWN] 0 2 0 0 0 [0 0 0 0] [24/Mar/2000:16:13:13] 127.0.0.1 [Demand Cache Hit]
<RealMedia> 10.253.221.18 - - [24/Mar/2000:16:13:56 +0000]  "GET rtsp://ticonderoga-
1.real.com:554/showcase/channels/screeningroom/background.rp RTSP/1.0" 200 9194
[WinNT_4.0_6.0.7.380_plus32_SP61_en-US_686] [00000000-0000-0000-0000-000000000000]
[UNKNOWN] 0 3 0 0 0 [0 0 0 0] [24/Mar/2000:16:13:13] 208.147.89.226 [Demand Pass-
Through]
<RealMedia> 10.253.221.18 - - [24/Mar/2000:16:14:48 +0000]  "GET
rtsp://duwamish.real.com:554/encoder/kingfm_g2.rm?end=1:00:00.0 RTSP/1.0" 200 42826
```

```
[WinNT_4.0_6.0.7.380_plus32_SP61_en-US_686] [00000000-0000-0000-0000-000000000000]
[UNKNOWN] 0 0 0 0 0 [0 0 0 0] [24/Mar/2000:16:14:27] 127.0.0.1 [Live Pass-Through]
```

The following table lists the format for each proxy log record:

<table>
<tr><th colspan="3">Proxy Log Format</th></tr>
<tr><th>Proxy Log Field</th><th colspan="2">Description</th></tr>
<tr><td>client_IP_address</td><td colspan="2">IP address of client, such as 123.45.123.45</td></tr>
<tr><td>- -</td><td colspan="2">Two hyphens for compatibility with standard Web server log formats.</td></tr>
<tr><td>timestamp</td><td colspan="2">Time that client accessed the file in the format:
dd/Mmm/yyyy:hh:mm:ss TZ
where TZ is the time zone expressed as the number of hours relative to the Coordinated Universal Time (Greenwich, England) and is relative to the server. For example:
[31/Oct/1996:13:44:32 -0800]</td></tr>
<tr><td>“GET filename or
“GET URL</td><td colspan="2">Requests for PNA will show the file name (and path) requested by the client.
Requests for RTSP will show the complete URL, beginning with rtsp://.
If the client requests a file that doesn't exist, UNKNOWN appears in place of a file name.</td></tr>
<tr><td rowspan="5">protocol/version”</td><td colspan="2">Application-layer protocol used to send the clip to the client. Possible values are:
RTSP
PNA
In addition, a letter at the end of the string indicates which transport type was used:</td></tr>
<tr><td>(blank)</td><td>UDP connection</td></tr>
<tr><td>T</td><td>TCP connection</td></tr>
<tr><td>M</td><td>Multicast</td></tr>
<tr><td colspan="2">For example, PNAT means that the clip was sent using the PNA protocol over a TCP connection. The version number indicates the edition of the protocol.</td></tr>
<tr><td>HTTP_status_code</td><td colspan="2">Return code using HTTP standard error codes. Usually returns 200.</td></tr>
<tr><td>bytes_sent</td><td colspan="2">Number of bytes transferred to the client.</td></tr>
<tr><td rowspan="3">[client_info]</td><td colspan="2">Describes the version and type of client being used. Client information appears in the following format,
[platform version client type dist_code language CPU]
If client information can't be gathered (the request came from a client that chose not to send statistics, or from a browser connecting to RealSystem Administrator pages), UNKNOWN appears within the brackets.</td></tr>
<tr><td>Field</td><td>Description</td></tr>
<tr><td>platform</td><td>Operating system RealPlayer runs on-Win16, WinNT, Mac, and so on.</td></tr>
</table>

Proxy Log Format		
Proxy Log Field	**Description**	
	version	Operating system version number.
	Client	Version number of RealPlayer.
	Type	Type of RealPlayer.
	dist_code	Distribution code of RealPlayer.
	language	Language setting in RealPlayer.
	CPU	Type of processor on which the client is running. If the processor does not have a hardware Floating Point Unit, the string "no-FPU" is appended to the end of the CPU field with no delimiter. For example: Win95_4.0_3.0.0.19_play32_PN01_EN_586
	RealAudio Player version 1.0 shows only two fields for [client_info]. They are platform and client.	
[client_GUID]	Unique ID generated during RealPlayer installation that enables you to track details for individual clients. If client information can't be gathered (the request came from a client that chose not to send statistics, or from a browser connecting to RealSystem Administrator pages), UNKNOWN appears within the brackets. If the user elects to suppress this information, this field will show a series of zeroes: 00000000-0000-0000-0000-000000000000 instead of a unique identifier. Refer to "Omitting Client Identifiers". Included when Logging Style is set to 2 or higher.	
[Stat1] See the Logging Stats Details section below.	Connection statistics sent by the client when it completes playing a clip. When the client blocks connection statistics, the field is replaced by [UNKNOWN]. Note that there is no space between the closing square bracket of this statistics type and the opening square bracket of the next statistics type. Included when Logging Stats is 1, 3, 5, or 7.	
[Stat2] See the Logging Stats Details section below.	Extended connection statistics sent by the client when it completes playing a clip. When the client blocks connection statistics, the field is replaced by [UNKNOWN]. Note that there is no space between the closing square bracket of this statistics type and the opening square bracket of the next statistics type. Included when Logging Stats is 2, 3, 6, or 7.	
[Stat3] See the Logging Stats Details section below.	Actions taken by the visitor while playing the clip. When the client preferences are set to block statistics, this field is replaced by [UNKNOWN]. Note that there is no space between the closing square bracket of the previous statistics type and the opening square bracket of this statistics type. Included when Logging Stats is 4, 5, 6, or 7.	

Proxy Log Format		
Proxy Log Field	**Description**	
file_size	Reserved for future use. Currently this information is not recorded. Included when Logging Style is set to 1 or higher.	
file_time	Reserved for future use. Currently this information is notrecorded. Included when Logging Style is set to 1 or higher.	
sent_time	Total length, in seconds, of the media sent to the client. Included when Logging Style is set to 1 or higher.	
resends	Number of packets successfully resent because of transmission errors. Included when Logging Style is set to 1 or higher.	
failed_resends	Number of packets not successfully resent in time to correct transmission errors. Included when Logging Style is set to 1 or higher.	
[stream_ components]	Type of material sent, indicated in the following pattern: RealAudio RealVideo Event RealImage 1 shows that the stream includes this type, 0 indicates that it does not. Thus, a stream that included RealVideo and RealAudio but no events or RealImages would appear in the proxy log as: 1 1 0 0. Included when Logging Style is set to 3 or 4.	
start_time	Timestamp of start time.Included when Logging Style is set to 3 or 4.	
server_address	IP address where clip came from. This may be the source RealServer, a RealServer which is acting as a receive splitter, or a RealProxy server which is acting as a receive splitter. In cache mode, RTSP requests will show the cache's address (usually 127.0.0.1). To find the address of the source RealServer, look in the GET field (see “GET filename or “GET URL). Included when Logging Style is set to 3 or 4.	
average_bitrate	Average bitrate of clip. Included when Logging Style is set to 4.	
packets_sent	Number of packets sent. Included when Logging Style is set to 4.	
presentation_id	Number used by other clips in a SMIL presentation. All elements from the same presentation use the same number. The SMIL file itself is also included in the log, and shares the number as well. The number is assigned by RealProxy at the time of transmission. Included when Logging Style is 5.	
[proxy_info]	Displays information about the type of proxied stream (always included):	
	Value	Meaning

Proxy Log Format		
Proxy Log Field	**Description**	
	Demand Pass-Through	The proxied stream was an on-demand clip, and it was sent in passthrough mode.
	Live Pass-Through	The proxied stream was a live clip, and it was sent in passthrough mode.
	Live Split	The proxied stream was a live clip, and it was sent via push splitting.
	Demand Cache Hit	The proxied stream as an on-demand clip, and CacheOS served it from the media cache.
	Unknown	Clip type and delivery were of unknown type.

Logging Stats Details

The information gathered by each of the three Statistics Types are listed in this section. Stat1 and Stat2 report information about the RealAudio portion of a clip. Even if a clip includes both RealAudio and RealVideo, these statistics report solely RealAudio information. Stat3 reports information about visitor and client behavior while playing all types of clips or presentations.

When Logging Stats is 0, two square brackets [ ] appear instead of the Stat1, Stat2, and Stat3 sections.

Stat1 Syntax

Statistics Type 1 gathers basic information about how successfully audio clips were received by the client. It also tells what the client used to decode the audio portion of the clip.

Fields included in this portion of the access log record are as follows:

```
[Stat1: packets_received out_of_order missing early late audio_format]
```

The table below details the information collected by statistics type 1:

Statistics Type 1 Information	
Field	**Description**
packets_received	Total number of packets received by the client.
out_of_order	Number packets received by the client out of order. These packets are reordered as they are being played by the client.
missing	Number of packets requested by the client, but that the client did not receive.
early	Number of requested packets received too early by the client.

Statistics Type 1 Information	
Field	**Description**
late	Number of packets received too late by the client.
audio_format	Name of the decoder used to play the clip. Possible values are: sipr RealAudio 5.0 formats dnet RealAudio 3.0 formats 28.8 RealAudio 2.0 28.8 format lpcJ RealAudio 2.0 14.4 format cook RealAudio G2 format

Stat2 Syntax

Statistics Type 2 provides details about the success of clip delivery, giving information about bandwidth requests. Re-sent packets are described in detail here. It identifies which transport type was used to make the connection and which video decoder played the clip. Fields included in this portion of the access log record are as follows:

```
[Stat2: bandwidth available highest lowest average requested received late rebuffering
transport startup format]
```

The table below explains what information is collected by statistics type 2:

Statistics Type 2 Information	
Field	**Description**
bandwidth	Bandwidth of the clip, in bits per second.
available	Average bits per second available to the user while the clip was playing.
highest	Highest time between the client resend packet request and the packet resend arrival, in milliseconds.
lowest	Lowest time between the client resend packet request and the packet resend arrival, in milliseconds.
average	Average time between the client resend packet request and the packet resend arrival, in milliseconds.
requested	Number of resend packets requested by the client.
received	Total number of re-sent packets received by the client.
late	Number of re-sent packets received by the client too late.
rebuffering	Rebuffering percentage for the clip.
transport	Transport type for the connection. Values are: 0: UDP 1: TCP

Statistics Type 2 Information	
Field	**Description**
	2: IP Multicast 3: PNAviaHTTP
startup	Time when the client receives the first clip data, in milliseconds. The data may arrive before the clip starts playing.
format	Name of the decoder used to play the clip. Possible values are: sipr RealAudio 5.0 formats dnet RealAudio 3.0 formats 28.8 RealAudio 2.0 28.8 format lpcJ RealAudio 2.0 14.4 format cook RealAudio G2 format

Stat3 Syntax

Statistics Type 3 provides detailed information about viewer action while listening or viewing clips. It addresses advanced features of the implementation, notably ads and image maps. You can find out at what point in the clip a viewer clicked on an image map or stopped watching the clip.

If Logging Stats is configured to gather statistics type 3 (Stat3), note that the access log file size will grow rapidly. If you configure Logging Stats to collect this information, be sure to review the log file frequently. This statistics type uses the following format:

```
[Stat3:timestamp|elapsed_time|action|;]
```

Records of activity are separated by a semicolon (;) and are in the following form:

```
timestamp|elapsed_time|action|;
```

Thus, the Stat3 record of a visitor pausing, resuming play, and watching to the clip's end would look like the following:

```
[Stat3:4360|2107|PAUSE|;8401|2107|RESUME|;12608|6321|STOP|;]
```

The table below describes the information collected by statistics type 3:

Statistics Type 3 Information	
Field	**Description**
timestamp	Time in milliseconds when action occurred. It is relative to the connect time of the client.
elapsed_time	Elapsed time of the clip when the behavior occurred, given in milliseconds.
action	The visitor's or client's behavior, where values are the following:

Statistics Type 3 Information					
Field	**Description**				
	ABORT	Abnormal client stop (not the natural end of clip play).			
	CLICK	Visitor clicked on the image map. Further information includes:			
		x-coord	Horizontal coordinate of click.		
		y-coord	Vertical coordinate of click.		
		action	Action that occurred. This is one of the following:		
			PLAYER="url"	The URL of the link the viewer clicked, as used in the client	
			URL="url"	The URL of the link the viewer clicked, as used in the Browser.	
			SEEK="destination"	The seek destination point, in milliseconds.	
	PAUSE	The visitor paused the client.			
	RESUME	Resume play after a pause, seek or stop.			
	SEEK	The seek destination point, in milliseconds.			
	STOP	End of clip reached.			
	RECSTART	RealPlayer Plus began recording the clip.			
	RECEND	RealPlayer Plus stopped recording the clip.			

Logging Style Record Formats

The format of the proxy log under each of the 6 different logging style values is shown in the table below. The default logging style is 3.

How Logging Style Value Effects Record Format	
Logging Style value	**Individual record format**
0	client_IP_address - - [timestamp] "GET filename protocol/version" HTTP_status_code bytes_sent [client_info] [client_GUID] [proxy_info]

How Logging Style Value Effects Record Format	
Logging Style value	**Individual record format**
1	client_IP_address - - [timestamp] “GET filename protocol/version” HTTP_status_code bytes_sent [client_info] [client_GUID] file_size file_time sent_time resends failed_resends [proxy_info]
2	client_IP_address - - [timestamp] “GET filename protocol/version” HTTP_status_code bytes_sent [client_info] [client_GUID] file_size file_time sent_time resends failed_resends [proxy_info]
3	client_IP_address - - [timestamp] “GET filename protocol/version” HTTP_status_code bytes_sent [client_info] [client_GUID] file_size file_time sent_time resends failed_resends [stream_components] start_time server_address [proxy_info]
4	client_IP_address - - [timestamp] “GET filename protocol/version” HTTP_status_code bytes_sent [client_info] [client_GUID] file_size file_time sent_time resends failed_resends [stream_components] start_time server_address average_bitrate packets_sent [proxy_info] [proxy_info]
5	client_IP_address - - [timestamp] “GET filename protocol/version” HTTP_status_code bytes_sent [client_info] [client_GUID] file_size file_time sent_time resends failed_resends presentation_id [proxy_info]

Customizing Information Reported by the Proxy Log

Logging Style and Logging Stats are used to customize the information gathered in the Content Accelerator access log for RealServer activity.

Note Information recorded by the source RealServer is similar to the information collected in the Content Accelerator access log. Information related to client requests is stored in both the Content Accelerator access log and on the RealServer. Content Accelerator RealMedia related log settings are are independent of the RealServer log settings. For example, CacheOS may be configured to record Logging Style 0, and RealServer may be collecting Logging Style 5 information.

Changing Information Gathered with Logging Stats

Logging Stats supplies more detailed information to the access log. This variable is optional. For a complete description of information collected by each statistics type, and the syntax of the types as they appear in the access log, see the *Logging Stats Details* section of this document.

If you omit a value for Logging Stats, the default value of 3 is used (gather statistics types 1 and 2).

Collecting Combinations of Logging Stats Information				
To gather this information...	**...set Logging Stats to this value**	**Statistics Type 1**	**Statistics Type 2**	**Statistics Type 3**
No additional statistics	0			
Statistics type 1 only	1	*		
Statistics type 2 only	2		*	
Both statistics types 1 and 2	3	*	*	
Statistics type 3 only	4			*
Both statistics types 1 and 3	5	*		*
Both statistics types 2 and 3	6		*	*
All statistics (types 1, 2, and 3)	7	*	*	*

Not all versions of RealPlayer supply the information requested by Logging Stats:

- Statistics type 2 is supplied by RealAudio Player versions 3.0 and later.
- Statistics type 3 is supplied by RealPlayer versions 5.0 and later.

Gathering Information with Logging Style

Logging Style provides six options, styles 0 through 5. Styles 1 through 4 include their own incremental information plus the information provided by logging styles with lower numbers. For example, Logging Style 3 provides the information collected by styles 0, 1, 2 and 3. Logging Style 5 consists of the fields in Logging Style 2, plus the presentation_id field.

See the table below and the preceeding *Proxy Log Format* section for more information.

Information Collected by Logging Style	
To gather this information...	**...set LoggingStyle to this value**
Bytes sent	0 or higher
Clip name including path	0 or higher
Client IP address and platform information	0 or higher

Timestamp	0 or higher
Packets successfully and unsuccessfully re-sent	1 or higher
Protocol (RTSP or PNA)	1 or higher
Send time (total media sent in seconds)	1 or higher
Transport method (TCP, UDP) and version	1 or higher
Client ID	2 or higher
Server IP Address	3 or 4
Stream components	3 or 4
Timestamp for start time	3 or 4
Average bitrate	4
Packets sent	4
Common presentation identifier	5

Error Logging

The Content Accelerator's event log records client connections and RealProxy errors. Each time a streaming media error is generated, a record is created in the Content Accelerator's event log.

Error Log Format

A Real Networks streaming media entry in the Content Accelerator's event log uses the following syntax. Details are provided in the following table:

```
date time "***date time plogplin(process_ID): error_message" plogplin.cpp:595
```

For example:

```
2000-03-27 17:05:34+00:00UTC  "***28-Mar-74 17:05:34.540 plogplin(1065757960): RTSP
Redirector: Could not initialize the plugin
"  0 230000:64   plogplin.cpp:595
```

Error Log Syntax	
Entry	**Meaning**
***	Three asterisks indicate an error. Informational messages are not preceded by asterisks.

Error Log Syntax	
Entry	**Meaning**
date	Date on which the error occurred. Given in the form d-Mmm-YY.
time	Time the error occurred, according to RealProxy. Given in the form HH:MM:SS:TT.hhh
plogplin	Proxy logging plugin
(process_ID)	The process ID in parentheses.
error_message	Text of error message

Installing Custom Real Networks Streaming Settings

The CacheOS default configuration is set to work with RealServer 7.0. No other related configuration is required unless you wish to use chaining. If you decide to change the default configuration, you must create or modify a Real Networks configuration file. Once the configuration file has been created, place it on an HTTP or FTP server visible to the Content Accelerator.

Note Always restart the Content Accelerator after changing any of the following settings in the configuration file:

- Port PNA Proxy
- Port RTSP Proxy
- Logging Stats
- Upstream Proxy RTSP
- Upstream proxy PNA
- Multicast AddressRange
- Multicast Enable
- Multicast SAP
- Multicast TTL
- Multicast DeliveryOnly
- Multicast Accept
- Multicast Port RTSP
- Multicast Port PNA
- Connection MaxProxyConnections

To install custom Real Networks streaming settings

1. Select Management from the CacheOS home page.

2. Select the Maintenance applet.
3. Select the Streaming tab.
4. In the Install Real Networks Streaming from field, enter the path to the configuration file to be installed. You can click View to display the configuration file before installing it.
5. Click Install to download the configuration file.
6. Click Apply to save changes.

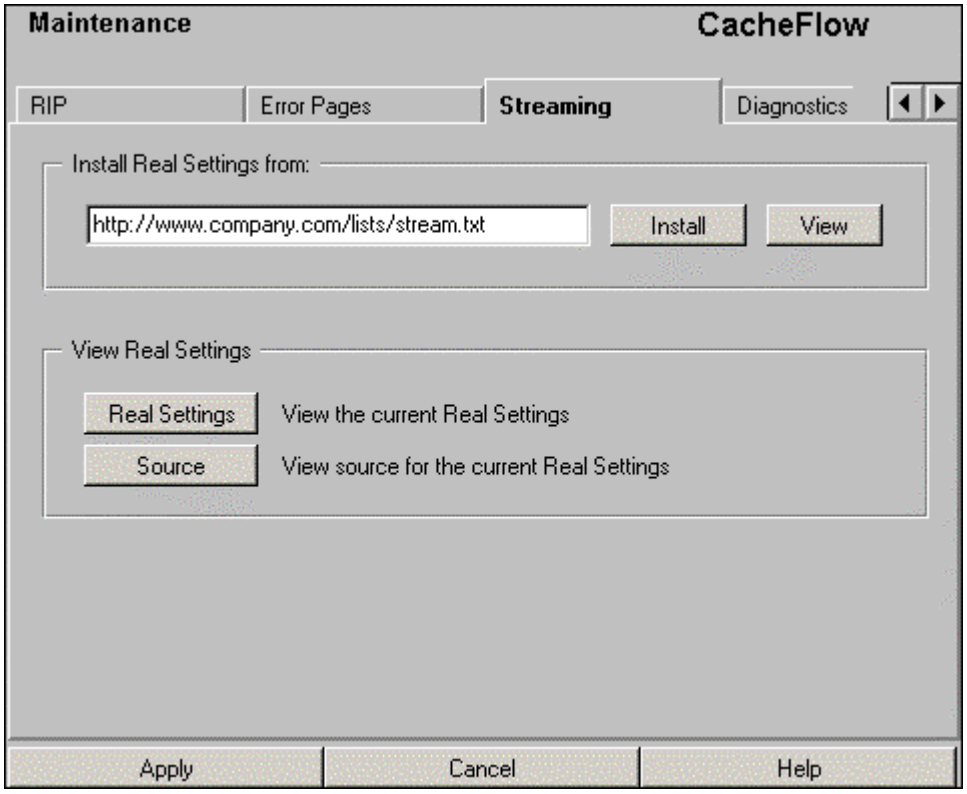


Figure 13-11 Installing a Real Networks streaming configuration

To install custom Real Networks streaming settings using the CLI

1. Open a terminal session with the Content Accelerator and type your Username and Password when prompted.
2. At the command prompt, type **enable** and type your Password when prompted.
3. At the command prompt, type **configure terminal** to enter terminal configuration mode.
4. At the `(config)` prompt, type **streaming real-networks path *URL*** to set the location of the RealProxy configuration file.
5. Type **load streaming real-networks** to install the configuration file.
6. Type **exit** to leave configuration mode.

To clear the download path to the Real Networks streaming configuration file using the CLI

1. Open a terminal session with the CacheFlow device and type your Username and Password when prompted.
2. At the command prompt, type **enable** and type your Password when prompted.
3. At the command prompt, type **configure terminal** to enter terminal configuration mode.
4. At the `(config)` prompt, type **streaming real-networks no path** to clear the download path.
5. Type **exit** to leave configuration mode.

Configuring Chaining

Chaining allows the connecting of several Content Accelerators on a network so all client requests for streamed media go through a single point. By forwarding requests handled by multiple Content Accelerators to a single Content Accelerator, all client requests for streaming media are funneled through one point.

To implement chaining on a Content Accelerator, two settings must be specified in the Real Networks configuration file, and the customized file installed on the Content Accelerator.

The first setting is Upstream Proxy RTSP. This sets the proxy to query if an RTSP request cannot be satisfied locally. The second setting is Upstream Proxy PNA This sets the proxy to query if a PNA request cannot be satisfied locally.

When creating or editing the configuration file, these settings appear as follows. Substitute the IP address and port number of your upstream RealProxy server.

Upstream Proxy PNA = ***IP address port***

Upstream Proxy RTSP = ***IP address port***

The following settings are used to clear the above chaining related parameters:

Upstream Proxy PNA = ***none 0***

Upstream Proxy RTSP = ***none 0***

Setting Up RealPlayer

To utilize Content Accelerator Real Networks media streaming services the client machine must have RealPlayer installed and configured to use RTP/RTSP streams. Use the following procedure to configure RealPlayer:

1. Start RealPlayer and Preferences from the View menu.

Figure 13-12 Configuring RealPlayer

2. Select the Proxy tab. Select Use PNA proxy and Use RTSP proxy and type the IP address of the related proxy server. The PNA proxy Port value should be set to 1090 and the RTSP proxy Port value should be set to 1091. For HTTP Options, select the Use my web browser's HTTP proxy radio button.

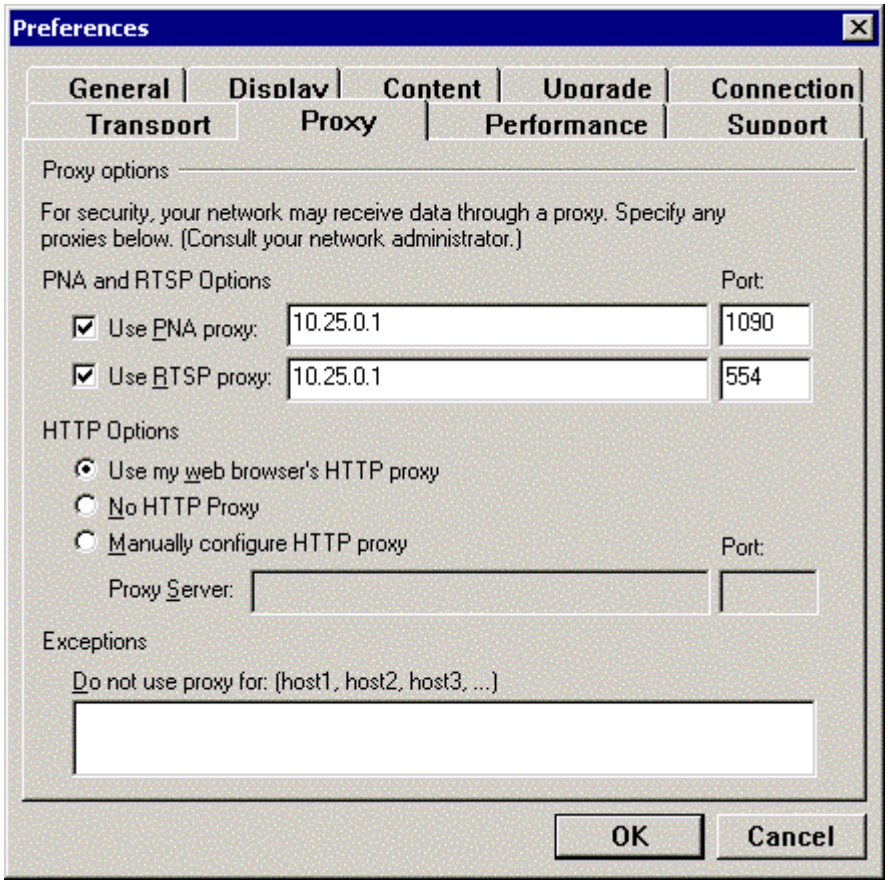


Figure 13-13 Configuring RealPlayer Proxy Settings

3. To configure RealPlayer transport settings, select the Transport tab. Then select the Use specified transports radio button and click RTSP Settings.

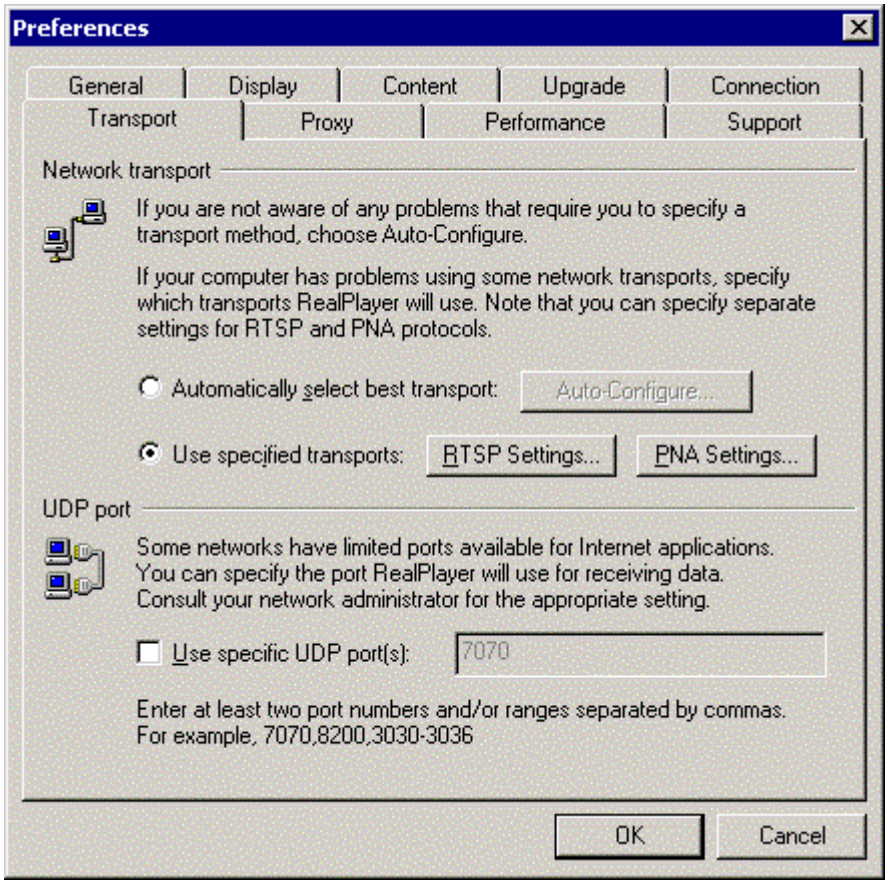


Figure 13-14 Configuring RealPlayer Transport Settings

4. Use the following RTSP transport settings:

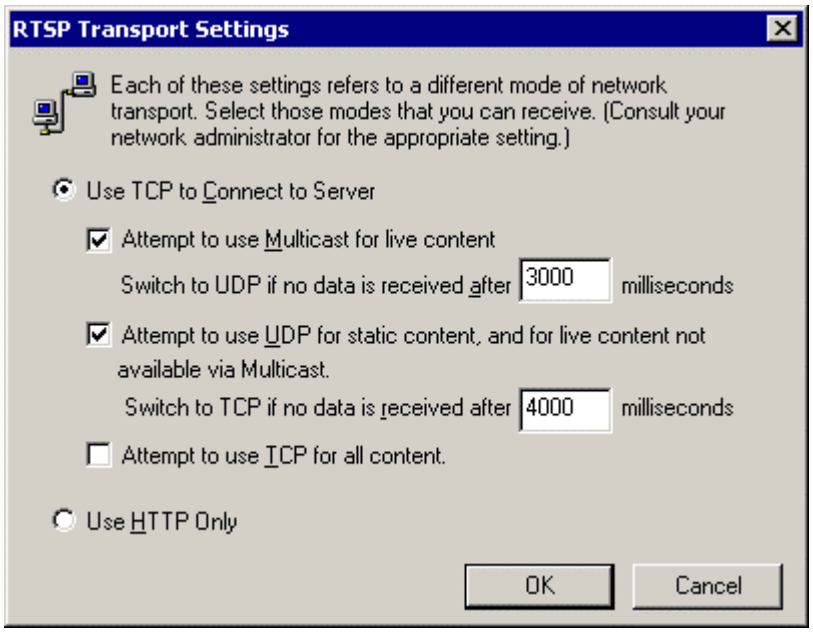


Figure 13-15 Configuring RealPlayer RTSP Settings

5. Use the following PNA transport settings:

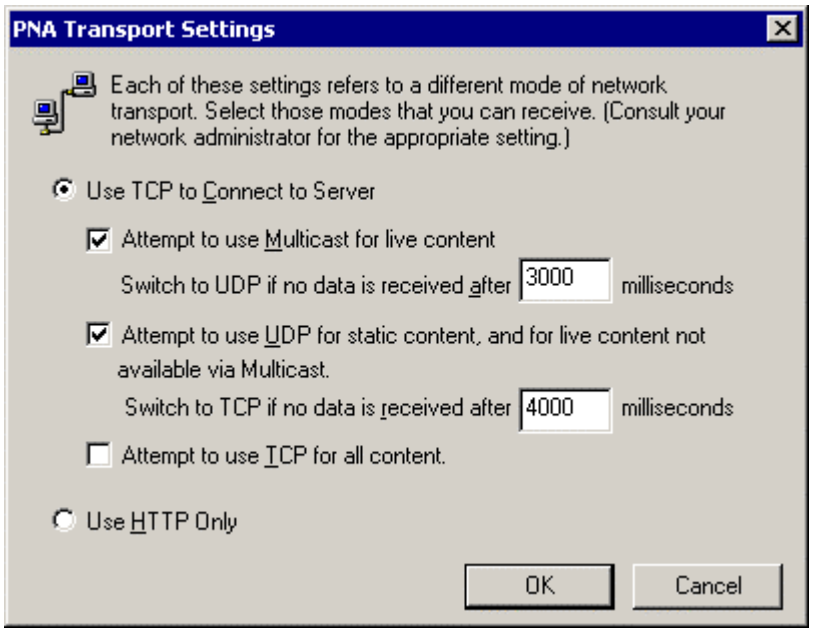


Figure 13-16 Configuring RealPlayer PNA Settings

Configuring Diagnostic Reporting

The Diagnostics tab allows you to control whether Daily Heartbeats and/or CacheFlow Monitoring are enabled or disabled.

Heartbeats are messages sent once every 24 hours. They contain the Content Accelerator's statistical and configuration data. Besides telling the recipient that the device is “alive”, heartbeats also show the Content Accelerator's “health.” System administrators and CacheFlow generally receive heartbeats.

CacheFlow Monitoring enables CacheFlow to gather heartbeat messages to track Content Accelerator “health” in the field. These data can then be used to troubleshoot system difficulties. Additionally, CacheFlow analyzes statistics for customer use through the CacheFlow Enterprise Manager. This option enables the sending of daily and emergency heartbeat messages to CacheFlow through HTTP or SMTP. If disabled, CacheFlow will not recieve any heartbeat messages, even if "heartbeat@mail.heartbeat.cacheflow.com" appears in the Maintenance/Events/Mail list. This option is enabled by default.

CacheFlow receives emergency heartbeats whenever a Content Accelerator is rebooted. Emergency heartbeats contain core dump and restart flags, in addition to daily heartbeat information.

To set daily hearbeats and/or CacheFlow Monitoring

This option enables the sending of daily heartbeat messages to everyone on the Maintenance/Events/Mail list. This option is enabled by default.

1. Select Management from the CacheOS home page.
2. Select the Maintenance applet.
3. Select the Diagnostics tab.
4. In the Monitoring box, enable or disable Daily Heartbeats and Cacheflow Monitoring as desired.
5. Click Apply to save changes.

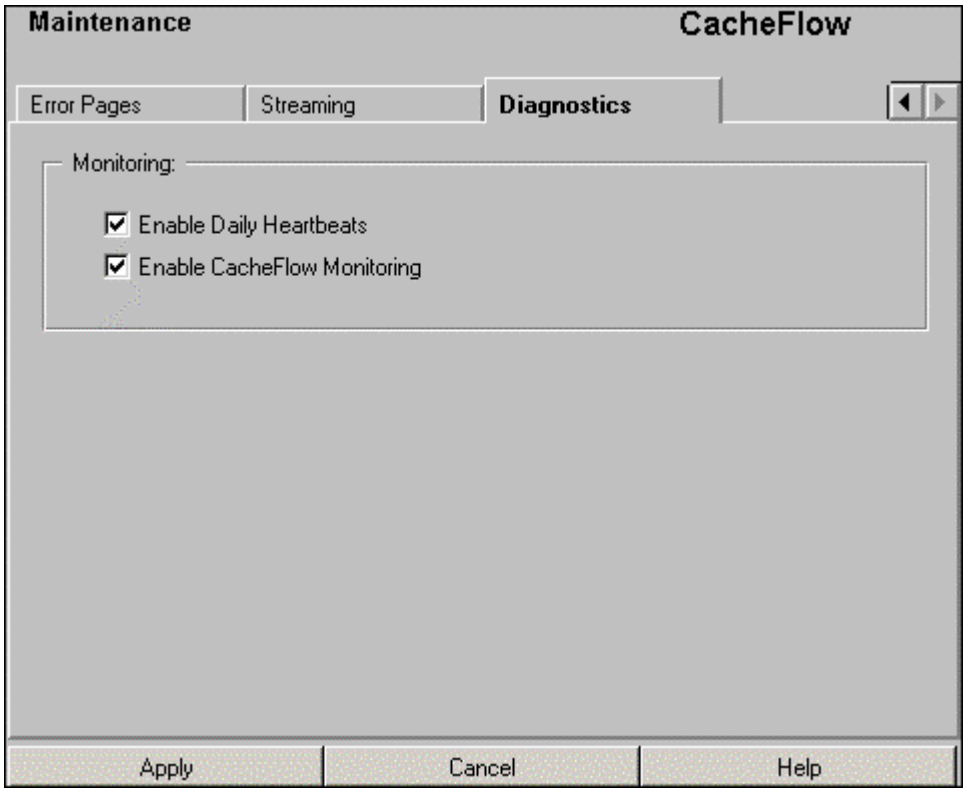


Figure 13-17 Setting diagnostic reporting options

To set daily hearbeats and/or CacheFlow Monitoring using the CLI

1. Open a terminal session with the Content Accelerator and type your Username and Password when prompted.
2. At the command prompt, type **enable** and type your Password when prompted.
3. At the command prompt, type **configure terminal**.
4. At the `(config)` prompt type **event log**.
5. At the `(config event-log)` prompt, type **mail add *email@address.com*** to add an email recipient to event log notifications.
6. At the `(config )` prompt, type **mail cacheflow-notify** to include CacheFlow in event log notifications.
7. At the `(config event-log)` prompt, type **exit** to leave event log configuration mode.

Chapter 14 - System Statistics

The Statistics section of the Web console allows you to graphically view the status of many system operations, as well as take disks offline, and put them online. Many statistics are available through the CLI, but without the benefit of graphical display.

The CLI also provides a great deal of detailed system information. Using the **show ?** command while in privileged mode lists the many subcommands to view a great deal of system configuration information in addition to the statistics discussed here. Refer to the CacheOS Command Reference appendix for detailed information on using the **show** command.

Setting the Graph Scale

Some graphs offer the option to switch between viewing statistics in bytes or objects. To switch between viewing modes, select byte or object mode from the Percentages reflect drop down list.

Some statistics are reported on the form of bar graphs. Most bar graphs offer the option to show all values in the graph, or to clip a percentage of the peak values. When you clip a percentage of the peak values, that percentage is allowed to fall off the top of the scale. For example, if you clip 25% of peaks, the top 25% of the values will be allowed to exceed the scale for the graph, showing greater detail for the remaining 75% of the values. To set the graph scale, select the value you want to display from the Graph scale drop down list.

General Statistics

The general statistics group of applets provide information about system configuration, the status of hardware sensors, and allows you to take disks offline, and put them online.

Viewing a System Summary

The device provides a variety of information on its status. The fields on the Summary tab are described below:

- Disks installed
  The number of disk drives installed in the device. The Disks tabs display the status of each drive.
- Memory installed
  The amount of RAM installed in the device.
- Software version
  The version of the device server image.
- Release ID
  Unique ID for the device release.
- Last access log upload
  The time and date the access log was last uploaded.
- Current access log size
  The current size of the access log.

- System started
  The time and date the device was started.
- CPU utilization
  The current utilization of the device CPU.

To view a system summary

1. Select Statistics from the CacheOS home page.
2. The system summary is displayed in the General applet.
3. Select the Environment tab to display the current hardware status and view system sensors.
4. Select the Disks tabs to display information on the installed disks, and to place disks on and offline.

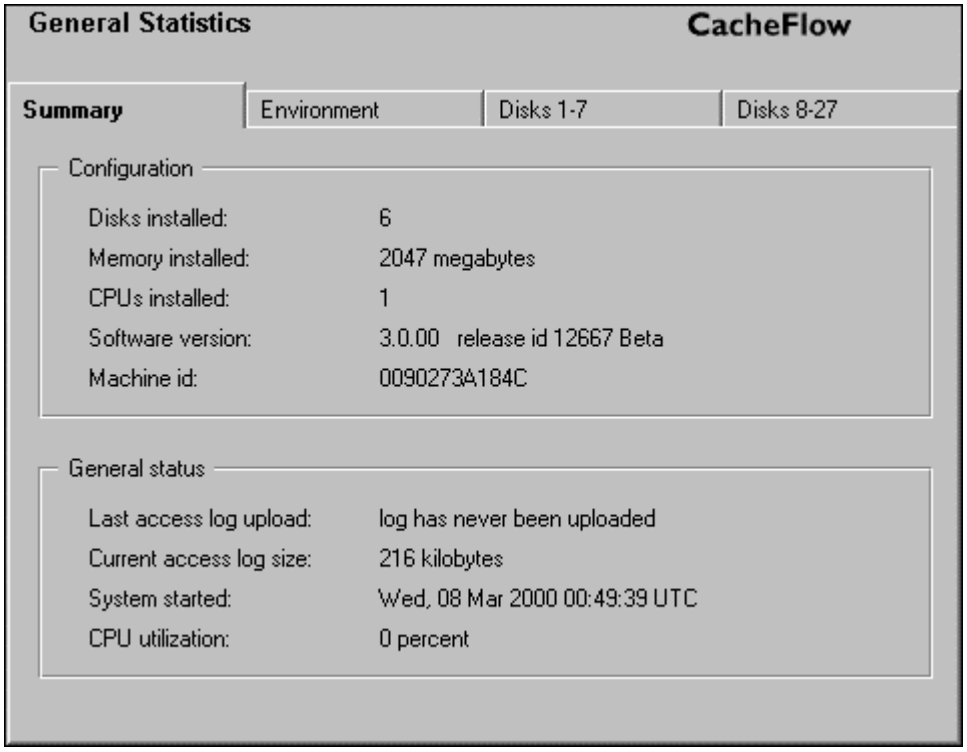


Figure 14-1 Displaying a system summary

The icons on the Environment tab are green when the related hardware environment is within acceptable parameters, and red indicates an out of tolerance condition. If an icon is red click View Sensors to view the sensor statistics to learn more about the determine more about the table. The Sensor statistics table shows you the status of the listed devices. The number out of tolerance condition.

Viewing the Volume of Data Traffic

The Volume group of applets allow you to view information about the data flow into and out of the Content Accelerator.

Viewing the Number of Objects Served

The Objects tab illustrates the device activity over the last 60 minutes, 24 hours, and 30 days. These charts illustrate the total number of objects served from either the cache, or from the Web. To review the number of cached objects versus non-cached objects, display the Efficiency page.

To view the number of objects served

1. Select Statistics from the CacheOS home page.
2. Select the Volume applet.

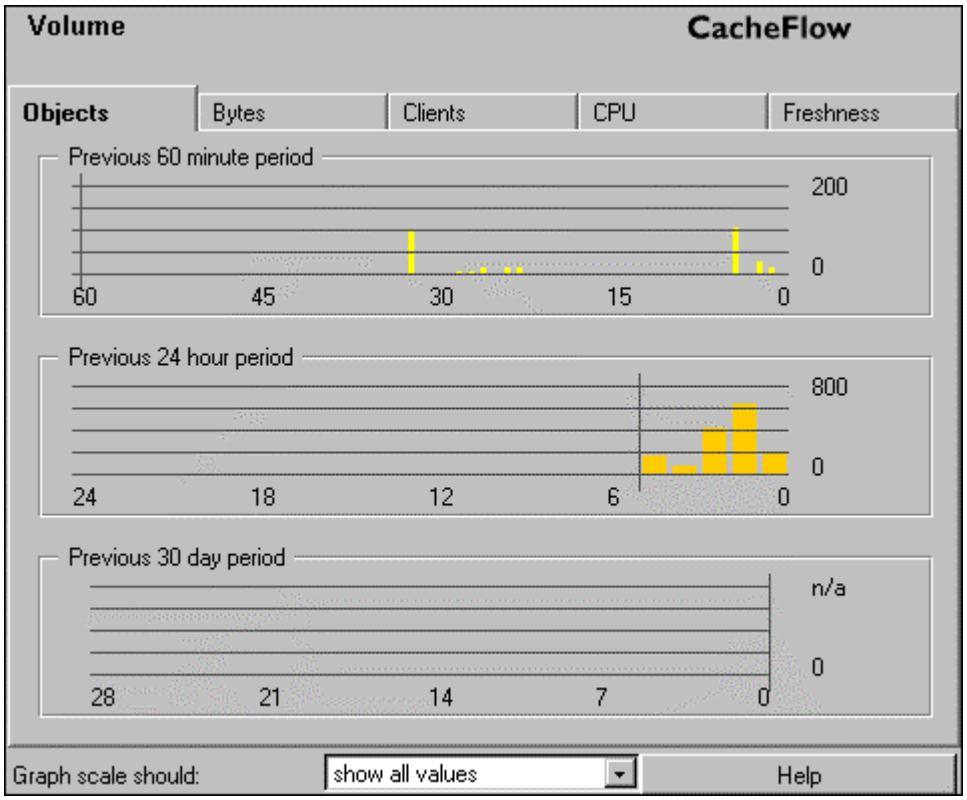


Figure 14-2 Displaying the number of objects served

Viewing the Number of Bytes Served

The Bytes tab shows the sum total of the number of bytes served from the device over the last 60 minutes, 24 hours, and 30 days. Thee chart shows the total number of bytes for objects served by the device, including both cache hits and cache misses.

To view the number of bytes served

1. Select Statistics from the CacheOS home page.
2. Select the Volume applet.
3. Select the Bytes tab.

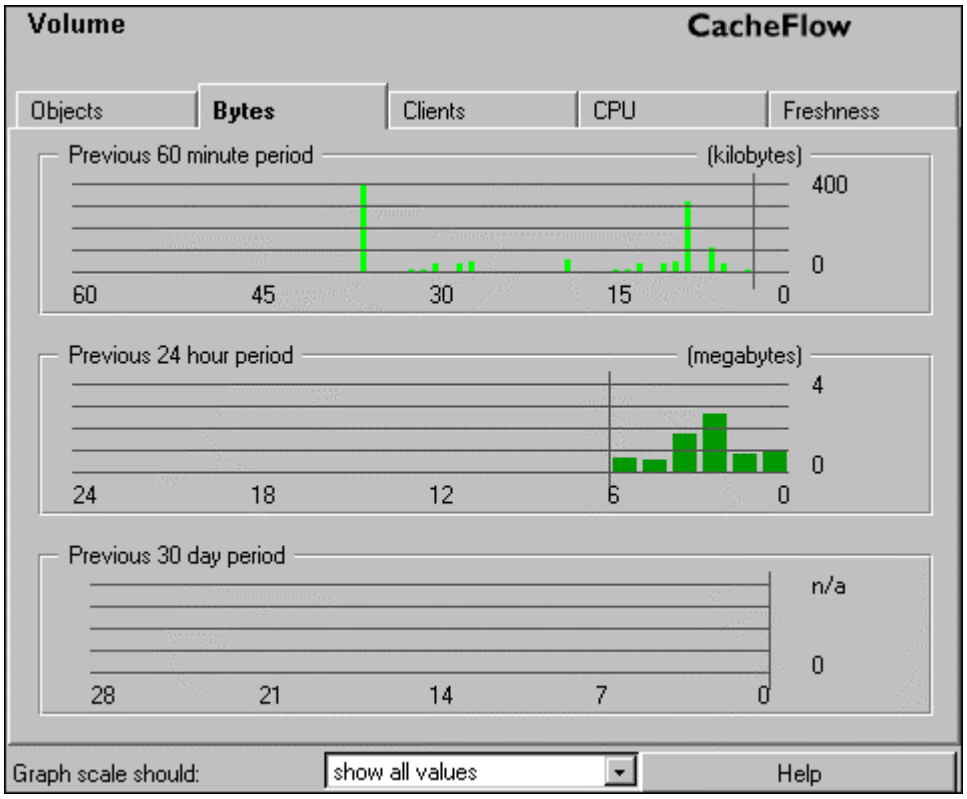


Figure 14-3 Displaying the number of bytes served

Viewing Active Client Connections

The Clients tab shows the maximum number of clients with requests processed over the last 60 minutes, 24 hours, and 30 days. This does not include idle client connections (connections that are open but which have not made a request). These charts allow you to monitor the maximum number of active clients accessing the Content Accelerator at any one time. In conjunction with the Objects and Bytes tabs, you can determine the the number of clients supported based on load, or load requirements for your site based on a specific number of clients.

To view the number of active clients

1. Select Statistics from the CacheOS home page.
2. Select the Volume applet.
3. Select the Clients tab.

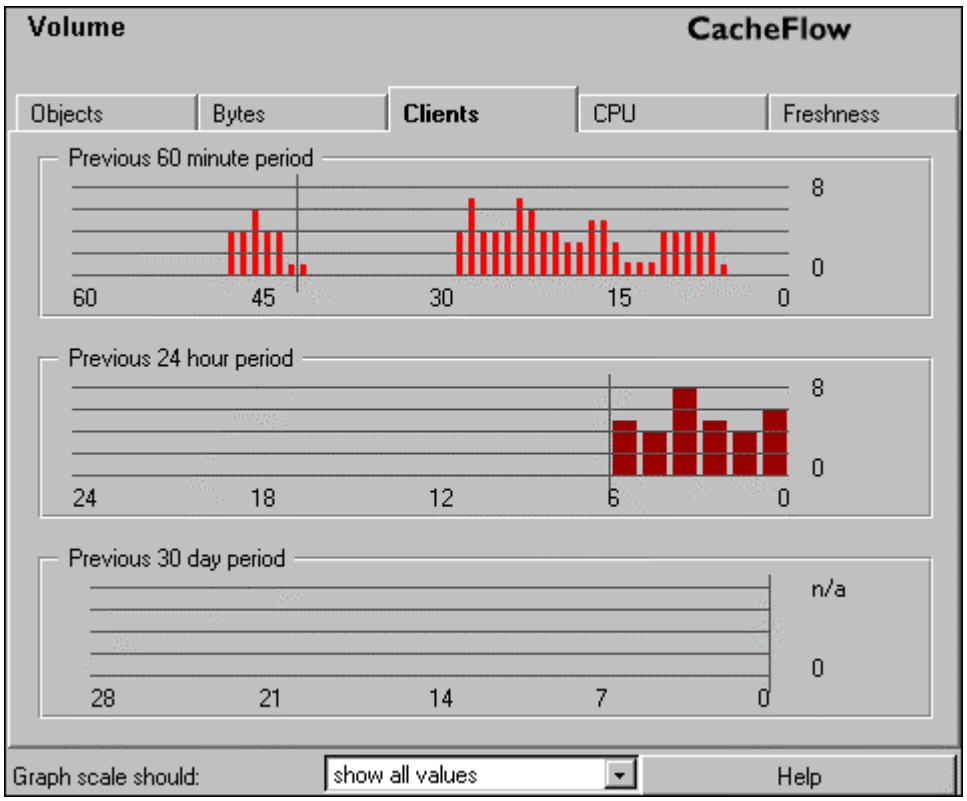


Figure 14-3 Displaying active clients

Viewing CPU Utilization

The CPU tab illustrates the CPU utilization for the device over the last 60 minutes, 24 hours, and 30 days.

To view CPU utilization

1. Select Statistics from the CacheOS home page.
2. Select the Volume applet.
3. Select the CPU tab.

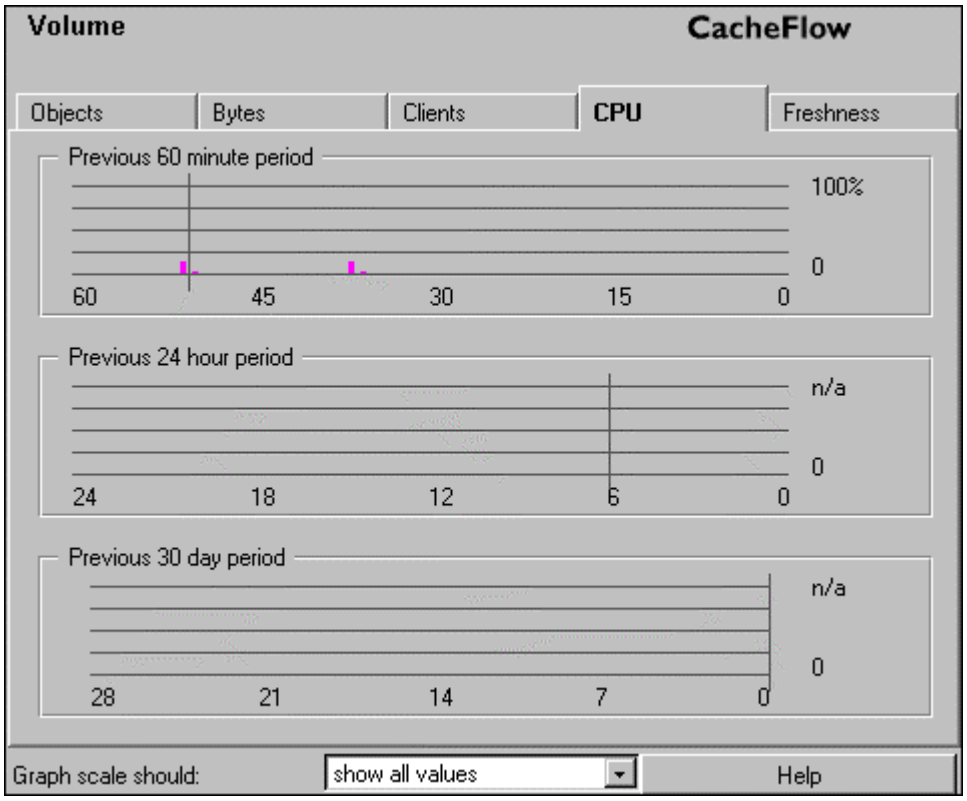


Figure 14-4 Displaying CPU utilization

Viewing Cache Freshness

The Freshness tab illustrates the estimated freshness of objects in the cache over the last 60 minutes, 24 hours, and 30 days.

The freshness applies only to objects that are cached (all objects that are not cached are always 100% fresh). Freshness describes statistically the percentage of objects in the cache that are fresh. For example, if the estimated freshness is 99%, that means when you request an object there is a 99% chance that object is fresh in the cache.

To view cache freshness

1. Select Statistics from the CacheOS home page.
2. Select the Volume applet.
3. Select the Freshness tab.

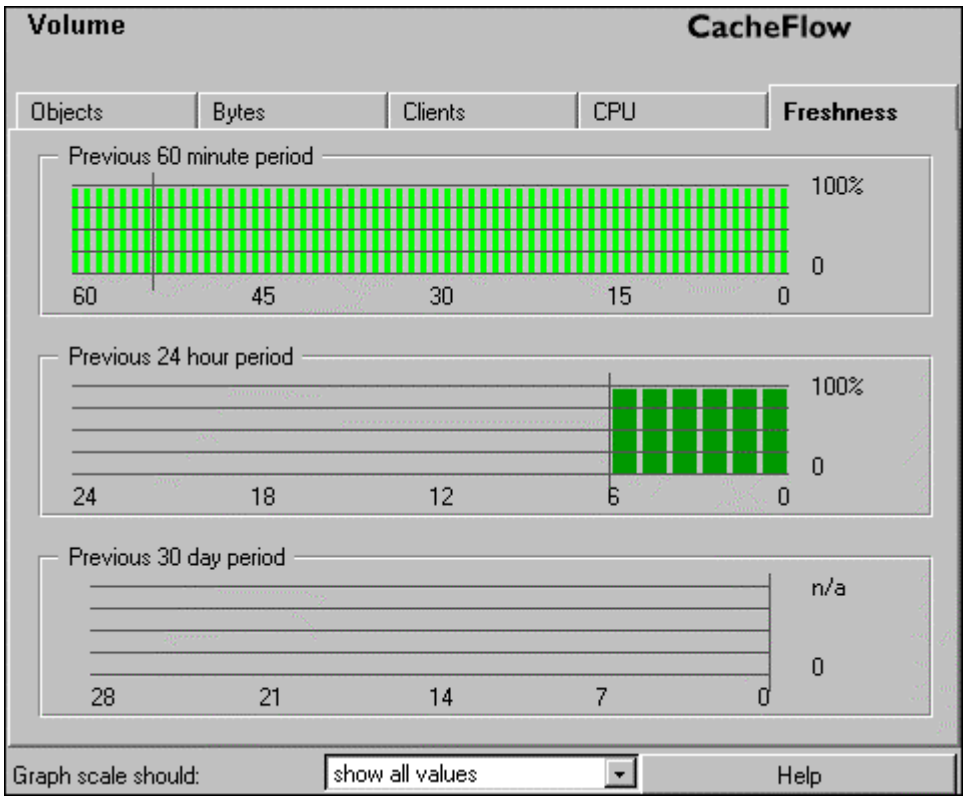


Figure 14-5 Displaying cache freshness

Viewing Streaming Client Statistics

These statistics do not appear if Real Networks has not been activated.

The Str. Clients tab shows the number of active streaming-client connections over the last 60 minutes, 24 hours and 30 days.

To view streaming client statistics

1. Select Statistics from the CacheOS home page.
2. Select the Volume applet.
3. Select the Str. Clients tab.

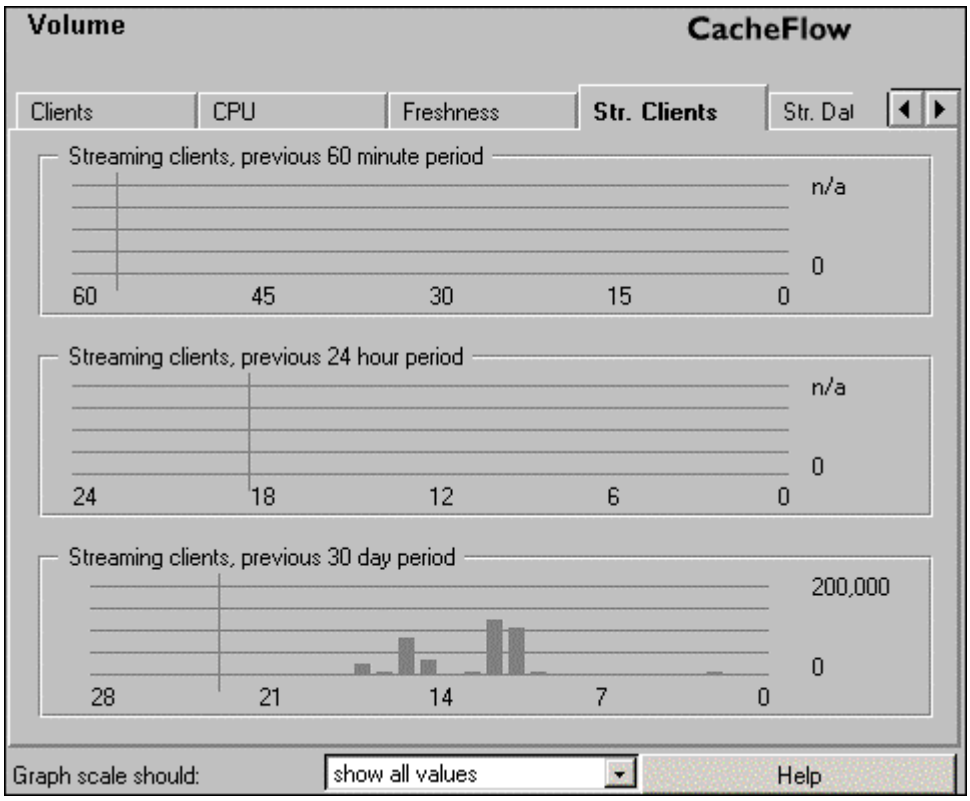


Figure 14-6 Displaying streaming client connections

Viewing Streaming Data Statistics

These statistics do not appear if Real Networks has not been activated.

The Str. Data tab shows real-time values for the number of connected streaming clients, data source, proxy, cache imports, splitter imports and the total streaming data volume. The Total clients served statistic is cumulative, and is reset only when the Content Accelerator is reset to factory defaults.

To view streaming data statistics

1. Select Statistics from the CacheOS home page.
2. Select the Volume applet.
3. Select the Str. Data tab.

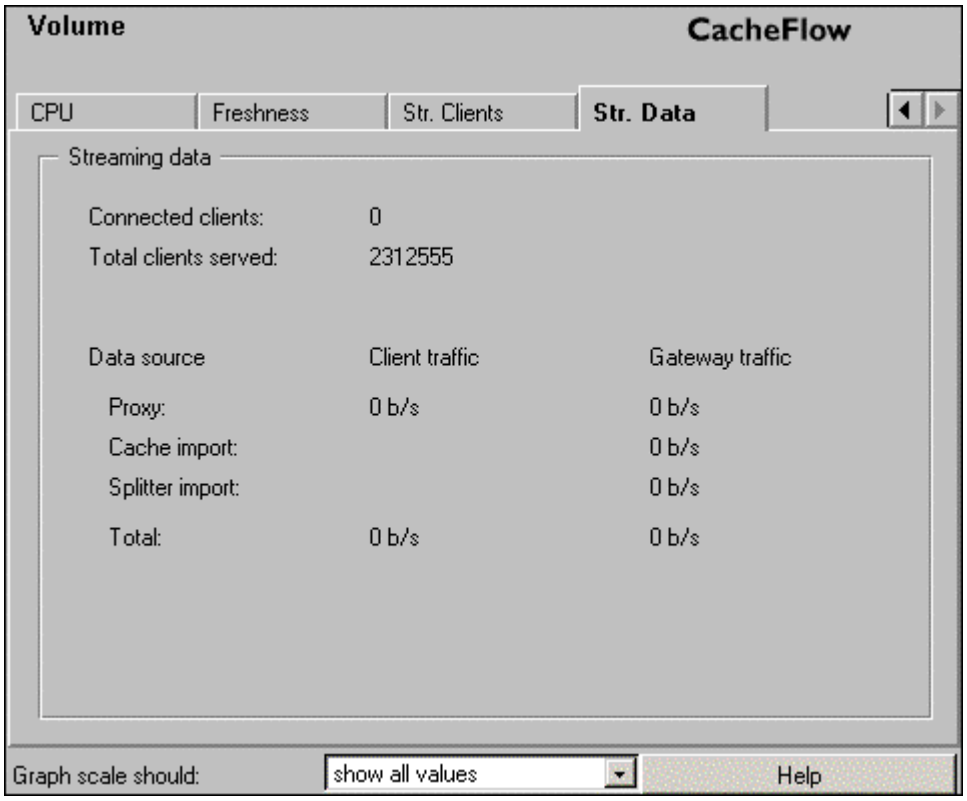


Figure 14-7 Displaying streaming data statistics

Viewing Resource Use

The Resources group of applets allow you to view information about how disk space and memory are being used, and how disk and memory space are allocated for CacheOS and cache data.

Viewing Disk Use

The Disk Use tab shows the Content Accelerator's disk usage. The fields on the Disk use tab are explained below:

- Cache available
  This is the amount of free space that can be used for caching.
- Cache in use
  This is the amount of disk space used for caching.
- System objects
  This is the amount of disk space used for device system objects.
- Access log
  This is the amount of disk space used for the access log.

To view disk use

1. Select Statistics from the CacheOS home page.
2. Select the Resources applet.

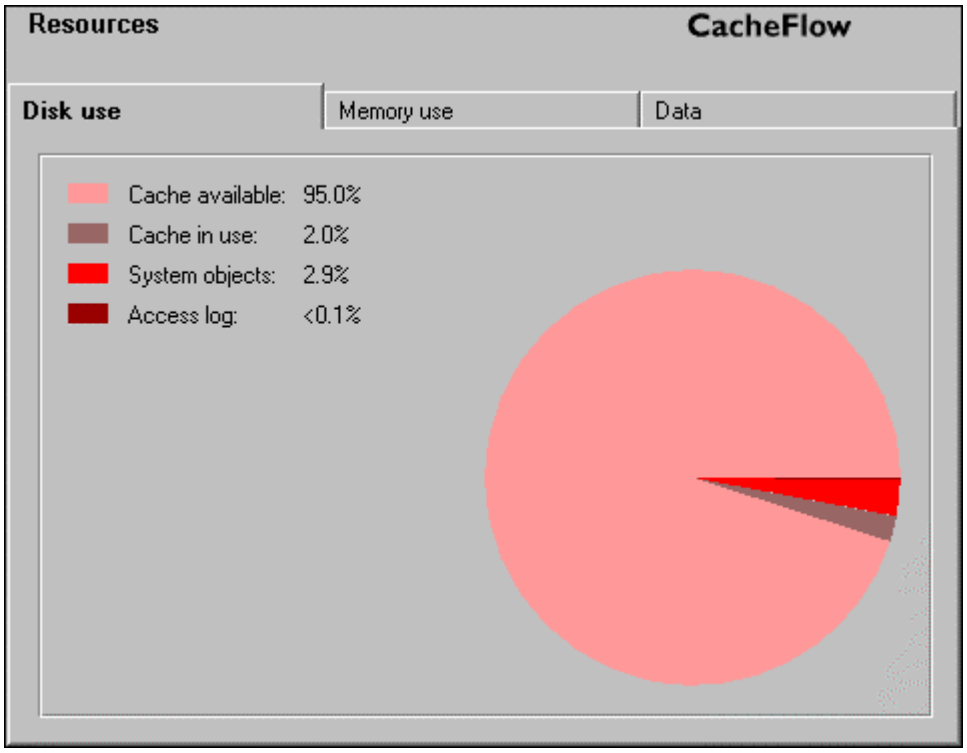


Figure 14-8 Displaying disk usage

Viewing Memory Use

The Memory Use tab shows the amount of memory used for RAM, the CacheOS itself, and for network buffers. The fields on the Memory use tab are explained below:

- RAM cache
  This is the amount of RAM that is used for caching.
- System allocation
  This is the amount of RAM allocated for the device system.
- Network buffers
  This is the amount of RAM currently allocated for network buffers.

To view memory use

1. Select Statistics from the CacheOS home page.
2. Select the Resources applet.
3. Select the Memory use tab.

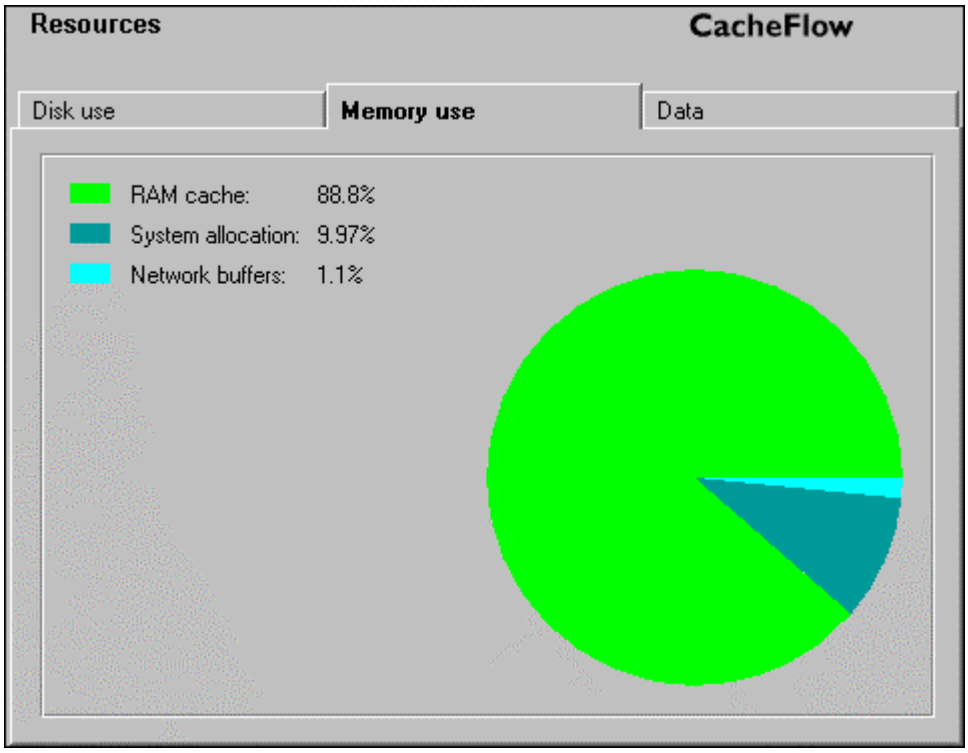


Figure 14-9 Displaying memory usage

Viewing Data Allocation in RAM and on Disk

The Data tab shows the total and available disk space and RAM, and how they are currently allocated. The fields on the Data tab are explained below:

- Disk available for cache
  This is the amount of free disk space that can be used for caching.
- Disk used by cache
  This is the amount of disk space used for caching.
- Disk used by system
  This is the amount of disk space used by the system objects.
- Disk used by access log
  This is the amount of disk space used for access logs.
- Total disk installed
  This is the total amount of disk space installed on the device.
- RAM used by cache
  This is the amount of RAM allocated for caching.
- RAM used by system

This is the amount of RAM allocated for system use.

- RAM used by network
  This is the amount of RAM allocated for network use.
- Total RAM installed
  This is total amount of RAM installed.

To view data allocation

1. Select Statistics from the CacheOS home page.
2. Select the Resources applet.
3. Select the Data tab.

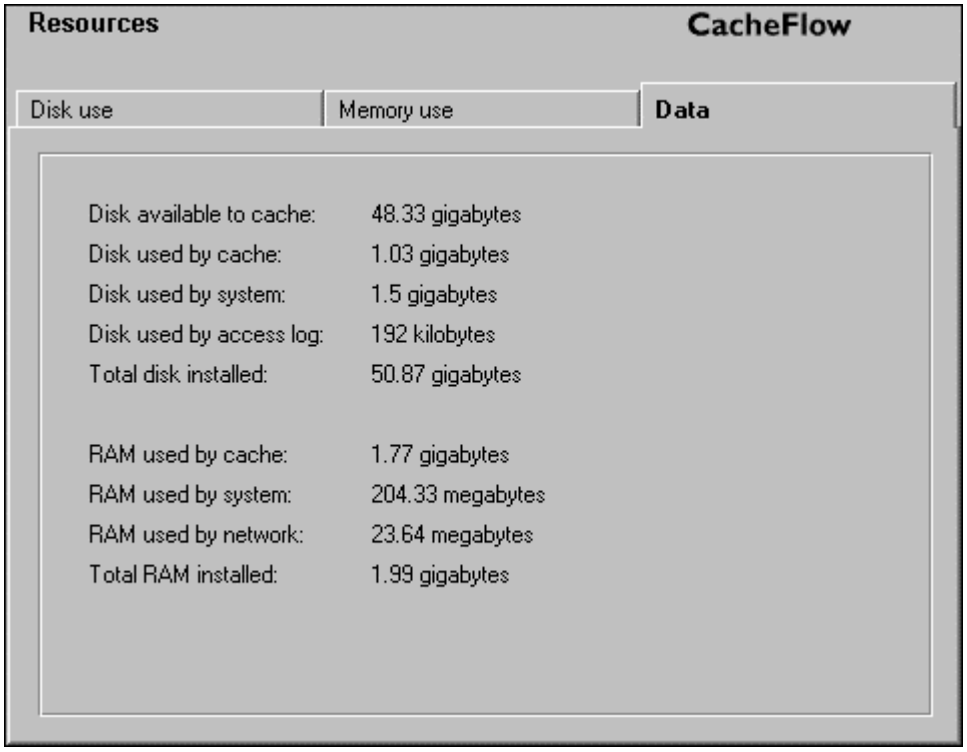


Figure 14-10 Displaying data allocation

Viewing Cache Efficiency

The Efficiency group of applets allow you to view information about the flow of both cacheable and non-cacheable data through the Content Accelerator. You can also view information about how data is being served (i.e., RAM, disk, origin).

Viewing the Cache Efficiency Summary

The Summary tab shows the percent of objects served from cache, the percent loaded from the network, and the percent that were non-cacheable. The data is since the last device reset. The values shown are either objects served or bytes served, based on the Values reflect field at the bottom of the tab. The fields on the Summary tab are explained below:

- Served from cache

  The percentage of requests the device was able to serve from the cache.
- In cache, verified fresh

  The percentage of requests the device was able to serve from the cache after verifying the object was still fresh.
- Loaded from source

  The percentage of requests the device had to retrieve from the Web, and was able to store in the cache.
- Non-cacheable

  The percentage of requests that were for non-cacheable objects. The Non-cacheable tab contains a breakdown of non-cacheable object types.

To view the cache efficiency summary

1. Select Statistics from the CacheOS home page.
2. Select the Efficiency applet.

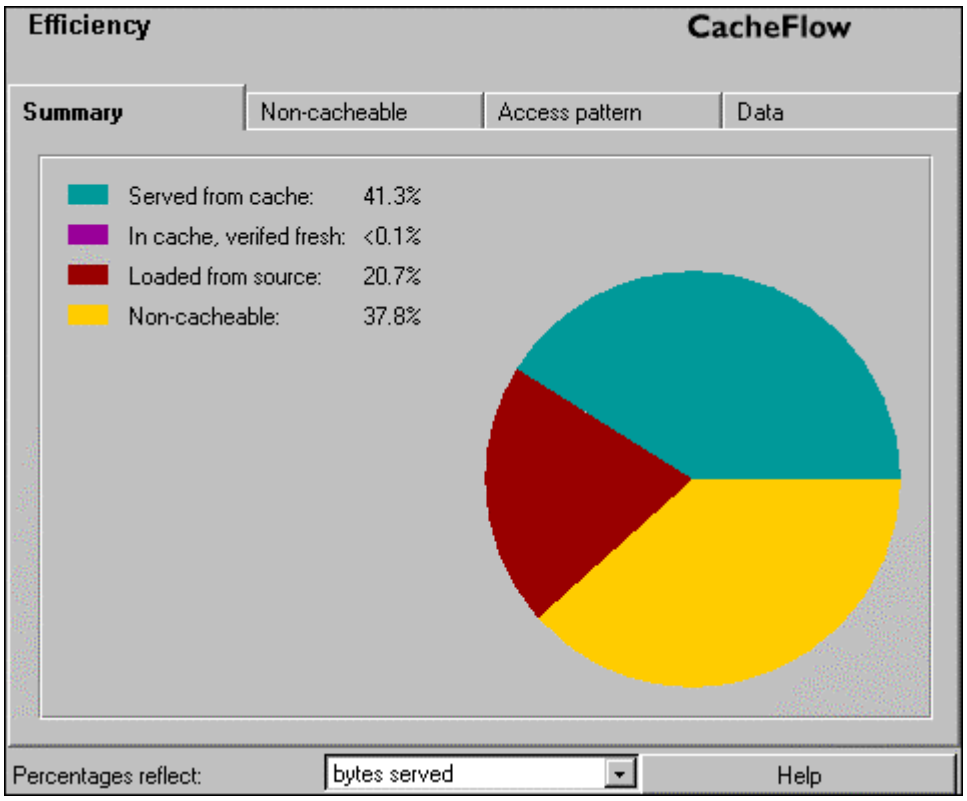


Figure 14-11 Displaying the cache efficiency summary

Viewing a Breakdown of Non-Cacheable Data

The Non-cacheable tab shows a breakdown of non-cacheable objects. It shows how many of the various types of non-cacheable requests have been handled. Each non-cacheable request type is described below:

- Pragma no-cache

  These are requests that specify non-cached objects, such as when a user clicks the refresh button in the Web browser.

- Password provided

  These are requests that include a client password.

- Data in request

  These are requests that include additional client data.

- Not a GET request

  Only the HTTP method Get request can be cached. These are all other methods (PUT, HEAD, POST, DELETE, LINK, and UNLINK).

- Cookie in response

  These are responses that include an HTTP cookie.

- Password required

  These are responses that require a client password.
- Negative response

  These are failed responses, such as when a server or object is not available. This value will be zero if the Cache Negative Responses option is enabled.
- Client unique CGI responses

  These are unique responses generated by a CGI application for a specific client.

To view a breakdown of non-cacheable data

1. Select Statistics from the CacheOS home page.
2. Select the Efficiency applet.
3. Select the Non-cacheable tab.

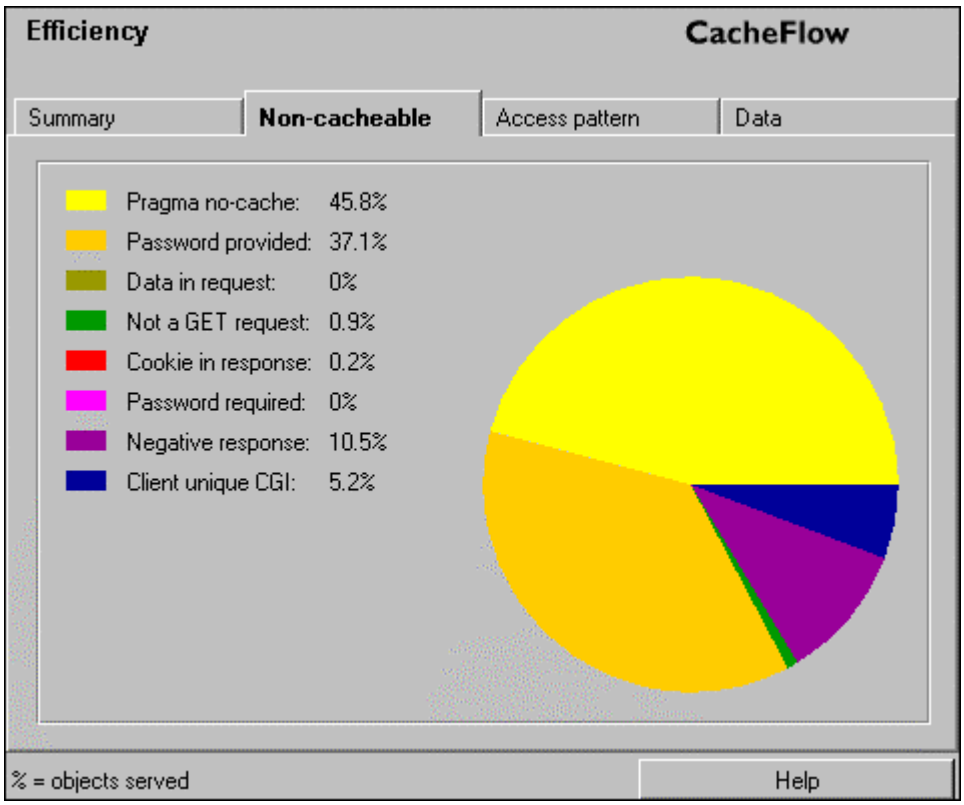


Figure 14-12 Displaying a breakdown of non-cacheable data

Viewing the Cache Data Access Pattern

The Access Pattern tab shows the number of cached requests served from RAM and disk. Cached objects are stored first in RAM. As time passes without additional requests for an object, the object is migrated to disk.

To view the cache data access pattern

1. Select Statistics from the CacheOS home page.
2. Select the Efficiency applet.
3. Select the Access pattern tab.

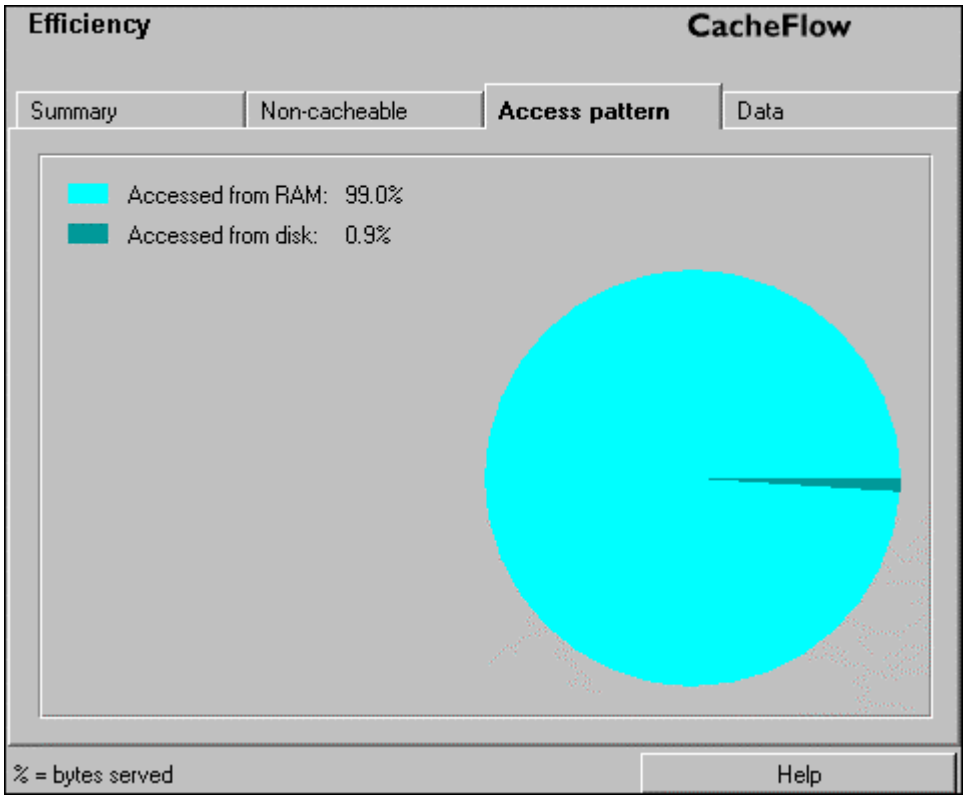


Figure 14-13 Displaying the cache data access pattern

Viewing Totals for Bytes Served

The Data tab lists a breakdown of all requests served. Each field is described below:

- Retrieved from server
  The number of objects that could not be served from the cache, and were retrieved from the Web.
- Retrieved from cache
  The number of objects served from the cache.
- Retrieved from console
  The number of console objects served.
- Non-cacheable objects
  The number of objects served that could not be cached.
- Pragma no-cache

These are requests that specify non-cached objects, such as when a user clicks the refresh button in a Web browser.

- Request authorize

  These are requests that include a client password.

- Request data

  These are requests that include additional client data.

- Non-cacheable method

  These are requests that include an invalid HTTP method.

- Response set cookie

  These are responses that include an HTTP cookie.

- Response authenticate

  These are responses that require a client password.

- Negative response

  These are failed responses, such as when a server or object is not available. This information will only be displayed if the Cache Negative Responses option is disabled.

- Unique CGI response

  These are responses that contain unique CGI data.

- Accessed from RAM

  The total number of bytes served from the RAM cache.

- Accessed from disk

  The total number of bytes served from the disk cache.

To view totals for bytes served

1. Select Statistics from the CacheOS home page.
2. Select the Efficiency applet.
3. Select the Data tab.

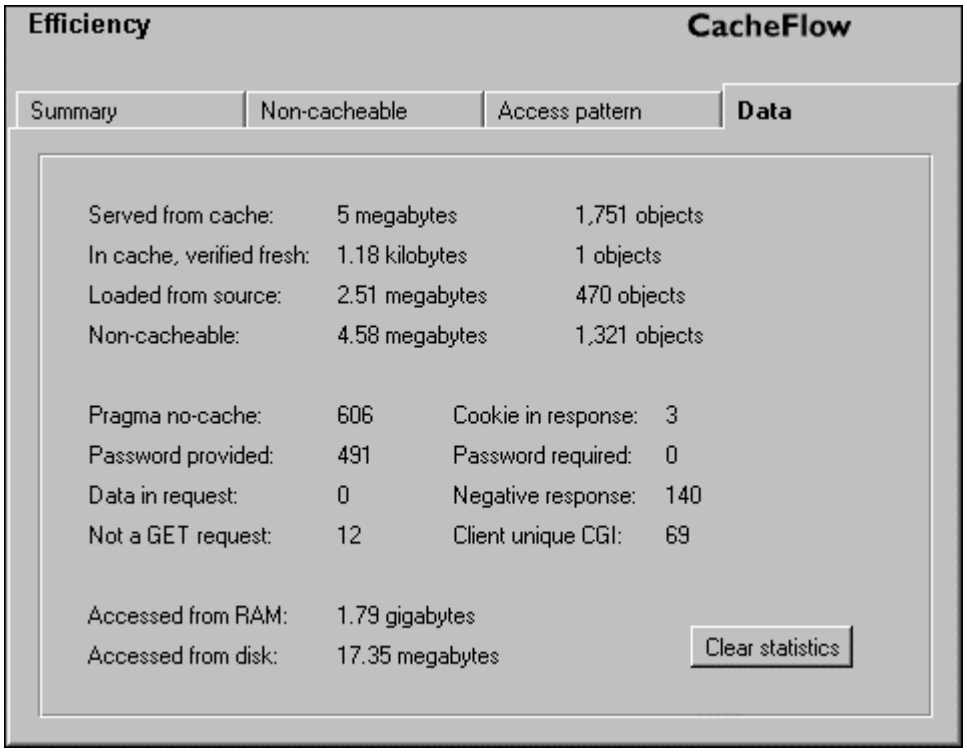


Figure 14-14 Viewing totals for bytes served

Viewing Cache Object Distribution by Size

The Content applets allow you to view information about objects currently stored or served organized by size. The cache contents include all objects currently stored by CacheOS. The cache contents are not cleared when the Content Accelerator is powered off.

Viewing Cached Objects by Size

The Distribution tab shows the objects currently stored by the Content Accelerator, ordered by size.

To view the distribution of cache contents

1. Select Statistics from the CacheOS home page.
2. Select the Contents applet.

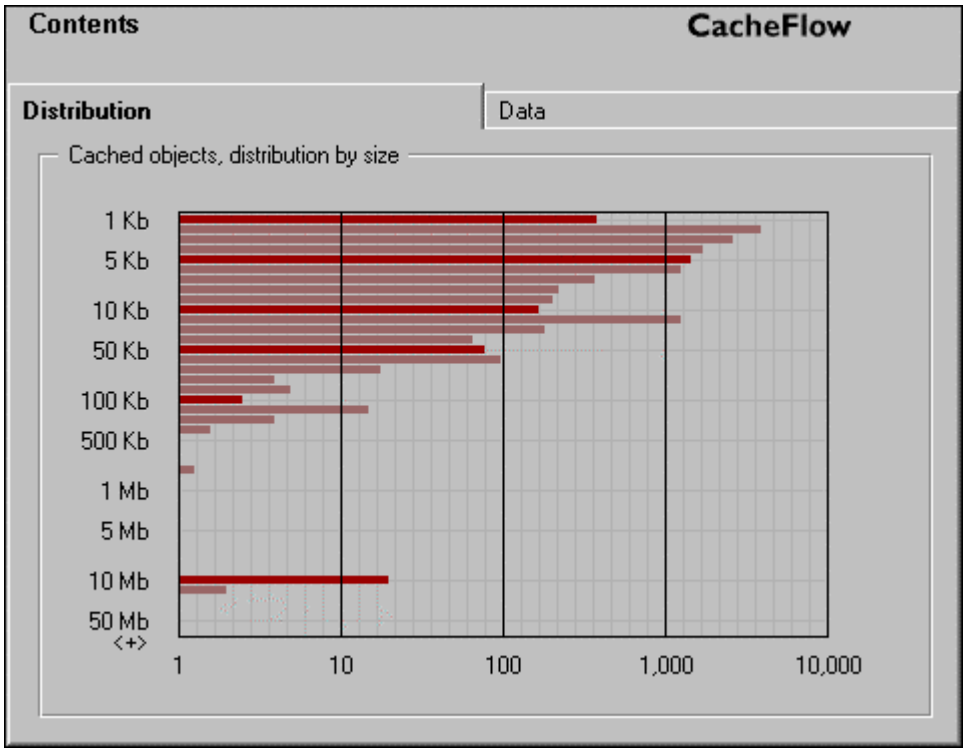


Figure 14-15 Viewing cache contents, organized by size

Viewing the Number of Objects Served by Size

The Data tab displays the number of objects served by the Content Accelerator, organized by size. This chart allows you to see how many objects of various sizes have been served.

To view the number of objects served

1. Select Statistics from the CacheOS home page.
2. Select the Contents applet.
3. Select the Data tab.

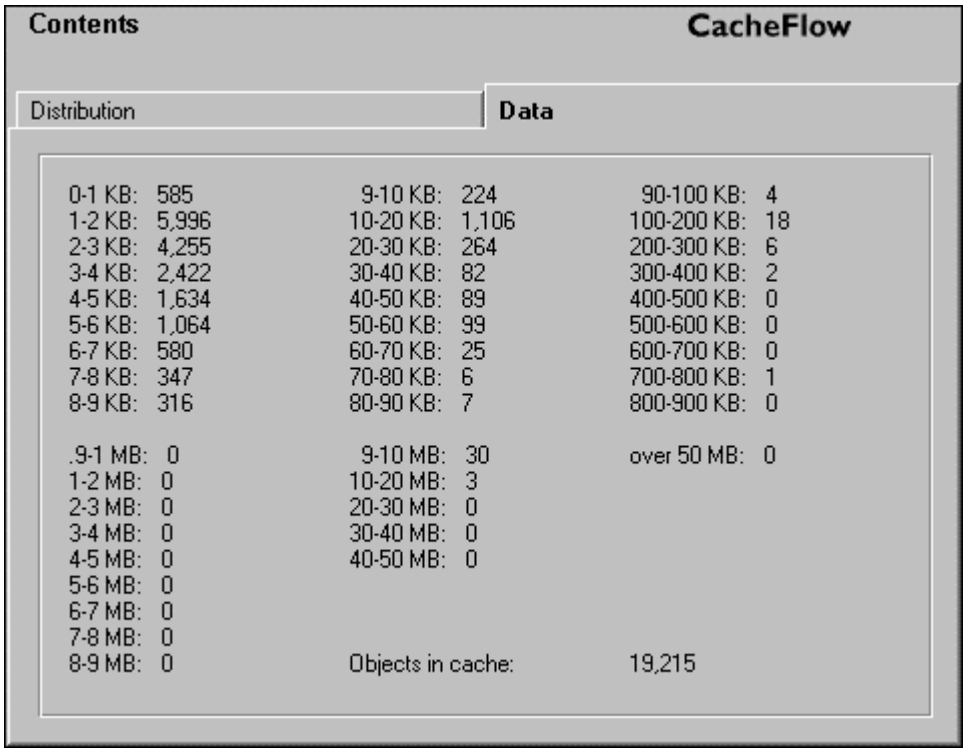


Figure 14-16 Viewing objects served, organized by size

Viewing the Event Log

The event log contains all events that have occurred on the Content Accelerator. The level of detail available in the event log event log is configured by selecting Events on the Management page.

Moving Through the Event Log

You can move to the top or bottom of the event log by clicking Log start or Log end. You can move forward or back one page at a time by clicking the forward arrow and back arrow.

Polling for New Events

The event log can poll for new events while the log is displayed. To enable polling, enable the Poll for new events checkbox. To disable polling, clear the Poll for new events checkbox. To save your polling preference, click Apply.

To display the event log

1. Select Statistics from the CacheOS home page.
2. Select the Event Viewer applet.
3. Click the forward arrow and back arrow buttons to move through the event list.

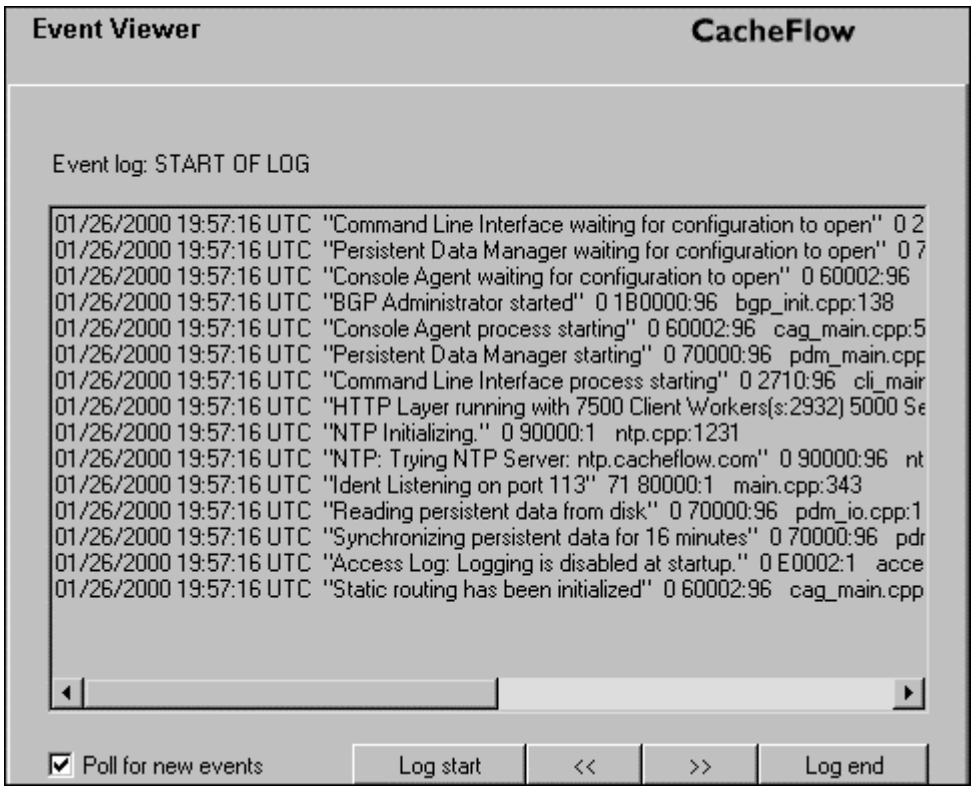


Figure 14-17 Displaying the Event Log

This page intentionally blank.

Appendix A - Access Log Formats

CacheOS and CacheOS/s can create access logs in one of three formats: NCSA common log format (default), Squid-compatible format, or custom format. When using the Squid or NCSA log format, a blank field is represented according to the standard of the format. When using a custom format, a blank field is represented by a dash character.

Common Access Log Format

The common log format contains one line for each request. The format of each log entry is shown below:

```
remotehost rfc931 authuser [date] "request" status bytes
```

Each field is described below:

Field Name	Description
remotehost	DNS hostname or IP address of remote server.
rfc931	The remote log name of the user.
authuser	The username as which the user has authenticated himself.
[date]	Date and time of the request.
"request"	The request line exactly as it came from the client.
Status	The HTTP status code returned to the client.
bytes	The content length of the document transferred.

Squid-Compatible Log Format

The Squid-compatible log format contains one line for each request. The format of each log entry is shown below:

```
timestamp  elapsed  src-address  type/code  size  method  URL
```

Each field is described below:

Field Name	Description
timestamp	The time the request is completed, with millisecond resolution.
elapsed	Elapsed time of the request, in milliseconds.
src-address	IP address of the requesting client.
Type	An indication of how the request was handled by the cache. These are described further below.
Code	The HTTP reply code when available. For ICP requests this is always "000". If the reply code was not given, it

Field Name	Description
	will be logged as “555”.
Size	For TCP requests, the amount of data written to the client. For UDP requests, the size of the request (in bytes).
method	The request method (GET, POST, etc). For ICP queries, the method is set to ICP_QUERY
URL	The URL of the request. For TCP misses that are handled by ICP, the URL includes the ICP source. The ICP source can be PARENT_HIT, SIBLING_HIT, FIRST_PARENT_MISS, DIRECT (indicating an ICP timeout or no target found), or NONE (indicating ICP was not used for a cache hit). The source is followed by the host name or IP address from which the object was retrieved, and the object MIME type.

Log Entry Types

The type field values are described below:

Value	Description
TCP_	Refers to requests on the HTTP port.
TCP_CLIENT_REFRESH	The client forces a revalidation with the origin server with a "Pragma: no-cache". If the server returns "304 Not Modified," this will show up in the Statistics:Efficiency file as, "In Cache, verified Fresh".
TCP_DENIED	Access to the requested object was denied by a filter. Includes Content-Filtering Service and CacheOS Filter File.
TCP_ERR_MISS	An error occurred while retrieving the object from the origin server.
TCP_EXPIRED	The object was in the cache, but it had expired.
TCP_HIT	A valid copy of the requested object was in the cache.
TCP_IFMODSINCE	An If-Modified-Since GET request.
TCP_MEM_HIT	The requested object was, in its entirety, in RAM.
TCP_MISS	The requested object was not in the cache.
TCP_NC_MISS	Object returned from origin server was non-cacheable.
TCP_PARTIAL_MISS	Object is in cache, but retrieval from origin server is in progress.
TCP_REFRESH	The user forced a refresh (“reload”).

Value	Description
TCP_REFRESH_HIT	A GIMS request to the server was forced and the response was, "304 Not Modified". This will show up in the Statistics:Efficiency file as, "In Cache, verified Fresh".
TCP_REFRESH_MISS	A GIMS rquest to the server was forced and new content was returned.
TCP_SWAPFAIL	The object was believed to be in the cache, but could not be accessed.
TCP_TUNNELED	The CONNECT method was used to tunnel this request (generally proxied HTTPS).
UDP_	Refers to requests on the ICP port (3130).
UDP_DENIED	Access was denied for this request.
UDP_HIT	A valid copy of the requested object was in the cache. This value is also used with ICP queries.
UDP_INVALID	The ICP request was corrupt, short or otherwise unintelligible.
UDP_MISS	The requested object was not in the cache. This value is also used with ICP queries.
UDP_MISS_NOFETCH	An ICP request was made to this cache for an object not in cache. The requestor was informed that it could not use this cache as a parent to retrieve the object. (This is not supported at this time.)
UDP_OBJ	An ICP request was made to this cache for an object that was in cache, and the object was returned through UDP. (This is not supported at this time. This functionality is deprecated in the current ICP specification.)

Using a Custom Format

To define your own log format, choose the Custom format string option and enter the format string using the codes described below:

Format Character	Description
space character	Multiple consecutive spaces are compressed to a single space.
/	A '/'.
"	A quote character (").
%a	Client IP Address.
%b	Number of bytes returned by the server (or the

Format Character	Description
	Cache).
%c	The type of object. Usually the MIME-type.
%d	Name or IP address of the server/cache from which the object was retrieved. The log entry is blank for a cache hit. The address or resolved name of the server is logged for a cache miss.
%e	Number of milliseconds the request took to process.
%f	Specifies the Websense or SmartFilter reasons for why the request was not acted upon (obscene materials, sports, humor, etc.) A "-" appears if no reason is given.
%g	UNIX type timestamp (GMT).
%h	Client IP address.
%i	The requested URL.
%m	HTTP Method. HTTP Methods are GET, PUT, POST, etc.
%p	Port on the destination server.
%r	First line of the client request.
%s	The code returned by the server (HTTP Code).
%t	UTC time of the user request.
%v	Name of the destination server.
%w	What type of action did CacheOS take to process this request (hit, miss, etc.).
%A	The browser's user agent.
%C	Log cookie data from the client request.
%H	How and where the object was retrieved from the cache hierarchy (DIRECT from the server, PARENT_HIT = from the parent cache, etc.).
%L	Local time of the user request.
%T	Number of seconds the request took to process.
%U	Path component of the requested URL.
%W	WebSense content filter processing status.

Examples for common access log formats are shown below:

```
Squid log format: %g %e %a %w/%s %b %m %i %u %H/%d %c
NCSA common log format: %h %l %u %t "%r" %s %b
NCSA extended log format: %h %l %u %t "%r" %s %b "%R" "%A"
```

You can separate the format codes with a space or slash. Multiple spaces are compressed to a single space in the actual access log. You can also enter a string such as "My default is %d". CacheOS goes through such strings and finds the relevant information. In this case, that information is %d.

This page intentionally blank.

Appendix B - Using WCCP

The CacheFlow device can be configured to participate in a WCCP (Web Cache Control Protocol) scheme, where a WCCP-capable router collaborates with a set of WCCP-configured CacheFlow devices to service requests. WCCP is a Cisco-developed protocol. For more information about WCCP, refer to the Cisco website.

The CacheFlow device can support WCCP protocol versions 1 and 2. Only one protocol version can be active on the CacheFlow device at a time. The active WCCP protocol must be matched by the version running on the WCCP router.

WCCP version 1 offers a subset of the functionality offered by version 2. In version 1, a single WCCP router, referred to as the *Home Router*, transparently redirects only TCP port 80 packets (common HTTP traffic) to a maximum of 32 CacheFlow devices. One of the caches participating in the WCCP protocol is automatically elected to configure the Home Router's redirection tables. As such, caches can be transparently added and removed from the WCCP group, without requiring operator intervention as shown in the following figure.

The WCCP version 2 protocol offers the same capabilities as version 1, along with protocol security and multicast protocol broadcasts. In addition, up to 32 WCCP-capable routers can transparently redirect traffic to a set of up to 32 CacheFlow devices. Whereas version 1 was only capable of redirecting TCP port 80 traffic, version 2 WCCP-capable routers can be configured to redirect IP traffic to a set of CacheFlow devices based on various fields within those packets.

This redirection policy and its administrative details comprise a *Service Group*. WCCP version 1 supports only a single Service Group. Version 2 allows routers and caches to participate in multiple simultaneous Service Groups. Thus, routers can transparently redirect IP packets based on their formats. For example, one Service Group could redirect HTTP traffic and another could redirect FTP traffic.

Using WCCP and Transparent Redirection

A WCCP-capable router operates in conjunction with CacheFlow devices to transparently redirect traffic to a set of caches which participate in the specified WCCP protocol. IP packets are redirected based on fields within each packet. For instance, WCCP version 1 only redirects destination TCP port 80 (default HTTP traffic) IP packets. The destination IP address is hashed to yield one of 256 buckets within a redirection hash table to determine which cache will be the recipient of the redirected packet. This hash table is configured by a dynamically elected cache participating in the *Service Group*.

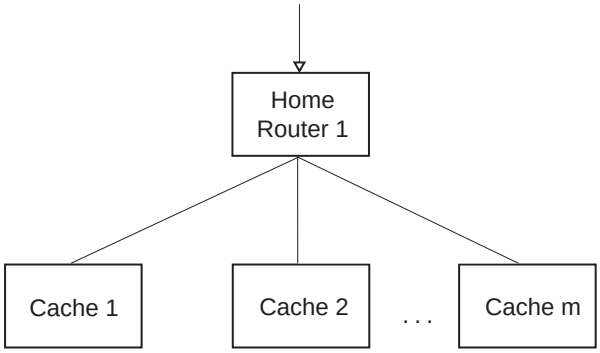
In version 2, each service group can be configured to use a security password. Both the routers and caches participating in the *Service Group* use this security password to verify the authenticity of WCCP protocol traffic. Protocol packets that fail the authenticity check are ignored.

Note that it is not recommended that WCCP compliant caches from different vendors participate in the same *Service Group*.

WCCP Version 1

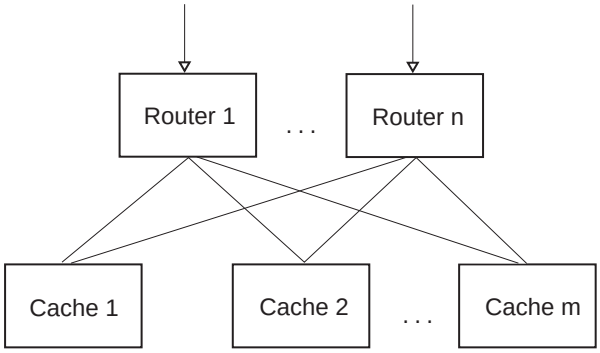
The following figure illustrates a typical WCCP implementation. Each applicable client IP packet received by the *Home Router* is transparently redirected to a cache. A cache from the group is selected to define the Home Router's

redirection hash table for all caches. All caches periodically communicate with the Home Router to verify WCCP protocol synchronization and cache availability within the *Service Group*. In return, the Home Router responds to each cache with information as to what caches are available in the Service Group.

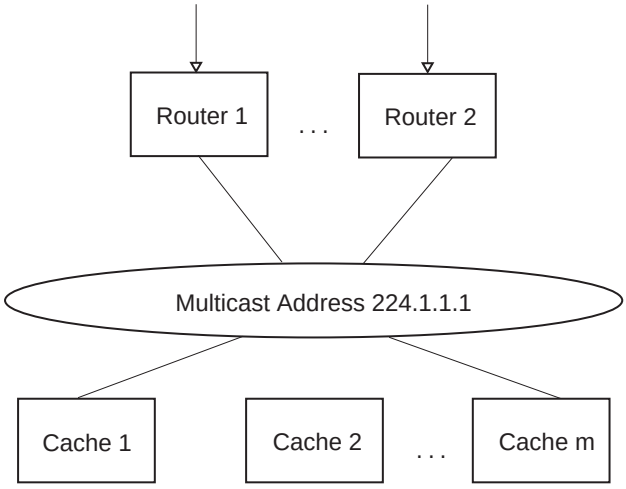


WCCP Version 2

The next figure illustrates a WCCP version 2 implementation using multiple routers and caches. In this scenario, routers 1 through N and caches 1 through M participate in the same Service Group. As in version 1, a cache from the group is selected to define the redirection hash table in all routers for all caches. All caches periodically communicate with all routers to verify WCCP protocol synchronization and cache and router availability within the Service Group. In return, each router responds to caches with information as to what caches and discovered routers are available in the *Service Group.*



WCCP communication between the routers and the caches can be performed by either directly addressing protocol packets to each router's and cache's IP address (as illustrated in the preceding figure) or by addressing these packets to a common multicast address as illustrated by the following figure:

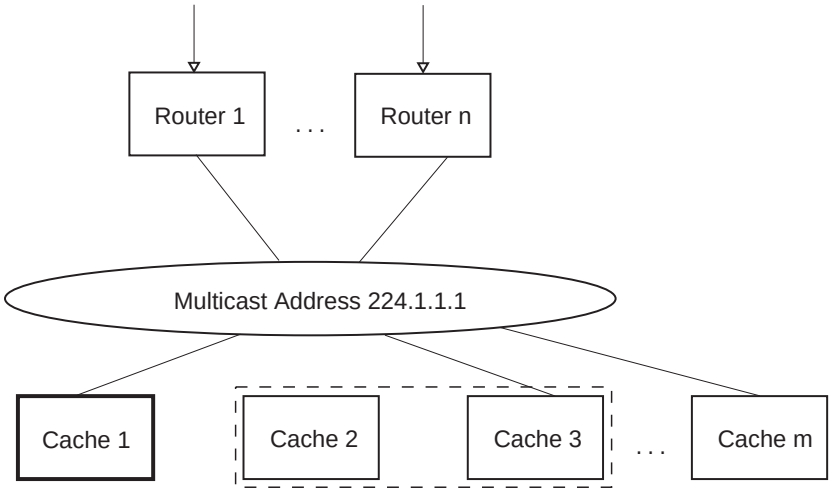


Some of the benefits of using a multicast address include reduced WCCP traffic and the ability to easily add and remove caches and routers from a Service Group without having to reconfigure all Service Group members. Multicast addresses fall within the range 224.0.0.0 to 239.255.255.255.

Multiple Network Cards within a CacheFlow Device

Multiple network cards within a CacheFlow device can participate in the same Service Group. To the routers and other caches, each interface appears as a unique cache. Thus, redirected traffic can be better distributed among network interfaces in a cache.

In the following illustration, Cache 2 and Cache 3 are physically located within the same CacheFlow device using two different network interface cards. Each of these caches will be assigned a unique portion of the redirection hash table and can act as an autonomous unit.



Service Group Security

A password can be applied to a WCCP version 2 *Service Group*. This password must match the configured password within each *Home Router*. Note that it is suggested that passwords be exactly 8 characters long.

Distribution of the Redirection Hash Table

As mentioned earlier, a cache within the *Service Group* defines the redirection hash table that it assigns to routers. Each element in this 256-entry table refers to an active cache within the Service Group. When a router receives an IP packet for redirection, it hashes fields within the packet to yield an index within the hash table. Finally, the packet is forwarded to the 'owner' cache for servicing. The proportion of redirection hash table assigned to each cache can be altered to provide a form of load balancing between caches in a *Service Group*.

By default, each cache is assigned roughly an even percentage of the 256-element redirection hash table. Using figure 4 above, all caches would be assigned 1/M of the redirection hash table but since **Cache 2** and **Cache 3** are physically located within the same CacheFlow device, the CacheFlow device would actually be assigned 2/M of the redirection hash table.

In WCCP version 2, the relative distribution of the redirection hash table can be specified for each cache. Each cache can be assigned a **primary-hash-weight** value (see the "Configuration File Syntax" section below) to determine the proportion of the 256 element hash table to be assigned to a cache. If all caches are configured with a 0 **primary-hash-weight** value (i.e. the default) then each cache will be assigned the same proportion of the redirection hash table. However, if any cache defines a non-zero **primary-hash-weight** then each cache will be assigned a relative proportion of the table. For instance, consider a configuration with five caches whose **primary-hash-weight** is defined as {25, 200, 0, 50, 25}. The total requested weight value is 25+200+0+50+25=300 and, thus, the proportion of the hash table assigned to each cache will be 25/300, 200/300, 0/300, 50/300, and 25/300. Note that since the third cache did not specify a non-zero **primary-hash-weight** it will not be assigned any elements within the redirection hash table and, therefore, will not receive any redirected traffic. Also note that the hash weight can be specified for each caching member within a CacheFlow device. In figure 4 above, **Cache 2** and **Cache 3** could be assigned different weight values.

Alternate Hash Table

In some cases, a web site becomes an Internet 'hot-spot' because it receives a disproportional number of client traffic relative to other sites. This situation can cause a larger request load on a cache relative to its peers within the *Service Group* since the hash element associated with the popular site receives more activity then other hash elements. To balance the redirection traffic load among the caches, a *Service Group* can be configured to use an alternate hash function. A hash element that is identified as a 'hot-spot' within the *Service Group* is reconfigured to use an alternate hashing function for computing a new hash table index. The new hashing function can be derived from other components of the IP packet to redirect. Thus, when a router receives an IP packet that hashes to an element flagged as being a 'hot-spot', the alternate hash function is computed. The cache as specified by the new index in the redirection hash table will receive the redirected packet.

Each CacheFlow device can dynamically determine a 'hot-spot' within its assigned portion of the redirection hash table. Hot-spots are identified as hash elements receiving an excessive amount of traffic over a period of time.

Alternate hash tables are only used for dynamic *Service Groups* that specify **alternate-hash** flags within their **service-flags** (refer to the **service-flags** commands in the "Configuration File Syntax" section below). Note that the default "**web-cache**" *Service Group* can not use an alternate hash table. Instead, a comparable dynamic *Service Group* must be created.

Configuration File Syntax

The settings configuration file uses three namespaces. The first namespace allows general WCCP commands to be configured. The second and third namespaces allow for specific configuration of *Service Groups*. By default, the Main Namespace is active. Focus can change to the *Service Group* Namespaces by executing a "**`service-group`**

[**web-cache** | ***number***]" command. Finally, focus can return to the Main Namespace by executing the "**end**" command. The *Service Group* namespace selected is dependent upon the specified WCCP version. As such, if WCCP version 1 is selected then the version 1 specific *Service Group* Namespace will be used. If no "**wccp version** [**1** | **2**]" is explicitly specified in the configuration file before a *Service Group* is defined then WCCP version 2 is selected.

Main Namespace

wccp [enable | disable]

This command will enable or disable WCCP. By default, WCCP protocol communication is disabled.

wccp version [1 | 2]

This command specifies that the following Service Group definitions refers to version 1 or 2 of WCCP. By default, WCCP version 2 Service Groups are created unless this command is specified. This command, which can appear at most once within the configuration file, should appear before any Service Groups are defined.

service-group [web-cache | number]

This command introduces the definition for a Service Group. The web-cache identifier refers to the standard HTTP redirection Service Group. If a number is specified, then a dynamic Service Group is being specified. Once this command is accepted, either the version 1 or version 2 Service Group command namespace will become active.

no service-group [web-cache | number]

This command will destroy a previously defined Service Group.

Version 1 Service Group Namespace

home-router address

This command allows a home router to be specified for a Service Group. The address must be a dotted decimal value and must not be a multicast address (within the range 224.0.0.0 to 239.255.255.255).

home-router domain-name

This command allows a home router to be specified for a Service Group. A DNS lookup is performed on the domain-name. If the lookup fails then an error will be reported. The domain-name must be a valid domain name string.

interface interfacenumber

This command specifies the network interface number to be associated with the Service Group. Multiple network interfaces within a CacheFlow device might participate within the same Service Group.

no interface interfacenumber

This command removes a network card interface from a Service Group.

end

This command returns focus back to the Main Namespace.

Version 2 Service Group Namespace

priority number

This command will set the priority value for the Service Group. The acceptable range is 0 to 255. Note that this command would be used for a dynamic Service Group (one specified as "service-group number").

protocol number

This command will set the protocol value for the Service Group. The acceptable range is 0 to 255. Note that this command would be used for a dynamic Service Group (one specified as "service-group number").

service-flags source-ip-hash

service-flags destination-ip-hash

service-flags source-port-hash

service-flags destination-port-hash

service-flags ports-defined

service-flags ports-source

service-flags source-ip-alternate-hash

service-flags destination-ip-alternate-hash

service-flags source-port-alternate-hash

service-flags destination-port-alternate-hash

These commands set the appropriate bit definitions within the service flags for Service Group.

Note that these commands would be used for a dynamic Service Group (one specified as "service-group number").

no service-flags source-ip-hash

no service-flags destination-ip-hash

no service-flags source-port-hash

no service-flags destination-port-hash

no service-flags ports-defined

no service-flags ports-source

no service-flags source-ip-alternate-hash

no service-flags destination-ip-alternate-hash

no service-flags source-port-alternate-hash

no service-flags destination-port-alternate-hash

These commands reset the appropriate bit definitions within the service flags for Service Group.

Note that these commands would be used for a dynamic Service Group (i.e. one specified as "service-group number").

ports *number number number number number number number number*

This command will set the port values for the *Service Group*. Each ***number*** is a 16 bit decimal value. Note that this command would be used for a dynamic *Service Group* (i.e. one specified as "**service-group *number***").

home-router *address*

This command allows multiple home routers to be specified for a *Service Group*. The ***address*** must be a dotted decimal value. For WCCP version 2, either a single multicast address (i.e. within the range 224.0.0.0 to 239.255.255.255) or up to 32 router IP addresses can be specified.

`home-router *domain-name*`

This command allows multiple home routers to be specified for a *Service Group*. A DNS lookup is performed on the ***`domain-name`***. If the lookup fails then an error will be reported. The ***`domain-name`*** must be a valid domain name string.

`interface *interfacenumber*`

This command specifies the network interface number to be associated with the *Service Group*. Multiple network interfaces within a CacheFlow device might participate within the same *Service Group*.

`no interface *interfacenumber*`

This command removes a network card interface from a *Service Group*.

`password *string*`

This command applies a password to a *Service Group*. It is suggested that the password ***string*** be exactly 8 characters long.

`no password`

This command removes the password used by a *Service Group*.

`primary-hash-weight *interfacenumber value*`

This command associates a weight factor of ***value*** for network interface ***interfacenumber*** within a *Service Group*. This weighting value is used in version 2 to alter the distribution of the primary hash table.

`end`

This command returns focus back to the Main Namespace.

Examples

Version 1 Standard HTTP Redirection

Configuring WCCP version 1 on the Router

The following example enables WCCP version 1 on a Cisco router. It is assumed that the router's Ethernet interface 0/0 will be used for redirecting traffic to cache members in the *Service Group*.

```
Router# configure terminal
Enter configuration commands, one per line. End with CTRL/Z.
Router(config)# ip wccp enable
Router(config)# interface ethernet 0/0
Router(config-if)# ip web-cache redirect
```

This configuration simply enables WCCP and assigns redirected traffic to sent out Ethernet interface 0/0.

Configuring the CacheFlow Device

To enable the WCCP version 1 *Service Group* within the CacheFlow device, the following configuration file could be loaded.

```
# Enable WCCP to allow WCCP protocol communication between the cache # and the Home
Router.
  wccp enable
# A WCCP version 1 Service Group will be configured. Note that the following line must
be appear before the Service Group definition.
wccp version 1
service-group web-cache
# specify the address for the router
  home-router 90.0.0.90
# network interface 0 will participate
  interface 0
  end
```

Version 2 Standard HTTP Redirection

Configuring WCCP version 2 on the Router

The following example will enable the standard HTTP traffic redirection on a WCCP version 2 capable Cisco router.

```
Router# configure terminal
Enter configuration commands, one per line. End with CNTR/Z.
Router(config)# ip wccp web-cache
Router(config)# interface ethernet 0/0
Router(config-if)# ip wccp web-cache redirect out
```

Configuring the CacheFlow Device

To enable the standard WCCP version 2 *Service Group* within the CacheFlow device, the following configuration file could be loaded.

```
# Enable WCCP to allow WCCP protocol communication between the cache and the Home
Router.
  wccp enable
# By default, the WCCP version 2 protocol is assumed. An explicit "wccp version 2"
command could be specified here.
  service-group web-cache
# specify the address for the router
  home-router 90.0.0.90
# network interface 0 will participate
  interface 0
  end
```

Version 2 Standard HTTP Redirection Using a Multicast Address

Configuring WCCP version 2 on the Router

The following example will enable the standard HTTP traffic redirection on a WCCP Version 2.0-capable Cisco router. In this case, WCCP protocol traffic will be directed to the multicast address 224.1.1.1.

```
Router# configure terminal
Enter configuration commands, one per line. End with CNTR/Z.
Router(config)# ip wccp web-cache group-address 224.1.1.1
Router(config)# interface ethernet 0/0
Router(config-if)# ip wccp web-cache group-listen
Router(config-if)# ip wccp web-cache redirect out
```

Configuring the CacheFlow Device

To enable the standard WCCP version 2 *Service Group* within the CacheFlow device, the following configuration file could be loaded. Note that in this example, both network interfaces 0 and 1 will participate within the *Service Group*. Both interfaces will send and receive WCCP protocol packets by way of the multicast address.

```
# Enable WCCP to allow WCCP protocol communication between the cache and the Home
Router.
  wccp enable
# By default, the WCCP version 2 protocol is assumed. An explicit "wccp version 2"
command could be specified here.
  service-group web-cache
# specify the multicast address
  home-router 224.1.1.1
# network interface 0 will participate
  interface 0
# network interface 1 will also participate
  interface 1
  end
```

Version 2 Standard HTTP Redirection Using a Security Password

Configuring WCCP version 2 on the Router

The following example will enable standard HTTP traffic redirection on a WCCP Version 2.0-capable Cisco router. A simple eight-character password is configured within the router. This password must match the password configured within the CacheFlow device.

```
Router# configure terminal
```

```
Enter configuration commands, one per line. End with CNTR/Z.
Router(config)# ip wccp web-cache password guesswat
Router(config)# interface ethernet 0/0
Router(config-if)# ip wccp web-cache redirect out
```

Configuring the CacheFlow Device

To enable the standard WCCP version 2 *Service Group* within the CacheFlow device, the following configuration file could be loaded.

```
# Enable WCCP to allow WCCP protocol communication between the cache and the Home
Router.
  wccp enable
# By default, the WCCP version 2 protocol is assumed. An explicit  "wccp version 2"
command could be specified here.
  service-group web-cache
# specify the address for the router
  home-router 90.0.0.90
# network interface 0 will participate
  interface 0
  password guesswat
  end
```

Version 2 Reverse Proxy Service Group

Configuring WCCP version 2 on the Router

The following example will enable the special reverse proxy Service Group on a WCCP Version 2.0-capable Cisco router. This *Service Group* redirects IP packets for TCP destination port 80 traffic by hashing the source IP address.

```
Router# configure terminal
Enter configuration commands, one per line. End with CNTR/Z.
Router(config)# ip wccp 99
Router(config)# interface ethernet 0/0
Router(config-if)# ip wccp 99 redirect out
```

Configuring the CacheFlow Device

To configure the special reverse proxy *Service Group* on the CacheFlow device, a dynamic *Service Group* must be created as illustrated by the following example.

```
# Enable WCCP to allow WCCP protocol communication between the cache and the Home
Router.
  wccp enable
# By default, the WCCP version 2 protocol is assumed. An explicit "wccp version 2"
command could be specified here.
# Service Group 99 is specially identified within the router as representing the
reverse proxy service.
```

```
  service-group 99
# specify the address for the router
  home-router 90.0.0.90
# network interface 0 will participate
  interface 0
# TCP protocol
  protocol 6
# hash based on source IP address
  service-flags source-ip-hash
  end
```

Version 2 Service Group with Alternate Hashing

Configuring WCCP version 2 on the Router

The following example will enable a special Service Group on a WCCP version 2 capable Cisco router that uses alternate hashing when hot-spots are detected. This *Service Group* redirects IP packets by hashing the source IP address.

```
Router# configure terminal
Enter configuration commands, one per line. End with CNTR/Z.
Router(config)# ip wccp 5
Router(config)# interface ethernet 0/0
Router(config-if)# ip wccp 5 redirect out
```

Configuring the CacheFlow Device

To configure this special *Service Group* on the CacheFlow device, a dynamic *Service Group* must be created as illustrated by the following example.

```
# Enable WCCP to allow WCCP protocol communication between the cache and the Home
Router.
  wccp enable
# By default, the WCCP version 2 protocol is assumed. An explicit "wccp version 2"
command could be specified here.
# Service Group 5 will be created to redirect standard HTTP traffic and use an
alternate hash function based on the source IP address if necessary.
  service-group 5
# specify the address for router 1
  home-router 90.0.0.90
# specify the address for router 2
  home-router 90.0.1.5
# network interface 0 will participate
  interface 0
```

```
# TCP protocol
  protocol 6
# The following two flags specify that a hash function based on the destination IP
address should be applied first. If a hot-spot is detected then an alternate hash
function using the source IP address should be used.
  service-flags destination-ip-hash
  service-flags source-ip-alternate-hash
  end
```

Appendix C - Using Regular Expressions

Regular expressions can be used for complex pattern matching. CacheOS supports regular expressions for URL matching with advanced forwarding and filters. The regular expression support in CacheOS described in this appendix is based on the Perl-compatible regular expression libraries (PCRE) by Philip Hazel. The text of this appendix is based on the PCRE documentation.

A regular expression (or RE) is a pattern that is matched against a subject string from left to right. Most characters stand for themselves in a pattern, and match the corresponding characters in the subject. The power of regular expressions comes from the ability to include alternatives and repetitions in the pattern. These are encoded in the pattern by the use of meta-characters, which do not stand for themselves, but instead are interpreted in some special way. For details of the theory and implementation of regular expressions, consult Jeffrey Friedl's "Mastering Regular Expressions", published by O'Reilly (ISBN 1-56592-257-3).

CacheOS uses a Regular Expression Engine (RE ENGINE) to evaluate regular expressions. In CacheOS, regular expressions can be used for filtering URLs which can match any portion of a URL. More specifically, a URL is considered to be of the form:

```
protocol://domain[:port]/relpath
```

In a CacheOS filter file, a line is considered to be a regular expression if it contains one or more regular expression metacharacters from the following set:

```
\  ^  $  [  |  (  ?  *  +  {
```

Portions of the regular expression which match against the protocol and the domain are converted to a canonical form so that matches are performed in a case-insensitive manner. Regular expressions used for filtering URLs can appear anywhere within a filter file; however, the order in which they appear is significant since the first regular expression matched is the one whose associated filter properties are used.

Similarly, in CacheOS Advanced Forwarding, the **icp_host_url_regex** command defines which requests are sent to which cache hosts or cache host groups based on a regular expression match of the requested object's URL.

Regular Expression Syntax

Regular expressions can contain both special and ordinary characters. Most ordinary characters, like 'A', 'a', or '3', are the simplest regular expressions; they simply match themselves. You can concatenate ordinary characters, so 'last' matches the characters 'last'. (In the rest of this section, we'll write REs `in` this special font, usually without quotes, and strings to be matched 'in single quotes'.)

Some characters, like | or (, are special. Special characters, called meta-characters, either stand for classes of ordinary characters, or affect how the regular expressions around them are interpreted. The meta-characters are shown below:

Metacharacters	Description
.	(Dot.) In the default mode, this matches any character except a newline

Metacharacters	Description
^	(Caret.) Matches the start of the string.
$	Matches the end of the string.
*	Causes the resulting RE to match 0 or more repetitions of the preceding RE, as many repetitions as are possible. ab* will match 'a', 'ab', or 'a' followed by any number of 'b's.
+	Causes the resulting RE to match 1 or more repetitions of the preceding RE. ab+ will match 'a' followed by any non-zero number of 'b's; it will not match just 'a'.
?	Causes the resulting RE to match 0 or 1 repetitions of the preceding RE. ab? will match either 'a' or 'ab'.
*?, +?, ??	The *, +, and ? qualifiers are all greedy; they match as much text as possible. Sometimes this behavior isn't desired. If the RE /page1/.*/ is matched against /page1/heading/images/, it will match the entire string, and not just /page1/heading/. Adding ? after the qualifier makes it perform the match in non-greedy or minimal fashion; as few characters as possible will be matched. Using .*? in the previous expression will match only /page1/heading/..
{m,n}	Causes the resulting RE to match from m to n repetitions of the preceding RE, attempting to match as many repetitions as possible. For example, a{3,5} will match from 3 to 5 'a' characters.
{m,n}?	Causes the resulting RE to match from m to n repetitions of the preceding RE, attempting to match as few repetitions as possible. This is the non-greedy version of the previous qualifier. For example, on the 6-character string 'aaaaaa', a{3,5} will match 5 'a' characters, while a{3,5}? will only match 3 characters.
\	Either escapes special characters (permitting you to match characters like '*?+&$'), or signals a special sequence; special sequences are discussed below.
[]	Used to indicate a set of characters. Characters can be listed individually, or a range of characters can be indicated by giving two characters and separating them by a '-'. Special characters are not active inside sets. For example, [akm$] will match any of the characters 'a', 'k', 'm', or '$'; [a-z] will match any lowercase letter and [a-zA-Z0-9] matches any letter or digit. Character classes such as \w or \S (defined below) are also acceptable inside a range. If you want to include a] or a - inside a set, precede it with a backslash. Characters not within a range can be matched by

Metacharacters	Description
	including a ^ as the first character of the set; ^ elsewhere will simply match the '^' character.
|	A|B, where A and B can be arbitrary REs, creates a regular expression that will match either A or B. This can be used inside groups (see below) as well. To match a literal '|', use |, or enclose it inside a character class, like [|].
(...)	Matches whatever regular expression is inside the parentheses, and indicates the start and end of a group; the contents of a group can be retrieved after a match has been performed, and can be matched later in the string with the \number special sequence, described below. To match the literals '(' or ')', use \(or \), or enclose them inside a character class: [(] [)].

Regular Expression Details

The syntax and semantics of the regular expressions supported by RE ENGINE are described below. Regular expressions are also described in the Perl documentation and in a number of other books, some of which have copious examples. Jeffrey Friedl's "Mastering Regular Expressions", published by O'Reilly (ISBN 1-56592-257-3), covers them in great detail. The description here is intended as reference documentation.

There are two different sets of meta-characters: those that are recognized anywhere in the pattern except within square brackets, and those that are recognized in square brackets. Outside square brackets, the meta-characters are as follows:

\ general escape character with several uses
^ assert start of subject (or line, in multiline mode)
$ assert end of subject (or line, in multiline mode)
. match any character except newline (by default)
[start character class definition
| start of alternative branch
(start subpattern
) end subpattern
? extends the meaning of "(" also 0 or 1 quantifier also quantifier minimizer
* 0 or more quantifier
+ 1 or more quantifier
{ start min/max quantifier

Part of a pattern that is in square brackets is called a "character class". In a character class the only metacharacters are:

\ general escape character
^ negate the class, but only if the first character

- indicates character range
] terminates the character class

The following sections describe the use of each of the metacharacters.

Backslash

The backslash character has several uses. Firstly, if it is followed by a non-alphanumeric character, it takes away any special meaning that character might have. This use of backslash as an escape character applies both inside and outside character classes.

For example, if you want to match a "*" character, you write "*" in the pattern. This applies whether or not the following character would otherwise be interpreted as a meta-character, so it is always safe to precede a non-alphanumeric with "\" to specify that it stands for itself. In particular, if you want to match a backslash, you write "\\".

An escaping backslash can be used to include a white space or "#" character as part of the pattern.

A second use of backslash provides a way of encoding non-printing characters in patterns in a visible manner. There is no restriction on the appearance of non-printing characters, apart from the binary zero that terminates a pattern, but when a pattern is being prepared by text editing, it is usually easier to use one of the following escape sequences than the binary character it represents. For example, \a represents "alarm", the BEL character (hex 07).

The handling of a backslash followed by a digit other than 0 is complicated. Outside a character class, RE ENGINE reads it and any following digits as a decimal number. If the number is less than 10, or if there have been at least that many previous capturing left parentheses in the expression, the entire sequence is taken as a *back reference*. A description of how this works is given later, following the discussion of parenthesized sub patterns.

Inside a character class, or if the decimal number is greater than 9 and there have not been that many capturing sub patterns, RE ENGINE re-reads up to three octal digits following the backslash, and generates a single byte from the least significant 8 bits of the value. Any subsequent digits stand for themselves. For example, \040 is another way of writing a space

Note that octal values of 100 or greater must not be introduced by a leading zero, because no more than three octal digits are ever read. All the sequences that define a single byte value can be used both inside and outside character classes. In addition, inside a character class, the sequence "\b" is interpreted as the backspace character (hex 08). Outside a character class it has a different meaning (see below).

The third use of backslash is for specifying generic character types:

\d any decimal digit
\D any character that is not a decimal digit
\s any white space character
\S any character that is not a white space character
\w any "word" character
\W any "non-word" character

Each pair of escape sequences partitions the complete set of characters into two disjoint sets. Any given character matches one, and only one, of each pair.

A "word" character is any letter or digit or the underscore character, that is, any character which can be part of a Perl "word". The definition of letters and digits is controlled by RE ENGINE's character tables, and might vary if locale-

specific matching is taking place (see "Locale support" above). For example, in the "fr" (French) locale, some character codes greater than 128 are used for accented letters, and these are matched by \w.

These character type sequences can appear both inside and outside character classes. They each match one character of the appropriate type. If the current matching point is at the end of the subject string, all of them fail, since there is no character to match.

The fourth use of backslash is for certain simple assertions. An assertion specifies a condition that has to be met at a particular point in a match, without consuming any characters from the subject string. The use of sub patterns for more complicated assertions is described below. The back slashed assertions are

\b word boundary

\B not a word boundary

\A start of subject (independent of multiline mode)

\Z end of subject or newline at end (independent of multiline mode)

\z end of subject (independent of multiline mode)

These assertions might not appear in character classes (but note that "\b" has a different meaning, namely the backspace character, inside a character class).

A word boundary is a position in the subject string where the current character and the previous character do not both match \w or \W (i.e. one matches \w and the other matches \W), or the start or end of the string if the first or last character matches \w, respectively.

The \A, \Z, and \z assertions differ from the traditional circumflex and dollar (described below) in that they only ever match at the very start and end of the subject string, whatever options are set. The difference between \Z and \z is that \Z matches before a newline that is the last character of the string as well as at the end of the string, whereas \z matches only at the end.

Circumflex and Dollar

Outside a character class, in the default matching mode, the circumflex character is an assertion which is true only if the current matching point is at the start of the subject string. Inside a character class, circumflex has an entirely different meaning (see below).

Circumflex need not be the first character of the pattern if a number of alternatives are involved, but it should be the first thing in each alternative in which it appears if the pattern is ever to match that branch. If all possible alternatives start with a circumflex, that is, if the pattern is constrained to match only at the start of the subject, it is said to be an "anchored" pattern. (There are also other constructs that can cause a pattern to be anchored.)

A dollar character is an assertion which is true only if the current matching point is at the end of the subject string, or immediately before a newline character that is the last character in the string (by default). Dollar need not be the last character of the pattern if a number of alternatives are involved, but it should be the last item in any branch in which it appears. Dollar has no special meaning in a character class.

Full Stop (Period, Dot)

Outside a character class, a dot in the pattern matches any one character in the subject, including a non-printing character, but not (by default) newline. The handling of dot is entirely independent of the handling of circumflex and dollar, the only relationship being that they both involve newline characters. Dot has no special meaning in a character class.

Square Brackets

An opening square bracket introduces a character class, terminated by a closing square bracket. A closing square bracket on its own is not special. If a closing square bracket is required as a member of the class, it should be the first data character in the class (after an initial circumflex, if present) or escaped with a backslash.

A character class matches a single character in the subject; the character must be in the set of characters defined by the class, unless the first character in the class is a circumflex, in which case the subject character must not be in the set defined by the class. If a circumflex is actually required as a member of the class, ensure it is not the first character, or escape it with a backslash.

For example, the character class [aeiou] matches any lower case vowel, while [^aeiou] matches any character that is not a lower case vowel. Note that a circumflex is just a convenient notation for specifying the characters which are in the class by enumerating those that are not. It is not an assertion: it still consumes a character from the subject string, and fails if the current pointer is at the end of the string.

When caseless matching is set, any letters in a class represent both their upper case and lower case versions, so for example, a caseless [aeiou] matches "A" as well as "a", and a caseless [^aeiou] does not match "A", whereas a caseful version would.

A class such as `[^a]` will always match a newline.

The minus (hyphen) character can be used to specify a range of characters in a character class. For example, [d-m] matches any letter between d and m, inclusive. If a minus character is required in a class, it must be escaped with a backslash or appear in a position where it cannot be interpreted as indicating a range, typically as the first or last character in the class. It is not possible to have the character "]" as the end character of a range, since a sequence such as [w-] is interpreted as a class of two characters. The octal or hexadecimal representation of "]" can, however, be used to end a range.

Ranges operate in ASCII collating sequence. They can also be used for characters specified numerically, for example [\000-\037]. If a range that includes letters is used when caseless matching is set, it matches the letters in either case. For example, [W-c] is equivalent to [][\^_`wxyzabc], matched caselessly, and if character tables for the "fr" locale are in use, [\xc8-\xcb] matches accented E characters in both cases.

The character types \d, \D, \s, \S, \w, and \W might also appear in a character class, and add the characters that they match to the class. For example, [\dABCDEF] matches any hexadecimal digit. A circumflex can conveniently be used with the upper case character types to specify a more restricted set of characters than the matching lower case type. For example, the class [^\W_] matches any letter or digit, but not underscore.

All non-alphanumeric characters other than \, -, ^ (at the start) and the terminating] are non-special in character classes, but it does no harm if they are escaped.

Vertical Bar

Vertical bar characters are used to separate alternative patterns. For example, the pattern

```
gilbert | sullivan
```

matches either "gilbert" or "sullivan". Any number of alternatives might appear, and an empty alternative is permitted (matching the empty string). The matching process tries each alternative in turn, from left to right, and the first one that succeeds is used. If the alternatives are within a subpattern (defined below), "succeeds" means matching the rest of the main pattern as well as the alternative in the subpattern.

Sub patterns

Sub patterns are delimited by parentheses (round brackets), which can be nested. Marking part of a pattern as a subpattern does two things:

1. It localizes a set of alternatives.

   For example, the pattern `cat(aract | erpillar|)` matches one of the words "cat", "cataract", or "caterpillar". Without the parentheses, it would match "cataract", "erpillar" or the empty string.

2. It sets up the subpattern as a capturing subpattern (as defined above). When the whole pattern matches, that portion of the subject string that matched the subpattern is passed back to the caller via the *ovector* argument of *RE Engine_exec()*. Opening parentheses are counted from left to right (starting from 1) to obtain the numbers of the capturing sub patterns.

   For example, if the string "the red king" is matched against the pattern `the ((red | white) (king | queen))` the captured substrings are "red king", "red", and "king", and are numbered 1, 2, and 3.

The fact that plain parentheses fulfill two functions is not always helpful. There are often times when a grouping subpattern is required without a capturing requirement. If an opening parenthesis is followed by "?:", the subpattern does not do any capturing, and is not counted when computing the number of any subsequent capturing sub patterns. For example, if the string "the white queen" is matched against the pattern the `((?:red | white) (king | queen))` the captured substrings are "white queen" and "queen", and are numbered 1 and 2. The maximum number of captured substrings is 99, and the maximum number of all sub patterns, both capturing and non-capturing, is 200.

As a convenient shorthand, if any option settings are required at the start of a non-capturing subpattern, the option letters might appear between the "?" and the ":". Thus the two patterns `(?i:saturday | sunday)` and (?:(?i)saturday | sunday) match exactly the same set of strings. Because alternative branches are tried from left to right, and options are not reset until the end of the subpattern is reached, an option setting in one branch does affect subsequent branches, so the above patterns match "SUNDAY" as well as "Saturday".

Repetition

Repetition is specified by quantifiers, which can follow any of the following items:

a single character, possibly escaped the . metacharacter

a character class

a back reference (see next section)

a parenthesized subpattern (unless it is an assertion - see below)

The general repetition quantifier specifies a minimum and maximum number of permitted matches, by giving the two numbers in curly brackets (braces), separated by a comma. The numbers must be less than 65536, and the first must be less than or equal to the second. For example `z{2,4}` matches "zz", "zzz", or "zzzz". A closing brace on its own is not a special character. If the second number is omitted, but the comma is present, there is no upper limit; if the second number and the comma are both omitted, the quantifier specifies an exact number of required matches. Thus `[aeiou]{3,}` matches at least 3 successive vowels, but might match many more, while `\d{8}` matches exactly 8 digits. An opening curly bracket that appears in a position where a quantifier is not allowed, or one that does not match the syntax of a quantifier, is taken as a literal character. For example, `{,6}` is not a quantifier, but a literal string of four characters.

The quantifier `{0}` is permitted, causing the expression to behave as if the previous item and the quantifier were not present. For convenience (and historical compatibility) the three most common quantifiers have single-character abbreviations:

* is equivalent to {0,}
+ is equivalent to {1,}
? is equivalent to {0,1}

It is possible to construct infinite loops by following a subpattern that can match no characters with a quantifier that has no upper limit, for example `(a?)*`

Earlier versions of Perl gave an error at compile time for such patterns. However, because there are cases where this can be useful, such patterns are now accepted, but if any repetition of the subpattern does in fact match no characters, the loop is forcibly broken.

By default, the quantifiers are "greedy", that is, they match as much as possible (up to the maximum number of permitted times), without causing the rest of the pattern to fail. The classic example of where this gives problems is in trying to match comments in C programs. These appear between the sequences /* and */ and within the sequence, individual * and / characters might appear. An attempt to match C comments by applying the following pattern fails, because it matches the entire string due to the greediness of the .* item.

```
  /\*.*\*/
```

to the string

```
  /* first command */  not comment  /* second comment */
```

However, if a quantifier is followed by a question mark, then it ceases to be greedy, and instead matches the minimum number of times possible, so the following pattern does the right thing with the C comments.

```
  /\*.*?\*/
```

The meaning of the various quantifiers is not otherwise changed, just the preferred number of matches. Do not confuse this use of question mark with its use as a quantifier in its own right. Because it has two uses, it can sometimes appear doubled, as below, which matches one digit by preference, but can match two if that is the only way the rest of the pattern matches.

```
  \d??\d
```

When a parenthesized subpattern is quantified with a minimum repeat count that is greater than 1 or with a limited maximum, more store is required for the compiled pattern, in proportion to the size of the minimum or maximum.

If a pattern starts with .* then it is implicitly anchored, since whatever follows will be tried against every character position in the subject string. RE ENGINE treats this as though it were preceded by \A.

When a capturing subpattern is repeated, the value captured is the substring that matched the final iteration. For example, after the following expression has matched "tweedledum tweedledee" the value of the captured substring is "tweedledee".

```
  (tweedle[dume]{3}\s*)+
```

However, if there are nested capturing sub patterns, the corresponding captured values might have been set in previous iterations. For example, after

```
  /(a|(b))+/
```

matches "aba" the value of the second captured substring is "b".

Back References

Outside a character class, a backslash followed by a digit greater than 0 (and possibly further digits) is a back reference to a capturing subpattern earlier (i.e. to its left) in the pattern, provided there have been that many previous capturing left parentheses.

However, if the decimal number following the backslash is less than 10, it is always taken as a back reference, and causes an error only if there are not that many capturing left parentheses in the entire pattern. In other words, the parentheses that are referenced need not be to the left of the reference for numbers less than 10. See the section entitled “Backslash” above for further details of the handling of digits following a backslash.

A back reference matches whatever actually matched the capturing subpattern in the current subject string, rather than anything matching the subpattern itself. So the following pattern matches “sense and sensibility” and “response and responsibility”, but not “sense and responsibility”.

```
(sens | respons)e and \1ibility
```

If caseful matching is in force at the time of the back reference, then the case of letters is relevant. For example, the following expression matches “rah rah” and “RAH RAH”, but not “RAH rah”, even though the original capturing subpattern is matched caselessly.

```
((?i)rah)\s+\1
```

There might be more than one back reference to the same subpattern. If a subpattern has not actually been used in a particular match, then any back references to it always fail. For example, the following pattern always fails if it starts to match “a” rather than “bc”. Because there might be up to 99 back references, all digits following the backslash are taken as part of a potential back reference number. If the pattern continues with a digit character, then some delimiter must be used to terminate the back reference.

```
(a | (bc))\2
```

A back reference that occurs inside the parentheses to which it refers fails when the subpattern is first used, so, for example, (a\1) never matches. However, such references can be useful inside repeated sub patterns. For example, the following pattern matches any number of “a”s and also “aba”, “ababaa” etc. At each iteration of the subpattern, the back reference matches the character string corresponding to the previous iteration. In order for this to work, the pattern must be such that the first iteration does not need to match the back reference. This can be done using alternation, as in the example above, or by a quantifier with a minimum of zero.

```
(a|b\1)+
```

Assertions

An assertion is a test on the characters following or preceding the current matching point that does not actually consume any characters. The simple assertions coded as \b, \B, \A, \Z, \z, ^ and $ are described above. More complicated assertions are coded as sub patterns. There are two kinds: those that look ahead of the current position in the subject string, and those that look behind it.

An assertion subpattern is matched in the normal way, except that it does not cause the current matching position to be changed. Lookahead assertions start with (?= for positive assertions and (?! for negative assertions. For example, the following expression matches a word followed by a semicolon, but does not include the semicolon in the match.

```
\w+(?=;)
```

The following expression matches any occurrence of “foo” that is not followed by “bar”.

```
foo(?!bar)
```

Note that the apparently similar pattern that follows does not find an occurrence of “bar” that is preceded by something other than “foo”; it finds any occurrence of “bar” whatsoever, because the assertion (?!foo) is always true when the next three characters are “bar”. A lookbehind assertion is needed to achieve this effect.

```
(?!foo)bar
```

Lookbehind assertions start with (?<= for positive assertions and (?<! for negative assertions. For example, the following expression does find an occurrence of “bar” that is not preceded by “foo”. The contents of a lookbehind assertion are restricted such that all the strings it matches must have a fixed length.

```
(?<!foo)bar
```

However, if there are several alternatives, they do not all have to have the same fixed length. Thus `(?<=bullock | donkey)` is permitted, but `(?<!dogs? | cats?)` causes an error at compile time. Branches that match different length strings are permitted only at the top level of a lookbehind assertion. This is an extension compared with Perl 5.005, which requires all branches to match the same length of string. An assertion such as `(?<=ab(c | de))` is not permitted, because its single branch can match two different lengths, but it is acceptable if rewritten to use two branches:

```
(?<=abc | abde)
```

The implementation of lookbehind assertions is, for each alternative, to temporarily move the current position back by the fixed width and then try to match. If there are insufficient characters before the current position, the match is deemed to fail.

Assertions can be nested in any combination. For example, the following expression matches an occurrence of “baz” that is preceded by “bar” which in turn is not preceded by “foo”.

```
(?<=(?<!foo)bar)baz
```

Assertion sub patterns are not capturing sub patterns, and might not be repeated, because it makes no sense to assert the same thing several times. If an assertion contains capturing sub patterns within it, these are always counted for the purposes of numbering the capturing sub patterns in the whole pattern. Substring capturing is carried out for positive assertions, but it does not make sense for negative assertions.

Assertions count towards the maximum of 200 parenthesized sub patterns.

Once-Only Sub patterns

With both maximizing and minimizing repetition, failure of what follows normally causes the repeated item to be re-evaluated to see if a different number of repeats allows the rest of the pattern to match. Sometimes it is useful to prevent this, either to change the nature of the match, or to cause it fail earlier than it otherwise might, when the author of the pattern knows there is no point in carrying on.

Consider, for example, the pattern `\d+foo` when applied to the subject line

```
123456bar
```

After matching all 6 digits and then failing to match “foo”, the normal action of the matcher is to try again with only 5 digits matching the \d+ item, and then with 4, and so on, before ultimately failing. Once-only sub patterns provide the means for specifying that once a portion of the pattern has matched, it is not to be re-evaluated in this way, so the matcher would give up immediately on failing to match “foo” the first time. The notation is another kind of special parenthesis, starting with (?> as in this example:

```
(?>\d+)bar
```

This kind of parenthesis “locks up” the part of the pattern it contains once it has matched, and a failure further into the pattern is prevented from backtracking into it. Backtracking past it to previous items, however, works as normal.

An alternative description is that a subpattern of this type matches the string of characters that an identical standalone pattern would match, if anchored at the current point in the subject string.

Once-only sub patterns are not capturing sub patterns. Simple cases such as the above example can be though of as a maximizing repeat that must swallow everything it can. So, while both \d+ and \d+? are prepared to adjust the

number of digits they match in order to make the rest of the pattern match, (?>\d+) can only match an entire sequence of digits.

This construction can of course contain arbitrarily complicated sub patterns, and it can be nested.

Conditional Sub patterns

It is possible to cause the matching process to obey a subpattern conditionally or to choose between two alternative sub patterns, depending on the result of an assertion, or whether a previous capturing subpattern matched or not. The two possible forms of conditional subpattern are

```
(?(condition)yes-pattern)
(?(condition)yes-pattern | no-pattern)
```

If the condition is satisfied, the yes-pattern is used; otherwise the no-pattern (if present) is used. If there are more than two alternatives in the subpattern, a compile-time error occurs.

There are two kinds of condition. If the text between the parentheses consists of a sequence of digits, then the condition is satisfied if the capturing subpattern of that number has previously matched. Consider the following pattern, which contains non-significant white space to make it more readable and to divide it into three parts for ease of discussion:

```
( \( )?    [^()]+    (?(1) \) )
```

The first part matches an optional opening parenthesis, and if that character is present, sets it as the first captured substring. The second part matches one or more characters that are not parentheses. The third part is a conditional subpattern that tests whether the first set of parentheses matched or not. If they did, that is, if subject started with an opening parenthesis, the condition is true, and so the yes-pattern is executed and a closing parenthesis is required. Otherwise, since no-pattern is not present, the subpattern matches nothing. In other words, this pattern matches a sequence of non-parentheses, optionally enclosed in parentheses.

If the condition is not a sequence of digits, it must be an assertion. This might be a positive or negative lookahead or lookbehind assertion. Consider this pattern, again containing non-significant white space, and with the two alternatives on the second line:

```
(?(?=[^a-z]*[a-z])
\d{2}[a-z]{3}-\d{2}  |  \d{2}-\d{2}-\d{2} )
```

The condition is a positive lookahead assertion that matches an optional sequence of non-letters followed by a letter. In other words, it tests for the presence of at least one letter in the subject. If a letter is found, the subject is matched against the first alternative; otherwise it is matched against the second. This pattern matches strings in one of the two forms dd-aaa-dd or dd-dd-dd, where aaa are letters and dd are digits.

Comments

The sequence (?# marks the start of a comment which continues up to the next closing parenthesis. Nested parentheses are not permitted. The characters that make up a comment play no part in the pattern matching at all.

Performance

Certain items that might appear in patterns are more efficient than others. It is more efficient to use a character class like `[aeiou]` than a set of alternatives such as `(a | e | i | o | u)`. In general, the simplest construction that provides the required behavior is usually the most efficient.

Regular Expression Engine Differences From PERL

The differences described here are with respect to Perl 5.005.

- Normally "space" matches space, formfeed, newline, carriage return, horizontal tab and vertical tab. Perl 5 no longer includes vertical tab in its set of white space characters. The \v escape that was in the Perl documentation for a long time was never in fact recognized. However, the character itself was treated as white space at least up to 5.002. In 5.004 and 5.005 it does not match \s.
- RE ENGINE does not allow repeat quantifiers on lookahead assertions. Perl permits them, but they do not mean what you might think. For example, `(?!a){3}` does not assert that the next three characters are not "a". It just asserts that the next character is not "a" three times.
- Capturing sub patterns that occur inside negative lookahead assertions are counted, but their entries in the offsets vector are never set. Perl sets its numerical variables from any such patterns that are matched before the assertion fails to match something (thereby succeeding), but only if the negative lookahead assertion contains just one branch.
- Though binary zero characters are supported in the subject string, they are not allowed in a pattern string because it is passed as a normal C string, terminated by zero. The escape sequence "\0" can be used in the pattern to represent a binary zero.
- The following Perl escape sequences are not supported: \l, \u, \L, \U, \E, \Q. In fact these are implemented by Perl's general string-handling and are not part of its pattern matching engine.
- The Perl \G assertion is not supported as it is not relevant to single pattern matches.
- RE ENGINE does not support the `(?{code})` construction.
- There are at the time of writing some oddities in Perl 5.005_02 concerned with the settings of captured strings when part of a pattern is repeated. For example, matching "aba" against the pattern `/^(a(b)?)+$/` sets $2 to the value "b", but matching "aabbaa" against `/^(aa(bb)?)+$/` leaves $2 unset. However, if the pattern is changed to `/^(aa(b(b))?)+$/` then $2 (and $3) get set. In Perl 5.004 $2 is set in both cases, and that is also true of RE ENGINE.
- Another as yet unresolved discrepancy is that in Perl 5.005_02 the pattern `/^(a)?(?(1)a | b)+$/` matches the string "a", whereas in RE ENGINE it does not. However, in both Perl and RE ENGINE `/^(a)?a/` matched against "a" leaves $1 unset.
- RE ENGINE provides some extensions to the Perl regular expression facilities: Although lookbehind assertions must match fixed length strings, each alternative branch of a lookbehind assertion can match a different length of string. Perl 5.005 requires them all to have the same length.

Regular Expression Examples

Common examples of URL matching used with CacheOS are listed below:

.*://.*\.edu$	Matches all URLs in the EDU domain.
http://.*\.(comp1 | comp2 | comp3)\.com.*	Matches URLs containing

	either comp1.com, comp2.com, or comp3.com.
ftp://.*	Matches all FTP requests.
.*cacheflow\.com:(8081 | 8084).*	Matches all requests to port 8081 or port 8084 in the cacheflow.com domain.

This page intentionally blank.

Appendix D - RIP Commands

net

```
net Nname[/mask] gateway Gname metric Value <passive|active|external>
```

Syntax

Parameters:	Description
Nname	Name of the destination network. It can be a symbolic network name, or an Internet address specified in dot notation.
/mask	Optional number between 1 and 32 indicating the netmask associated with Nname.
Gname	Name or address of the gateway to which RIP responses should be forwarded.
Value	The hop count to the destination host or network. A net Nname/32 specification is equivalent to the host Hname command.
passive|active|external	Indicates whether the gateway should be treated as passive or active, or whether the gateway is external to the scope of the RIP protocol.

host

```
host Hname gateway Gname metric Value <passive|active|external>
```

Syntax

Parameters:	Description
Hname	Name of the destination network. It can be a symbolic network name, or an Internet address specified in dot notation.
Gname	Name or address of the gateway to which RIP responses should be forwarded. It can be a symbolic network name, or an Internet address specified in dot notation.
Value	The hop count to the destination host or network. A net Nname/32 specification is equivalent to the host Hname command.
passive|active|external	Indicates whether the gateway should be treated as passive or active, or whether the gateway is external to the scope of the RIP

Parameters:	Description
	protocol.

RIP Parameters

Lines that do not start with net or host commands, must consist of one or more of the following parameter settings, separated by commas or blanks:

Parameters:	Description
if=[0|1|2|3]	Indicates that the other parameters on the line apply to the interface numbered 0,1,2, or 3 in CacheOS terms.
passwd=XXX	Specifies a RIPv2 password that will be included on all RIPv2 responses sent and checked on all RIPv2 responses received. The password must not contain any blanks, tab characters, commas or '#' characters.
no_ag	Turns off aggregation of subnets in RIPv1 and RIPv2 responses.
no_super_ag	Turns off aggregation of networks into supernets in RIPv2 responses.
passive	Marks the interface to not be advertised in updates sent via other interfaces, and turns off all RIP and router discovery through the interface.
no_rip	Disables all RIP processing on the specified interface.
no_ripv1_in	Causes RIPv1 received responses to be ignored.
no_ripv2_in	Causes RIPv2 received responses to be ignored.
ripv2_out	Turns off RIPv1 output and causes RIPv2 advertisements to be multicast when possible.
ripv2	Is equivalent to no_ripv1_in and no_ripv1_out. **This parameter is set by default.**
no_rdisc	Disables the Internet Router Discovery Protocol. **This parameter is set by default.**
no_solicit	Disables the transmission of Router Discovery Solicitations.
send_solicit	Specifies that Router Discovery solicitations should be sent, even on point-to-point links, which by default only listen

Parameters:	Description
	to Router Discovery messages.
no_rdisc_adv	Disables the transmission of Router Discovery Advertisements.
rdisc_adv	Specifies that Router Discovery Advertisements should be sent, even on point-to-point links, which by default only listen to Router Discovery messages.
bcast_rdisc	Specifies that Router Discovery packets should be broadcast instead of multicast.
rdisc_pref=N	Sets the preference in Router Discovery Advertisements to the integer N.
rdisc_interval=N	Sets the nominal interval with which Router Discovery Advertisements are transmitted to N seconds and their lifetime to 3*N.
trust_gateway=rname	Causes RIP packets from that router and other routers named in other trust_gateway keywords to be accept, and packets from other routers to be ignored.
redirect_ok	Causes RIP to allow ICMP Redirect messages when the system is acting as a router and forwarding packets. Otherwise, ICMP Redirect messages are are overridden.

CacheOS-Specific RIP Parameters

The following RIP parameters are unique to CacheOS configuration:

Parameters:	Description
no_rip_out	Disables the transmission of all RIP packets. This setting is the default.
no_ripv1_out	Disables the transmission of RIPv1 packets.
no_ripv2_out	Disables the transmission of RIPv2 packets.
rip_out	Enables the transmission of RIPv1 packets.
ripv1_out	Enables the transmission of RIPv1 packets.
rdisc	Enables the transmission of Router Discovery Advertisements.
ripv1	Causes RIPv1 packets to be sent.
ripv1_in	Causes RIPv1 received responses to be handled.

Using Passwords with RIP

The first password specified for an interface is used for output. All passwords pertaining to an interface are accepted on input. For example, with the following settings:

```
if=0 passwd=aaa
if=1 passwd=bbb
passwd=ccc
```

Interface 0 would accept passwords aaa and ccc, and would transmit using password aaa. Interface 1 would accept passwords bbb and ccc, and would transmit using password bbb. The other interfaces would accept and transmit the password ccc

Appendix E - Severe Error Message Reference

This appendix provides a list of severe error messages that can be generated by CacheOS. Severe error messages are defined as an error or errors that can potentially impact the reliability or availability of a Content Accelerator. CacheFlow documents only severe error messages.

When configuring Event Logging, CacheFlow recommends that you set Event logging to log only severe errors only, unless you have a specific need for a more detailed log.

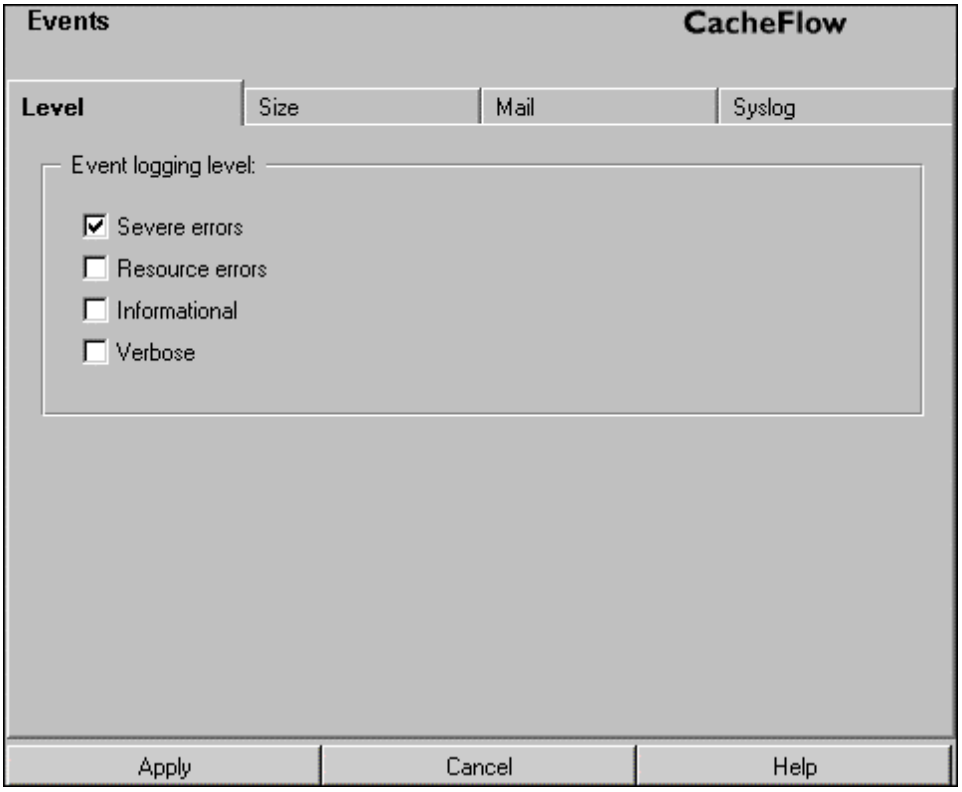


Figure E-1 Setting the event logging level

Event Log Format

The Event Log format is as follows:

```
<UTC Time> "<Text Description>" <Additional Hex Information> <Event_Code>:<Event
Severity> <Source Filename> : <Source File Line Number>
```

Event Severity is a number from 0 to 255 showing the importance of the message. 0 is of the highest importance, 255, the lowest.

Severe Error Messages

Error/Message	Explanation
AUTH_EVENT_AUTHLDAP_FAIL.	In the case of AUTH_EVENT_AUTHLDAP_FAIL, LDAP has returned an error code that indicates a problem on the LDAP server . Contact Technical Support.
AUTH_EVENT_AUTHRADIUS_FAIL.	In the case of AUTH_EVENT_AUTHRADIUS_FAIL, RADIUS has attempted to communicate with either the primary or the alternate RADIUS server specified in the Content Accelerator configuration. The communication attempt failed. This problem could be caused by one of the following reasons: • The configuration for the primary RADIUS server IP address is incorrect. • The configuration for the alternate RADIUS server IP address is incorrect. • The configuration for the primary RADIUS server UDP port is incorrect. • The configuration for the alternate RADIUS server UDP port is incorrect. • The primary or alternate RADIUS server is offline. • The primary and secondary RADIUS servers are not reachable using the current adapter configuration (see configuration of network interface). Contact Technical Support.
AUTH_EVENT_AUTHRADIUS_TIMEOUT.	Almost identical to AUTH_EVENT_AUTHRADIUS_FAIL. In this case, RADIUS has attempted to communicate with either the primary or the alternate RADIUS server specified in the Content Accelerator configuration. The communication attempt timed-out because neither the primary not the alternate RADIUS server responded to the request. Contact Technical Support.
Booting is being done from suspect disk DISKID in slot SLOT.	When this occurs, the boot disk was declared unusable, probably just prior to the current restart. Since declaring the boot disk unusable causes an immediate restart, there is no opportunity to report this prior to the restart. This message is intended to report the problem after the restart. No related action is necessary; however, since the boot disk was previously declared unusable, there might be something seriously wrong with it. Because this message was sent, the disk was used successfully for rebooting. This, in turn, means that the error, causing the disk to be declared unusable, might have been a transient error. In any case, the administrator should investigate the problem. DISKID and SLOT are the same as for the above message. Contact Technical Support.
Connection refused on DNS lookup to DNSSERVER for HOSTNAME.	When this occurs, the DNS server has refused a lookup request. This probably means that the DNS service has been disabled. Only the first in a

Error/Message	Explanation
DNSSERVER for HOSTNAME.	series of refusals is reported. On the next successful DNS lookup after a refusal, the number of refusals is reported in a normal severity message. The administrator should re-enable the DNS server. DNSSERVER and HOSTNAME are the same as for the above message. Contact Technical Support.
CONTENT FILTER: The automatic database download feature of the content filtering service has not been configured. This feature MUST be configured for effective service. Please configure the automatic database download feature.	This message will be logged when the content filtering database "auto-download" feature is disable, through either the CLI or the GUI, or during system startup. Contact Technical Support.
Disk DISKID in slot SLOT has only FREE%% free space (FREEBLOCKS blocks free out of TOTALBLOCKS) for TOTALOBJECTS objects.	When this error occurs, the system displays this message, which provides early warning that the free space on the specified disk has fallen below a critical threshold, which is currently 5%. (Normally, free disk space should be maintained around 20%.) Even though no immediate related action is necessary, the fact that free space has fallen so low indicates that some serious problem has occurred or is occurring. If free space reaches zero, the Content Accelerator will reboot. If free space does not eventually recover its normal level, the offending disk should be removed and returned to CacheFlow for analysis. DISKID is the hexadecimal identifier of the disk (a unique identifier). SLOT is the slot number (starting from 1, at left), in which the disk resides in FREE. FREE is the current free space percentage. FREEBLOCKS is the number of free blocks on the disk. TOTALBLOCKS is the total number of blocks on disk. TOTALOBJECTS is the current number of objects on disk. Contact Technical Support.
Disk DISKID in slot SLOT has only FREEBLOCKS free blocks out of TOTALBLOCKS for TOTALOBJECTS objects. (Last status SCSISTATUS.)	When this occurs, the eight largest objects are reported (as described for the above severe error) after this message is given. This message was a fatal error (which wouldn't have been logged because the Content Accelerator was rebooted immediately). Now it is a severe message, which can be recorded in the event log. DISKID, SLOT, FREEBLOCKS, TOTALBLOCKS and TOTALOBJECTS are the same as for the above messages. SCSISTATUS is the most recent non-zero status reported by SCSI. The free space on the specified disk has been exhausted. FREEBLOCKS might be non-zero because there could be free blocks unable to be reused. If the specified disk is not the boot disk, it will be re-initialized. If the specified disk is the boot disk, then the Content Accelerator is rebooted. Contact Technical Support.
Disk DISKID in slot SLOT is invalid because STRING IO status is CASTATUS (SCSISTATUS).	When this occurs, the specified disk has been declared invalid. This means that some non-recoverable IO error has occurred on it. The disk is re-initialized. The specified disk is a non-boot disk. If the boot disk is declared invalid,

Error/Message	Explanation
	the Content Accelerator is restarted. Since the disk is re-initialized, or the Content Accelerator restarted, no related action is necessary. The error might, however, indicate a disk that is about to fail in a more serious way. DISKID, SLOT, STRING, CASTATUS and SCSISTATUS are the same as for the above message. Contact Technical Support.
Disk DISKID in slot SLOT is unusable because STRING. IO status is CASTATUS (SCSISTATUS).	When this occurs, the specified disk has been declared unusable. The disk is taken offline and won't be used until after the next reboot. The STRING shows when the error was detected. CASTATUS and SCSISTATUS indicate which error occurred. DISKID and SLOT are one of the following, indicating the reason for invalidity: • the allocation map cannot be read • a block cannot be written • the R-block address is invalid • the R-block cannot be read • an internal object cannot be read • a hash block cannot be read • a boot component cannot be read • the allocation map cannot be written • an initial block cannot be written • a hash block cannot be rewritten • the R-sector cannot be read • the R-sector cannot be written • a boot component cannot be written CASTATUS is the cache administrator status; SCSISTATUS is the SCSI status. SCSISTATUS is usually interpreted by referring to the hardware documentation for the corresponding disk. The specified disk is a non-boot disk. If the boot disk is declared unusable, the Content Accelerator is restarted. The Content Accelerator should probably be rebooted (but not necessarily immediately) to determine if the error recurs on the disk. After the reboot, the disk is re-initialized, if it is not the boot disk. If re-initialization fails, the disk should be returned to CacheFlow; if re-initialization succeeds, then the error may have been some type of transient error. Contact Technical Support.
DNS lookup to DNSSERVER for HOSTNAME timed out.	When this occurs, a DNS lookup has timed out. This probably means that it is no longer accessible, perhaps because of a network error or some other problem. Only the first timeout in a series of timeouts is reported. On the next successful DNS lookup after a timeout, the number of failed lookups due to timeouts will be reported in a normal severity message. The administrator should attempt to determine why the DNS server is no longer accessible. DNSSERVER is the name or IP address of the DNS server for the Content

Error/Message	Explanation
	Accelerator. HOSTNAME is the hostname being looked up when the timeout occurred. Contact Technical Support.
LDAP_EVENT_CONNECT_FAILED.	In the case of LDAP_EVENT_CONNECT_FAILED, LDAP has attempted to connect to either the primary or the alternate LDAP server specified in the Content Accelerator configuration. The connection attempt failed. This problem could be caused by one of the following reasons: • The configuration for the primary LDAP server IP address is incorrect. • The configuration for the alternate LDAP server IP address is incorrect. • The configuration for the primary LDAP server TCP port is incorrect. • The configuration for the alternate LDAP server TCP port is incorrect. • The primary or alternate LDAP server is offline. • The primary and secondary LDAP servers are not reachable using the current adapter configuration. (See configuration of network interface.) Contact Technical Support.
LDAP_EVENT_CONNECT_FAILED.	In the case of LDAP_EVENT_CONNECT_FAILED, LDAP has attempted to connect to either the primary or the alternate LDAP server specified in the Content Accelerator configuration. The connection attempt failed. This problem could be caused by one of the following reasons: • The configuration for the primary LDAP server IP address is incorrect. • The configuration for the alternate LDAP server IP address is incorrect. • The configuration for the primary LDAP server TCP port is incorrect. • The configuration for the alternate LDAP server TCP port is incorrect. • The primary or alternate LDAP server is offline. • The primary and secondary LDAP servers are not reachable using the current adapter configuration. (See configuration of network interface.) Contact Technical Support.
SMTP: DNS error looking up gateway.	This means that the SMTP gateway that is configured can't be found through a DNS lookup. Either the name is wrong, or there is something wrong with the DNS configuration. Contact Technical Support.
SMTP_ERROR_DEFINITION.	This means that the SMTP gateway that is configured can't be found through a DNS lookup. Either the name is wrong, or there is something wrong with the DNS configuration. Contact Technical Support.
SMTP_worker: Error connecting to mail gateway %s.	This means that the SMTP process can't connect to the SMTP server. Either there is a network problem or the server isn't running. Contact Technical Support.
The NTH largest object on disk SLOT is URL, which is LENGTH bytes long. (Flags OHTFLAGS, ODFLAGS)	This occurs when the free space on disk is low or when the free space on disk is exhausted, the eight largest objects on the disk are located and each is reported by a message like the one above. This might allow you to discover why the free space is exhausted on some disks at certain installations.

Error/Message	Explanation
	Even though the message is a debugging message to some extent, it is severe enough that it will not be suppressed on a normal production system. No action is needed for these messages, but appropriate action should be taken for the associated low-free-space or out-of-free-space message. NTH is one of the "1st", "2nd" or "3rd", indicating how large this object is. SLOT is the slot number of the disk containing the object URL. LENGTH is the length of the object in bytes. OHTFLAGS is the object hash table flags in hexadecimal. ODFLAGS is the object descriptor flags in hexadecimal. Contact Technical Support.
WCCP: Cannot add multicast membership.	An internal error prevents WCCP from creating a multicast UDP socket (with a multicast address between 224.0.0.0 to 239.255.255.255). As such, the Content Accelerator would be unable to participate in a WCCP service group. Contact Technical Support
WCCP: Cannot alloc memory.	An internal error prevents WCCP from allocating memory from its internal memory heap manager. Depending upon the circumstance, the Content Accelerator's ability to participate in WCCP service groups might be jeopardized. Contact Technical Support.
WCCP: Cannot bind socket	An internal error prevents WCCP fro binding a UDP socket to a local IP address. As such, the Content Accelerator would be unable to participate in a WCCP service group. Contact Technical Support
WCCP: cannot create namespace.	An internal error prevents WCCP from registering a command syntax definition with the command parser. As such, the Content Accelerator would be unable to parse WCCP configuration objects. The Content Accelerator would not be able to participate in WCCP sessions with routers. Contact Technical Support.
WCCP: cannot create parser handle.	An internal error prevents WCCP from registering a command syntax definition with the command parser. As such, the Content Accelerator would be unable to parse WCCP configuration objects. The Content Accelerator would not be able to participate in WCCP sessions with routers. Contact Technical Support.
WCCP: Cannot create receive thread.	An internal error prevents WCCP from creating a thread for receiving WCCP protocol packets. As such, the Content Accelerator would be unable to participate in a WCCP service group. Contact Technical Support.
WCCP: Cannot create socket	An internal error prevents WCCP from creating a UDP socket necessary for initiating a WCCP sessions. As such, the Content Accelerator would be unable to participate in a WCCP service group.

Error/Message	Explanation
	Contact Technical Support.
WCCP: cannot register main commands.	An internal error prevents WCCP from registering a command syntax definition with the command parser. As such, the Content Accelerator would be unable to parse WCCP configuration objects. The Content Accelerator would not be able to participate in WCCP sessions with routers. Contact Technical Support.
WCCP: cannot register version 1 service-group commands.	An internal error prevents WCCP from registering a command syntax definition with the command parser. As such, the Content Accelerator would be unable to parse WCCP configuration objects. The Content Accelerator would not be able to participate in WCCP sessions with routers. Contact Technical Support.
WCCP: cannot register version 2 service-group commands.	An internal error prevents WCCP from registering a command syntax definition with the command parser. As such, the Content Accelerator would be unable to parse WCCP configuration objects. The Content Accelerator would not be able to participate in WCCP sessions with routers. Contact Technical Support.

This page intentionally blank.

Appendix F - CacheOS Command Reference

The CacheOS command-line interface allows you to configure and manage the Content Accelerator using Telnet or the Serial Console interface.

The command-line interface has two modes: standard mode and privileged mode. Standard mode commands allow you to view the configuration settings. Privileged mode commands allow you to both view and change the configuration. When you first connect to the command-line interface, you are in standard mode. To enter privileged mode, type **enable** and then enter the enable password (if required), when prompted:

```
telnet> open 10.25.36.47
username: admin
password: ******
CacheOS> enable
password: ******
CacheOS#
```

When you enter enabled mode, a pound (#) character is added to the command prompt. To leave privileged mode, type **exit** or press Ctrl-Z.

Standard Mode Commands

disable

Turn off privileged commands

Syntax

disable

The disable command does not have any parameters or subcommands.

Example

```
CacheOS#disable
```

display

Display a text-based HTTP URL.

Syntax

display ***url***

The display has the parameter url.

Example

```
CacheOS>display www.cacheflow.com
```

enable

Turn on privileged commands

Syntax

enable

The enable command does not have any parameters or subcommands.

Example

```
CacheOS>enable
Password:******
CacheOS#
```

exit

Exit command-line interface

Syntax

exit

The exit command does not have any parameters or subcommands.

Example

```
CacheOS>exit
```

help

Information on displaying help

Syntax

help

The help command does not have any parameters or subcommands.

Example

```
CacheOS>help
Help may be requested at any point in a command
by typing a question mark '?'.
1. For a list of available commands, enter '?' at
   the prompt.
2. For a list of arguments applicable to a command,
   precede the '?' with a space (e.g. 'show ?')
```

```
3. For help completing a command, do not precede
   the '?' with a space (e.g. 'sh?')
10.25.36.47 CacheFlow>
```

ping

Send echo messages

Syntax

ping

The help command does not have any parameters or subcommands.

Example

```
CacheOS>ping 10.25.36.47
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 10.25.36.47, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 0/0/0 ms
Number of duplicate packets received = 0
```

show

Show running system information

Syntax

show

The help command has several parameters and subcommands. Please refer to the table presented later in this appendix.

traceroute

Trace route to destination

Syntax

traceroute ***ip*** **|** ***hostname***

The traceroute command has the parameters/subcommands ip and hostname.

Example

```
CacheOS>traceroute 10.25.36.47
```

Type escape sequence to abort.

Tracing the route to 10.25.36.47

1 10.25.36.47 0 0 0

Privileged-Mode Commands

acquire-utc

Acquires UTC from the NTP server.

Syntax

acquire-utc

The acquire-utc command does not have any parameters or subcommands.

Example

```
CacheOS#acquire-utc
  ok
```

clear-arp

Clears the ARP table.

Syntax

clear-arp

The clear-arp command does not have any parameters or subcommands.

Example

```
CacheOS#clear-arp
  ok
```

clear-cache

Clear the contents of the cache.

Syntax

clear-cache

Example

```
CacheOS#clear-cache
  ok
```

configure

The configure command allows you to configure CacheOS settings from your current terminal session, or by loading a text file of configuration settings from the network.

Syntax

configure terminal | network *url*

Example

```
CacheOS#config t
Enter configuration commands, one per line.  End with CTRL-Z.
CacheOS#(config)
```

To configure CacheOS from the terminal, type **configure terminal**. The prompt changes to `#(config)`. At the `#(config)` prompt, you can enter any of the configure commands. For a list of available commands when using the command-line interface, enter a question mark. To exit configuration mode, press Ctrl-Z or type **exit** at the command prompt.

The **configure terminal** command allows you to modify the following settings:

Command	Description
access-log	Configure access log
archive-configuration	Save system configuration
authentication	Configure authentication
banner	Define a login banner
bypass-list	Installation parameters for bypass list
caching	Modify caching parameters
clock	Modify clock settings
content-filter	Configure content filter
direct-deny-list	Installation parameters for direct or deny list
dns	Modify DNS settings
dynamic-bypass	Modify dynamic bypass configuration
error-pages	Configure HTTP error pages
event-log	Configure event log parameters
filter-list	Installation parameters for filter list
forwarding	Forward requests to another cache
hostname	Set system hostname
http-proxy-port	Specify proxy port for HTTP requests
icp	Installation parameters for ICP settings
inline	Install configurations from console input
interface	Select an interface to configure

Command	Description
ip-default-gateway	Specify the default IP gateway
line-vty	Configure a terminal line
management-port	Specify port for WEB console
no	Clear certain parameters
ntp	Modify NTP parameters
restart	System restart behavior
return-to-sender	IP “return to sender” behavior
rip	Modify RIP configuration.
rtsp	Modify RTSP configuration
security	Modify security parameters
snmp	Modify SNMP parameters
socks-machine-id	Specify machine ID for SOCKS
static-routes	Installation parameters for static routes table
streaming	Configure streaming
telnet-management	Enable or disable Telnet access to CLI
timezone	Set local timezone
transparent-proxy	Enable or disable transparent proxy
wccp	Configure WCCP parameters
upgrade-path	Network path to download system software
web-management	Enable or disable Web console

The commonly used configure commands are described below. For syntax help on commands not included here, use the command-line interface help.

access-log

Allows you to configure the access log settings.

The CacheOS can maintain an access log for each HTTP request made. The access log can be stored in one of three formats, which can be read by a variety of reporting utilities. See the *Access Log Formats* chapter for additional information on log formats.

Syntax

access-log action | alternate | disable | enable | filename-prefix | format | no | primary | upload

When you enter the access-log command, the interface displays the config access-log prompt, where you can enter access-log commands:

Command	Parameters	Description
action	stop | upload	What to do if access log exceeds allotted size.

Command	Parameters	Description
alternate	*host* | *password* | *path* | *username*	Configure secondary access log upload site.
disable		Disable access logging.
enable		Enable access logging.
filename-prefix	*prefix string*	Configure upload filename prefix
format	common | squid-compatible | custom	Configure access log format.
no	alternate | primary	Delete primary or alternate upload site.
primary	*host* | *password* | *path* | *username*	Configure primary access log upload site.
threshold	*percent*	Percent of disk access log can consume.
upload	daily | hourly	Specify access log upload interval.

Example

```
CacheOS#config)access-log
CacheOS#(config access-log)enable
    ok
CacheOS#(config access-log)format squid-compatible
    ok
CacheOS#(config access-log)
```

archive-configuration

Configures the archive.

Syntax

archive-configuration host | password | path | protocol | username

Command	Parameter/Subcommand	Description
host	*host name*	Upload configuration to this FTP host
password	*password*	Password for FTP upload host
path	*path*	Path on FTP upload host
protocol	ftp | tftp	Sets the upload protocol.
username	*username*	Username for FTP upload host

Example

```
CacheOS#(config)archive-configuration password wallyworld
```

```
ok
```

authentication

Provides authentication for certain protocols.

Syntax

authentication admin-verification | ldap | protocol | radius | user-verification

Command	Parameter/Subcommand	Description
admin-verification	disable | enable	Enable or disable authentication for administrators
ldap	See table	Configure LDAP authentication
protocol	ldap | none | radius | unix-password-file	Select authentication protocol
radius	See table	Configure radius authentication
user-verification	disable | enable	Enable or disable authentication for users

Example

```
CacheOS#(config)authentication admin-verification disable
    ok
```

ldap

This configures LDAP authentication.

Syntax

ldap admin-attribute | alternate-server | cache-duration | distinguished-name | grant-access-on-bind | no | primary-server | user-attribute

Command	Parameter/Subcommand	Description
admin-attribute	*type* | *value*	Configure the administrator attributes
alternate server	*ip* | *port*	Alternate LDAP server configuration
cache-duration	*minutes*	Length of time to cache user credentials
distinguished-name	*prefix* | *suffix*	Configure LDAP distinguished name prefix and suffix
grant-access-on-bind		Grant proxy user access on bind only
no	*grant-access-on-bind*	Negate certain LDAP parameters
primary-server	*ip* | *port*	Primary LDAP server configuration

Command	Parameter/Subcommand	Description
user-attribute	*type* | *value*	Configure the user attributes

radius

This configures radius authentication.

Syntax

radius alternate-server | primary-server | query-timeout | server-retry

Command	Parameter/Subcommand	Description
alternate server	*ip* | *port* | *secret*	Alternate Radius server configuration
primary-server	*ip* | *port* | *secret*	Primary Radius server configuration
query timeout	*seconds*	Radius server query timeout. Possible values are 0 – 65535 seconds.
server retry	*count*	Number of authentication attempts to attempt. Possible values are 0 – 65535.

banner

Defines a login banner.

Syntax

banner login | no

Command	Parameter/Subcommand	Description
login	*string*	Set login banner
no	login	Negate banner commands

Example

```
CacheOS#(config)banner no login
   ok
```

bypass-list

Sets bypass list options. The bypass list is only used for transparent caching.

Bypass routes are used to prevent the Content Accelerator from transparently proxying requests to servers that perform IP authentication with clients. The bypass list contains a list of IP addresses, subnet masks, and gateways.

When a request matches an IP address and subnet mask specification in the bypass list, the request is sent to the designated gateway.

To use bypass routes, create a text file that contains a list of address specifications. The file should be named with a .txt extension. Once you have created the bypass list, place it on an HTTP server so it can be installed onto the Content Accelerator.

There are two types of bypass lists: the local bypass list, and the central bypass list.

You can create your own central bypass list to manage multiple Content Accelerators, or you can use the central bypass list maintained by CacheFlow Technical Support at http://www.cacheflow.com/support/subscriptions/CentralBypassList.txt

The central bypass list maintained by CacheFlow contains addresses CacheFlow has identified as using client authentication.

Syntax

bypass-list central-path | local-path | no | notify | poll-now | subscribe

Command	Parameter/Subcommand	Description
central-path	*url*	Network path to download central bypass list
local-path	*url*	Network path to download local bypass list
no	central-path | local-path | notify | subscribe	Negate bypass list parameter/subcommands
notify		Send e-mail when central bypass list changes
poll-now		Check if central bypass list has changed
subscribe		Update bypass list when central list changes

Example

```
CacheOS#(config)bypass-list local-path 10.25.36.47/files/bypasslist.txt
    ok
```

caching

The caching command allows you to configure how CacheOS manages the cached objects.

When the Content Accelerator retrieves an object from the Web and returns it to the client, the object is considered fresh--the Web cache knows it is fresh because it just retrieved the object from the source. The goal of the Web cache is to keep fresh as many of the objects in the cache as possible, so when the objects are requested CacheOS can deliver them to the client without having to retrieve them from the source.

Syntax

caching always-verify-source | ftp | max-cache-size | negative-response | no | refresh

When you enter the caching command, the interface displays the config caching prompt, where you can enter caching commands:

Command	Parameter/Subcommand	Description
always-verify-source		Always verify object freshness with source
ftp	See table	FTP caching parameter/subcommands
max-cache-size	*megabytes*	Maximum size object to cache
negative-response	*minutes*	Cache negative responses
no	always-verify-source	Negate always-verify-source
refresh	See table	Refresh parameter/subcommands

Example

```
CacheOS#(config)caching
CacheOS#(config caching)always-verify-source
    ok
```

ftp

Syntax

ftp disable | enable | max-cache-size | type-m-percent | type-n-initial

Command	Parameter/Subcommand	Description
disable		Disable caching FTP objects
enable		Enable caching FTP objects
max-cache-size	*megabytes*	Maximum size FTP object to cache
type-m-percent	*percent*	Time to live for objects with last modified time
type-n-initial	*percent*	Time to live for objects without expiration time

refresh

Syntax

refresh automatic | bandwidth | desired-freshness | no

Command	Parameter/Subcommand	Description
automatic		Let CacheOs manage refresh bandwidth
bandwidth	*kbps*	Bandwidth in kilobits to use for

Command	Parameter/Subcommand	Description
		refresh
desired-freshness	*percent*	Desired freshness for refreshed objects
no	automatic	Negate “automatic”

clock

Displays current time.

Syntax

clock *day* | *hour* | *month* | *minute* | *second* | *year*

Example

```
CacheOS#(config)clock minute 59
  ok
```

content-filter

Configures the content filter.

Syntax

content-filter disable | enable | select-provider | smartfilter | test-url | websense

Command	Parameter/Subcommand	Description
disable		Disable content-filter
enable		Enable content-filter
select-provider	smartfilter | websense	Select service provider
show		Show running system information
smartfilter	See table	Configure SmartFilter
test-url	*url*	Test URL against content-filter
websense	See table	Configure WebSense

Example

```
CacheOS#(config)content-filter enable
  ok
```

smartfilter

Syntax

smartfilter category | download | no

Command	Parameter/Subcommand	Description
category	block | unblock	Configure content categories
download	See table	Configure download parameter/subcommands
no	*control-file* | *DNR-control-file* | *path* | *password* | *username*	Negate certain parameter/subcommands
show		Show running system information

download

Syntax

download control-file | day-of-week | disable-auto | dnr-control-file | enable-auto | get-now | path | password | time-of-day | username

Command	Parameter/Subcommand	Description
control-file	*filename*	Control database file
day-of-week	all | *day*	Day of week for automatic downloads
disable-auto		Disable automatic downloads
dnr-control-file	*filename*	Domain resolved control data
enable-auto		Enable automatic downloads
get-now		Initiate database download
path	*url*	Network path to download database
password	*text*	Network password
time-of-day	*hour* (0-23)	Time of day for automatic downloads
username	*text*	Network username

Websense

Syntax

websense category | download | no

Command	Parameter/Subcommand	Description
category	block | unblock	Configure content categories
download	See table	Configure download parameter/subcommands
show		Show running system information

download

Syntax

download control-file | day-of-week | disable-auto | dnr-control-file | enable-auto | get-now | server | path | password | time-of-day | username

Command	Parameter/Subcommand	Description
control-file	*filename*	Control database file
day-of-week	all | *day*	Day of week for automatic downloads
disable-auto		Disable automatic downloads
dnr-control-file	*filename*	Domain resolved control data
enable-auto		Enable automatic downloads
get-now		Initiate database download
server	*server name* or *IP address*	Sets the WebSense download server address.
path	*url*	Network path to download database
password	*text*	Network password
time-of-day	*hour* (0-23)	Time of day for automatic downloads
username	*text*	Network username
username		Negate network username

direct-deny-list

Configures the direct or deny settings for forwarding.

When using forwarding, CacheOS forwards requests for objects not found in the cache to the forwarding gateway. The forwarding gateway then determines what to do with the request.

Direct addresses are addresses CacheOS should send directly on the network rather than to the forwarding gateway. Deny addresses are addresses to which CacheOS should deny access. The direct and deny address specifications are made up of a subnet and mask. Requested addresses are compared to the subnet and mask to determine a match. If the request does not match an address in the direct or deny list, CacheOS sends the request to the gateway.

The direct and deny list is a simple text file containing a list of IP addresses, subnet masks and commands. A sample direct or deny list is illustrated below:

```
10.25.36.47   255.255.0.0   DENY
10.25.36.48   255.255.0.0   DENY
10.25.36.40   255.255.0.0   DIRECT
```

To enter a direct or deny list, create a text file with the direct or deny commands, then place the file on an HTTP server. To download the list to CacheOS, use the load command.

Syntax

direct-deny-list path | no

Command	Parameter/Subcommand	Description
no	*path*	Clears the direct-deny-list URL
path	*url*	Network path to download "direct or deny" list

Example

```
CacheOS#(config)direct-deny http://10.25.36.47/files/dodlist.txt
    ok
```

dns

Allows you to modify the DNS settings for the Web cache. Note that the alternate DNS servers are only checked if the servers in the standard DNS list return: "Name not found."

Syntax

dns alternate | clear | imputing | server | no

Command	Parameter/Subcommand	Description
alternate	*ip address*	Add entry to alternate dns server list
clear	alternate | imputing | server	Remove all entries from a list
imputing	*name*	Add an entry to the name imputing list
server	*ip address*	Add an entry to the primary dns server list
no	alternate | imputing | server	Remove a single entry from a list

Example

```
CacheOS#(config)dns server 10.25.36.47
    ok
```

dynamic-bypass

Modifies dynamic bypass configuration.

Syntax

dynamic-bypass clear | disable | enable | no | trigger

Command	Parameter/Subcommand	Description
clear		Remove all entries from the

Command	Parameter/Subcommand	Description
		dynamic bypass list
disable		Disable the dynamic bypass list
enable		Enable the dynamic bypass list
no	trigger	Negate dynamic bypass settings
trigger	See table	Specify dynamic bypass criteria

Example

```
CacheOS#(config)dynamic-bypass clear
    ok
```

trigger

Syntax

trigger all | non-http | 400 | 401 | 403 | 405 | 406 | 500

Command	Parameter/Subcommand	Description
all		Enable all bypass list triggers
non-http		Enable dynamic bypass for non-HTTP responses
400		Enable dynamic bypass for HTTP 400 responses
401		Enable dynamic bypass for HTTP 401 responses
403		Enable dynamic bypass for HTTP 403 responses
405		Enable dynamic bypass for HTTP 405 responses
406		Enable dynamic bypass for HTTP 406 responses
500		Enable dynamic bypass for HTTP 500 responses

error-pages

Configures HTTP error pages.

Syntax

error-pages | no | path

Command	Parameter/Subcommand	Description
no	path	Clear network path to download

Command	Parameter/Subcommand	Description
		error pages
path	*url*	Network path to download HTTP error pages

Example

```
CacheOS#(config)error-pages no 10.25.36.47
    ok
```

event-log

Allows you to configure the event log settings.

Syntax

event-log level | log-size | mail | syslog | when-full

Command	Parameter/Subcommand	Description
level	See table	Event log level
log-size	*megabytes*	Specify event log size
mail	See table	Send e-mail when specific events occur
show		Show running system information
syslog	See table	Specify syslog configuration
when-full	See table	What to do when event log reaches max size

Example

```
CacheOS#(config)event-log
CacheOS#(config event-log)syslog enable
    ok
```

You must replace the default CacheFlow SMTP gateway with your gateway. If you do not have access to an SMTP gateway, you can use the CacheFlow gateway to send event messages to CacheFlow (the CacheFlow SMTP gateway will only send mail to CacheFlow, it will not forward mail to other domains).

level

Syntax

level severe | resource | informational | verbose

Command	Parameter/Subcommand	Description
severe		Log severe errors only
resource		Log above plus resource errors

Command	Parameter/Subcommand	Description
informational		Log above plus information messages
verbose		Log all messages

mail

Syntax

mail add | cacheflow-notify | clear | no | remove | smtp-gateway

Command	Parameter/Subcommand	Description
add	*e-mail address*	Add e-mail recipient to event log notification
cacheflow-notify		Include CacheFlow in event log notification
clear		Remove all e-mail recipients from event notification
no	cacheflow-notify | smtp-gateway	Negative event log commands
remove	*e-mail address*	Remove e-mail recipient from event log notification
show		Show running system information
smtp-gateway	*domain name* | *ip address*	Configure SMTP gateway for notifications

syslog

Syntax

syslog disable | enable | loghost | no

Command	Parameter/Subcommand	Description
disable		Disable syslog notification
enable		Enable syslog notification
loghost	*domain name* | *ip address*	Configure syslog loghost for notifications
no	loghost	Negative syslog commands

when-full

Syntax

when-full overwrite | stop

Command	Parameter/Subcommand	Description
overwrite		When log is full, overwrite oldest events
stop		When log is full, stop logging events

filter-list

Specifies the location of the filter list, as well as options for updating the list on the Content Accelerator.

Syntax

filter-list central-path | local-path | primary-gateway | no | notify | poll-interval | poll-now | subscribe

Command	Parameter/Subcommand	Description
central-path	*URL* or *IP address*	Network path to download central filter list
local-path	*URL* or *IP address*	Network path to download local filter list
primary-gateway	*URL* or *IP address*	Configure primary forwarding gateway
no		Negate filter list parameters
notify		Send email when central filter list changes
poll-interval	*minutes*	Interval to check for central list changes
poll-now		Check if central filter list has changed
subscribe		Update filter list when central list changes

Example

```
CacheOS#(config)filter list poll-interval 30
    ok
```

forwarding

Forwards requests to another cache.

CacheOS can be configured to forward requests to another Web cache or proxy. If a gateway is specified, when an object is requested that is not in the cache, the Web cache will forward the request to the gateway rather than retrieve the object from the network. A primary and alternate gateway can be specified. For detailed information on forwarding, see “Configuring Hierarchical Caches.”

Syntax

forwarding alternate-gateway | no | primary-gateway

Command	Parameter/Subcommand	Description
alternate-gateway		Configure alternate forwarding gateway
no	alternate-gateway | primary-gateway	Remove primary or alternate gateway
primary-gateway		Configure primary forwarding gateway

Example

```
CacheOS#(config)forwarding primary-gateway
CacheOS#(config forwarding primary)address 10.25.36.47
    ok
```

hostname

Sets the Web cache hostname.

You can assign a name to the Web cache. If you have entered a host name for the Web cache in your DNS server, you can use the same name. To set the name, follow the steps outlined below:

Syntax

hostname ***name***

name The name to use for this Web cache.

Example

```
CacheOS#(config)hostname CacheFlow Demo
    ok
```

http-proxy-port

Sets the proxy port for HTTP requests.

The proxy port is the port on which the Web cache listens for HTTP requests. The default port is 8080.

Syntax

http-proxy-port ***port number***

Example

```
CacheOS#(config)http-proxy-port 8084
    ok
```

icp

Sets the ICP configuration options. For information on creating an ICP or advanced forwarding configuration, see "Configuring Hierarchical Caches."

Once you have created the ICP configuration file, place the file on an HTTP server so it can be downloaded to the CacheFlow Web cache. To download the ICP configuration to the Web cache, use the load command.

Syntax

icp no | path

Command	Parameter/Subcommand	Description
no	path	Negate certain ICP settings
path	*url*	Network path to download ICP settings

Example

```
CacheOS#(config)icp path 10.25.36.47/files/icpconfig.txt
    ok
```

inline

The inline command allows users to configure many CacheOS settings from the CacheOS CLI with text from a general CacheOS configuration file. A general CacheOS configuration file can be created and loaded using the **upload configuration** command. You can then edit the file and load settings with the inline command. You can also use the **show configuration** CLI command to get configuration information needed for including into the inline command.

For example, if you previously defined static routes, output from either the **upload configuration** or **show configuration** commands will include an inline command to load the static-route-table.

If you do not have a configuration file with which to work, you can also type input to the inline command directly from the keyboard.

When the inline command is entered, the CLI buffers all input until the string represented by the end-of-input marker is encountered in the input stream. Once the end-of-input marker is detected, the buffered input is sent to the appropriate component of CacheOS for parsing and validation.

Syntax

inline bypass-list | error-pages | filter-list | icp-settings | rip-settings | static-route-table | streaming | wccp-settings ***end-of-input-marker***

Inline Subcommands	Parameters	Description
bypass-list	local | central	Configure bypass list
error-pages		Configure HTTP error pages
filter-list	local | central	Configure filter list

Inline Subcommands	Parameters	Description
icp-settings		Configure ICP settings
rip-settings		Configure RIP configuration
static-route-table		Configure static routes table
streaming		Configure streaming media settings
wccp-settings		Configure WCCP parameters

Example

```
CacheOS/s#(config)inline wccp-settings end-of-input-marker
wccp enable
other wccp configuration commands
end-of-input-marker
  ok
```

When entering input for the inline command, you can correct mistakes on the current line using the backspace key. If you detect a mistake in a line that has already been terminated using the Enter key, you can abort the inline command by typing Ctrl-C. If the mistake is detected after you terminate input to the inline command, enter the inline command again and enter the correct configuration information. The corrected information replaces the information from the last inline command.

The end-of-input marker is an arbitrary string chosen by the user to mark the end of input for the current inline command. The string can be composed of standard characters and numbers, and cannot contain any spaces. Punctuation marks and other symbols are not accepted.

Important Care should be taken to choose an end-of-input string which does not match any string of characters in the configuration information.

interface

Allows you to configure the network interfaces.

The built-in Ethernet adapter is configured for the first time using the setup console. If you want to modify the built-in adapter configuration, or you have multiple adapters, you can configure each one using the command-line interface.

Syntax

interface fast-ethernet

When you enter the interface command, the command-line interface displays the config interface prompt, where you can enter interface configuration commands:

Command	Parameter/Subcommand	Description
fast-ethernet	0 | 1 | 2 | 3	Configure FastEthernet interfaces

Example

```
CacheOS#(config)interface 0
CacheOS#(config interface 0)
```

fast-ethernet

Syntax

fast-ethernet accept-inbound | full-duplex | half-duplex | ip-address | instructions | link-autosense | no | speed | subnet-mask

Command	Parameter/Subcommand	Description
accept-inbound		Allow inbound connections on this interface
full-duplex		Configure interface for full duplex
half-duplex		Configure interface for half duplex
ip-address	*ip address*	Set IP address for interface
instructions	proxy | default-pac | central-pac	Configure client proxy instructions
link-autosense		Interface should autosense speed and duplex
no	accept-inbound | link-autosense	Negative command variations
show		Show running system information
speed	10 | 100	Configure speed for interface
subnet-mask	*mask*	Set subnet mask for interface

ip-default-gateway

Sets the default IP gateway.

Syntax

ip-default-gateway *address*

The IP address of the default gateway to be used by the Web cache.

Example

```
CacheOS#(config)ip-default-gateway 10.25.36.47
   <ip address> [preference group (1-10)] [weight (1-100)]
```

line-vty

Allows you to configure the terminal settings for the command-line interface.

Syntax

line-vty length | telnet | timeout

Command	Parameters	Description
length	*number of lines*	Set number of lines on a screen
telnet	no | transparent	Telnet protocol specific configuration
timeout	*minutes*	Configure line timeout in minutes

Example

```
CacheOS#(config)line-vty
CacheOS#(config line vty)length 60
    ok
```

load

Syntax

load bypass-list | direct-deny-list | error-pages | filter-list | icp-settings | rip-settings | static-route-table | upgrade | wccp-settings

Command	Parameter/Subcommand	Description
bypass-list	central | local	Download new bypass list
direct-deny-list		Download new “direct or deny” list
error-pages		Download new HTTP error pages
filter-list	central | local	Download new filter list
icp-settings		Download new ICP settings
rip-settings		Download new RIP settings
static-route-table		Download new static route table
upgrade		Download new system image
wccp-settings		Download new WCCP settings

Example

```
CacheOS#(config)load bypass-list central
    ok
```

management-port

Sets the IP port to which the Web cache listens for Web console connections.

Syntax

management-port ***port***

port The port to use for HTTP requests. The default port is 8081.

Example

```
CacheOS#(config)management-port 8086
    ok
```

no

Syntax

no ip-default-gateway | socks-machine-id | upgrade-path

Command	Parameter/Subcommand	Description
archive-configuration		Clear archive configuration upload site
ip-default-gateway		Set the default ip gateway to zero
socks-machine-id		Remove the SOCKS machine ID
upgrade-path		Clear the upgrade image download path

Example

```
CacheOS#(config)no socks-machine-id
    ok
```

ntp

Sets NTP parameters.

CacheOS sets UTC time by connecting to an NTP server. CacheOS includes a list of NTP servers available on the Internet. If an NTP server is not available, you can set the time manually using the Web interface.

Syntax

ntp clear | enable | disable | server | no

Command	Parameter/Subcommand	Description
clear		Remove all entries from NTP server list
enable		Enable NTP
disable		Disable NTP
server	*domain name*	Add entry to NTP server list
no	server	Remove entry from NTP server list

Example

```
CacheOS#(config)ntp server clock.tricity.wsu.edu
    ok
```

restart

Sets restart options for the Web cache. To restart the Content Accelerator, enter priveliged mode and enter the command.

Syntax

restart compress | core-image |mode | no

Command	Parameter/Subcommand	Description
compress		Specify compressed core image
core-image	context | full | none	Specify type of core image to write
mode	hardware | software	Configure hard or soft restart
no	compress	Negative restart commands

Example

```
CacheOS#(config)restart mode software
    ok
```

return-to-sender

The return-to-sender feature of CacheOS can help eliminate unnecessary network traffic when the three following conditions are met:

- A Content Accelerator has connections to clients or servers on a different subnet.
- The shortest route to the clients or servers is not via the default gateway.
- There are no static routes or RIP routes defined that apply to the IP addresses of the clients and servers.

Under these conditions, if the return-to-sender feature is enabled, CacheOS remembers the MAC address of the last hop for a packet from the client or server and sends any responses/requests to the MAC address instead of the default gateway.

Under the same conditions, if return-to-sender is disabled, CacheOS sends requests/responses to the default gateway, which then sends the packets to the gateway representing the last hop to the Content Accelerator for the associated connection. This effectively doubles the number of packets transmitted on the LAN compared to when return-to-sender is enabled.

Inbound return-to-sender affects connections initiated to the Content Accelerator by clients. Outbound return-to-sender affects connections initiated by the Content Accelerator to origin servers.

Note Return-to-sender functionality should only be used if static routes cannot be defined for the clients and servers or if routing information for the clients and servers is not available via RIP packets.

Syntax

return-to-sender inbound | outbound

Command	Parameter/Subcommand	Description
inbound	disable | enable	Configure "return to sender" for incoming sessions
outbound	disable | enable	Configure "return to sender" for outgoing sessions

Example

```
CacheOS#(config)return-to-sender inbound enable
    ok
```

rip

Sets the RIP configuration options.

The RIP configuration is defined in a configuration file. To configure RIP, first create a text file of RIP commands and then load the file by using the load command.

Syntax

rip enable | disable | no | path

Command	Parameter/Subcommand	Description
enable		Enable RIP
disable		Disable RIP
no	path	Negate certain RIP settings
path	*url*	Network path to download RIP settings

Example

```
CacheOS#(config)rip path 10.25.36.47/files/rip.txt
    ok
```

rtsp

Sets the RTSP configuration options.

Syntax

rtsp parent-proxy-ip-address | parent-proxy-port | proxy-port

Command	Parameter/Subcommand	Description
parent-proxy-ip-address	***ip*** | *hostname*	Specify parent proxy IP address

Command	Parameter/Subcommand	Description
parent-proxy-port	*port number*	Specify parent proxy port
proxy-port	*port number*	Specify proxy port for RTSP requests

Example

```
CacheOS#(config)rtsp parent-proxy-ip-address 10.25.36.47
    ok
```

security

Sets security options for the Web cache.

Note that the Content Accelerator can limit proxy services to only those users with proper credentials. See the Technical Note available on the CacheFlow website for details that describe how to create and upload a password file. Once the password file is loaded into the Content Accelerator, you can enable Client Authentication.

Syntax

security allowed-access | enforce-console-acl | enable-password | front-panel-pin | no | password | user-name

Command	Parameter/Subcommand	Description
allowed-access	*source ip* | *ip mask*	Add IP address to console access control list
enforce-console-acl		Enforce console access control list
enable-password	*password*	Specify console enable password
front-panel-pin	*PIN*	Specify the PIN for the front panel console; this does not affect modules that allow configuration for the front panel
no	allowed-access | enforce-console-acl | enable-password | password | user-name	Remove username or password
password	*password*	Specify console account password
username	*user name*	Specify console account username

Example

```
CacheOS#(config)security enable-password wallyworld
    ok
```

show

Syntax

show access-log | bypass-list | caching | clock | configuration | content-distribution | content-filter | cpu | direct-deny-list | disk | download-paths | dynamic-bypass | efficiency | event-log | filter-list | forwarding | hostname | http-stats | icp-settings | interface | ip-default-gateway | ip-route-table | ip-stats | ntp | ports | resources | restart | return-to-sender | rip | security | sessions | snmp | socks-machine-id | sources | static-routes | status | telnet-management | terminal | timezones | user-authentication | version | wccp | web-management

Command	Parameter/Subcommand	Description
access-log		Access log settings
bypass-list		Bypass list
caching		Caching settings
clock		Current time
configuration		Current configuration, as different from default
content-distribution		Sizes of objects in cache
content-filter		Content filter settings
cpu		CPU usage
direct-deny-list		Direct or deny list
disk		Disk status and information
dns		DNS servers and name imputing
download-paths		Downloaded configuration paths
dynamic-bypass		Dynamic bypass configuration
efficiency		Efficiency statistics
event-log		Event log setting
filter-list		Current filter list
forwarding		Forwarding settings
hostname		Hostname
http-stats		HTTP statistics
icp-settings		ICP settings
interface		Interface status and configuration
ip-default-gateway		Default IP gateway
ip-route-table		Route table information
ip-stats		TCP/IP statistics
ntp		NTP servers and information
ports		HTTP and console port

Command	Parameter/Subcommand	Description
resources		Allocation of system resources
restart		System restart settings
return-to-sender		"Return to sender" settings
rip		RIP settings
security		Security Parameter/Subcommands
sessions		Information about Telnet connections
snmp		SNMP statistics
socks-machine-id		Machine ID for SOCKS
sources		Source listings for installable lists
static-routes		Static route table information
status		Current system status
telnet-management		Telnet management status
terminal		Terminal configuration Parameter/Subcommands
timezones		Display timezones used
user-authentication		User authentication information
version		System hardware and software status
wccp		WCCP configuration
web-management		Web management status

Example

```
CacheOS#(config)show bypass-list
TCP/IP Bypass List Information
Destination   Mask   Source   Mask   Gateway   Interface   Life(secs)   UseCount
```

snmp

Sets SNMP options for the Web cache.

The CacheFlow Web cache can be viewed using an SNMP management station. The CacheOS supports MIB-2 (RFC 1213).

Syntax

snmp authorize-traps | disable | enable | no | reset-configuration | read-community | sys-contact | sys-location | trap-address | trap-community | write-community

Command	Parameter/Subcommand	Description
authorize-traps		Enable SNMP authorize traps

Command	Parameter/Subcommand	Description
disable		Disable SNMP
enable		Enable SNMP
no	authorize-traps | sys-contact | sys-location | trap-address	Clear certain SNMP parameter/subcommands
reset-configuration		Reset SNMP configuration to default settings
read-community	*password*	Specify read community string
show		Show running system information
sys-contact	*string*	Set "sysContact" MIB variable
sys-location	*string*	Set "sysLocation" MIB variable
trap-address	1 | 2 | 3	Specify IP address to receive traps
trap-community	*password*	Specify trap community string
write-community	*password*	Specify write community string

Example

```
CacheOS#(config)snmp
CacheOS#(config snmp)authorize-traps
    ok
```

socks-machine-id

Sets the machine ID for SOCKS.

If you are using a SOCKS server for the primary or alternate gateway, you must specify the CacheFlow Web cache's machine ID for the Identification (Ident) protocol used by the SOCKS gateway.

Syntax

socks-machine-id ***machine id***

Example

```
CacheOS#(config)socks-machine-id 10.25.36.47
    ok
```

static-routes

Sets the network path to download the static routes configuration file.

The Content Accelerator can be configured to use static routes. To use static routes you must create a routing table and place it on an HTTP server accessible to the Content Accelerator. The routing table is a text file that contains a list of IP addresses, subnet masks, and gateways. When you download a routing table, the table is stored in the device until it is replaced by downloading a new table.

The routing table is a simple text file containing a list of IP addresses, subnet masks, and gateways. A sample routing table is illustrated below:

```
10.63.0.0    255.255.0.0 10.63.158.213
10.64.0.0    255.255.0.0  10.63.158.213
10.65.0.0    255.255.0.0  10.63.158.226
```

When a routing table is loaded, all requested addresses are compared to the list, and routed based on the best match.

Once the routing table is created, place it on an HTTP server so it can be downloaded to the device. To download the routing table to CacheOS, use the load command.

Syntax

static-routes no | path

Command	Parameter/Subcommand	Description
no	path	Negate certain static route settings
path	*url*	Network path to download static routes

Example

```
CacheOS#(config)static-routes path 10.25.36.47/files/routes.txt
    ok
```

streaming

Configures streaming.

Syntax

streaming real-networks

Command	Parameter/Subcommand	Description
real-networks	no | path	Specify Real Networks streaming configuration

Example

```
CacheOS#(config)streaming real-networks 10.25.36.47/files/routes.txt
    ok
```

telnet-management

Enables or disables Telnet management to CLI.

Syntax

telnet-management disable | enable

Example

```
CacheOS#(config)telnet-management enable
```

```
ok
```

timezone

Sets local time zone.

Syntax

timezone *timezone* #

Example

```
CacheOS#(config)timezone 3
  ok
```

transparent-proxy

Enables or disables the transparent proxy.

Syntax

transparent-proxy disable | enable

Command	Parameter/Subcommand	Description
disable		Disable transparent proxy on port 80
enable		Enable transparent proxy on port 80

Example

```
CacheOS#(config)transparent-proxy enable
    ok
```

upgrade path

Network path to download system software.

Syntax

upgrade path url

Example

```
CacheOS#(config)upgrade-path 10.25.36.47
  ok
```

wccp

The Content Accelerator can be configured to participate in a WCCP (Web Cache Control Protocol) scheme, where a WCCP-capable router collaborates with a set of WCCP-configured Content Accelerators to service requests. WCCP is a Cisco-developed protocol. For more information about WCCP, refer to the *Appendix B: WCCP (Web Cache Control Protocol)*.

Once you have created the WCCP configuration file, place the file on an HTTP server so it can be downloaded to the CacheFlow Web cache. To download the WCCP configuration to the Web cache, use the load command.

Syntax

wccp enable | disable | no | path

Command	Parameter/Subcommand	Description
enable		Enable WCCP
disable		Disable WCCP
no	path	Negate certain WCCP settings
path	*url*	Network path to download WCCP settings

Example

```
CacheOS#(config)wccp path 10.25.36.47/files/wccp.txt
    ok
```

web-management

Enables and disables the Web interface management console. When web-management is disabled, you can still access CacheOS homepage and online documentation. Only the management and statistics applications are disabled.

Syntax

web-management disable | enable

Command	Parameter/Subcommand	Description
disable		Disable WEB management console
enable		Enable WEB management console

Example

```
CacheOS#(config)web-management disable
    ok
```

disable

Turns off privileged commands.

Syntax

disable

The disable command does not have any parameters or subcommands.

Example

```
CacheOS#disable
```

display

Displays a text-based URL.

Syntax

display ***url***

The display command has the parameter url.

Example

```
CacheOS#display www.company1.com
<!DOCTYPE HTML PUBLIC "-//IETF//DTD HTML 2.0//EN">
<HTML><HEAD>
<TITLE>302 Found</TITLE>
</HEAD><BODY>
<H1>Found</H1>
The document has moved <A HREF="http://lc2.law5.company1.passport.com/cgi-bin/log
in">here</A>.<P>
</BODY></HTML>
```

enable

Activates privileged command mode. When in priveliged mode, a pound sign (#) is displayed as part of the command prompt.

Syntax

enable

The enable command does not have any parameters or subcommands.

Example

```
CacheOS#enable
```

```
ok
```

exit

Exits from configuration mode to priveliged mode, from privileged mode to standard mode, and from standard mode closes the command-line interface session.

Syntax

exit

The exit command does not have any parameters or subcommands.

Example

```
CacheOS#exit
```

help

Displays help information.

Syntax

help

The help command does not have any parameters or subcommands.

Example

```
CacheOS#help
Help may be requested at any point in a command
by typing a question mark '?'.
1. For a list of available commands, enter '?' at
   the prompt.
2. For a list of arguments applicable to a command,
   precede the '?' with a space (e.g. 'show ?')
3. For help completing a command, do not precede
   the '?' with a space (e.g. 'sh?')
```

kill

Terminates a telnet session.

Syntax

kill *session* #

The kill command has the parameter session #.

Example

```
CacheOS#kill 123
  ok
```

load

Loads installable lists or system upgrade images.

Syntax

load bypass-list | direct-deny-list | error-pages | filter-list | icp-settings | rip-settings | static-route-table | streaming | upgrade | wccp-settings

Command	Parameter/Subcommand	Description
akamizer-settings	*eof marker*	Install Akamizer settings from console input
bypass-list	central | local	Install bypass list from console input
direct-deny-list	*eof marker*	Install "direct or deny" list from console input
error-pages	*eof marker*	Install HTTP error pages from console input
filter-list	central | local	Install filter list from console input
icp-settings	*eof marker*	Install ICP settings from console input
rip-settings	*eof marker*	Install RIP settings from console input
static-route-table	*eof marker*	Install static route table from console input
streaming	*eof marker*	Install streaming configuration from console input
wccp-settings	*eof marker*	Install WCCP settings from console input

Example

```
CacheOS#load akamizer-settings #123
  ok
```

offline-disk

Takes a disk offline.

Syntax

offline-disk ***disk number***

The offline-disk command has the parameter disk number.

Example

```
CacheOS#offline-disk 3
  ok
```

ping

Sends echo messages.

Syntax

ping ***IP*** **|** ***hostname***

Example

```
CacheOS#ping 10.25.36.47
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 10.25.36.47, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 0/0/0 ms
Number of duplicate packets received = 0
```

purge-dns-cache

Clears the DNS cache.

Syntax

purge-dns-cache

The purge-dns-cache command does not have any parameters or subcommands.

Example

```
CacheOS#purge-dns-cache
  ok
```

restart

Restarts the system.

Syntax

restart regular | upgrade

Example

```
CacheOS#restart regular
  ok
```

restore-defaults

Restores system to default configuration.

Syntax

restore-defaults

The restore-defaults command does not have any parameters or subcommands.

Example

```
CacheOS#restore-defaults
  ok
```

show

Shows running system information.

Syntax

show

Command	Parameter/Subcommand	Description
access-log		Access log settings
archive-configuration		Archive configuration settings
arp-table		ARP information
authentication		Authentication settings
bypass-list		Bypass list
caching		Caching settings
clock		Current time
configuration		Current configuration, as different from default
content-distribution		Sizes of objects in cache
content-filter		Content filter settings
cpu		CPU usage
direct-deny-list		Direct or deny list
disk		Disk status and information
dns		DNS servers and name imputing
download-paths		Downloaded configuration paths

Command	Parameter/Subcommand	Description
dynamic-bypass		Dynamic bypass configuration
efficiency		Efficiency statistics
event-log		Event log setting
filter-list		Current filter list
forwarding		Forwarding settings
hostname		Hostname
http-stats		HTTP statistics
icp-settings		ICP settings
interface		Interface status and configuration
ip-default-gateway		Default IP gateway
ip-route-table		Route table information
ip-stats		TCP/IP statistics
ntp		NTP servers and information
ports		HTTP and console port
resources		Allocation of system resources
return-to-sender		"Return to sender" settings
rip		RIP settings
rtsp		RTSP settings
security		Security parameters
sessions		Information about Telnet connections
snmp		SNMP statistics
socks-machine-id		Machine ID for SOCKS
sources		Source listings for installable lists
static-routes		Static route table information
status		Current system status
streaming		Streaming information
telnet-management		Telnet management status
terminal		Terminal configuration parameters
timezones		Displays timezones used
user-authentication		User authentication information
version		System hardware and software status
wccp		WCCP configuration
web-management		Web management status

Using the Show Command

To use the show command, enter the parameter you want to display. For example, to display the current DNS configuration settings, enter the following show command:

```
CacheOS#show dns
DNS servers:
  10.25.36.47
Imputed names:
```

Commonly used show commands are described below. For syntax help on commands not included here, use the command-line interface help.

show access-log

Displays the current access-log settings.

Syntax

show access-log

The show access-log command does not have any parameters or subcommands.

Example

```
Settings:
  Access logging is enabled
  Format: squid log format
  Filename prefix:
Primary upload site:
  Host:
  Path:
  Username:
Alternate upload site:
  Host:
  Path:
  Username:
Upload schedule:
  Upload access log daily at 02:00
Access log size:
  Log may occupy 2 percent of total disk space
  If log reaches maximum size, upload ahead of schedule
```

show archive-configuration

Displays the current archive configuration settings for the cache.

Syntax

show archive-configuration

The show archive-configuration command does not have any parameters or subcommands.

Example

```
CacheOS#show archive-configuration
Archive configuration
  Host:
  Path:
  Username:
  Password:
```

show arp-table

Displays the current ARP information.

Syntax

show arp-table

The show arp-table command does not have any parameters or subcommands.

Example

```
CacheOS#show arp-table
TCP/IP ARP Information
IP Address       MAC Address        Interface
10.25.220.165   00:C0:4F:38:CA:CC  Adapter0
10.25.220.249   00:A0:C9:5E:14:CB  Adapter0
10.25.221.16    00:C0:4F:2E:B6:9F  Adapter0
10.25.221.35    00:C0:4F:28:68:35  Adapter0
```

show authentication

Displays the current archive configuration settings for the cache.

Syntax

show authentication ldap | radius | settings

Example

```
CacheOS#show authentication settings
Authentication:
  Authentication protocol: none
  Verify administrators:   disabled
  Verify proxy users:      disabled
```

show bypass-list

Displays the current bypass list settings for the cache.

Syntax

show bypass-list

The show bypass-list command does not have any parameters or subcommands.

Example

```
CacheOS#show bypass-list
TCP/IP Bypass List Information
Destination   Mask   Source   Mask   Gateway   Interface   Life(secs)   UseCount
```

show caching

Displays the current configuration settings for the cache.

Syntax

show caching

The show caching command does not have any parameters or subcommands.

Example

```
CacheOS#show caching
Refresh:
  Desired access freshness is 99.0%
  Let CacheOS manage refresh bandwidth
Policies:
  Do not cache objects larger than 50 megabytes
  Cache negative responses for 0 minutes
  Let CacheOS manage freshness
FTP caching:
  Caching FTP objects is enabled
  Do not cache FTP objects larger than 50 megabytes
  FTP objects with last modified date, cached for 10% of last modified time
  FTP objects without last modified date, initially cached for 24 hour
```

show clock

Displays the current time settings for the cache.

Syntax

show clock

The show clock command does not have any parameters or subcommands.

Example

```
CacheOS#show clock
   Thu, 24 Feb 2000 23:12:08 UTC
```

show configuration

Displays the current CacheOS configuration as it differs from the default settings. You can capture the output of this command to a text file for future reference, or to restore the configuration by using the configure command.

Syntax

show configuration

The show configuration command does not have any parameters or subcommands.

You can use the show configuration command to display the current configuration and then save the output to a file that can later be used to restore the configuration. When you enter the configure command from enable mode, the command-line interface prompts you to configure from a terminal or the network. When you enter **network**, you can specify the URL where the configuration file is located. The output of the show configuration file can be used with the configure command to restore the configuration.

Example

```
CacheOS#show configuration
interface 0
ip-address 10.25.36.47
subnet-mask 255.255.0.0
exit
ip-default-gateway 10.25.0.1 1 100
dns clear server
dns server 10.25.0.2
!
authentication ldap
primary-server ip 10.25.0.1
alternate-server ip 10.25.0.2
alternate-server port 366
admin-attribute value read-only
cache-duration 0
exit
!
authentication radius
server-retry 15
cache-duration 0
exit
!
access-log
enable
upload hourly 23
threshold 10
exit
!
```

```
caching
negative-response 10
refresh no automatic
exit
!
reveal http
http persistent server
!
snmp
enable
exit
!
hostname 10.25.36.47 - CacheFlow 5000
security allowed-access 10.25.0.1 255.255.255.255
!
upgrade-path 10.25.0.2/cf_1000/v3/wdir/100.chk
```

show content-distribution

Displays the number of objects in the cache by size.

Syntax

show content-distribution

The show-content-distribution command does not have any parameters or subcommands.

Example

```
CacheOS#show content-distribution
Object distribution, objects smaller than 10Kb
  0Kb - 1Kb: 29702
.
.
.
Object distribution, objects between 10Kb and 100Kb
  10Kb - 20Kb: 13925
  20Kb - 30Kb: 2931
  30Kb - 40Kb: 1140
.
.
.
Object distribution, objects between 100Kb and 1Mb
  100Kb - 200Kb: 186
  200Kb - 300Kb: 42
  300Kb - 400Kb: 24
```

```
.
.
.
Object distribution, objects larger than 1Mb
  1Mb - 2Mb: 12
  2Mb - 3Mb: 14
  3Mb - 4Mb: 16
  4Mb - 5Mb: 19
  5Mb - 6Mb: 1
  6Mb - 7Mb: 3
.
.
.
```

show cpu

Displays CPU usage in the cache.

Syntax

show cpu

The show cpu command does not have any parameters or subcommands.

Example

```
CacheOS#show cpu
Current cpu usage: 0.0 percent
```

show direct-deny-list

Displays the direct or deny list by the Web cache.

Syntax

show direct-deny-list

The show direct-deny-list command does not have any parameters or subcommands.

Example

```
CacheOS#show direct-deny-list
CacheFlow Direct Deny List
;Subnet             Mask                Command
The list is empty
```

show disk

Displays disk status and information.

Syntax

show disk disk number | all

Example

```
CacheOS#show disk 3
Disk in slot 3
  Vendor: SEAGATE
  Product: ST39102LC
  Revision: 0006
  Serial number: LJU33241
  Capacity: 9105018368 bytes
  Status: present
```

show dns

Displays DNS servers and name imputing information.

Syntax

show dns

The show dns command does not have any parameters or subcommands.

Example

```
CacheOS#show dns
Primary DNS servers:
  10.253.220.249
Alternate DNS servers:
Imputed names:
```

show download-paths

Displays the downloaded configuration path.

Syntax

show download-paths

The show download-paths command does not have any parameters or subcommands.

Example

```
CacheOS#show download-paths
Filter list
  Local:
  Central: www.cacheflow.com/support/subscriptions/CentralFilterList.txt
    Update when changed: no
    Notify when changed: yes
Bypass list
```

```
  Local:
  Central: www.cacheflow.com/support/subscriptions/CentralBypassList.txt
    Update when changed: no
    Notify when changed: yes
Direct Deny list:
HTTP error pages:
ICP settings:
Real Networks settings:
RIP settings:
Static route table:
Upgrade image:           10.25.36.47/Builds/flipper.12581/cf_5000_ali.chk
WCCP settings:
```

show dynamic-bypass

Displays dynamic bypass configuration.

Syntax

show dynamic-bypass

The show dynamic-bypass command does not have any parameters or subcommands.

Example

```
CacheOS#show dynamic-bypass
Dynamic bypass: disabled
  Non-HTTP trigger: disabled
  HTTP 400 trigger: disabled
  HTTP 401 trigger: disabled
  HTTP 403 trigger: disabled
  HTTP 405 trigger: disabled
  HTTP 406 trigger: disabled
  HTTP 500 trigger: disabled
```

show efficiency

Displays statistics on objects handled by the Web cache.

Syntax

show efficiency

The show efficiency command does not have any parameters or subcommands.

Example

```
CacheOS#show efficiency
Efficiency (by objects):
  Served from cache:       81256 (46%)
```

```
  Retrieved from source:   16262 (9%)
  Noncacheable:            76737 (43%)
  Verified fresh:          2121 (1%)
Efficiency (by bytes):
  Served from cache:       327987315 (56%)
  Retrieved from source:   187995105 (32%)
  Noncacheable:            57571087 (9%)
  Verified fresh:          10832879 (1%)
Noncacheable object breakdown:
  Pragma no-cache:         15142
  Password provided:       24139
  Data in request:         0
  Not a GET request:       1231
  Cookie in response:      7627
  Password required:       0
  Negative response:       21271
  Client unique CGI:       7327
Access pattern:
  Accessed from RAM:       502337536 (92%)
  Accessed from disk:      41197568 (7%)
```

show event-log

Displays the event-log setting.

Syntax

show event log

The show event-log command does not have any parameters or subcommands.

Example

```
CacheOS#show event-log
Settings:
  Event level: severe + resource + informational
  Event log size: 1 megabytes
  If log reaches maximum size, overwrite earlier events
  Send events to CacheFlow
  Syslog loghost: <none>
  Syslog notification: disabled
Event recipients:
  heartbeat@mail.heartbeat.cacheflow.com
SMTP gateway:
  mail.heartbeat.cacheflow.com
```

show filter-list

Displays the current filter list.

Syntax

show filter-list

The show filter-list command does not have any parameters or subcommands.

Example

```
CacheOS#show filter-list
```

show forwarding

Displays the forwarding settings.

Syntax

show forwarding

The command show forwarding command does not have any parameters or subcommands.

Example

```
CacheOS#show forwarding
Primary gateway
  Domain name:
  Port:        0
  Socks:       no
Secondary gateway
  Domain name:
  Port:        0
  Socks:       no
```

show hostname

Displays the hostname.

Syntax

show hostname

The show hostname command does not have any parameters or subcommands.

Example

```
CacheOS#show hostname
 Hostname: 10.25.36.47 - CacheFlow 5000
```

show http-stats

Displays HTTP statistics.

Syntax

show http-stats

The show http-stats command does not have any parameters or subcommands.

Example

```
CacheOS#show http-stats
CacheFlow Inc, HTTP Statistics
HTTP Statistics version  4
There have been 0 connections accepted by HTTP.
Persistent connections were reused for an additional 0 requests.There are curren
tly 0 active client connections.
The high-water-mark of concurrent client connection is 0
```

show icp-settings

Displays ICP settings.

Syntax

show icp-settings

The show icp-settings command does not have any parameters or subcommands.

Example

```
CacheOS#show icp-settings
# Current ICP Configuration
# Written on Wed, 26 Jan 2000 22:43:57 UTC
# ICP Port to listen on (0 to disable ICP)
icp_port 0
# Neighbour timeout (seconds)
neighbor_timeout 2
# ICP and HTTP failure counts
icp_failcount 20
http_failcount 5
# Host failure/recovery notification flags
host_recover_notify off
host_fail_notify off
# 0 neighbours defined, 32 maximum
# ICP host configuration
# icp_host hostname peertype http_port icp_port [options]
# Forwarding host configuration
# fwd_host hostname http_port [options]
# 0 groups defined, 16 maximum
# Forwarding host URL regex configuration
# fwd_host_url_regex targetname url_regex
```

```
# targetname of 'deny' means deny access
# targetname of 'direct' means no forwarding
# 0 forwarding host URL regexes defined, 256 maximum
# Forwarding host domain configuration
# fwd_host_domain targetname domainname
# targetname of 'deny' means deny access
# targetname of 'direct' means no forwarding
# 0 forwarding host domains defined, 256 maximum
# Forwarding host ip configuration
# fwd_host_ip targetname IP[/netmask]
# targetname of 'deny' means deny access
# targetname of 'direct' means no forwarding
# 0 IPs defined, 256 maximum
# ICP access domain configuration
```

show interface

Displays the status and configuration of the network interfaces.

Syntax

show interface # | all

Example

```
CacheOS#show interface 0
Ethernet interface 0
  Internet address: 10.25.36.47
  Internet subnet: 255.255.255.0
  Interface autosensed full duplex, 100 Mb/s network
Inbound connections will be accepted by this interface
```

show ip-default-gateway

Displays the default IP gateway.

Syntax

show ip-default-gateway

The show ip-default-gateway command does not have any parameters or subcommands.

Example

```
CacheOS#show ip-default-gateway
Default IP gateway: 10.25.36.47
```

show ip-route-table

Displays route table information.

Syntax

show ip-route-table

The show ip-route-table command does not have any parameters or subcommands.

Example

```
CacheOS#show ip-route-table
10.25.36.47   10.253.0.1    UGHW     0    4   27   0  0  Adapter0
10.25.36.48   10.253.0.1    UGHW     0    2   26   0  0  Adapter0
10.25.36.49   10.253.0.1    UGHW     0    4   27   0  0  Adapter0
10.25.36.50   10.253.0.1    UGHW     0    2   25   0  0  Adapter0
10.25.36.51   10.253.0.1    UGHW     0   16   25   0  0  Adapter0
10.25.36.52   10.253.0.1    UGHW     0    2   25   0  0  Adapter0
10.25.36.53   10.253.0.1    UGHW     0    2   25   0  0  Adapter0
10.25.36.54   10.253.0.1    UGHW     0    2   25   0  0  Adapter0
10.25.36.55   10.253.0.1    UGHW     0    2   25   0  0  Adapter0
10.25.36.56   10.253.0.1    UGHW     0   90    6   0  0  Adapter0
10.25.36.57   10.253.0.1    UGHW     0    2   22   0  0  Adapter0
10.25.36.58   10.253.0.1    UGHW     0    2   27   0  0  Adapter0
```

show ip-stats

Displays TCP/IP statistics.

Syntax

show ip-stats

The show ip-stats command does not have any parameters or subcommands.

Example

```
CacheOS#show ip-stats
TCP/IP General Statistics
Entries in TCP queue: 135
Maximum entries in TCP queue: 217
Entries in TCP time wait queue: 65
Maximum entries in time wait queue: 158
Number of time wait allocation failures: 0
Entries in UDP queue: 3
Memory Statistics
Bytes in use 5,034,352
Maximum bytes in use 5,144,072
Bytes assigned 7,911,232
Maximum bytes assigned 7,911,232
Number of failed allocations 0
Malloc bytes used 1,888,624
```

```
Malloc max bytes used 1,998,344
Malloc buffers assigned 634
Large buffer used bytes 147,952
Large buffer max used bytes 186,760
Large buffer assigned bytes 2,128,064
Net-write buffers used 0
Max net-write buffers used 11
Time wait bytes used 3,145,728
Interface Statistics
Interface:  Adapter0
Packets received 32,002
Total number of bytes received 5,056,134
Packets sent 21,886
Total number of bytes sent 3,641,667
Input errors 0
Output errors 0
Collisions 0
Packets received via multicast 19,150
Packets sent via multicast 141
Dropped on input 0
Destined for unsupported protocol 0
Number of receive lockups 0
Send queue length 0
Dropped on output 0
Number of times interface was down 0
No route to destination 0
No route to gateway 0
Route rejected 0
Bad protocol family 0
Can't add ethernet header 0
Packets sent using return-to-sender 0
Transmit maximum collisions 0
Transmit late collisions 0
Transmit underruns 0
Transmit lost CRS 0
Transmit deffered 0
Transmit single collisions 0
Transmit multiple collisions 0
Receive CRC errors 0
Receive alignment errors 0
Receive resource errors 0
Receive overrun errors 0
Receive CDT errors 0
```

```
Receive short frames 0
Changes in interface health 0
Active TCP connections 4
Link is up
```

show ntp

Displays NTP servers and related information.

Syntax

show ntp

The show ntp command does not have any parameters or subcommands.

Example

```
CacheOS#show ntp
NTP is enabled
NTP servers:
  ntp.cacheflow.com
  ntp2.cacheflow.com
```

show ports

Displays HTTP and console port information.

Syntax

show ports

The show ports command does not have any parameters or subcommands.

Example

```
CacheOS#show ports
 HTTP:        8080
 Transparent: 80
 Console:     8081
```

show resources

Displays allocation of system resources.

Syntax

show resources

The show resources command does not have any parameters or subcommands.

Example

```
CacheOS#show resources
Disk resources:
```

```
  Available to cache:   25026998272
  In use by cache:      672813056
  In use by system:     1612600320
  In use by access log: 0
  Total disk installed: 27312411648
Memory resources:
  In use by cache:      439975936
  In use by system:     88979584
  In use by network:    7915392
  Total RAM installed:  536870912
```

show restart

Displays system restart settings.

Syntax

show restart

The show restart command does not have any parameters or subcommands.

Example

```
CacheOS#show restart
Restart settings
  Restart:     software only
  Core image:  context only
  Compression: disabled
```

show return-to-sender

Displays "Return to sender" settings.

Syntax

show return-to-sender

The show return-to-sender command does not have any parameters or subcommands.

Example

```
CacheOS#show return-to-sender
Return to sender:
  Incoming sessions: disabled
  Outgoing sessions: disabled
```

show rip

Displays RIP settings.

Syntax

show rip routes | statistics

Example

```
CacheOS#show rip routes
Destination        Gateway           Metric Interface
```

show rtsp

Displays RTSP settings.

Syntax

show rtsp

The show rtsp command does not have any parameters or subcommands.

Example

```
CacheOS#show rtsp
Proxy port:             1091
Parent proxy address:   0.0.0.0
Parent proxy port:      1091
```

show sessions

Displays information about Telnet connections.

Syntax

show sessions

The show sessions command does not have any parameters or subcommands.

Example

```
CacheOS#show sessions
Sessions:
  #   state  type    start                     elapsed
  01  NORML  serial  24 Feb 2000 21:55:21 UTC  03:30:31
  02  NORML  telnet  25 Feb 2000 00:30:00 UTC  00:55:52
  03* NORML  telnet  25 Feb 2000 01:25:46 UTC  00:00:06
  04  IDLE
  05  IDLE
```

show snmp

Displays SNMP statistics.

Syntax

show snmp

The show snmp command does not have any parameters or subcommands.

Example

```
CacheOS#show snmp
General info:
  SNMP is disabled
MIB variables:
  sysContact:       Rita
  sysLocation:
Traps:
  Trap address 1:
  Trap address 2:
  Trap address 3:
  Authorization traps:  disabled
```

show socks-machine-id

Displays machine ID for SOCKS.

Syntax

show socks-machine-id

The show socks-machine-id command does not have any parameters or subcommands.

Example

```
CacheOS#show socks-machine-id
 SOCKS machine id: 10.25.36.47
```

show sources

Displays source listings for installable lists.

Syntax

show sources akamizer-settings | bypass-list | direct-deny-list | filter-list | icp-settings | rip-settings | static-route-table | streaming | wccp-settings

Example

```
CacheOS#show sources akamizer-settings
; Empty Akamizer configuration object
```

show static-routes

Displays static route table information.

Syntax

show static-routes

The show static-routes command has no parameter or subcommands.

Example

```
CacheOS#show static-routes
TCP/IP Static Route Information
Destination      Mask            Gateway          Interface
default          0.0.0.0         10.253.0.1       Adapter0
```

show status

Displays current system status.

Syntax

show status

The show status command does not have any parameters or subcommands.

Example

```
CacheOS#show status
Configuration:
  Disks installed:         6
  Memory installed:        512 megabytes
  CPUs installed:          1
  Software version:        2.9.99
  Release id:              99999
  Machine id:              00A0C960288F
General status:
  Last access log upload:  log has never been uploaded
  Current access log size: 0
  System started:          Thu, 24 Feb 2000 02:10:40 UTC
  CPU utilization:         0%
```

show telnet-management

Displays current telnet-management status.

Syntax

show telnet-management

The show telnet-management command does not have any parameters or subcommands.

Example

```
CacheOS#show telnet-management
Telnet management: enabled
```

show terminal

Displays terminal configuration parameters and subcommands.

Syntax

show terminal

The show terminal command does not have any parameters or subcommands.

Example

```
CacheOS#show terminal
Terminal characteristics:
  Line timeout:       5 minutes
  Line width:         80 characters
  Screen length:      24 lines
  Telnet transparent: no
```

show timezone

Displays local timezone being currently used.

Syntax

show timezone

The show timezone command does not have any parameters or subcommands.

Example

```
CacheOS#show timezone
Current timezone:
  21 (UTC) [UTC] UTC Standard Time
Supported timezones:
   0 (UTC-12:00) [MHT] Dateline Standard Time
   1 (UTC-11:00) [SST] Samoa Standard Time
   2 (UTC-10:00) [HST] Hawaiian Standard Time
   3 (UTC-09:00) [AKST,ADKT] Alaskan Standard Time
   4 (UTC-08:00) [PST,PDT] Pacific Standard Time
   5 (UTC-07:00) [MST,MDT] Mountain Standard Time
   6 (UTC-07:00) [MST] US Mountain Standard Time
   7 (UTC-06:00) [CST] Canada Central Standard Time
   8 (UTC-06:00) [CST,CDT] Central Standard Time
   9 (UTC-06:00) [MST] Mexico Standard Time
  10 (UTC-05:00) [EST,EDT] Eastern Standard Time
  11 (UTC-05:00) [PET] SA Pacific Standard Time
  12 (UTC-05:00) [EST] US Eastern Standard Time
  13 (UTC-04:00) [AST,ADT] Atlantic Standard Time
```

```
14 (UTC-04:00) [VET] SA Western Standard Time
15 (UTC-03:30) [NST,NDT] Newfoundland Standard Time
16 (UTC-03:00) [BRT,BRST] E. South America Standard Time
17 (UTC-03:00) [ART] SA Eastern Standard Time
18 (UTC-02:00) [MAST,MADT] Mid-Atlantic Standard Time
19 (UTC-01:00) [AZOT,AZOST] Azores Standard Time
20 (UTC+00:00) [GMT,BST] Greenwich Mean Time
21 (UTC) [UTC] UTC Standard Time
22 (UTC+01:00) [CET,CEST] Central Europe Time
23 (UTC+02:00) [EET,EEST] E. Europe Time
24 (UTC+02:00) [EET,EEST] Egypt Standard Time
25 (UTC+02:00) [EET,EEST] FLE Standard Time
26 (UTC+02:00) [EET,EEST] GFT Standard Time
27 (UTC+02:00) [EET] Israel Standard Time
28 (UTC+02:00) [SAST] South Africa Standard Time
29 (UTC+03:00) [EAT] E. Africa Time
30 (UTC+03:00) [MSK,MSD] Russian Standard Time
31 (UTC+03:00) [AST] Saudi Arabia Standard Time
32 (UTC+03:30) [IRT,IRST] Iran Standard Time
33 (UTC+04:00) [GST] Arabian Standard Time
34 (UTC+04:00) [GET] Caucasus Standard Time
35 (UTC+04:30) [AFT] Afghanistan Standard Time
36 (UTC+05:00) [YEKT,YEKST] Yekaterinburg Standard Time
37 (UTC+05:00) [PKT] West Asia Standard Time
38 (UTC+05:30) [IST] India Standard Time
39 (UTC+06:00) [BDT] Central Asia Standard Time
40 (UTC+06:00) [LKT] Sri Lanka Standard Time
41 (UTC+07:00) [ICT] Bangkok Standard Time
42 (UTC+08:00) [CST] China Standard Time
43 (UTC+08:00) [SGT] Singapore Standard Time
44 (UTC+08:00) [CST] Taipei Standard Time
45 (UTC+08:00) [WST] W. Australia Standard Time
46 (UTC+09:00) [KST] Korea Standard Time
47 (UTC+09:00) [JST] Tokyo Standard Time
48 (UTC+09:00) [YAKT,YAKST] Yakutsk Standard Time
49 (UTC+09:30) [CST] AUS Central Standard Time
50 (UTC+09:30) [CST,CST] Cen. Australia Standard Time
51 (UTC+10:00) [EST] E. Australia Standard Time
52 (UTC+10:00) [EST,EST] Sydney Standard Time
53 (UTC+10:00) [EST,EST] Tasmania Standard Time
54 (UTC+10:00) [VLAT,VLAST] Vladivostok Standard Time
55 (UTC+10:00) [GST] West Pacific Standard Time
56 (UTC+11:00) [NCT] Central Pacific Standard Time
```

```
57 (UTC+12:00) [MHT] Fiji Standard Time
58 (UTC+12:00) [NZT,NZST] New Zealand Standard Time
```

show user-authentication

Displays user authentication information.

Syntax

show user-authentication

The show user-authentication command does not have any parameters or subcommands.

Example

```
CacheOS#show user-authentication
Title: User Authentication General Statistics
Version: 1.0
Number of users in the credential cache: 0
Number of cache buffers used to store cache: 9
Number of authentication requests processed: 0
Number of authentication requests rejected: 0
Length of longest chain in the hash table: 0
```

show version

Displays system hardware and software status.

Syntax

show version

The show version command does not have any parameters or subcommands.

Example

```
CacheOS#show version
Version: 3.00
Release id: 99999
PIC: 1.001
```

show wccp

Displays WCCP configuration settings.

The show wccp command does not have any parameters or subcommands.

Syntax

show wccp configuration | statistics

Example

```
CacheOS#show wccp configuration
```

```
; WCCP Settings
; CacheOS WCCP version 1.2
wccp disable
```

show web-management

Displays Web management status.

Syntax

show web-management

The show web-management command does not have any parameters or subcommands.

Example

```
CacheOS#show web-management
  Web management: enabled
```

static-route

Replaced by the command temporary-route.

temporary-route

Manages temporary route entries.

Syntax

temporary-route add | delete <destination_address> <net_mask> <gateway_address>

Example

```
CacheOS#temporary-route delete 10.25.36.47
  ok
```

test

Tests subsystems.

Syntax

test http

Command	Parameter/Subcommand	Description
get	*url*	Get HTTP object
loopback		Perform loopback test

Example

```
CacheOS#test http get 10.25.36.47
HTTP response code: HTTP/1.0 503 Service Unavailable
Throughput rate is non-deterministic
HTTP get test passed
```

traceroute

Traces route to destination.

Syntax

traceroute *IP* | *hostname*

Example

```
CacheOS#traceroute 10.25.36.47
Type escape sequence to abort.
Executing HTTP get test
HTTP response code: HTTP/1.0 503 Service Unavailable
Throughput rate is non-deterministic
HTTP get test passed
10.25.36.47#traceroute 10.25.36.47

Type escape sequence to abort.
Tracing the route to 10.25.36.47
1 10.25.36.47 212 0 0 0
```

upload

Uploads access log or running configuration.

Syntax

upload access-log | configuration

Example

```
CacheOS#upload configuration 10.25.36.47
 ok
```

Index