Equalizer
Installation and Administration Guide

Version 7.2.4

July 2008

Coyote Point Systems, Inc.
675 North First Street
Suite 975
San Jose, California 95112

THE SPECIFICATIONS AND INFORMATION REGARDING THE PRODUCTS IN THIS MANUAL ARE SUBJECT TO CHANGE WITHOUT NOTICE. ALL STATEMENTS, INFORMATION, AND RECOMMENDATIONS IN THIS MANUAL ARE BELIEVED TO BE ACCURATE BUT ARE PRESENTED WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED. USERS MUST TAKE FULL RESPONSIBILITY FOR THEIR APPLICATION OF ANY PRODUCTS.

See Appendix F for complete License and Warranty information for this product.

Contents

Preface

The *Equalizer Installation and Administration Guide* is intended for people who are installing, configuring, or administering Equalizer™ systems.

In This Guide

This guide contains the following chapters and appendices:

- Chapter 1, *Overview*, contains detailed descriptions of Equalizer concepts and terminology. This chapter includes information to help you plan your Equalizer configuration. If you are setting up Equalizer for the first time, be sure to read the *Overview* chapter before attempting to install and configure your system.
- Chapter 2, *Installing Equalizer*, provides comprehensive instructions for installing Equalizer.
- Chapter 3, *Configuring Equalizer Hardware*, instructs you in setting up Equalizer to work with your networks and servers.
- Chapter 4, *Accessing Browser Controls*, discusses how to use Equalizer's HTML-based administration interface to check Equalizer status and to change settings within Equalizer.
- Chapter 5, *Configuring Equalizer Operation*, tells you how to configure through the Equalizer Administration Interface, including setting up a failover configuration.
- Chapter 6, *Monitoring Equalizer Operation*, describes how to view information, statistics, and graphical displays about Equalizer's operation.
- Chapter 7, *Administering Virtual Clusters*, tells you how to add and remove virtual clusters and servers, changing load balancing options, and shutting down servers.
- Chapter 8, *Working with Match Rules*, shows you to create match rules that distribute requests based on a request's attributes.
- Chapter 9, *Administering Geographic Clusters*, shows you how to use the Envoy add-in to add and remove geographic clusters and sites and change geographic load balancing and targeting options.
- Appendix A, *Using Server Agents,* describes how to develop custom server agents.
- Appendix B, *Using Reserved IP Addresses,* describes how to configure Equalizer to distribute requests to servers assigned IP addresses on reserved, non-routable networks.
- Appendix C, *Regular Expression Format*, discusses Equalizer's regular expressions, components, formats, and usage.
- Appendix D, *HTTPS Cluster Certificates,* shows you how to obtain and install certificates for HTTPS clusters.
- Appendix E, *Troubleshooting,* helps you to diagnose problems with Equalizer.

- Appendix F, *License and Warranty*, contains the complete License and Warranty information.
- Appendix G, *Additional Requirements*, lists additional hardware related requirements for Equalizer installations.
- The *Glossary* defines the technology-specific terms used throughout this book.
- Use the *Index* to help find specific information in this guide.

Typographical Conventions

The following typographical conventions appear throughout this guide:

Italics indicates the introduction of new terms, is used to emphasize text, and indicates variables.

Boldface text highlights field, key, or button names in instructions.

Courier text denotes commands, file names, directory names, keywords, and syntax from text.

1. Numbered lists show steps that you must complete in the numbered order.

- Bulleted lists identify items that you can address in any order.

Note – Highlights important information and special considerations.

Caution – Warns when an action could result in loss of data or damage to your equipment.

⚠ Emphasizes information critical to Equalizer operation.

Where to Go for More Help

Customer Support contact information is available from the **Support** link on our main web page at **http://www.coyotepoint.com**. Register today for access to the **Coyote Point Support Portal** at:

http://support.coyotepoint.com

Registration provides you with a login so you can access these benefits:

- **Support FAQ**: answers to our customer's most common questions.
- **Moderated Customer Support Forum**: ask questions and get answers from our support staff and other Equalizer users.
- **Software upgrades and security patches**: access to the latest software updates to keep your Equalizer current and secure.
- **Online device manuals, supplements, and release notes**: the latest Equalizer documentation and updates.

1 Overview

Introducing Equalizer

This chapter provides an overview of Equalizer's features and discusses some common configurations.

Overview of Equalizer

Equalizer™ is a high-performance content switch that features:

- Intelligent load balancing based on multiple, user-configurable criteria.
- Real-time server and cluster performance monitoring.
- Server and cluster administration from a single interface.
- Session persistence using cookies or IP addresses
- Hot-backup configurations (requires a second Equalizer) featuring no single point of failure.
- Layer 7 content-sensitive routing.
- Geographic load balancing (requires the optional Envoy add-in).

This document describes the features and capabilities of the Equalizer units available at the time this document was prepared. For a current list of products and their features, please visit Coyote Point's website at (`www.coyotepoint.com`).

Intelligent Load Balancing

Equalizer functions as a *gateway* to one or more sets of *servers* known as *virtual clusters*. When a client submits a request to a site that Equalizer manages, Equalizer identifies the virtual cluster for which the request is intended, determines the server in the cluster that will be best able to handle the request, and forwards the request to that server for processing.

To route the request, Equalizer modifies the header of the request packet and forwards the modified packet to the selected server. When operating in *Layer 7 (L7)*, Equalizer can evaluate and, in some cases, modify the contents of both the request and response headers.

To determine the best server to route a request to, Equalizer uses intelligent *load balancing* algorithms that take into account the configuration options set for the cluster and servers, real-time server status information, L7 rules, and information from the request itself.

Load Balancing Configuration

When you configure your virtual cluster, you can select one of the following load-balancing algorithms to control how Equalizer balances the load across your servers: **round robin**, **static weight**, **adaptive**, **fastest response**, **least connections**, or **server agent**.

When you configure the servers in a virtual cluster, you assign a static weight between 20 and 200 for each server. When you select one of the adaptive load-balancing algorithms, Equalizer uses the servers' static weights as a starting point to determine the percentage of requests to route to each server. Each server handles a percentage of the total load based on its fraction of the total weights in the server cluster. Equalizer dynamically adjusts server weights according to real-time conditions to ensure that Equalizer routes requests to the server that is best able to respond. A server with a weight of zero (0) is considered down or unavailable: Equalizer does not route new requests to servers in this state.

Real-Time Server Status Information

Equalizer can gather real-time information about a server's status using Server Agents and Active Content Verification (*ACV)*.

You can install a *server agent* on each server to provide Equalizer with periodic performance statistics. This enables Equalizer to adjust the dynamic weights of the servers in a cluster according to their actual performance characteristics. If the server is overloaded and you have enabled adaptive load balancing, Equalizer responds by reducing the server's dynamic weight so that the server receives fewer requests. Coyote Point provides APIs useful for creating these agents. For more information see “Using Server Agents” on page 161.

Equalizer's *active content verification (ACV)* provides a way to check the validity of a server's response using most network services that support a text-based request/response protocol, such as HTTP. When you enable ACV for a cluster, Equalizer requests data from each server in the cluster (using an *ACV Probe string)* and verifies the returned data (against an *ACV Response string)*. If Equalizer receives no response or the response string is not in the response, the verification fails and Equalizer stops routing new requests to that server. (Note that you cannot use ACV with UDP-based services.) For more information, see “Using Active Content Verification (ACV)” on page 83.

Network Address Translation and Spoofing

Equalizer's *Network Address Translation (NAT)* subsystem distributes incoming Layer 4 or Layer 7 (with *spoofing*) client requests among the available servers. The NAT subsystem records the existence of the request, selects the best available server, rewrites the TCP/UDP and IP headers of the request packet, and then forwards the translated packet to the selected server. Because the servers are configured to use Equalizer to gateway all packets, Equalizer performs the reverse translation as the server response packets leave the cluster.

When IP spoofing is enabled, the servers see their client's actual IP address. However any response must be gatewayed through the Equalizer because clients will only recognize the Equalizer's address—they did not communicate directly with the server. (For more information about configuring spoofing see “Adding a Virtual Cluster” on page 67.)

When Equalizer receives an incoming packet that is not destined for a virtual cluster address, Equalizer passes the packet through unaltered. Similarly, when Equalizer receives an outgoing packet that is not a response to an existing virtual cluster connection, Equalizer passes the packet through to the external network.

Load Balancing UDP Services

You can configure Equalizer virtual clusters to provide load balancing and server failure detection for many UDP (User Datagram Protocol) based services. UDP load balancing is ideal for *stateless* protocols such as DNS and RADIUS, can load-balance WAP (Wireless Application Protocol) gateways, and can even load-balance certain types of NFS server cluster that provide a single-system image.

Equalizer does not support Active Content Verification for UDP clusters.

Maintaining Persistent Sessions

The *persistence* of *session* data is important when the client and server need to refer to data previously generated during the same session. For example, a web-based shopping cart application may depend on persistent session information between the client and server; that is, the details in the shopping cart potentially need to persist across many individual TCP connections before the data is no longer needed and the transaction is complete. Equalizer supports two mechanisms for maintaining persistent sessions: cookie-based and IP-address-based persistence.

Cookie-Based Persistence

Equalizer can use cookie-based persistence for HTTP and HTTPS clusters that support Layer 7 load balancing. In cookie-based persistence, Equalizer “stuffs” a cookie into the server’s response header on its way back to the client. This cookie uniquely identifies the server to which the client was just connected. The client includes (sends) the cookie in subsequent requests to the Equalizer. Equalizer uses the information in the cookie to route the requests back to the same server.

Equalizer can direct requests from a particular client to the same server, even if the connection is to a different virtual cluster. For example, if a user switches from an HTTP cluster to an HTTPS cluster, the persistent cookie will still be valid if the HTTPS cluster contains a server with the same IP address.

If the server with which a client has a persistent session is unavailable, Equalizer automatically selects a different server. Then, the client must establish a new session; Equalizer stuffs a new cookie in the next response.

IP-Address Based Persistence

For generic TCP and UDP clusters that support Layer 4 load balancing, Equalizer supports IP-address based persistent sessions. With the *sticky connections* feature enabled, Equalizer identifies clients by their IP addresses when they connect to a cluster. Equalizer routes requests received from a particular client during a specified period of time to the same server in the cluster.

A *sticky timer* measures the amount of time that has passed since there was a connection from a particular IP address to a specific cluster. The sticky time period begins to expire as soon as there are no longer any active connections between the client and the selected cluster. Equalizer resets the timer whenever a new connection occurs. If the client does not establish any new connections to the same cluster, the timer continues to run until the sticky time period expires. At expiration, Equalizer handles any new connection from that client like any other incoming connection and routes to an available server based on the selected load-balancing criteria.

To correctly handle sticky connections from ISPs that use multiple proxy servers to direct user connections, Equalizer supports *sticky network aggregation* with which only the network portion of a client's IP address maintains a sticky connection. Sticky network aggregation directs the user to the same server no matter which proxy he or she connects through.

You can also configure Equalizer to ensure that it directs requests from a particular client to the same server even if the incoming connection is to a different virtual cluster. When you enable *inter-cluster stickiness* for a cluster, Equalizer checks the cluster for a sticky record as it receives each connection request, just like it does for ordinary sticky connections. If Equalizer does not find a sticky record, Equalizer proceeds to check all of the other clusters that have the same IP address. If Equalizer still does not find a sticky record, it connects the user based on the incoming request.

Layer 7 Load Balancing and Server Selection

Equalizer’s support for Layer 7 content-sensitive load balancing (not available for the E250si) enables administrators to define rules for routing HTTP and HTTPS requests, depending on the content of the request. Layer 7 load balancing routes requests based on information from the application layer. This provides access to the actual data payloads of the TCP/UDP packets exchanged between a client and server. For example, by examining the payloads, a program can base load-balancing decisions for HTTP requests on information in client request headers and methods, server response headers, and page data.

Equalizer’s Layer 7 load balancing allows administrators to define rules in the administration interface for routing HTTP and HTTPS requests according to the request content. These rules are called *match* rules. For example, you can use Layer 7 rules to specify routing preferences such as,

- send all requests for graphics files to servers A, B and E
- send all requests for Perl scripts to servers C and D
- send all other requests to server Z

This enables administrators to create extremely flexible cluster configurations. Administrators can use Layer 7 technology to implement client-server persistence based on HTTP cookies.

For HTTP requests, Layer 7 load balancing can make decisions based on the following:

- HTTP protocol version
- Host name
- Pathname of the request
- Filename of the request
- Pattern matches against arbitrary HTTP request headers

Go to “Match Functions” on page 132 for a complete list of match functions.

For HTTPS requests, load balancing decisions can be based on the SSL protocol level the client uses to connect.

Geographic Load Balancing

The optional Envoy add-on supports *geographic load balancing*, which enables requests to be automatically distributed across Equalizer sites in different physical locations. An Equalizer *site* is a cluster of servers under a single Equalizer's control. A *geographic cluster* is a collection of sites that provide a common service, such as Web sites. The various sites in a geographic cluster can be hundreds or even thousands of miles apart. For example, a geographic cluster might contain two sites, one in the eastern U.S. and one on the U.S.'s west coast (Figure 1).

Geographic load balancing can dramatically improve reliability by ensuring that your service remains available even if a site-wide failure occurs. Equalizer can also improve performance by routing requests to the location with the least network latency.

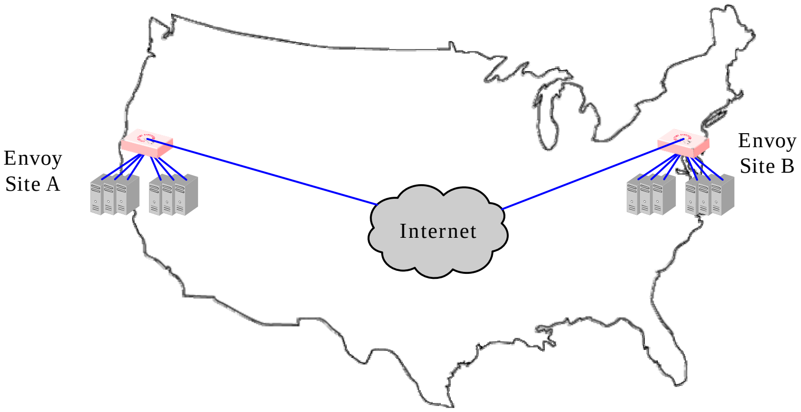


Figure 1 Geographic cluster with two sites

Geographic Load Balancing Routing

Envoy routes each incoming request to the site best able to handle it. If a site is unavailable or overloaded, Envoy routes requests to the other sites in the geographic cluster. When you enable geographic load balancing, Envoy directs incoming client requests to one of the sites in the geographic cluster based on the following criteria:

- **Availability:** If a site is unavailable due to network outage, server failure, or any other reason, Equalizer stops directing requests to that site.
- **Performance:** Envoy tracks the load and performance at each site and uses this information to determine the site that can process the request most efficiently.
- **Distance:** Envoy notes the site that is *closest* to the client (in network terms) and offers the least network latency.

Distributing the Geographic Load

Envoy uses the Domain Name System (DNS) protocol[1] to perform its geographic load distribution. DNS translates fully-qualified domain names such as `www.coyotepoint.com` into the IP addresses that identify hosts on the Internet. For Envoy, the authoritative name server for the domain is configured to query the Equalizers in the geographic cluster to resolve the domain name. When Envoy receives a resolution request, it uses the load-balancing algorithms configured for the

[1] For more information about DNS, see Paul Albitz and Cricket Liu, DNS and BIND, 3rd ed. (O'Reilly & Associates, 1998).

geographic cluster to determine the site that is best able to process the request and then returns the address of the selected site.

For example, the geographic cluster `www.coyotepoint.com` might have three sites (see Figure 2): one on the east coast of the U.S., one on the west coast of the U.S., and one in Europe. The servers at each site are connected to an Equalizer with the Envoy add-on installed.

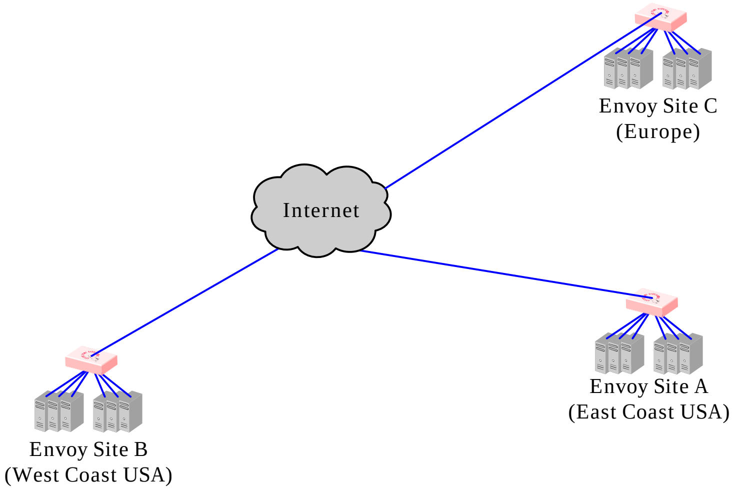


Figure 2 Three-site geographic cluster configuration

When a client in California attempts to connect to `coyotepoint.com`:

1. The client queries the its local name server to resolve the domain name (see Figure 3).

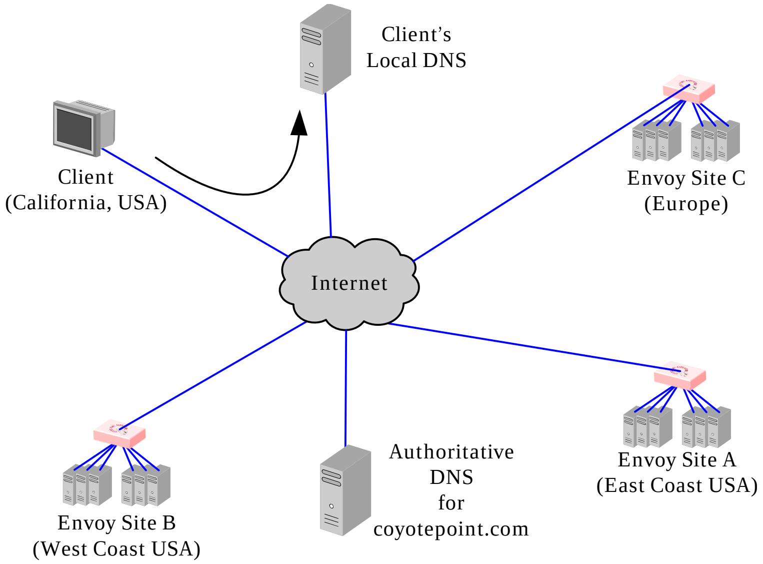


Figure 3 Client queries its local DNS for coyotepoint.com

2. The local name server queries the authoritative name server for `coyotepoint.com` (see Figure 4).

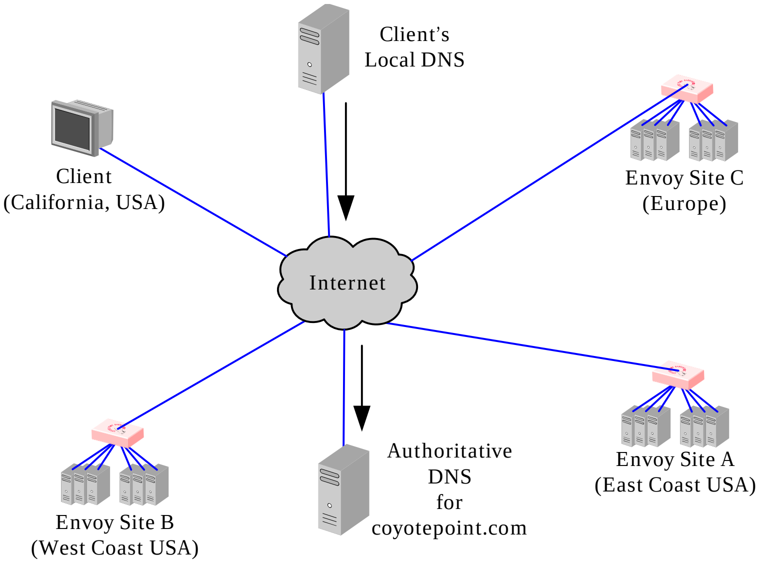


Figure 4 Client's local DNS queries the authoritative name server for coyotepoint.com

3. The authoritative name server provides a list of Envoy-enabled Equalizers and returns this list to the client's local DNS (see Figure 5).

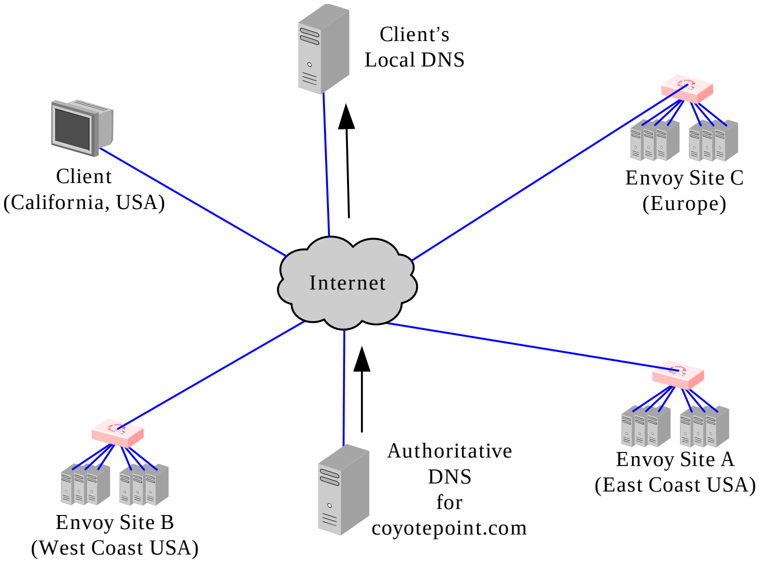


Figure 5 The authoritative name server for coyotepoint.com returns a list of delegates

4. The client’s DNS selects one of the Equalizers in the list and queries it. If the queried site doesn’t respond, the client tries each of the other sites.
5. Envoy returns the IP Address of the virtual cluster best able to handle the client’s request.

For more information on geographic load balancing using Envoy, see “Administering Geographic Clusters” on page 149.

Configuring the Equalizer Network

Equalizer is a versatile traffic management solution. It works in a single or dual network mode. If you have a second unit, you can use it as a hot-backup unit. Equalizer also works with servers placed on a reserved, non-routable network and allows for IP address aliasing.

You can use Equalizer in a number of configurations. Before you install Equalizer, you need to determine where it will fit into your network and how you will configure it. This section describes some configuration choices. The following section provides a worksheet to help you plan your configuration.

Equalizer’s Network Ports

All Equalizers have two types of network ports: *external* and *server*. The external port is always a single port labeled **External** or **Ext**. The server ports are labeled **Int** on dual-port models or labeled with numbers on switch-based models. Depending on the Equalizer switch-based model, there may be four or more of these ports.

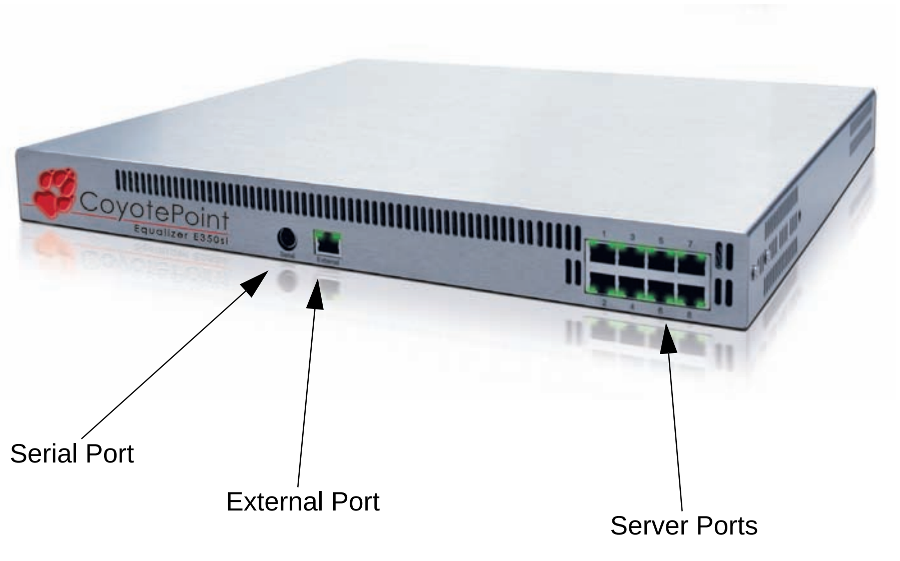


Figure 6 Equalizer E350si

Equalizer’s External Port

The external port is connected to the network to which the client machines and possibly the Internet or an Intranet are connected. This *external network* receives the client request packets that Equalizer distributes across the available servers. Equalizer also uses the external network to transmit response packets to clients. This port is only used for dual network (external and internal) configurations and single network configurations on dual-port models. It is not used for single network configurations on multi-server port models, see “Using Equalizer in a Single Network Environment” on page 11 for more information.

Hosts or routers on the external network can have routes to the internal network that are gatewayed through Equalizer's external address. Equalizer’s external address is also its *administration address*, the IP address used to connect to Equalizer’s browser-based administration interface.

Note – When using dual-port Equalizers in single network mode, use the external port to connect to the network to which the client machines, Intranet, or Internet are connected.

Equalizer’s Server Port

Servers that process the incoming requests connect to the server ports: either directly or through a network device such as a switch. These physical servers provide services on specific IP addresses and ports and are organized into clusters. Equalizer's load-balancing subsystem translates client request packets and then forwards them to the selected server. When a server machine sends a response packet back to a client, Equalizer processes it and forwards it to the appropriate client across the external network.

When using Equalizer with NAT in layer 4 or spoofing in layer 7, you must configure the servers’ routing tables so that Equalizer is the gateway for any outbound packets that leave the internal network. If the servers do not use Equalizer’s internal address as the gateway when they send responses to clients, the reply packets will not be translated on their way to the client, causing the clients to reject the reply packets because they do not belong to an established connection. (From the client side, it would look like the server was not responding.) If you are using Equalizer without spoofing, you do not need to use Equalizer as a gateway.

When using Equalizer in single network mode, the client machines, Intranet, or Internet must connect to one of the server ports. In this instance one of the server ports is the external port.

Using Equalizer as a Gateway Between Networks

The most common Equalizer configuration is to have Equalizer function as the gateway between two separate networks—the internal network where the servers reside and the external network on which clients and the Internet or an Intranet reside. Figure 7 shows this configuration in detail.

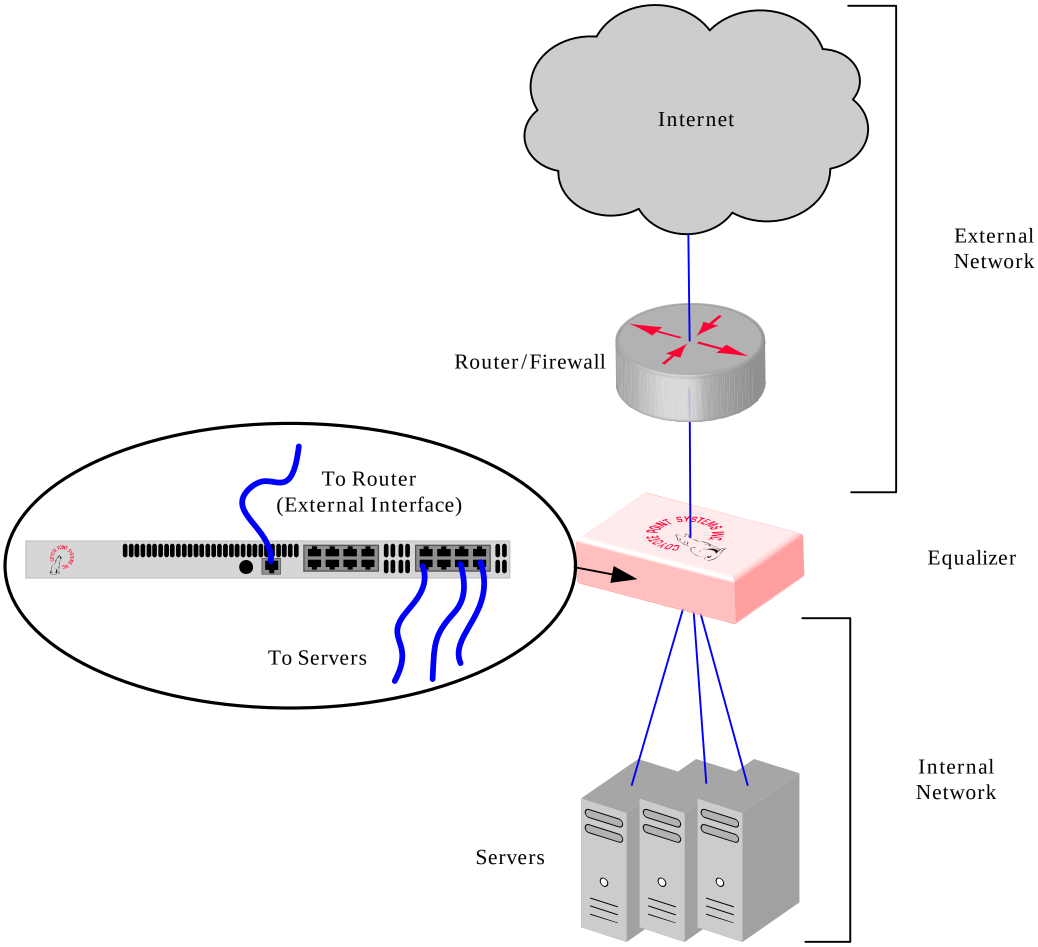


Figure 7 Sample two-network configuration

Using Equalizer in a Single Network Environment

If you do not want to split your network into internal and external networks, you can configure Equalizer to use a single-network mode, effectively placing both the clients and servers on the same network. Figure 8 on page 11 shows this configuration in detail. Certain protocols that use dynamic port mapping or multiple TCP/UDP ports work best in a single network environment. For example, use a single network configuration if you need your servers on your internal network to communicate with a Windows file server or a machine running pcAnywhere™.

You implement single-network configurations differently depending on the Equalizer model.

For switch-based Equalizer models, connect one of Equalizer's server ports to the network and do not use the external port. Servers connect to the other server ports as usual. You must configure servers, which must have valid network addresses on the external network, to use Equalizer's internal address as the gateway for outbound packets. You do not configure an IP address on the external port when using a single network configuration.

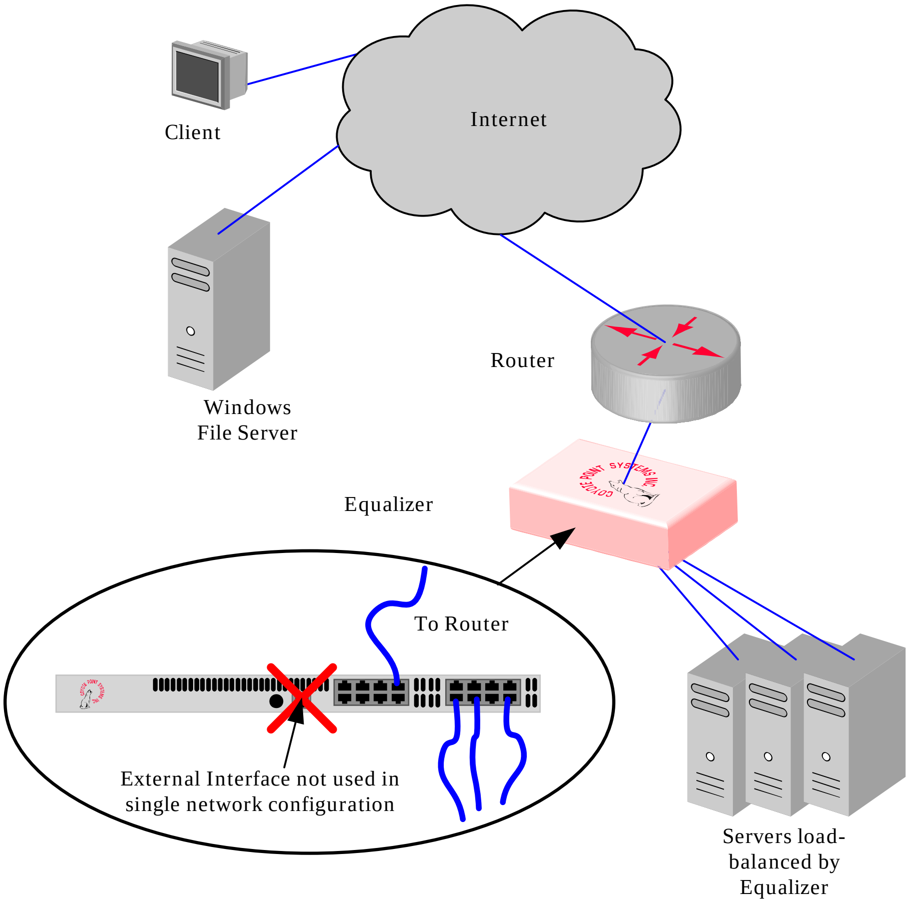

Client
Internet
Windows File Server
Router
Equalizer
To Router
External Interface not used in single network configuration
Servers load-balanced by Equalizer

Figure 8 Sample single network configuration for a switch-based Equalizer

For dual-port Equalizer models, the reverse is true. You leave the server (INT) port disconnected and connect the external (EXT) port to a switch that maintains the connections to the servers and to the external network.

Most operating systems allow you to specify a host route (gateway) for packets destined for specific hosts. If you want your virtual clusters to accept connections from clients on the same network as the servers, you must configure the servers to route packets destined for these clients through Equalizer. The clients on the local network must also be configured to use the Equalizer as their gateway; clients that do not have such routes configured connect to the server's IP address directly and not through a virtual cluster (that is, they are not routed through Equalizer).

Using a Second Equalizer as a Backup Unit

You can configure a second Equalizer as a backup unit that will take over in case of failure. This is known as a *hot-backup* configuration. The two Equalizers are siblings (or *peers),* the *primary* unit and the *backup* unit. If the primary Equalizer stops functioning, the backup unit adopts the primary unit's IP addresses (clusters) and begins servicing connections. In a failover configuration, the servers in a virtual cluster use a separate *failover alias* as their default gateway, rather than the IP address of the cluster or external port on a particular Equalizer. The failover alias migrates between the primary and backup unit as needed, automatically ensuring that the servers have a valid gateway in the event of a failure.

In a hot-backup configuration, both the primary and backup Equalizers are connected to the same networks; the backup unit's cluster and external ports must be connected to the same hubs or switches to which the primary Equalizer's ports are connected. Figure 9 on page 13 shows a sample failover configuration.

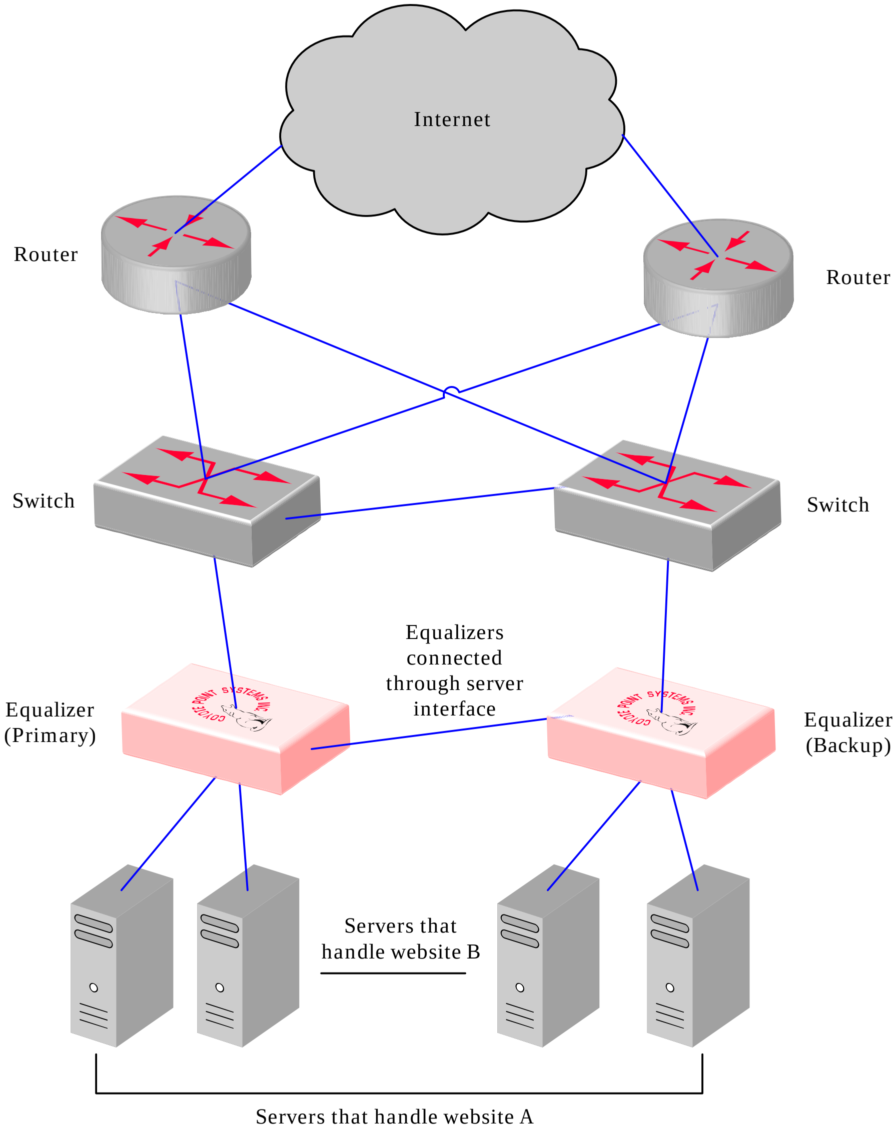

Internet
Router
Router
Switch
Switch
Equalizers connected through server interface
Equalizer (Primary)
Equalizer (Backup)
Servers that handle website B
Servers that handle website A

Figure 9 Sample failover configuration

In the sample failover configuration, the is no single point of failure. If a router goes down, the other router takes over or if a link fails, requests are routed through another link.

Figure 10 shows a sample of the cabling of the Equalizers shown in Figure 9.

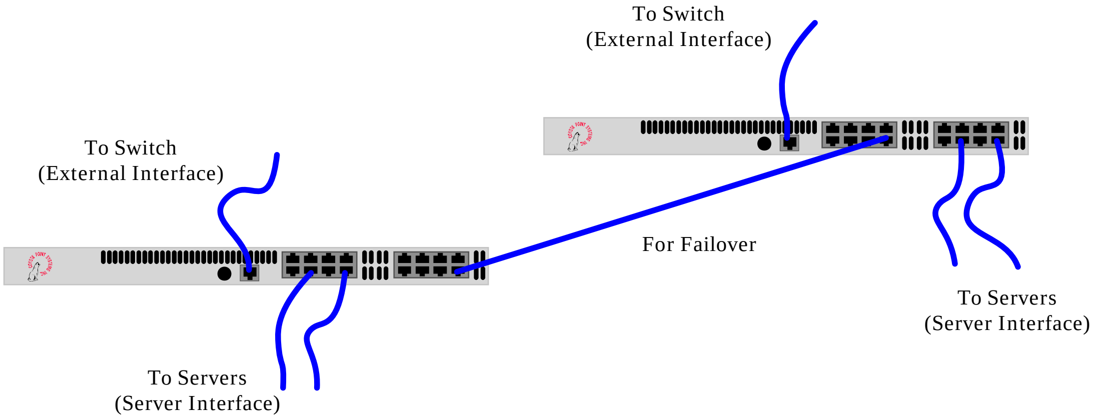


Figure 10 Cabling example from the sample failover configuration

The backup-unit Equalizer monitors all traffic to and from the primary unit; both Equalizers periodically exchange status messages over the local area network. The sibling Equalizers also exchange current configuration information. When you update the configuration on either machine, the configuration on its sibling is automatically updated.

Should either Equalizer fail to respond to a status message probe, the survivor begins a diagnostic cycle and attempts to contact its sibling via the other network ports. If these attempts fail, the sibling is considered to be *down*.

When the backup Equalizer determines that its sibling is down, it initiates a failover process:

1. The backup Equalizer configures the virtual cluster aliases on the external port and sends out “gratuitous ARP” packets that instruct any external-network routers to replace ARP table entries that point to the physical address of the failed Equalizer with the physical address of the backup unit.
2. The backup Equalizer configures a *failover gateway alias* on the port that is local to the servers.
   - With no backup configuration, the servers use the IP address of the cluster or external port as their default gateway.
   - In a hot-backup environment, the gateway address can migrate between the primary and backup unit. This requires an additional address.
3. The Equalizer kernel changes from BACKUP mode to PRIMARY mode. The PRIMARY-mode Equalizer performs gateway routing of packets between its cluster and external ports, address translation, and load balancing.

   When a failed unit is brought back online, it begins to exchange status messages with its sibling. Once both Equalizers have synchronized, the newly-started unit assumes the backup role.

Using Reserved IP Addresses

In environments in which conserving IP addresses is important, using reserved IP addresses can minimize the number of “real” IP addresses needed. Equalizer supports placing servers on *reserved*, non-routable networks such as the class A network 10.0.0.0 and the class C network 192.168.2.0.

For example, an ISP hosting several hundred unique web sites replicated on three servers might not want to assign real IP addresses for all of them because each virtual cluster would consume four addresses: three on the back-end servers and one for the virtual cluster. In this case, the ISP might use 10.0.0.0 (the now-defunct Arpanet) as the internal network and assign virtual server addresses out of this network for the servers.

Figure 11 shows a reserved network configuration in detail.

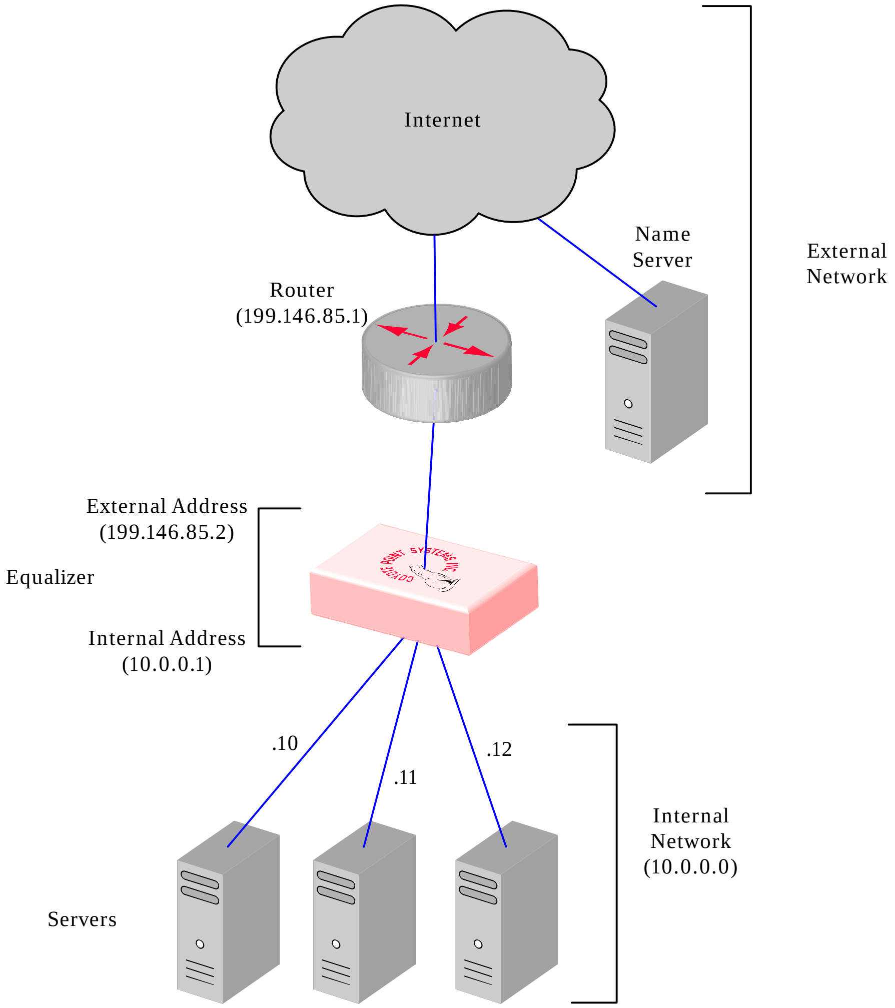

Internet
Name Server
External Network
Router (199.146.85.1)
External Address (199.146.85.2)
Equalizer
Internal Address (10.0.0.1)
.10
.11
.12
Internal Network (10.0.0.0)
Servers

Figure 11 Reserved internal network configuration

If servers placed on a non-routable network need to communicate with hosts on the Internet for any reason (such as performing DNS resolution or sending e-mail), you need to configure Equalizer to perform *outbound NAT*. When you enable outbound NAT, Equalizer translates connections originating from the servers on the reserved network so that external hosts will not see packets originating from non-routable addresses. If you use a failover configuration, you must use the same outbound NAT setting on both Equalizers. For more information, see “Setting Up a Failover Configuration” on page 47.

> **Note –** If outbound NAT is enabled, Equalizer processes each server response. If your servers do not need to initiate outbound connections, disable outbound NAT for improved performance.

Equalizer Configuration Worksheets

This section includes two configuration worksheets:

- use the “Standard Configuration Worksheet”, below, to prepare to install and configure Equalizer
- use the “Special Configuration Worksheet for Using Reserved IP Addresses” on page 18 only if you plan to use reserved IP addresses (rather than real IP addresses) when you set up Equalizer

Standard Configuration Worksheet

Before you install and configure Equalizer, write down the answers to all the following questions:

1. **What is your physical network layout?**

   Will all your servers, Equalizer, and your Internet router reside on a single network? Or will you use a two-network configuration and split your network into multiple subnets? If you use two-network configuration, Equalizer will function as the gateway between them and must be connected to both.

   If you don’t have a subnet or separate network available to devote to Equalizer’s internal network, you can use a single-network topology. For information about using Equalizer with a single network, refer to “Using Equalizer in a Single Network Environment” on page 11.

2. **Which network will be used as the external network?**

   Equalizer’s external port is connected to this network, which is connected to the Internet.

   Example 1: Single Network

   For the class C network `199.146.85.0` with a default netmask of `255.255.255.0`, the external network would be `199.146.85.0`. (See Figure 8 on page 11.)

   Example 2: Two Class C Networks

   If you use two class C networks, `199.146.85.0` and `199.145.90.0`, and choose the first as the external network, the external network would be `199.146.85.0`, with a netmask of (for example) `255.255.0.0`. (See Figure 7 on page 10.)

3. **What is Equalizer’s address on the external network?**

   You can assign any suitable IP address on your external network as Equalizer’s external and administration address. To administer Equalizer, enter this address in your browser’s URL field.

   Example 1: Single Network

   Equalizer Administration Address: `199.146.85.2`. (See Figure 8 on page 11.)

   Example 2: Two Class C Networks

Equalizer Administration Address: `199.145.85.2`. (See Figure 7 on page 10.)

4. **What network will be used as the internal network?**

   This is the network on which the physical servers will reside. If you use separate external and internal networks, the internal network is connected to Equalizer's server port. You should configure routers within your site's network (the external network) to use Equalizer's external port as the gateway to the internal network.

   Example 1: Single Network - Switch-based Equalizer (more than two ports)

   External port (labeled Ext): Not Used. (See Figure 8 on page 11.)

   Example 2: Single Network - Dual-port Equalizer

   Server port (labeled Int): Not Used. (See Figure 8 on page 11.)

5. **What is Equalizer's address on the internal network?**

   Typically, assign the lowest numbered address on the internal network as Equalizer's address. Configure the back-end servers to use this address as their default gateway.

   Example 1: Single Network

   Equalizer Internal Network Address: Not applicable. (See Figure 8 on page 11.)

   Example 2: Two Class C Networks

   Equalizer Internal Network Address: `199.146.90.1`. (See Figure 7 on page 10.)

6. **How many physical server machines will you be configuring? What are their IP addresses on the internal network?**

   If you plan to use IP aliases on the server hosts (virtual hosting), decide the addresses that will be configured on each of the server machines. All server IP addresses and aliases must be unique; you can configure a particular server IP address or alias for only one server machine.

7. **What virtual cluster addresses will you be configuring?**

   Choose the IP addresses, protocols, and ports you will assign to the virtual clusters you create using Equalizer. These are the addresses on the external network that will be visible to clients.

   For example, `199.146.85.4:HTTP` is a virtual cluster on port 80, and `199.146.85.4:FTP` is a virtual cluster on port 21.

8. **What is the address of your internet router on the external network?**

   Equalizer uses this gateway when transmitting packets to hosts that are not on the internal network.

9. **What is the IP address of the name server that Equalizer will use?**

   If you configure a name server, Equalizer displays virtual cluster and server addresses by name rather than by IP address. If no name server is available, set the name server address to `0.0.0.0`.

10. **Where are your Name Servers?**

    If you configure Equalizer to use Envoy, determine the DNS servers in your organization that you need to configure to refer fully qualified domain lookups to your Equalizer machine(s).

Special Configuration Worksheet for Using Reserved IP Addresses

Equalizer supports placing servers on reserved, non-routable networks such as the class A network `10.0.0.0` and the class C network `192.168.2.0`. In environments in which conservation of IP addresses is important, using reserved IP addresses can minimize the number of “real” IP addresses needed. However, due to the additional overhead introduced by enabling outbound NAT, approach using reserved internal addresses with caution. For more information about using reserved IP addresses, see Appendix B.

Before you install and configure Equalizer using reserved IP addresses, write down the answers to both of the following questions:

1. **What is the reserved network to be used for the internal network?**

   Equalizer uses this set of addresses to forward connections to the HTTP daemons running on the servers.

   Example:

   `10.0.0.0` (netmask `255.0.0.0`) or `192.168.2.0` (netmask `255.255.255.0`)

2. **What is Equalizer's address on the internal network?**

   This is the address that the servers will use as their default gateway. This address must be on the reserved network (see above). Usually, the lowest address in the range is used for Equalizer.

   Example:

   `10.0.0.1` or `192.168.2.1`

2 Installing Equalizer

Before You Install Equalizer

The first step in setting up Equalizer is to connect it to the local area network and a power source. Once you have installed Equalizer, you need to configure it as described in Chapter 3, “Configuring Equalizer Hardware”.

Please review the warnings located in Appendix G , “Additional Requirements”, on page 197 for precautions you must take before installing your Equalizer hardware.

Stepping Through the Hardware Installation

To install Equalizer, follow these steps:

1. Carefully remove the Equalizer rack-mount enclosure and cables from the shipping container.

   (Save the original packaging in case you need to ship the Equalizer for any reason, such as sending it in for warranty service. The Equalizer chassis does not contain any parts that you can service. If you open the chassis or attempt to make repairs, you may void your warranty. See Appendix F , “License and Warranty”, on page 193.)

2. Place the Equalizer in its intended position in an EIA equipment rack or on a flat surface. Please see Appendix G , “Additional Requirements”, on page 197, for a list of environmental limits and power requirements for your Equalizer.

3. If you have an optional Xcel SSL Accelerator Card, install the card following the instructions that came with the device. (Instructions can also be downloaded from the **Device Manuals** section of the **Support Portal**; go to `http://www.coyotepoint.com/support.php` for more information.)

4. Using the supplied serial cable, connect a serial terminal or a workstation running terminal emulator software to the serial port on the front panel of the Equalizer (see Figure 6 on page 8).

5. Connect Equalizer to the network with a quality category 5 network cable:

   a. To use Equalizer as an intermediary between an external and internal network, connect Equalizer to the external network using the RJ-45 network connector marked *Ext* and connect Equalizer to the internal network using one or more of the numbered internal network connectors.

   b. For a single-network topology with a switch-based Equalizer (more than two ports), connect Equalizer to the external network using one of the numbered RJ-45 network connectors on the front panel of the Equalizer and connect Equalizer to the internal network using one or more of the other numbered network connectors.

   c. For a single-network topology with a dual-port Equalizer, connect Equalizer using the RJ-45 network connectors labeled Ext on the front panel of the Equalizer to a switch connected to both the external network and the internal network.

6. Connect Equalizer to an appropriate power source using the supplied power cord, which plugs into the 3-pin connector on the rear of the Equalizer enclosure. This system uses an auto-sensing power supply that can operate at 50Hz or 60Hz, 110-240 VAC input.
7. Turn on the power using the switch on the rear panel.

Once you have installed and started Equalizer, follow the directions in Chapter 3, “Configuring Equalizer Hardware” to configure the hardware for your network.

3 Configuring Equalizer Hardware

After you install the Equalizer hardware as shown in Chapter 2, “Installing Equalizer”, use the procedures in this chapter to perform basic hardware and network configuration. This chapter contains:

- “Setting Up a Terminal or Terminal Emulator for Equalizer” on page 21
- “Performing Basic Equalizer Configuration” on page 22
- “Managing Remote Access to the Equalizer” on page 27
- “Configuring DNS and Firewalls for Geographic Load Balancing” on page 29
- “Configuring Routing on Servers” on page 29
- “Configuring a Second Equalizer As a Backup (Failover)” on page 29
- “Testing Your Basic Configuration” on page 30

Setting Up a Terminal or Terminal Emulator for Equalizer

After the installation of the Equalizer hardware, you need to use a terminal or terminal emulator to complete the hardware configuration.

Serial Connection

When you set up Equalizer for the first time, you must use a serial connection in order to configure Equalizer’s network with the **eqadmin** interface. Connect the serial port on the Equalizer (see Figure 6) to the serial port on a terminal, or any system (such as a Windows or Unix PC) running terminal emulation software.

Configure your terminal or terminal emulator software to use the following settings:

- 9600 baud
- 8 data bits
- no parity
- one stop bit
- VT100 terminal emulation
- ignore hang-ups (if supported); this allows a single terminal session to continue running even if Equalizer restarts

On Windows systems, you can use the Windows built-in terminal emulator, **HyperTerminal**, or the **Tera Term Pro** terminal emulator to log in to Equalizer over the serial port. On Unix systems, you can use the **cu**(1) command or any other Unix serial communication program.

If you use **HyperTerminal**, in addition to the settings shown above, select **File > Properties > Settings** from HyperTerminal's menu, select **VT100** in the **Emulation** drop-down box, and then **Terminal Setup** to enable these options:

- keyboard application mode
- cursor keypad mode

Tera Term Pro version 2.3 is freely available at:

```
http://hp.vector.co.jp/authors/VA002416/teraterm.html
```

Performing Basic Equalizer Configuration

Use the Equalizer Configuration Utility (**eqadmin**) to specify the following:

- **Network Interfaces**: Equalizer's external and internal network interfaces and the netmasks associated with these networks.
- **Hostname/IP Address**: The DNS hostname or IP address that is assigned to Equalizer for an interface.
- **Default Gateway**: The IP address of the router or other ntwork device that Equalizer will use to forward packets to the Internet or Intranet.
- **DNS Server**: The Domain Name Server Equalizer will use.
- Current date, time, and time zone.
- Passwords for the Equalizer console and administration interface. The default values are included with the documentation that comes with Equalizer.

You must at least configure one of the network interfaces with an IP address in order to access the Equalizer browser based Administration Interface.

Starting to Configure Equalizer

As Equalizer boots, the terminal displays a series of device probe and startup messages. Normally, you can ignore these diagnostic messages. However, if you do not configure the terminal emulation software to ignore hang-ups, the terminal session might exit twice during the boot process. If this happens, restart the terminal session.

To begin configuration, follow these steps:

1. When the boot process is complete, press **Enter** on the terminal keyboard to display the login prompt.
2. When the login prompt appears, type **eqadmin** and press **Enter**.
3. When the password prompt appears, enter the **eqadmin** password and press **Enter**. Equalizer automatically launches the **Equalizer Configuration Utility**, which provides a character-based interface for setting and changing Equalizer configuration parameters.
4. If the terminal display is not readable or not formatted properly, press **Esc** and make sure that your terminal emulator is set for VT100 emulation. Start over at Step 2.
5. To select a menu item within the configuration utility, press one or more arrow keys until you highlight the desired item. If the arrow keys do not operate within your terminal emulator, you

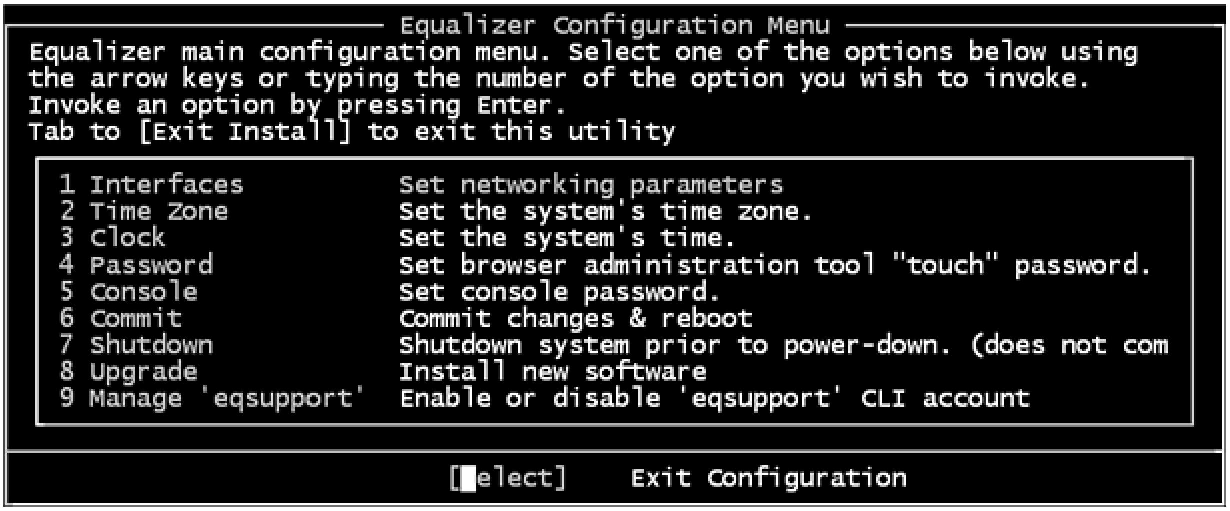
can use **Ctrl-n** to select the *next* menu item or **Ctrl-p** to select the *previous* menu item. Press the **Tab** key to highlight one of the menu actions (such as Select or Cancel) displayed at the bottom of the window. Then press **Enter** to continue.

Continue with “Configuring the Network Parameters” on page 23.

Configuring the Network Parameters

To configure the Hostname, Network Interfaces, Default Router, and DNS, use the following steps. Even if you are using your Equalizer in a single network configuration, you need to enter information for both the external and internal (server) interfaces.

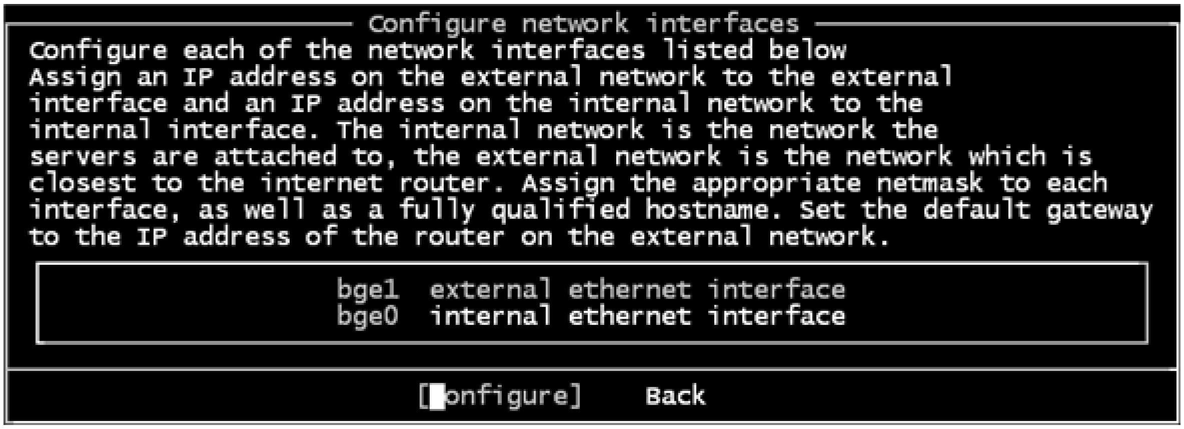
1. Once you log into Equalizer as shown in the previous section, the system displays the Equalizer Configuration Menu:



```
                      Equalizer Configuration Menu
Equalizer main configuration menu. Select one of the options below using
the arrow keys or typing the number of the option you wish to invoke.
Invoke an option by pressing Enter.
Tab to [Exit Install] to exit this utility

  1 Interfaces           Set networking parameters
  2 Time Zone            Set the system's time zone.
  3 Clock                Set the system's time.
  4 Password             Set browser administration tool "touch" password.
  5 Console              Set console password.
  6 Commit               Commit changes & reboot
  7 Shutdown             Shutdown system prior to power-down. (does not com
  8 Upgrade              Install new software
  9 Manage 'eqsupport'   Enable or disable 'eqsupport' CLI account

                       [█elect]    Exit Configuration
```

Figure 12 Equalizer Configuration Utility: Main Menu

2. In the Equalizer Configuration Menu window, select option 1, **Interfaces**, and press **Enter**. Equalizer displays the **Configure network interfaces** window (see Figure 13).

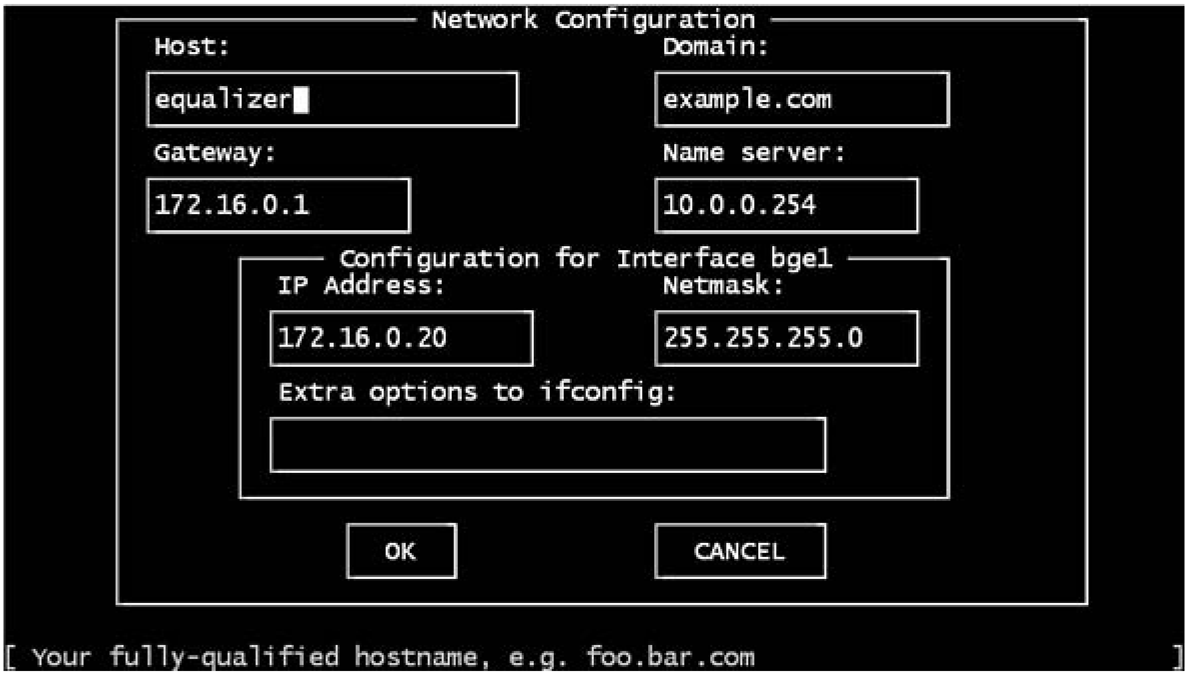


```
                    Configure network interfaces
Configure each of the network interfaces listed below
Assign an IP address on the external network to the external
interface and an IP address on the internal network to the
internal interface. The internal network is the network the
servers are attached to, the external network is the network which is
closest to the internet router. Assign the appropriate netmask to each
interface, as well as a fully qualified hostname. Set the default gateway
to the IP address of the router on the external network.

                  bge1  external ethernet interface
                  bge0  internal ethernet interface

                     [█onfigure]    Back
```

Figure 13 Equalizer Configuration Utility: Sample Interfaces

The interfaces shown in the screen above are examples only; the interfaces displayed for your system depend on your hardware configuration.

3. Press one or more arrow keys until you highlight **External Ethernet interface**; then press **Enter**. The Equalizer Configuration Utility displays the Network Configuration window (see Figure 14).



```
Network Configuration
Host:                         Domain:
equalizer                     example.com
Gateway:                      Name server:
172.16.0.1                    10.0.0.254
   Configuration for Interface bge1
   IP Address:                Netmask:
   172.16.0.20                255.255.255.0
   Extra options to ifconfig:

        OK                    CANCEL
[ Your fully-qualified hostname, e.g. foo.bar.com ]
```

Figure 14 Equalizer Configuration Utility: Network Configuration

4. In the **Host** field (required), enter the name for the Equalizer on your network. This can be the system node name (such as “eq-ext”), or the fully qualified domain name (FQDN, such as “eq-ext.customer.com”). If you supply the FQDN in the **Host** field, the **Domain** field will automatically be filled in using the domain of the FQDN.

5. In the **Domain** field (required), enter the domain name for the Equalizer. (For example, for the fully qualified domain name, `eq-ext.customer.com`, you would enter “customer.com” in the **Domain** field.

6. In the **Gateway** field (required), enter the IP address of the router on the external network. This router is the gateway for all the packets Equalizer sends to the outside world through the external network. For example, if your external network router is located at IP address `192.22.33.1`, enter “192.22.33.1” in the Gateway field.

7. In the **Name Server** field, enter the IP address of the domain name server that Equalizer will use. To indicate that no name server is available, leave the field blank (or, on the Equalizer 450 only, type `NONE).`

8. If you will be using the external port (that is, using either a dual-network configuration for a switch-based Equalizer or any configuration on a two-port Equalizer) you need to assign an IP address to the external interface. In the **IP address** and **Netmask** fields, respectively, specify

the IP address and netmask for the external interface. Use the address and netmask from your configuration worksheet (see “Equalizer Configuration Worksheets” on page 16).

For single network configurations using a switch-based Equalizer, leave the IP address for the external interface blank (or, on the Equalizer 450, type `NONE)` to disable the port.

9. When you’re finished, highlight **OK**. Then press **Enter**.

   Follow the next two steps only if you are using a switch-based Equalizer or a two-port Equalizer in a dual-network mode.

10. To specify the internal interface parameters, select **Internal Ethernet interface**. Then press **Enter**.
11. Specify the **IP Address** and **Netmask**. For example, if the internal interface will have the address `192.22.34.2`, enter `192.22.34.2` in the **IP Address** field. Leave the **IP address** field blank or type `NONE` to disable the server ports. The **Netmask** used will depend on how your network is configured.
12. Highlight **OK**. Then press **Enter**.
13. Highlight **Back**. Then press **Enter** to return to the main configuration menu.

For the new settings to take effect, you must commit these changes and reboot Equalizer, as shown in the following section.

Committing Changes to the Configuration Parameters

For the changes you make to the Network Configuration as shown in the previous section to take effect, you must commit the changes and reboot Equalizer, as shown in the following steps:

1. In the **Equalizer Configuration Menu** window, select option 6, **Commit**; then press **Enter**. The system commits your changes and automatically reboots.
2. When the boot process is complete, do the following to test your configuration changes:

- ping the assigned internal and external interface addresses from the Equalizer to check network connectivity
- ping the external address from a host on the external network
- ping the internal address from a host in the internal network
- if DNS is configured, ping a host on the Internet (e.g., `www.coyotepoint.com`) from the Equalizer to ensure that DNS and the Equalizer gateway are functioning properly

Setting the Time Zone

To set the current time zone, follow these steps:

1. In the **Equalizer Configuration Menu** window, select option 2, **Time Zone**, and press **Enter**.
2. Use the menus to specify your time zone.
3. Highlight **OK;** then press **Enter**.

Setting the Date and Time

To set the current date and time, follow these steps:

1. In the **Equalizer Configuration Menu** window, select option 3, **Time**; then press **Enter**.
2. Specify the current date and time, based on a 24-hour clock, in the format MM/DD/YY HH:MM.
3. Highlight **OK;** then press **Enter**.

Changing Equalizer's Console Password

The console password is the password for the **eqadmin** account, which automaticallly displays the Equalizer Configuration Utility when you log in. The factory-installed password for this account is **equalizer**. To change Equalizer's console password, follow these steps:

1. In the **Equalizer Configuration Menu** window, select option 5, **Console**. Then press **Enter**.
2. Type the new password; it can include any combination of printable characters (except spaces) up to 20 characters.
3. When prompted, enter the password again to confirm the change. The new password takes effect immediately.

Changing the Administration Interface Password

Use this option if you've forgotten the Administrative Interface's **touch** login password.

The Administrative Interface supports two logins: **touch** and **look**. When you log in to the administration interface using the **touch** login, you are in *edit mode*, and can update Equalizer's configuration (including the passwords for the **touch** and **look** logins). The **look** login does not allow any changes to be made. Therefore, if you forget the password for the **touch** login, you cannot reset it through the Administrative Interface, and must reset it using the **Equalizer Configuration Utility** via the serial console or remote access (see the section "Managing Remote Access to the Equalizer" on page 27).

To change the administration password, follow these steps:

1. In the **Equalizer Configuration Menu** window, select option 4, **Password**, and press **Enter**.
2. Type the new password. The password can include any combination of printable characters (except spaces) and can be no more than 20 characters in length (*note that spaces are accepted by the interface, but will not work when attempting to log in*).
3. When prompted, enter the password again to confirm the change. The new password takes effect immediately.

Upgrading Equalizer Software

After you have finished setting up your Equalizer to access the Internet, you can use the Equalizer Configuration Utility to install the latest Equalizer software upgrade from Coyote Point. The procedure below contains upgrade instructions that apply to any release. Check the Support Portal at **support.coyotepoint.com** for detailed upgrade instructions for the latest release, release notes, and version compatibility issues.

> **Note –** Before you can upgrade your Equalizer, you must first license it See "Licensing Equalizer" on page 39 for more information.

1. In the **Equalizer Configuration Menu** window, select option 8, **Upgrade**, and press **Enter**.
2. Highlight **OK**; then press **Enter**. The upgrade utility prompts you to enter the upgrade URL (see Figure 15):

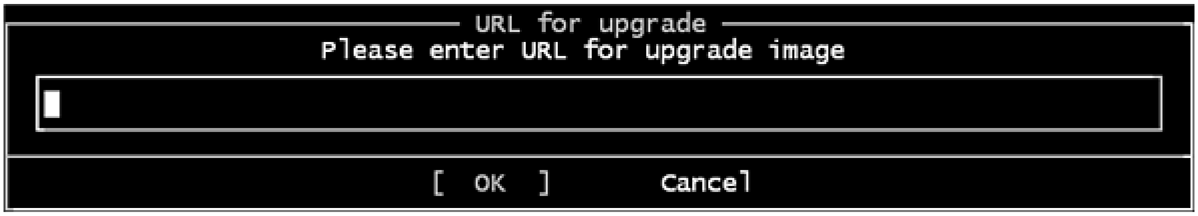


Figure 15 Equalizer Configuration Utility: Upgrade URL

Enter the URL provided to you by Coyote Point, select **OK**, and press **Enter**. The latest release of Equalizer software is always located at the following URL:

```
ftp://ftp.coyotepoint.com/pub/patches/upgrades/latest/upgrade.tgz
```

Equalizer downloads the upgrade file and runs the upgrade script.

3. When prompted, confirm that you want to upgrade the Equalizer software. The script then installs the software upgrade. Upgrades may take as long as five minutes. After the upgrade is installed, you will be prompted to reboot the system.

Shutting Down Equalizer

You can shut down Equalizer from the configuration utility. *Note that shutting down Equalizer does not automatically commit changes made to the configuration*. To shut down, follow these steps:

1. In the **Equalizer Configuration Menu** window, select option 7, **Shutdown**; then press **Enter**.
2. After the shutdown process completes, power off the system.

Managing Remote Access to the Equalizer

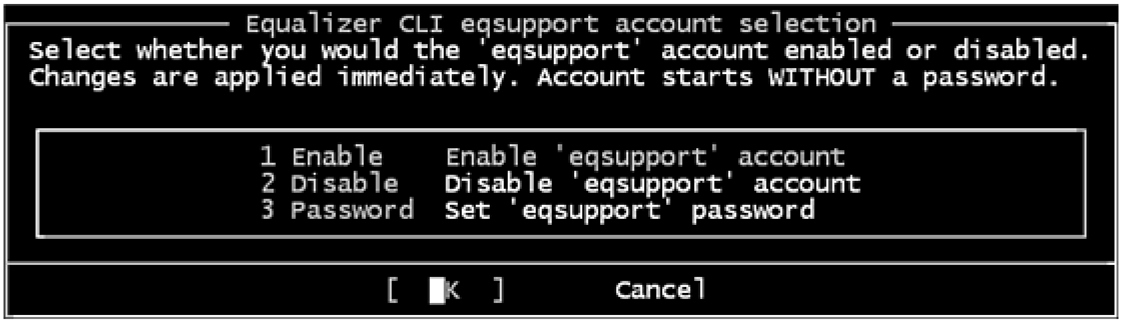
Remote access, when enabled, provides a user account (**eqsupport**) which allows you to log into Equalizer over a Secure Shell (SSH) connection.

Note – By default, the password for the **eqsupport** account is blank. If you enable the account, change the password when you enable it.

Managing the Remote Access Account

To enable, disable, or change the password for this account, use the hardware configuration utility as follows:

1. Log into the Equalizer hardware configuration utility using a terminal or terminal emulator (see “Setting Up a Terminal or Terminal Emulator for Equalizer” on page 21 and “Starting to Configure Equalizer” on page 22.
2. In the **Equalizer Configuration Menu**, select option 9, **Manage ‘eqsupport’**, and press **Enter** (see Figure 16). Equalizer displays the **Equalizer CLI eqsupport account selection** window.



```
─────── Equalizer CLI eqsupport account selection ───────
Select whether you would the 'eqsupport' account enabled or disabled.
Changes are applied immediately. Account starts WITHOUT a password.

              1 Enable     Enable 'eqsupport' account
              2 Disable    Disable 'eqsupport' account
              3 Password   Set 'eqsupport' password

                    [ █K ]        Cancel
```

Figure 16 Equalizer CLI eqsupport account selection

3. The following selections are available:
    a. To enable the remote access account, use the arrow keys to highlight **Enable** and press **Enter**. The account is now enabled.
    b. To disable the remote access account, use the arrow keys to highlight **Disable** and press **Enter**. The account is now disabled.
    c. To change the password, use the arrow keys to highlight **Password** and press **Enter**. Follow the prompts to change the password.

    If you modify the password for the account when it is disabled, Equalizer will display a reminder that the account must be enabled before you can use it.
4. When you are done, highlight OK on the account selection window and press **Enter** to return to the **Equalizer Configuration Menu**.

Using the Remote Access Account

Use the Secure Shell Client (SSH) to log in with the remote access account user name (**eqsupport**) and password. The account must be enabled in order for this to work. For the best visual output, the following are recommended:

- The PuTTY terminal emulator, freely available from

    `http://www.chiark.greenend.org.uk/~sgtatham/putty/`

- An SSH client running from a Windows Command window; for example, OpenSSH, which is freely available from:

    `http://sshwindows.sourceforge.net/download/`

- An SSH client running from a Cygwin window. Cygwin is a UNIX shell environment that includes versions of many UNIX utilities, including SSH; it is freely available from:

    `http://cygwin.com/`

When you run the Setup program to install, make sure that SSH (under “Net”), the Xorg Server and xterm (under “X11”) are selected for installation. To run, open a Cygwin window and enter ‘startx’; once the Xterm window opens, enter ‘ssh eqsupport@*equalizer-ip*’.

Configuring a Second Equalizer As a Backup (Failover)

You can configure a second Equalizer as a hot backup (or hot spare) so that if the Equalizer that currently handles requests (the *primary unit*) fails, the *backup unit* automatically takes over. This is called a *failover configuration*.

Both Equalizers are configured to default to either the primary or backup role. When a failed unit comes back online, it assumes the backup role, even if it is designated the default primary.

If you are going to use two Equalizers in a failover configuration, perform the basic configuration for both units now as described in the previous section.

Additional configuration for failover is performed through the Equalizer Administration Interface, as described in the section “Setting Up a Failover Configuration” on page 47.

Configuring Routing on Servers

To use Equalizer, you must configure your servers so that Equalizer gateways the packets the servers send to their clients. If you do not adjust the routing on your servers, a client may not receive a response when it attempts to contact a virtual cluster. Then, the connection will time out.

When you configure the servers, the *default* route gateway depends on your Equalizer configuration:

- If you use a **two-network configuration**, the gateway for the default route should be Equalizer’s internal address regardless of the Equalizer model.
- If you use a **single-network configuration** on switch-based Equalizers, the gateway for the default route should be Equalizer’s internal address.
- If you use a **single-network configuration** on dual-port Equalizers, the gateway for the default route should be Equalizer’s external address.
- If you use a **failover configuration**, set the gateway for the default route to the failover alias. For more information, see “Setting Up a Failover Configuration” on page 47.

The way that you configure routing on a server depends on the server’s operating system. To verify that you have configured a server’s routing correctly, trace the route from the server to a destination address outside the internal network to ensure that Equalizer gets used as a gateway. On UNIX systems, use the `traceroute` utility; on Windows, use `tracert`.

Configure each server from the system console, not through a telnet session. This will avoid any disconnects that might otherwise occur as you change the network settings on a server.

Configuring DNS and Firewalls for Geographic Load Balancing

If you are configuring Equalizer to use Envoy for geographic load balancing, you need to configure your authoritative domain name server to delegate authority to the Envoy sites. If you will use

Envoy across firewalled networks, you also need to configure the firewalls to allow traffic between Envoy sites and between the Equalizer and clients.

Configuring the Authoritative Name Server to Query Envoy

To delegate authority to the Envoy sites, you must configure the authoritative name server(s) for the domains that are to be geographically load-balanced. You also must delegate each of the fully-qualified subdomains to be balanced.

For example, assume that you want to balance `www.coyotepoint.com` across a geographical cluster with two Envoy sites, `east.coyotepoint.com` and `west.coyotepoint.com`. In this case, you configure the name servers that handle the `coyotepoint.com` domain to delegate authority for `www.coyotepoint.com` to both `east.coyotepoint.com` and `west.coyotepoint.com`. When a client asks to resolve `www.coyotepoint.com`, the name servers should return name server (NS) and alias (A) records for both sites.

Using Geographic Load Balancing with Firewalled Networks

Equalizer sites communicate with each other using Coyote Point’s UDP-based Geographic Query Protocol. Similarly, Equalizer sites communicate with clients using the DNS protocol. If a network firewall protects one or more of your sites, you must configure the firewall to permit Equalizer packets to pass through.

To use geographic load balancing with firewalled networks, you need to configure the firewalls so that the following occurs:

- Equalizer sites communicate with each other on UDP ports 5300 and 5301. The firewall must allow traffic on these ports to pass between Envoy sites.
- Equalizer sites and clients can exchange packets on UDP port 53. The firewall must allow traffic on this port to flow freely between an Equalizer server and any Internet clients so that clients trying to resolve hostnames via the Equalizer DNS server can exchange packets with Equalizer sites.

Equalizer sites can send ICMP echo request packets (i.e., a ‘ping’) through the firewall and receive ICMP echo response packets from clients outside the firewall. (When a client attempts a DNS resolution, Equalizer sites send an ICMP echo request packet to the client; the client might respond with an ICMP echo response packet.)

Testing Your Basic Configuration

Once you have installed and configured Equalizer and your servers, perform tests to verify that Equalizer is working properly.

To perform these tests, you need the following:

- A test machine on the internal network (the same physical network as the servers; one of the server machines can be used for this purpose).
- If you have a two-network configuration, a test machine on the external network.
- A client machine somewhere on the Internet, to simulate a “real-world” client. This machine should be set up so that the only way it can communicate with your servers or Equalizer is through your Internet router.

Then follow these steps:

1. From the internal-network test machine, ping the physical IP address of each server. You should be able to successfully ping all of the servers from the test machine.
2. From the internal-network test machine, ping the server aliases on each of the servers. You should be able to successfully ping all of the servers from the test machine using their aliases.
3. From the internal test machine and each of the servers, ping the Equalizer address that you use as the default gateway on your servers. (If you use a two-network topology, this will be Equalizer's internal address or failover alias.)
4. From the internal-network test machine, connect to the server aliases on service ports of running daemons (you may need to configure telnet or ssh services on Windows servers). You should be able to connect successfully to the server aliases.
5. If you use a two-network configuration: From the external-network test machine, ping a physical server IP address using `ping -R` to trace the route of the ping. The Equalizer IP address should appear in the list of interfaces that the ping packet traverses. You can also use the `traceroute` (UNIX) or `tracert` (Windows) tools to perform this test.

For help in resolving configuration problems, see Appendix E, ”Troubleshooting”.

4 Accessing Browser Controls

Introducing the Equalizer Administration Interface

You use Equalizer’s HTML-based administration interface for routine monitoring and administrative tasks. Access the administration interface from a Javascript-enabled web browser to perform the following actions, described in the remainder of the chapters in this book):

- Monitor the status of Equalizer and the configured clusters and servers
- View cluster and server performance statistics graphically
- Add virtual clusters
- Modify cluster parameters
- Delete clusters
- Add servers to a cluster
- Adjust server static weights
- Delete servers
- Shut down a server gracefully
- Shut down Equalizer

The sections below show you how to access and use the Administration Interface.

Accessing the Equalizer Administration Interface

You must access the Equalizer Administration Interface through a Javascript-enabled browser.

The Equalizer Administration Interface supports the following two user modes:

View mode enables you to *view*, but not edit, Equalizer configuration and status information. Use the **look** login to log into Equalizer in **View** mode.

Edit mode enables you to *view* Equalizer configuration and status information, and *edit* the configuration. Use the **touch** login to log into Equalizer in **Edit** mode.

Logging In

To access the administration interface and log into Equalizer, follow these steps:

1. Launch a Javascript-enabled web browser. We recommend you use one of these browsers:
    - Internet Explorer Version 6 or later

- Firefox Version 2 or later

2. From the browser, load the URL that corresponds to Equalizer's external address, using either the `http` or `https` protocols. If you are using a redundant pair of Equalizers, use the failover alias to ensure that the browser connects to the Equalizer that has the primary role.

   For example, if the external or failover address is `199.146.85.2`, open the Equalizer Administration Interface by typing `http://199.146.85.2 or https://199.146.85.2` in the appropriate location in the browser. Use the `https` protocol to access the interface using SSL and a server certificate. This is recommended when accessing Equalizer over a public network (such as the Internet).

   Equalizer displays the login screen (see Figure 17):

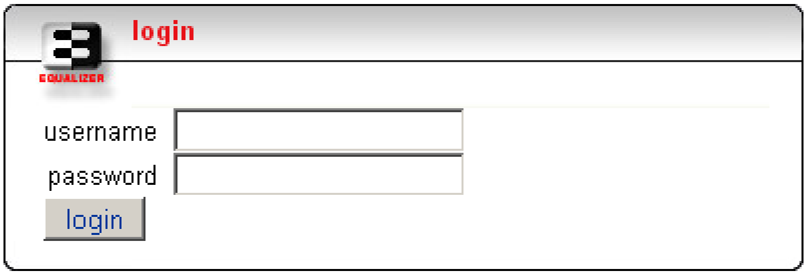


Figure 17 The login screen

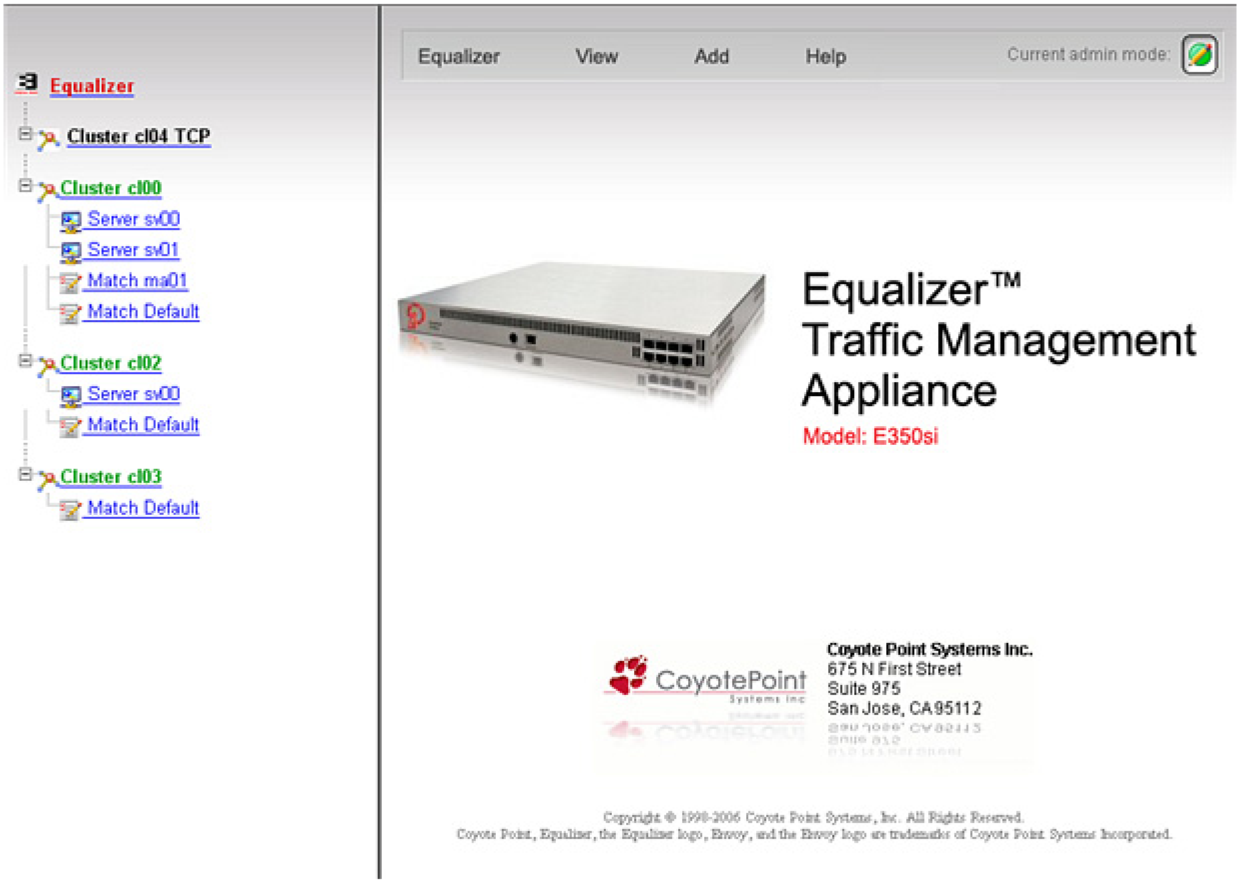
3. Enter **touch** or **look** as `username`; then, enter the `password`. Click the **login** button.

> **Note –** Initial passwords for the **touch** and **look** logins are provided in a separate, single-page document shipped with Equalizer, and available on the Support Portal (support.coyotepoint.com). The **touch** login password is set through the Equalizer Configuration Utility. For more information refer to “Changing the Administration Interface Password” on page 26.

Navigating Through the Interface

The Equalizer Administration Interface (see Figure 18) provides two navigation mechanisms: links and menus.

You can access status information and current parameters of any of the items in the hierarchical list in the left frame by clicking the name of the item you want to view. The hierarchical list contains all the currently configured clusters, servers, geographic clusters, and sites. Equalizer displays the status information and current parameters in the right frame.



Coyote Point Systems Inc.
675 N First Street
Suite 975
San Jose, CA 95112

Copyright © 1998-2006 Coyote Point Systems, Inc. All Rights Reserved.
Coyote Point, Equalizer, the Equalizer logo, Envoy, and the Envoy logo are trademarks of Coyote Point Systems Incorporated.

Figure 18 Equalizer’s Administration Interface

Using the Main Menu Bar

Use the menus in the main menu bar (see Figure 19) in the top frame and the local menus on the parameters pages to access Equalizer’s reporting options, modify the configuration, or view help information.

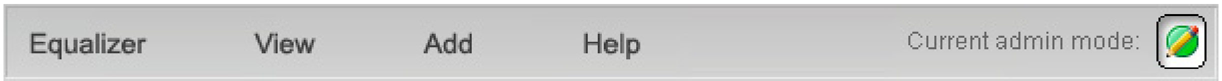


Figure 19 Main menu bar

- **Equalizer**: provides the following commands:

  Global Configuration: displays the **modify global parameters** screen, which allows you to view and change Equalizer’s operating parameters that affect all Equalizer clusters. A submenu provides access to additional global configuration options.

  Shut Down Equalizer: starts a clean shutdown of the Equalizer system so you can safely turn off the power. Note that this option works only when you are logged in under Edit mode. If you try to do this while you are logged in under View mode, Equalizer displays an error message.

Reboot: reboots the Equalizer. If you try to do this while you are logged in under view mode, Equalizer displays an error message.

Log Out: exits the Equalizer Administration Interface.

- **View**: provides access to the following global status information:

  Equalizer Status: displays the Equalizer software and hardware information, basic configuration, and recent statistics.

  Cluster Summary: displays summary information for all the configured clusters.

  Event Log: displays the Equalizer event log. When you have finished viewing the event log, moving to another location automatically closes the event log.

- **Add**: provides the following commands for adding clusters under Edit mode:

  Virtual Cluster, to add a new virtual cluster.

  Geographic Cluster, to add a new geographic cluster to a site. This command is only displayed if Envoy is installed.

- **Help**: provides access to the following information about using Equalizer:

  View Guide: displays the PDF file that contains the *Equalizer Installation and Administration Guide* (this book).

  View Release Notes: displays the PDF file that contains the *Release Notes* for the currently active version of Equalizer.

  Context Help: displays the section in the *Equalizer Installation and Administration Guide* PDF file corresponding to the screen currently displayed in the right frame.

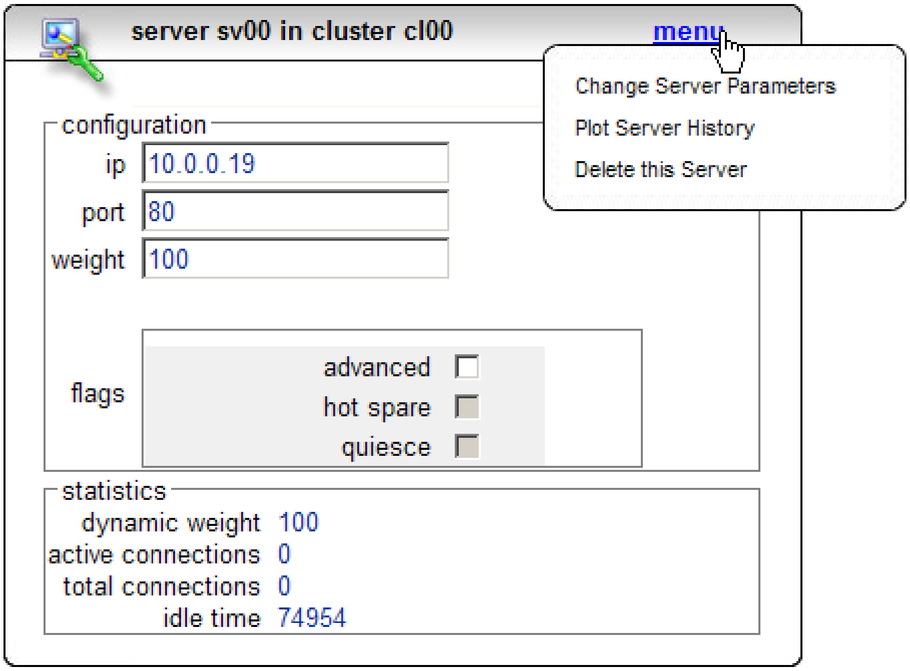
  Save System Info: creates an archive of Equalizer’s current state, various configuration files, logs, and other information. This archive is used as a diagnostic aid by Coyote Point Support. See “Troubleshooting” on page 187.

  About Equalizer: displays version and copyright information for Equalizer.

The icon displayed in the top right corner of the administration interface indicates the current user mode: **View** or **Edit**. When you are logged in under view mode, the configuration functions, such as adding a server or modifying a cluster’s parameters, are not available.

Accessing Local Menus

You can access local menus (see Figure 20) from the parameter screens that appear when you click an item in the left frame. Typically, you can find local menus in the upper right corner of the page. To activate a local menu, roll over it with your mouse. With local menus, you can view and change information about the currently-viewed item. For example, when you are in edit mode, the local menu in the Server Parameters page enables you to change the server’s parameters, plot the server’s history, and delete the server.



server sv00 in cluster cl00

configuration

ip 10.0.0.19

port 80

weight 100

flags

advanced □

hot spare □

quiesce □

statistics

dynamic weight 100

active connections 0

total connections 0

idle time 74954

Figure 20 The local menu on the Server Parameters page

5 Configuring Equalizer Operation

You can modify Equalizer’s configuration through the Equalizer Administration Interface and perform the following actions, described in this chapter:

> **Note –** The procedures in this chapter assume that you have already set up your Equalizer hardware and performed the initial configuration according to the instructions found in Chapter 2 and Chapter 3.

Licensing Equalizer

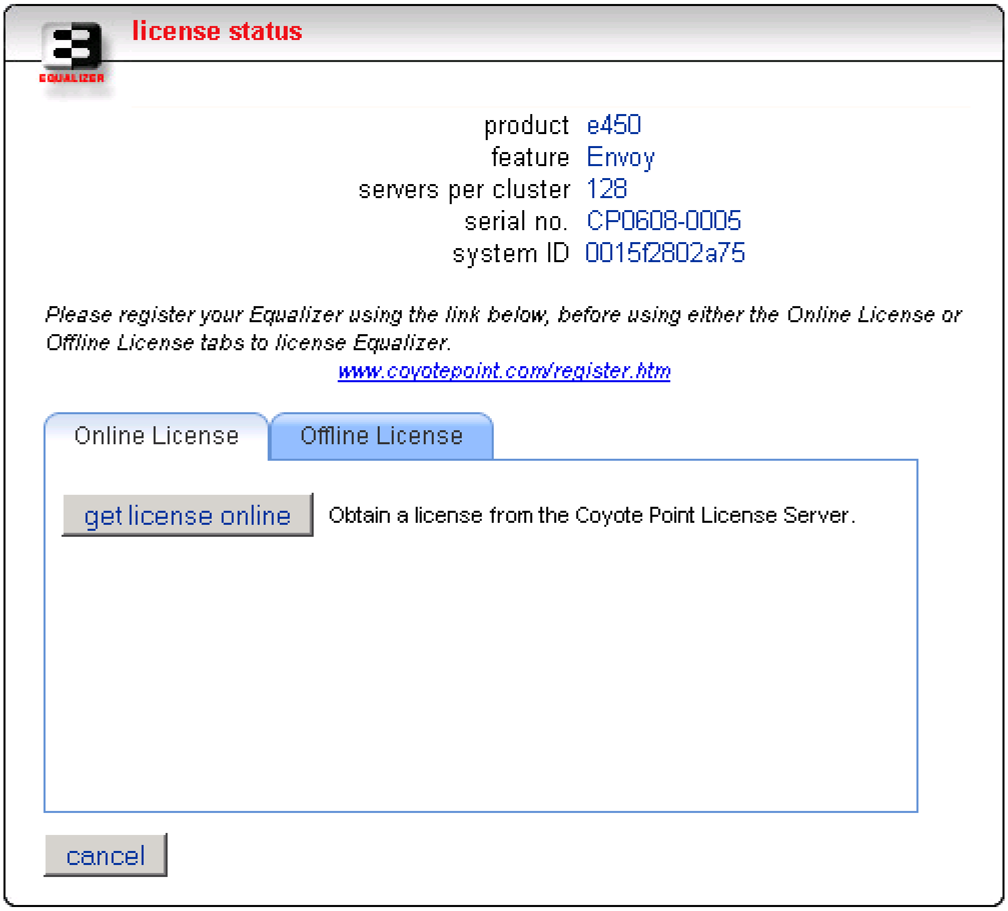
You must register and license your Equalizer before performing any other configuration using the Equalizer Administration Interface (described in Chapter 4). The License Manager is used to view your current license information and to request a license from the Coyote Point License Server.

You'll need to request a license if:

- The left frame of the Equalizer Administrative Interface displays an unlicensed system error.
- You add the Envoy Geographic Clustering product to Equalizer.

Follow this procedure to view license information or to request a license.

1. Log into the Equalizer Administration Interface in Edit mode.
2. Select **Equalizer** > **Global Configuration** from the main menu bar; then select **menu** > **Manage Licenses** from the **modify system parameters** screen. The **license status** screen appears in the right frame.



license status

product e450
feature Envoy
servers per cluster 128
serial no. CP0608-0005
system ID 0015f2802a75

Please register your Equalizer using the link below, before using either the Online License or Offline License tabs to license Equalizer.
www.coyotepoint.com/register.htm

Obtain a license from the Coyote Point License Server.

Figure 21 The license status screen

The top section of **license status** screen shows the following information for an already licensed system:

product	Equalizer product model number.
feature	Optional features enabled by your license.
servers per cluster	The number of servers per cluster allowed, as specified by your license.
serial no.	The serial number of the Equalizer unit (also printed on the back or bottom of the unit).

system ID	The internal system identifier (the MAC address of your primary network card). [Note: in previous releases, the system ID was shown with a colon (:) separating each pair of numbers.]

If you don’t need to license Equalizer, select **Cancel** to return to the **modfiy system parameters** screen. Otherwise, continue with the next step.

3. If your Equalizer is already registered with Coyote Point, skip this step.

   Otherwise, you must register your Equalizer before you can license it. Click on the link shown in the screen above to register Equalizer. Follow the prompts displayed by the Registration Web Site. You will need to copy the **system ID** and the **system serial number** shown in the **license status** screen into the registration form.

4. Do ***one*** of the following:

   a. If Equalizer is connected to the Internet and a DNS server is configured, click on the **get license online** button to request a license online. The license server will download your license automatically, and ask you if you want to reboot to apply the license. Select **Yes** to reboot.

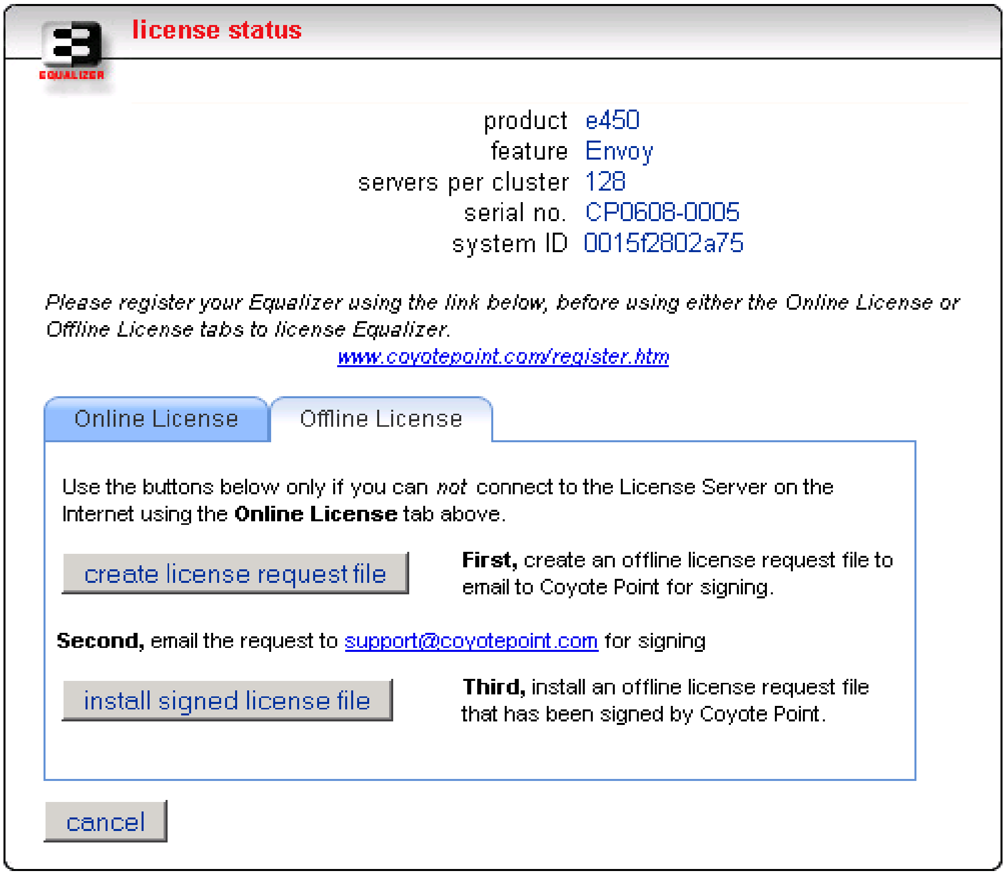
   b. If Equalizer is not connected to the Internet or DNS is not configured, then see the section “Requesting a License Offline” on page 41, below.

After the system comes back up, there should be no unlicensed error in the left frame. If you licensed Envoy, the Equalizer Status screen (select **Equalizer** in the left frame) should show **Envoy geographic load balancing enabled**.

Requesting a License Offline

If your Equalizer is not currently connected to the Internet or if DNS is not configured for Equalizer, then you will need to request a license offline. To do this, follow this procedure:

1. Follow Steps 1 through 3 of the procedure above.
2. Select the **Offline License** tab on the **license status** screen; the **license status** screen now appears as shown below:



product e450
feature Envoy
servers per cluster 128
serial no. CP0608-0005
system ID 0015f2802a75

Please register your Equalizer using the link below, before using either the Online License or Offline License tabs to license Equalizer.
www.coyotepoint.com/register.htm

Use the buttons below only if you can *not* connect to the License Server on the Internet using the **Online License** tab above.

First, create an offline license request file to email to Coyote Point for signing.

Second, email the request to support@coyotepoint.com for signing

Third, install an offline license request file that has been signed by Coyote Point.

3. Select **create license request file** and save the file to an appropriate location on your local system.
4. Select the **support@coyotepoint.com** link to open your browser's mail client, or open your email client manually and specify this address in the **To:** field of a new mail message. Specify **license request** in the **Subject** field, and attach the license request file you saved in the previous step. Send the email.
5. Once Coyote Point processes your request, you will receive a signed license file in a return email from Coyote Point. Save the licensing file you receive from Coyote Point to an appropriate location on your local system.
6. Select **install signed licensed file** and use the browse box to select the signed license file you saved in the previous step.
7. Equalizer installs the license and asks you if you want to reboot to apply the license. Select **Yes** to reboot.

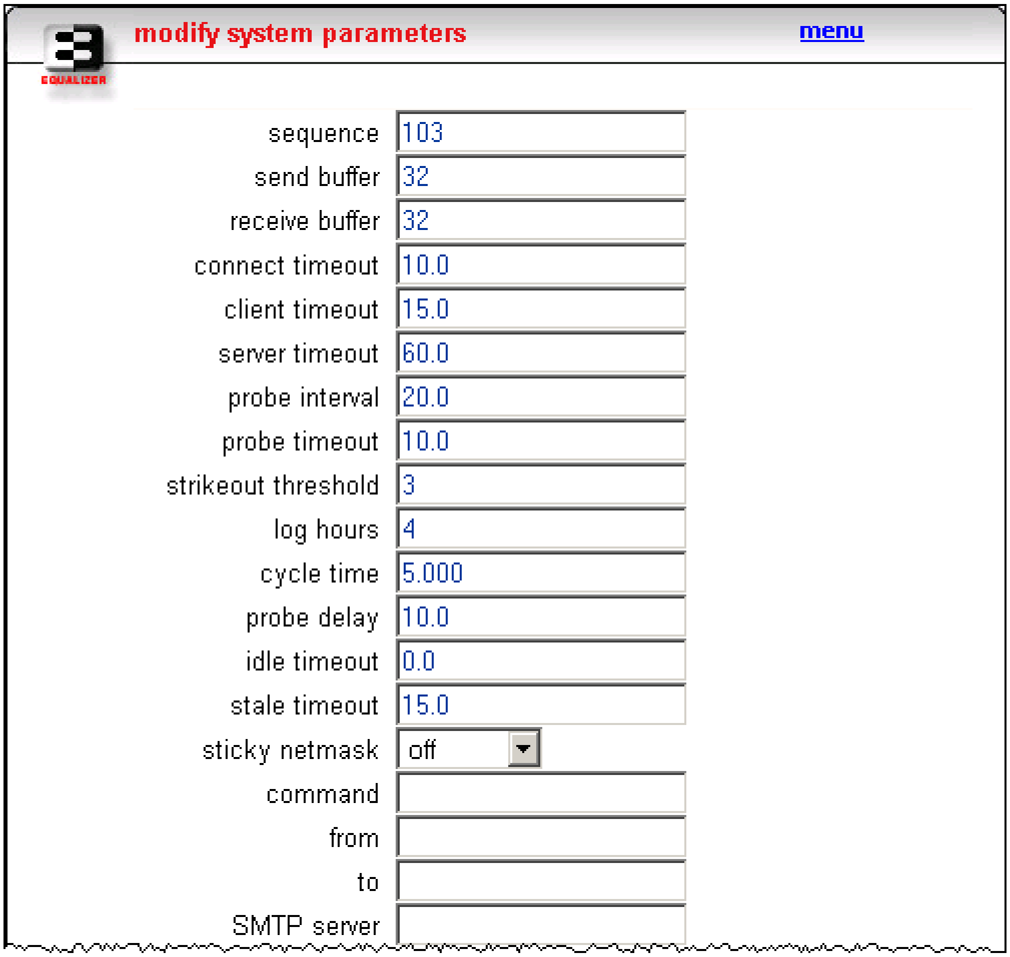
After the system comes back up, there should be no unlicensed error in the left frame. If your license includes the ability to use Envoy, the Equalizer Status screen (select **Equalizer** in the left frame) should show **Envoy geographic load balancing enabled**.

Modifying System Parameters

To view and modify the Equalizer's system parameters, follow these steps:

1. Log into the Equalizer Administration Interface in Edit mode.
2. Select **Equalizer** > **Global Configuration** from the main menu bar; Equalizer displays the **modify system parameters** screen in the right frame; see Figure 22 on page 43, Figure 23 on page 45, and Figure 24 on page 46.
3. Change the appropriate fields.
4. Click the **commit** button.

The **modify system parameters** screen displays the following parameter values that affect Equalizer's operation:



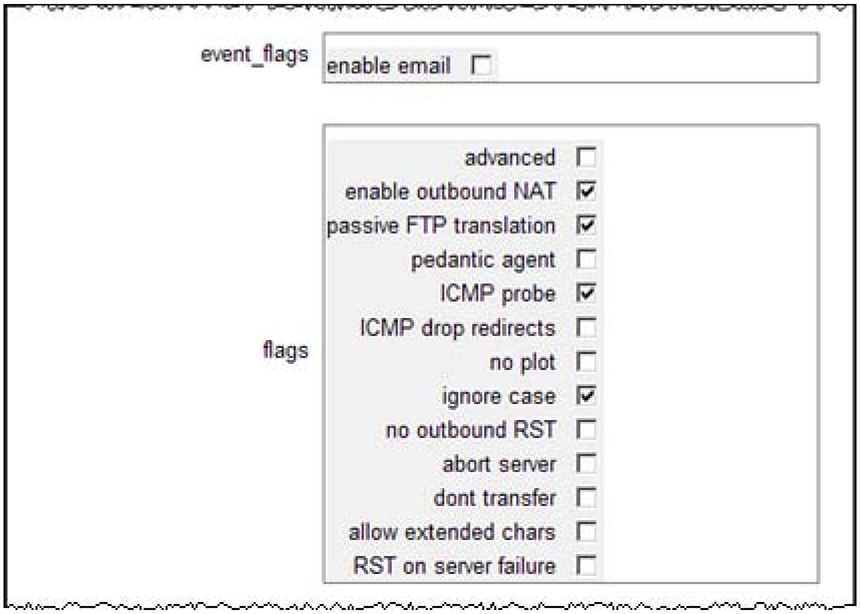
sequence	103
send buffer	32
receive buffer	32
connect timeout	10.0
client timeout	15.0
server timeout	60.0
probe interval	20.0
probe timeout	10.0
strikeout threshold	3
log hours	4
cycle time	5.000
probe delay	10.0
idle timeout	0.0
stale timeout	15.0
sticky netmask	off
command	
from	
to	
SMTP server	

Figure 22 The modify system parameters screen (parameter values)

- **sequence** is the Equalizer-assigned number for the current configuration. This number is assigned by Equalizer and cannot be edited. It can be useful in a failover configuration to determine if the latest configuration has been transferred to the default backup Equalizer when a change is made to the configuration on the default primary Equalizer.

- **send buffer** applies to L7 clusters and is the amount of memory in kilobytes reserved by each L7 proxy process to store outgoing data before it is placed on the network interface.
- **receive buffer** applies to L7 clusters and is the amount of memory in kilobytes reserved by each L7 proxy process to store data that has been received on an interface before it is processed by an L7 proxy process.
- **connect timeout** applies to L7 clusters and is the time in seconds that Equalizer waits for a server to respond to a connection request.
- **client timeout** applies to L7 clusters and is the time in seconds that Equalizer waits before closing an idle client connection.
- **server timeout** applies to L7 clusters and is the time in seconds that Equalizer waits before closing an idle server connection.
- **probe interval** is the target time in seconds at which the internal load balancing daemon checks the internal server status list that shows the result of server probes (performed by the probe daemon). If the list shows the server as down after **strikeout threshhold** attempts to read the table, then the server is marked down. This value is solely a target; the monitoring process adjusts itself based on load. The default value is 20 seconds.
  Also see **probe delay**, below.
- **probe timeout** is the time in seconds that the probe daemon waits for a response once a TCP or ACV probe has been issued.
- **strikeout threshold** is the number of failures (strikes) to respond to a TCP or ACV probe before a server is declared down.
- **log hours** is the target number of hours of plot log data to retain. A zero in this field allots the number of hours based on available memory. Note that the number of hours of log data retained is limited by the amount of memory and disk space. If you define a large number of clusters and servers, this will limit the amount of time over which log data can be retained on Equalizer. For example, a system with 10 clusters each with 10 servers might only be able to retain about 4 hours of log data.
- **cycle time** is time in seconds for the master daemon to make one pass through all of the clusters. This value should not be modified unless recommended by Coyote Point Support.
- **probe delay** is the time in seconds (default is 10) between successive probes of servers by the probe daemon. If a server fails to respond to a probe, the probe daemon marks it as down in its internal server status table. This applies to both TCP probes (always performed by Equalizer) and ACV probes (if enabled). You can override this value for each cluster.
- **idle timeout** applies to L4 clusters and is the time in seconds before reclaiming idle Layer 4 connection records.
- **stale timeout** the length of time that a partially open or closed connection is maintained; see “Managing Stale Connections” on page 53.
- **sticky netmask** enables sticky network aggregation for a subnet (all the connections coming from a particular subnet are directed to the same server in the cluster). See “Enabling Sticky Network Aggregation” on page 54.
- **command**, **from**, **to**, and **subject** enable event handling on the Equalizer; see “Configuring Custom Event Handling” on page 114, in Chapter 7, for more information about these parameters.

The **modify system parameters** screen displays the following flags that affects Equalizer’s operation:



event_flags
enable email □

flags
advanced □
enable outbound NAT ☑
passive FTP translation ☑
pedantic agent □
ICMP probe ☑
ICMP drop redirects □
no plot □
ignore case ☑
no outbound RST □
abort server □
dont transfer □
allow extended chars □
RST on server failure □

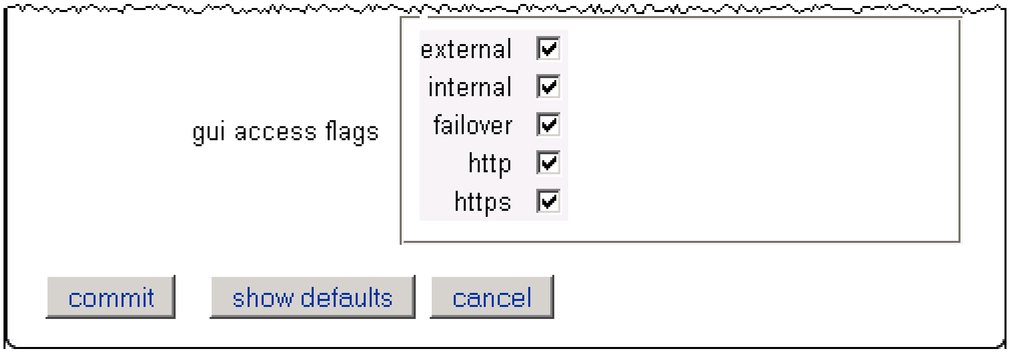
Figure 23 The modify system parameters screen (flags)

- **enable email** enables and disables event triggered email on the Equalizer; see “Configuring Custom Event Handling” on page 114, in Chapter 7.
- **advanced** enables display of advanced parameters and flags by default on all menus for this cluster (i.e., any screen that has an **advanced** check box will have the check box enabled, and all advanced parameters and flags are displayed when the screen is opened).
- **enable outbound NAT** is described under “Enabling Outbound NAT” on page 53.
- **passive FTP translation** is described under “Enabling Passive FTP Connections” on page 53.
- **pedantic agent** applies only when clusters use server agents. When you check this box, Equalizer will treat a server as down when it can probe a server but receives no response from the server’s agent. See Appendix A, ”Using Server Agents”.
- **ICMP probe** enables probing servers using a mix of L4, L7, and ICMP echo probes.
- **ICMP drop redirects** tells Equalizer to drop (i.e., ignore) incoming ICMP redirect messages.
- **no plot** disables the recording of plotting data.
- **ignore case** applies to L7 clusters and is the global setting to ignore case in match expressions. You can override this value per cluster and per match rule. See Chapter 8.
- **no outbound RST** applies to L4 clusters only and causes Equalizer to disable forwarding of untranslated TCP RST (reset) packets. You may want to enable this flag if other network devices (e.g., firewalls, routers, etc.) are logging unexpected source IP messages for the real IPs of servers behind Equalizer (and not the cluster IP). When Equalizer manages a cluster connection, it keeps a record of the connection so it can translate the source IP in a server response before forwarding it. If a client connected to a server IP directly, or if the server sends a RST after Equalizer has already removed the connection record, the RST packet will

not be translated by Equalizer. Enabling this option tells Equalizer to drop any RST packets from servers that do not currently have a Layer 4 connection record that matches the RST packet; with this option disabled (the default) Equalizer will forward all RST packets.

- **abort server** causes Equalizer to terminate server connections without waiting for the server to quiesce.
- **don't transfer** disables the transfer of the Equalizer configuration between failover peers (siblings) when a failure occurs. This is generally used only when using two different Equalizer models/configurations as failover peers (siblings). See "Using Failover with Different Equalizer Models or Versions" on page 51.
- **allow extended chars** is described under "Configuring Support for Extended Characters" on page 55.
- **RST on server failure** applies to L4 clusters only and enables the sending of TCP RST (reset) packets to clients on established connections when the server on the other end of the connection goes down. By default, Equalizer does not send RST packets to clients. This is useful when used with applications like Network File System (NFS) that require a TCP RST to close a connection.

The **modify system parameters** screen displays the following flags that control access to the Administrative Interface. All are enabled by default.



gui access flags

external ☑
internal ☑
failover ☑
http ☑
https ☑

Figure 24 The modify system parameters screen (gui access flags)

- **external** enables access via the currently configured external interface IP address.
- **internal** enables access via the currently configured internal interface IP address.
- **failover** enables access via the currently configured failover IP address.
- **http** enables access via the http:// protocol.
- **https** enables access via the https:// protocol.

You can check the currently configured IP addresses on the **View > Equalizer** status screen.

If you should attempt to disable access for the IP address or protocol that you are using for the current browser session, an error is displayed. If you should disable all access to the Administrative Interface (e.g., by manually editing the eq.conf file), see "Restoring Access to the Administrative Interface" on page 191 for how to re-enable access.

Setting Up a Failover Configuration

You can set up two Equalizers in a hot backup, or failover, configuration. In such a configuration, one of the systems handles incoming requests (the primary system), while the other (the backup system) waits for a failure to occur and automatically takes over if the Equalizer that is currently handling requests fails. The two Equalizers are called *failover peers* or *siblings* in such a configuration.

To use a second Equalizer as a hot backup or failover peer, you need to install both Equalizers so their network interfaces have corresponding configurations (see Figure 9 on page 13):

- You must plug the external interface of the backup unit into the same hub or switch into which the external interface of the primary unit is plugged.
- You must plug the server (or internal) interface of the backup unit into the same hub or switch into which the server interface of the primary unit is plugged.
- For failover configuration between two switch models, connect a cable from one Equalizer's switch interface to the others (see Figure 10 on page 14).

Note – Be sure that you do *not* create a loop between the external and internal interfaces.

You must designate one of the Equalizers as the *preferred primary*; the second is the *preferred backup*. When you boot both Equalizers at the same time, the preferred primary Equalizer is activated. If the primary Equalizer fails, the backup takes over. When you bring the failed unit back online, it assumes the backup role until another failure occurs or you reboot its peer.

A failover configuration requires one or two additional IP addresses, called the *failover aliases*. In a dual network configuration, failover aliases must be supplied for both the internal and external interfaces; in a single network configuration, only an internal alias is needed. These IP addresses are initially assumed by the preferred primary system and are used as the network-visible interfaces of the Equalizer, instead of the addresses assigned to the individual Equalizers via the **eqadmin** interface. When a failover occurs, the failover aliases are then assumed by the backup system.

When Equalizer is brought online, it checks to make sure that the configured network interfaces are link active. In the case of the internal interface, Equalizer attempts to ping a configured server or failover peer. If the interfaces are not active, Equalizer sits in a loop waiting for them to become active (and sends comments to the console). Once the network interfaces are active, the failover peers begin a negotiation in which one system becomes the primary unit and the other becomes the backup unit. This is accomplished by the backup system performing a reboot.

When a backup Equalizer loses contact with its failover peer, it tries to determine the cause. If it cannot identify the cause, it will try to assume the primary role. It checks that no other system has configured the gateway IP address or virtual cluster addresses. If these tests are successful, the Equalizer assumes those IP addresses and starts handling traffic.

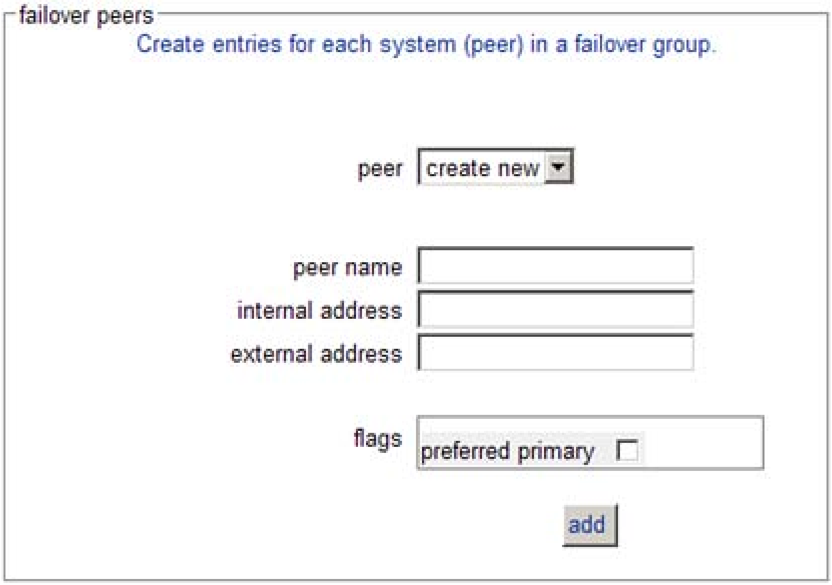
A *partition* occurs when both systems are unable to communicate with each other and both Equalizers enter primary mode. When the partition is healed and both units regain communication, the two systems resolve this dispute by choosing one system to reboot itself. Generally, this means that the system that is configured as the default backup will reboot; upon coming back up, it will enter backup mode.

> **Note –** Any switch, such as one from Cisco or Dell, that comes with Spanning Tree enabled by default can cause a communication problem in a failover configuration when one or both of the Equalizers are dual-port models. This problem occurs at bootup because the switch disables its ports for roughly 30 seconds to listen to BPDU (bridge protocol data unit) traffic. The 30-second pause causes both Equalizers to attempt to become the primary unit; the default backup continually reboots.
>
> To repair this condition, either disable Spanning Tree or enable PortFast for the ports connected to the Equalizers. This enables the ports to act as normal hubs and accept all traffic immediately.

Since different Equalizer models and software revisions have varying configuration parameters, it is recommended that both of the failover peers are the same model Equalizer running the same software version. See the section “Using Failover with Different Equalizer Models or Versions” on page 51 for more information on setting up a failover pair with two different Equalizer models.

You’ll need to create the two failover peer definitions, set failover timing parameters, and define the failover aliases on both systems. The following procedure leads you through the failover setup process on both Equalizers in the failover pair.

1. Log into the Equalizer Administration Interface in Edit mode on the failover peer that will assume the *preferred primary* role. (Configuring and rebooting the preferred primary Equalizer first ensures that it assumes the primary role.)
2. Select **Equalizer** > **Global Configuration** from the main menu bar; then select **menu** > **Failover** from the **modify system parameters** screen. The **failover configuration** screen appears in the right frame.
3. In the failover peers section (see Figure 25 on page 48), make sure that **create new** is selected in the **peer** drop-down.



failover peers

Create entries for each system (peer) in a failover group.

peer: create new

Figure 25 Failover peers section of the failover configuration screen for a dual network configuration

4. Enter a unique name for the new peer in the **peer name** field.

5. Enter the internal and external addresses for this peer in the **internal address** and **external address** fields.

   In single network mode, you will see one address and corresponding network mask depending on the type of Equalizer. Equalizer models with an integrated switch use the internal address when in single network mode. All other models use the external address when in single network mode.

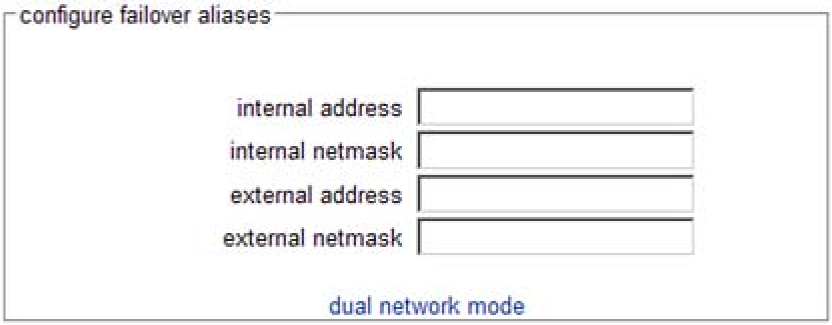
6. Check **preferred primary** if the peer being defined is the preferred primary. If the peer being defined is the backup peer, do *not* check this flag.

7. Click the **add** button to add the peer.

   (Clicking the **add** button adds the peer and disables failover until you **commit** the changes in Step 11.)

8. Configure the second backup peer by repeating Step 3 through Step 7.

9. Specify the **address** and **netmask** for the failover aliases (see Figure 26 on page 49).



configure failover aliases

internal address

internal netmask

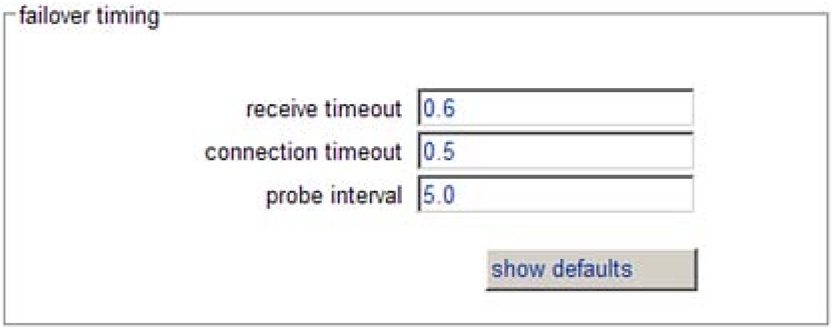
external address

external netmask

Figure 26 Failover aliases section of the failover configuration screen for a dual network configuration

The alias addresses are unique IP addresses assigned to the failover pair, and are passed between them whenever a failover occurs. The netmask can be left blank and it will default to the same as the associated interface. The Equalizer that is running in primary mode assumes these aliases; the servers should use the internal address (when in dual network mode) or the single address (when in single network mode) as their default gateway. If you are in dual network mode and running Envoy, the external failover alias is used for DNS queries to Envoy.

10. You should accept the default failover timing parameters. These parameters affect how the peers try to heartbeat each other.



failover timing

receive timeout 0.6

connection timeout 0.5

probe interval 5.0

Figure 27 Failover timing section of the failover configuration screen for a dual network configuration

The **receive timeout** is the time in seconds that Equalizer allows to receive a response from its sibling before it times out. The **connection timeout** is the time in seconds allowed to establish a TCP connection with its sibling. When either of these timeouts occur, that counts as one of the strikeouts that occurs before the backup becomes the primary (three strikeouts must occur before the backup takes the primary role). The **probe interval**. is the number of seconds Equalizer waits between attempts to exchange status information.

Normally the default values are the best to use; however, if you notice the log files contain too many false positives (messages that Equalizer has regained contact with its peer) you may want to increase the values.

11. Click the **commit & reboot** button.

    Errors are reported when a failover configurations is not successfully committed. If successful, you will be prompted to reboot immediately. (Click the **cancel** button if you want to wait to reboot the Equalizer.)

    > **Note –** Both Equalizers must reboot in order for the failover configuration to work. Also note that selecting the **commit & reboot** button on one of the peers does not cause the second Equalizer (the peer that is not the system being configured) to reboot.

    As Equalizer reboots:

    - Watch the console for messages indicating that the Equalizer has successfully assumed the primary or backup role.
    - Check the event logs (**View** > **Event Log** in the Administrative Interface) for each Equalizer to see that there are no related errors.
    - Make sure that “Successfully assumed PRIMARY role” appears in the log for the preferred primary system; the default backup system’s log should contain “Successfully assumed BACKUP role”.

12. Repeat this procedure on the default *backup* Equalizer peer starting at Step 2.

13. If you have not already rebooted the two Equalizers as part of the above procedure, reboot the *preferred primary* system first, then the backup system.

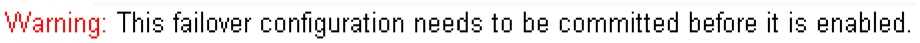
Modifying or Deleting a Failover Configuration

To make changes to a peer’s address or to delete a peer, select it from the **peer** drop-down. The buttons **modify** and **delete** appear. You can make changes and click the **modify** button. To delete a peer, click the **delete** button.

Using either the **modify** or **delete** button *disables failover* until you **commit** the changes. If the system reboots while failover is disabled, it will start up in standalone mode.

> **Note –** If both systems in a failover pair start in standalone mode, each will assume the cluster aliases and neither will assume a failover alias, resulting in nothing working. To resolve this type of problem, configure and commit failover on both Equalizers, and then reboot both.

If failover is disabled, the following appears at the top of the **failover configuration** screen:



Warning: This failover configuration needs to be committed before it is enabled.

Using Failover with Different Equalizer Models or Versions

We recommend that you use the same model Equalizer (e.g., E350si, E450si, etc.) for both systems in a failover pair and that both Equalizers are running the same version of the software (e.g., 7.2.4). This is recommended because the default behavior of Equalizer is to maintain the same configuration files on both systems in a failover pair (so that you don't need to manually update both Equalizers with the same configuration changes). Changes committed to one system are copied to the configuration files on the other system.

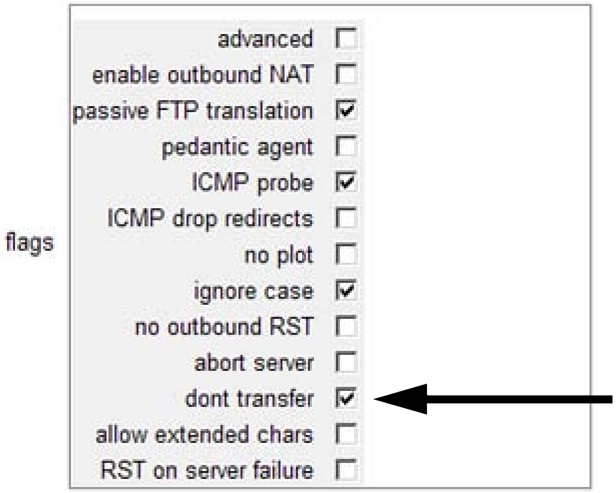
For this reason, it is *not* generally recommended to deploy two different Equalizer models in a failover pair. However, some sites prefer to upgrade failover pairs to new hardware one at a time rather than deploying new models for both failover systems at the same time. If you are pairing an older model with a newer model (such as a newer switch-integrated E350si or E450si system with an older E350 or E450 non-switch system in single network mode), the differences in hardware configuration on these models *require* that the systems do not share changes to their configuration files by setting a special flag (**dont transfer**) on *both* Equalizers.

Similarly, some sites prefer to upgrade one Equalizer in a failover pair to a major new software revision and leave the other running the previous release for a limited period of time, in case there are any unforeseen configuration problems.

> **Note –** Whenever the **dont transfer** flag is enabled, you must manually perform any changes to your Equalizer and cluster configuration (such as adding/removing clusters or servers, changing system parameters, etc) on *both* Equalizers in the failover pair.

To prevent Equalizers in a failover pair from sharing changes to configuration files, perform the following procedure on both systems:

1. Select **Equalizer > Global Configuration** from the main menu.
2. At the bottom of the **modify system parameters** screen, in the **flags** section, check the box labeled **dont transfer**.



flags	advanced	☐
	enable outbound NAT	☐
	passive FTP translation	☑
	pedantic agent	☐
	ICMP probe	☑
	ICMP drop redirects	☐
	no plot	☐
	ignore case	☑
	no outbound RST	☐
	abort server	☐
	dont transfer	☑
	allow extended chars	☐
	RST on server failure	☐

Figure 28 Modify system parameters flags

3. Click the **commit** button to save the flag change.

4. Perform Steps 1 to 3 on the other Equalizer in the failover pair.

Upgrading Failover Configurations from Versions prior to 7.2.1

The upgrade script contains facilities to migrate a version 7.1 format failover configuration (stored in */etc/eq.static*) to the new format used in 7.2 and later systems.

When the upgrade script runs, it will detect the presence of a valid configuration in the *eq.static* file. If it finds this file, the script prompts you whether to migrate the failover configuration.

If you respond ‘y’ to the upgrade script’s prompt, the configuration file will be migrated to the upgrade partition, and the following message displayed:

```
IMPORTANT NOTE: configuration file transfers will be disabled when the
system reboots. You may re-enable configuration sharing by clearing
the dont transfer checkbox in the equalizer global parameters page.

If you are configuring failover between two different types of
Equalizers, where one contains a built-in switch and the other does
not, configuration file transfers must remain disabled between the two
systems. (See release notes)
```

This indicates that when the system reboots, the **dont transfer** flag is set and any changes that are made to the configuration of this system will not be shared with the failover peer. You may clear the **dont transfer** flag once the system reboots, provided the failover pair is not both operating in single network mode and a combination of a switch-integrated system with a non-switch system. See “Using Failover with Different Equalizer Models or Versions” on page 51 for more information.

Changing the Network Mode between Single and Dual

It is important to delete the failover configuration before changing the network mode between single and dual network on an Equalizer that is already configured for failover. If the network mode is changed before the failover configuration is deleted, the web browser interface will become unusable because the configuration parser generates error messages stating that the failover configuration does not match the network mode.

Troubleshooting Changes between Network Modes without Deleting Failover Configurations First

The following manual procedure deletes the failover configuration and should only be used if the network mode was changed without first deleting the failover configuration. You should follow this procedure with the assistance of a member of Coyote Point’s technical support team.

1. Log into the Equalizer via SSH using the eqsupport account or as root via the serial port.
2. `# mount –w /` (if using the eqsupport account, you must use **su** first)
3. Edit the file to remove the *interface* stanza and save it.

   # ee /var/eq/eq.conf (vi may be used as well)
4. `# shadow /var/eq/eq.conf`
5. `# shutdown -r now`

   This command reboots Equalizer.

After Equalizer comes back up, you can create a failover configuration.

Enabling Outbound NAT

If you use a reserved network configuration and the servers on the non-routable network must be able to communicate with hosts on the Internet, you must configure Equalizer to perform outbound network address translation (NAT). When outbound NAT is enabled, Equalizer translates the source IP address in all packets originating from the servers on the reserved network to Equalizer's external IP address, so that clients do not see packets originating from non-routable IP addresses.

> **Note –** If you use outbound NAT in a failover configuration, you should enable outbound NAT on both units in case a failover actually occurs.

To enable Equalizer to perform outbound NAT, follow these steps:

1. Log into the Equalizer Administration Interface in edit mode.
2. Select **Equalizer > Global Configuration** from the Equalizer menu in the main menu bar. The **modify system parameters** screen appears in the right frame (see Figure 22 on page 43).
3. Check the **enable outbound NAT** checkbox.
4. Click the **commit** button.

Enabling Passive FTP Connections

If your servers are on a network the outside world cannot reach, consider enabling Equalizer’s passive FTP translation option. This option causes the Equalizer to rewrite outgoing FTP PASV control messages from the servers so they contain the IP address of the virtual cluster rather than that of the server.

To enable passive FTP translation, follow these steps:

1. Log into the Equalizer Administration Interface in Edit mode.
2. Select **Equalizer > Global Configuration** from the Equalizer menu in the main menu bar. The **modify system parameters** screen appears in the right frame (see Figure 22 on page 43).
3. Check the **passive FTP translation** checkbox.
4. Click the **commit** button.

Managing Stale Connections

The stale connection timeout is the length of time that a partially open or closed connection is maintained. If a client fails to complete the TCP connection termination handshake sequence or sends a SYN packet but does not respond to the server’s SYN/ACK, Equalizer marks the connection as incomplete. Equalizer reclaims connections in the incomplete state when the stale connection timeout expires. When Equalizer reclaims a connection, it sends a TCP RST (reset) packet to the server, enabling the server to free any resources associated with the connection. Stale connections apply to Layer 4 (L4) only.

If you change the stale timeout setting while partially established connections are currently in the queue, those connections will be affected by the new setting.

Note – Reducing the stale connection timeout can be an effective way to counter the effects of SYN flood attacks on server resources. A stale connection timeout of 10 seconds would be an appropriate value for a site under SYN flood attack.

To set the stale connection timeout, follow these steps:

1. Log into the Equalizer Administration Interface in Edit mode.
2. Select **Equalizer** > **Global Configuration** from the Equalizer menu in the main menu bar. The **modify system parameters** screen appears in the right frame (see Figure 22 on page 43).
3. Enter a value, in seconds, for **stale timeout** (default is 15 seconds).
4. Click the **commit** button.

Enabling Sticky Network Aggregation

Sticky network aggregation enables Equalizer to correctly handle sticky connections from ISPs that use multiple proxy servers to direct user connections. When you enable sticky network aggregation, all the connections coming from a particular network are directed to the same server. (Typically, all the servers in a proxy farm are on the same network.)

When you enable sticky network aggregation, Equalizer routes all the connections from a particular network to the same server. The netmask value indicates which portion of the address Equalizer should use to identify particular networks. The mask corresponds to the number of bits in the network portion of the address:

- 8 bits corresponds to a Class A network
- 16 bits corresponds to a Class B network
- 24 bits corresponds to a Class C network

In previous versions of Equalizer, enabling sticky network aggregation was the equivalent of setting the sticky network aggregation mask to 24 bits (that is, Equalizer routed all connections from the same class C network to the same server).

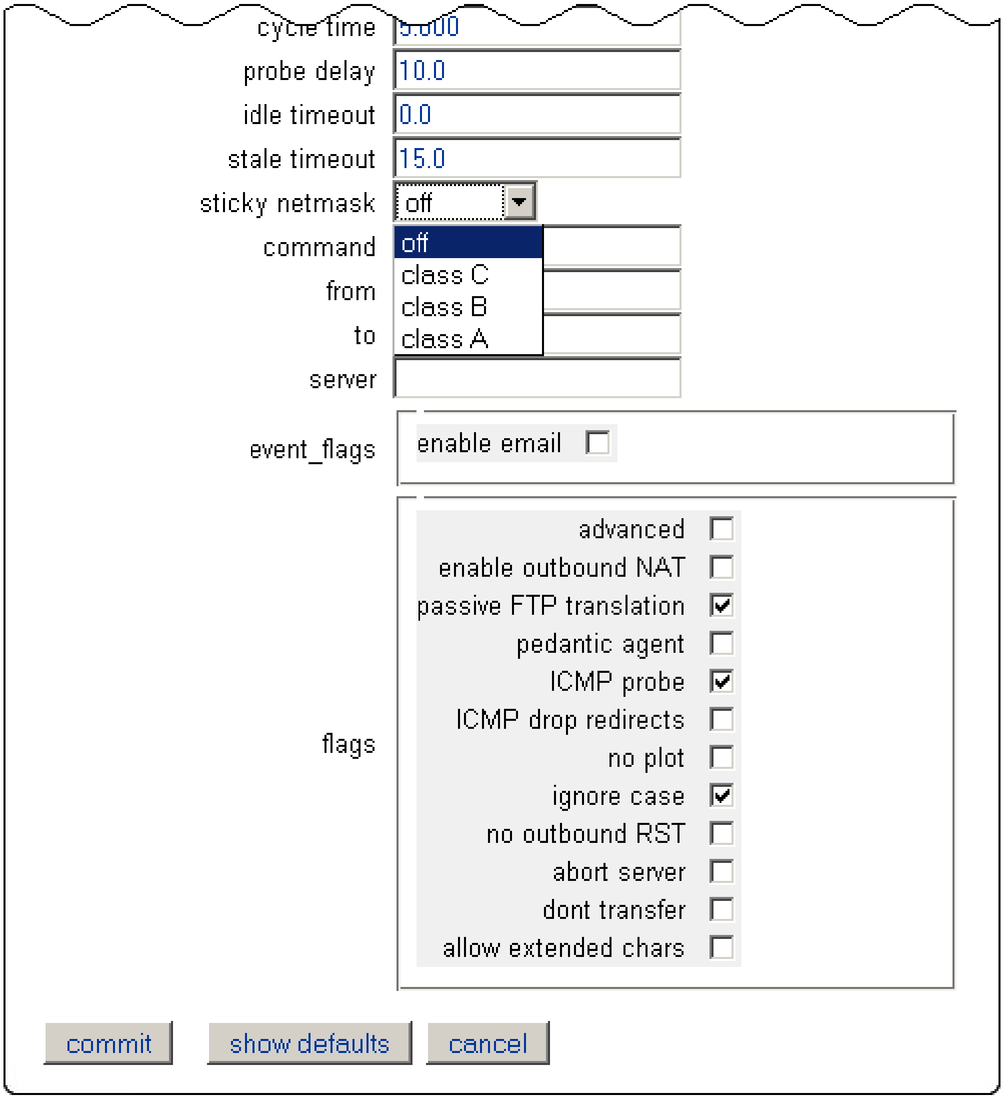
Note – A potential drawback of using sticky network aggregation is that all users connecting through a particular proxy farm might be directed to the same server. In practice, this has not been a problem. Equalizer’s load-balancing algorithms direct other visitors to different servers to keep the load balanced.

Sticky network aggregation is applicable only for Layer 4 TCP and UDP clusters. For Layer 4 clusters with the **spoof** flag disabled and for Layer 4 clusters configured for FTP, a sticky record is maintained for each connection whether sticky network aggregation is enabled or not.

To enable sticky network aggregation, follow these steps:

1. Log into the Equalizer Administration Interface in Edit mode.
2. Select **Equalizer** > **Global Configuration** from the Equalizer menu in the main menu bar. The **modify system parameters** screen appears in the right frame (see Figure 22 on page 43).

3. Enable sticky network aggregation by selecting a **sticky netmask** from the pull-down menu shown below.



cycle time 5.000
probe delay 10.0
idle timeout 0.0
stale timeout 15.0
sticky netmask off
command
from
to
server

event_flags
enable email □

flags
advanced □
enable outbound NAT □
passive FTP translation ☑
pedantic agent □
ICMP probe ☑
ICMP drop redirects □
no plot □
ignore case ☑
no outbound RST □
abort server □
dont transfer □
allow extended chars □

Figure 29 Enabling sticky network aggregation

4. Click the **commit** button.

Note – If you are using two Equalizers in a failover configuration, you must set the sticky network aggregation mask identically for both Equalizers.

Configuring Support for Extended Characters

By default, support for 8-bit ASCII and multibyte UTF characters in URIs is disabled; Equalizer returns a **400 Bad Request** error when a request URI contains 8-bit or multibyte characters. To enable support for 8-bit and multibyte characters in URIs, turn on the **allow extended characters** flag in the **global parameters** as shown in the procedure below.

> **Caution –** There are potential risks to enabling this option, because it allows Equalizer to pass requests that violate RFC2396; load-balanced servers may be running software that is incapable of handling such requests. Therefore, ensure that your server software is capable of handling URIs containing extended characters and will not serve as a potential weak point in your network *before* you enable extended characters.

1. Log into the Equalizer Administration Interface in Edit mode.
2. Select **Equalizer** > **Global Configuration** from the main menu bar; the **modify system parameters** screen is displayed (see Figure 22 on page 43).
3. In the **flags** section at the bottom of the **modify system parameters** screen, enable the **allow extended chars** check box.
4. Select the **commit** button.

Changing the Administration Passwords

An administrator logged in under Edit mode can change both the View password (**look** login) and Edit password (**touch** login).

To change the view or edit a password (see Figure 30 on page 56), follow these steps:

1. Log into the Equalizer Administration Interface in Edit mode.
2. Select **Equalizer** > **Global Configuration** from the main menu bar; then select **menu** > **Change Passwords** from the **modify system parameters** screen.

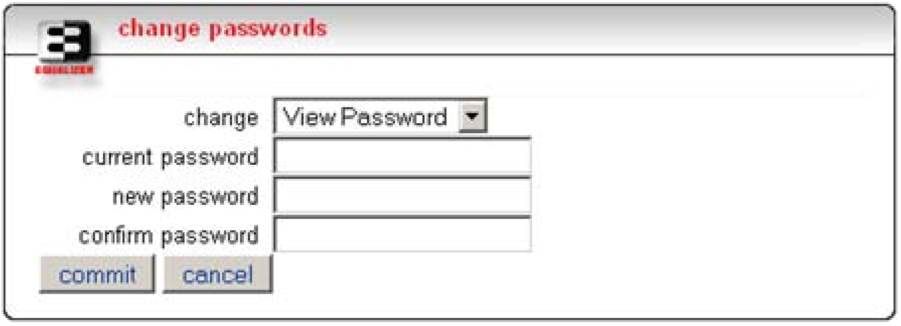


Figure 30 Change passwords screen

3. Select the password to be changed: **View Password** or **Edit Password**.
4. For **Edit password** only, enter the current password in the **current password** field. You can change the View password without specifying the current password.
5. Enter the new password in the **new password** field and then confirm it by entering it again in the **confirm password** field.

6. Select the **commit** button.

> **Note –** If you have lost or forgotten the Edit mode password, you can set it through the console-based Equalizer Configuration Utility. For more information, refer to "Changing the Administration Interface Password" on page 26.

Configuring Static Routes

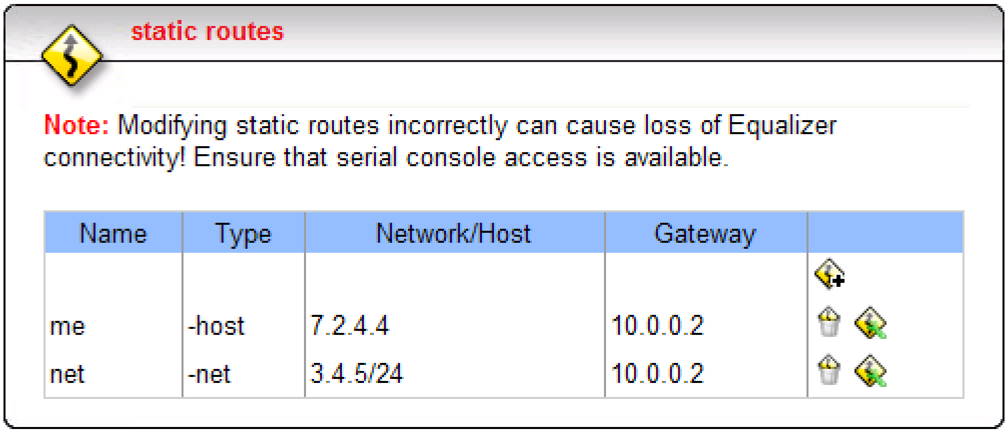
Static routes are commonly used to specify routes to IP addresses via gateways other than the default.

A default gateway is specified when you configure Equalizer via the **eqadmin** character based interface. If you need to access systems on a subnet that cannot be reached via this gateway, then you need to specify a **static route** to those systems through the gateway for that subnet.

Static routes on Equalizer are specified using the browser-based Administration Interface. Static routes can also be defined from the command line via the serial interface, but we recommend you use the browser interface exclusively to manage static routes on Equalizer. The interface manages changes to the */var/etc/rc.conf-eq* file for you, and updates Equalizer's routing tables (displayed using the **netstat -nr** shell command) as you add and delete them.

Adding a Static Route

1. Log into the Equalizer Administration Interface.
2. Select **Equalizer** > **Global Configuration** from the main menu bar; then select **menu** > **Manage Static Routes** from the **modify system parameters** screen. The **static routes** screen appears in the right frame.



static routes

Note: Modifying static routes incorrectly can cause loss of Equalizer connectivity! Ensure that serial console access is available.

Name	Type	Network/Host	Gateway	
me	-host	7.2.4.4	10.0.0.2	
net	-net	3.4.5/24	10.0.0.2	

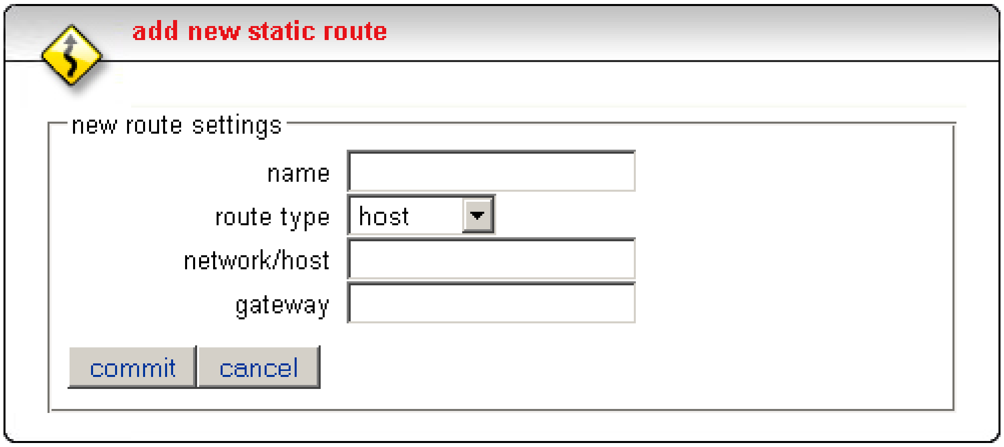
Figure 31 The static routes screen

The table contains the following information for each configured static route on the system::

Name	An identifier for the route.
Type	Either **host** to specify a route to a host address, or **net** to specify an address for a subnet.

Network	The IP address for the host or subnet. Can be specified as a Classless Internet Domain Routing (CIDR) address to specify a netmask; for example: 192.168.1.0/24.
Gateway	The IP address of the gateway used to reach the host or subnet.

3. Click on the Add icon . The **add static route** screen appears:



add new static route

new route settings

name

route type host

network/host

gateway

Figure 32 The add static route screen

4. Enter the parameters for the route, and select **commit**. You are returned to the **static routes** screen, which now displays the route you added.

Modifying a Static Route

1. Log into the Equalizer Administration Interface.
2. Select **Equalizer** > **Global Configuration** from the main menu bar; then select **menu** > **Manage Static Routes** from the **modify system parameters** screen. The **static routes** screen appears in the right frame.
3. Highlight the route you want to change in the table and select the Edit icon . The **edit static route** screen is displayed:
4. Edit the values shown as needed and select **commit** to submit your changes. You are returned to the **static routes** screen, which now displays the updated route.

Deleting a Static Route

1. Log into the Equalizer Administration Interface.
2. Select **Equalizer** > **Global Configuration** from the main menu bar; then select **menu** > **Manage Static Routes** from the **modify system parameters** screen. The **static routes** screen appears in the right frame.
3. Highlight the route you want to delete in the table and select the Delete icon . A confirmation screen appears.

4. Select **delete** to delete the route. You are returned to the **static routes** screen, from which the route has been removed.

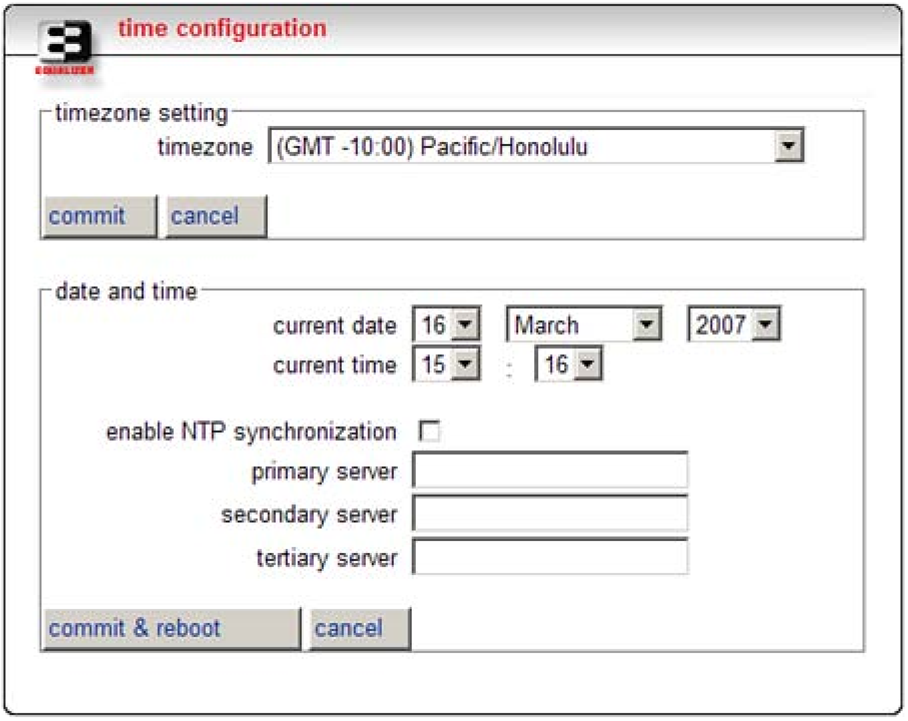
Managing System Time and NTP

Through Equalizer's Administrative Interface, you can:

- set the time zone
- set the system date and time
- set up to three Network Time Protocol (NTP) servers, and enable or disable synchronization with these servers

Setting the Time Zone

1. Select **Equalizer > Global Configuration**.
2. Select **menu > Manage System Time**. The **time configuration** screen is displayed:



time configuration

timezone setting

timezone (GMT -10:00) Pacific/Honolulu

date and time

current date 16 March 2007

current time 15 : 16

enable NTP synchronization □

primary server

secondary server

tertiary server

Figure 33 The time configuration screen

3. Make a selection from the drop down box in the **timezone setting** section, then select the **commit** button in that section.

Setting the System Date and Time

1. Select **Equalizer** > **Global Configuration**.
2. Select **menu > Manage System Time**. The **time configuration** screen is displayed (Figure 33 on page 59)
3. Use the drop-down boxes at the top of the **date and time** field to manually set the date and time.
4. Select the **commit & reboot** button; then select **OK** to confirm and reboot Equalizer.

Enabling or Disabling NTP

1. Select **Equalizer** > **Global Configuration**.
2. Select **menu > Manage System Time**. The **time configuration** screen is displayed (Figure 33 on page 59)
3. Do one of the following:
   a. **To disable NTP**: turn off the **enable NTP synchronization** check box.
   b. **To enable NTP**: turn on the **enable NTP synchronization** check box and type in the name of an NTP server into the **primary server** text box (you can also specify two additional servers to be used in sequence if the first is unavailable). See the section “Selecting an NTP Server” on page 60 for help choosing an appropriate NTP server.
4. Select the **commit & reboot** button; then select **OK** to confirm and reboot Equalizer.

Selecting an NTP Server

We recommend that you specify NTP pool servers appropriate for your geographic location. Selecting a pool server means that you are specifying an alias that is assigned by **ntp.isc.org** to a list of time servers for a region. Thus, NTP pool servers are specified by geography. The following table shows the naming convention for servers specified by continent:

Worldwide	pool.ntp.org
Asia	asia.pool.ntp.org
Europe	europe.pool.ntp.org
North America	north-america.pool.ntp.org
Oceania	oceania.pool.ntp.org
South America	south-america.pool.ntp.org

To use the continent-based NTP pool servers for Europe, for example, you could specify the following pool servers in Equalizer’s **time configuration** screen:

```
0.europe.pool.ntp.org
1.europe.pool.ntp.org
2.europe.pool.ntp.org
```

You can also specify servers by country. So, for example, to specify a UK based time server pool, you would use:

```
0.uk.pool.ntp.org
1.uk.pool.ntp.org
2.uk.pool.ntp.org
```

Or, for the US, you would use:

```
0.us.pool.ntp.org
1.us.pool.ntp.org
2.us.pool.ntp.org
```

Be careful when using country based NTP pool servers, since some countries contain a very limited number of time servers. In these cases, it is best to use a mix of country and continent based pool servers. If a country has only one time server, then it is recommended you use a time server pool based in another nearby country that supports more servers, or use the continent based server pools.

For example, Japan has 6 (six) time servers as of the date this document was published. The organization that maintains time server pools recommends using the following to specify time server pools for Japanese locations:

```
2.jp.pool.ntp.org
0.asia.pool.ntp.org
2.asia.pool.ntp.org
```

For more information on choosing NTP pool servers, please see the NTP pool server web pages at:

```
http://ntp.isc.org/bin/view/Servers/NTPPoolServers
```

Saving or Restoring Your Configuration

Equalizer enables you to save or back up a configuration or restore a saved configuration.

> **Note –** Equalizer passwords are not saved or restored, but IP configuration, clusters, and failover information are saved.

Saving Your Configuration

Use the Backup/Restore Configuration command to save your Equalizer configuration to a file or to load a saved configuration.

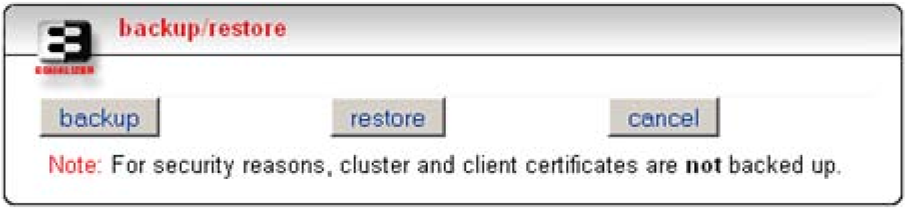
When you save your configuration, Equalizer wraps up the following information in a binary file:

- `/var/eq/eq.conf`, which contains the cluster/server configurations that appear in the left pane of the administrative interface.
- `/var/eq/envoy.conf`, which is the Envoy configuration (if Envoy is installed); it contains geographic cluster and site information from the left pane of the administrative interface.
- `/var/eq/licenses`, which contains licensing information.
- Configuration files from `/etc` (including `hosts`, `master.passwd`, `ntp.conf`, `passwd`, `rc.conf-eq`, `resolv.conf`, `syslog.conf`) and `/etc/ssh` (including `ssh_config`, `sshd_config`, and host keys).

Backing Up Your Configuration

To back up your current configuration to a file, follow these steps:

1. Log into the Equalizer Administration Interface in Edit mode.
2. Select **Equalizer** > **Global Configuration** from the main menu bar; then select **menu** > **Backup/Restore Configuration** from the **modify system parameters** screen. The **backup/restore** screen (see Figure 34 on page 62) appears in the right frame.



Note: For security reasons, cluster and client certificates are **not** backed up.

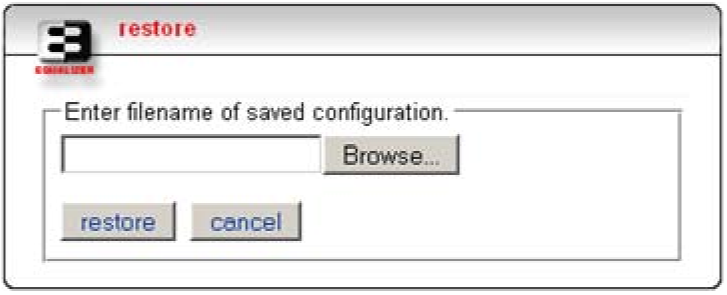
Figure 34 Backing up your Equalizer configuration

3. Click the **backup** button.
4. When prompted, specify the location where you want to save the configuration file; then click **OK**.

Restoring a Saved Configuration

To restore a saved configuration, follow these steps:

1. Log into the Equalizer Administration Interface in Edit mode.
2. Select **Equalizer** > **Global Configuration** from the main menu bar; then select **menu** > **Backup/Restore Configuration** from the **modify system parameters** screen. The **backup/restore** screen (see Figure 34 on page 62) appears in the right frame.
3. Click the **restore** button.



Enter filename of saved configuration.

Figure 35 Restoring a saved configuration

4. Click **Browse...** to locate and select the configuration file that you want to use to restore the Equalizer configuration.

5. Click **restore** to upload the configuration file. Equalizer automatically reboots to update the configuration.

> **Note –** Be very careful when restoring configurations. The saved IP information could cause conflicts on the network if the restored file comes from another Equalizer (for example, its backup). If this happens, use the console-based Equalizer Configuration Utility to re-configure the restored configuration's IP addresses. See "Configuring Equalizer Hardware" on page 21.

Shutting Down Equalizer

Before turning off Equalizer or disconnecting the power, you should perform a clean shutdown. To shut down Equalizer cleanly, follow these steps:

1. Log into the Equalizer Administration Interface in Edit mode.
2. Select **Equalizer > Shut Down Equalizer** from the Equalizer menu in the main menu bar. A confirmation dialog box appears (see Figure 36 on page 63).

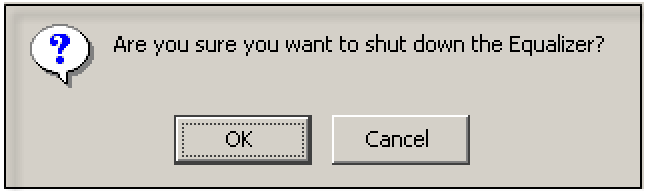

Are you sure you want to shut down the Equalizer?

Figure 36 The Shutdown confirmation dialog box

3. In the confirmation dialog box, click **OK** to confirm that you really want to shut down Equalizer (or click **Cancel** to abort the shutdown request). If you click **OK**, Equalizer immediately initiates the shutdown cycle. After waiting 30 seconds, you can safely power down the Equalizer.

Rebooting Equalizer

Rebooting Equalizer shuts it down cleanly and then restarts the system. To reboot the Equalizer:

1. Log into the Equalizer Administration Interface in edit mode.
2. Select **Equalizer > Reboot** from the Equalizer menu in the main menu bar. A confirmation dialog box appears.
3. In the confirmation dialog box, click **OK** to confirm that you really want to reboot Equalizer. The system will shutdown and then reboot; wait a few minutes before attempting to log into the system again.

6 Administering Virtual Clusters

A virtual cluster is a collection of servers with a single network visible IP address. All client requests come into Equalizer through a cluster IP address, and are routed by Equalizer to the appropriate server in the cluster, according to the load balancing options set on the cluster.

The following sections show you how to create and manage virtual clusters and the servers they contain:

Working with Virtual Clusters

A virtual cluster acts as the network-visible front-end for a group of servers. Use the Equalizer Administration Interface to add, configure, or remove virtual clusters and the servers that belong to them. The figure below shows a conceptual diagram of an Equalizer with three clusters.

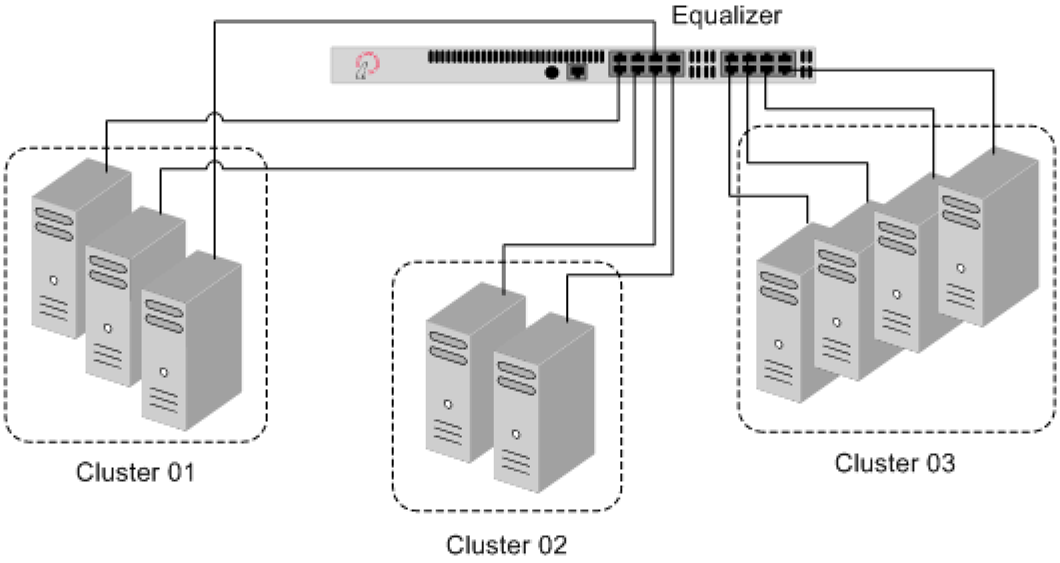


Figure 37 An Equalizer with three defined clusters

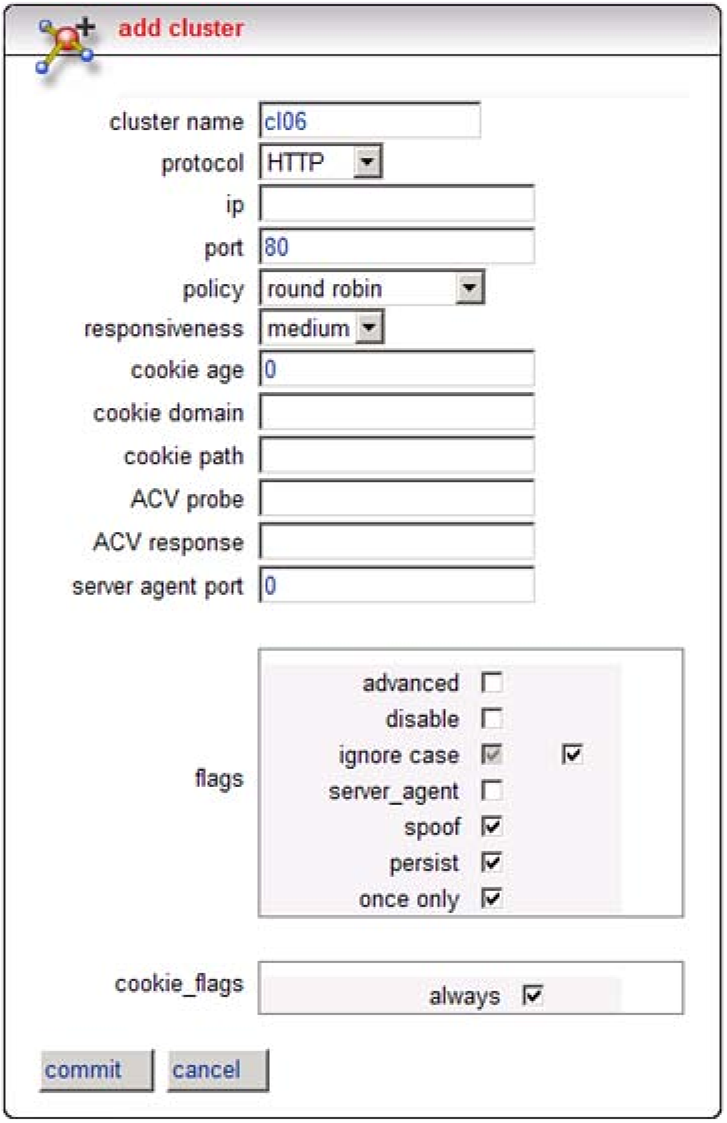
The parameters you specify when setting up a virtual cluster determine how the Equalizer manages connections between the Equalizer and the servers in a cluster, and how incoming requests are routed through the Equalizer to the cluster. Before beginning to define a cluster, we recommend that you read this chapter in its entirety so that you can:

- Select an IP address to use for the cluster and for each server in the cluster.
- Determine the protocol (HTTP, HTTPS, Layer 4 TCP, or Layer 4 UDP) that will be used to communicate between the Equalizer and the servers in the cluster.
- Determine the load balancing policy (round robin, static weight, adaptive, fastest response, least connections, or server agent) that the Equalizer will use to decide how to route incoming requests to the servers in the cluster.
- Determine the responsiveness of the Equalizer to changing loads; that is, how often and to what degree does the Equalizer adjust the dynamic weights of the servers in the cluster.
- Determine the optional settings and flags to be used (if any) on the cluster and its servers.

Adding a Virtual Cluster

To add a new virtual cluster, follow these steps:

1. Log into the Equalizer Administration Interface in edit mode.
2. Select **Add** > **Virtual Cluster** from the main menu bar. The **add cluster** screen appears in the right frame (see Figure 38). Another way to display this screen is to view the Equalizer status and select **Add Virtual Cluster** from the local menu.



add cluster

cluster name	cl06
protocol	HTTP
ip	
port	80
policy	round robin
responsiveness	medium
cookie age	0
cookie domain	
cookie path	
ACV probe	
ACV response	
server agent port	0

flags

advanced	□	
disable	□	
ignore case	☑	☑
server_agent	□	
spoof	☑	
persist	☑	
once only	☑	

cookie_flags

always ☑

Figure 38 Adding a virtual cluster

3. Enter the **cluster name**, which is the logical name for the cluster, or accept Equalizer’s default. Each cluster must have a unique name that begins with an alphabetical character (for example, *CPImages*).
4. Select one of the following **protocol** types for the cluster:

- **HTTP**, Equalizer passes web server requests and route requests to particular servers based on the content of the request and various load-balancing criteria. (This protocol supports Layer 7 load balancing.)
- **HTTPS**, Equalizer passes secure web server requests and route requests to particular servers based on the content of the request and various load-balancing criteria. (This protocol supports Layer 7 load balancing.)
- **L4 TCP**, Equalizer passes TCP-based requests and route requests based on configured load balancing criteria, the IP address, and TCP port number. Load balancing based on generic connection protocols can be quite efficient; however, routing decisions cannot take into account the content of the request. (This protocol supports Layer 4 load balancing.)
- **L4 UDP**, Equalizer passes TCP-based requests and route requests based on configured load balancing criteria, the IP address, and UDP port number. Load balancing based on the generic connection protocols can be quite efficient, but routing decisions cannot take into account the content of the request. (This protocol supports Layer 4 load balancing.)

When you first open the **add cluster** screen, it displays the fields for the HTTP protocol, as shown in Figure 38 on page 67. When you change the protocol in the drop down box, the available fields on the form change to include the fields appropriate for the chosen protocol.

> **Note –** On the Equalizer E250si, Layer 7 content-based load balancing is not supported; HTTP and HTTPS are not available choices for **protocol**. Load balancing of HTTP and HTTPS packets on the E250si is accomplished through Layer 4 TCP load balancing.

5. Enter the **ip** address, which is the dotted decimal IP address of the cluster. The IP address of the cluster is the external address (for example, `199.146.85.0`) with which clients connect to the cluster.

6. For HTTP and HTTPS protocol clusters, enter the **port**: the numeric port number on the Equalizer to be used for traffic between the clients and the cluster. For HTTP clusters, the port defaults to 80. For HTTPS clusters, the port defaults to 443. This port also becomes the default port for servers added to the cluster (though servers can use a different port number than the one used by the cluster).

   For L4 UDP and L4 TCP protocol clusters, a *port range* can be defined using the **start_port** and **end_port** fields. These are the ports on the Equalizer to be used to send traffic to the servers in the cluster. Port ranges allow Equalizer users to create a single cluster to control the traffic for multiple, contiguous ports. There are two typical uses for port ranges:

   - Specific applications that require a range of ports.
   - The need to open up access to servers behind the Equalizer for all ports.

   Enter the first port number in the **start_port** field (which is required). Enter the end port number in the **end_port** field. (If **end_port** is not visible, check the **advanced** flag.)

   When the **end_port** field is left with a value of zero (the default), Equalizer disables the port range feature and uses the **start_port** as the server port. The **start_port** cannot be higher than **end_port** when **end_port** is nonzero.

The **port** defined for a *server* in the cluster for which a port range is defined indicates the port on the server that starts the range of ports to be opened. See Step 6 on page 89, under “Adding a Server to a Cluster”

> **Note –** Old configuration files will still work—the port section for clusters will be interpreted as having a port range of start port being the same as end port. The port section will remain the same in the configuration file until the cluster is changed, at which point **start_port** and **end_port** will be written to the file.

7. For all cluster protocols, choose the appropriate load-balancing **policy** to be used by this cluster. Choose from **round robin** (default), **static weight**, **adaptive**, **fastest response**, **least connections**, or **server agent**. For more information, refer to “Configuring a Cluster’s Load-Balancing Options” on page 74.

8. Enter values for:
   - **responsiveness** sets the load-balancing response setting for this cluster. For more information, refer to “Configuring a Cluster’s Load-Balancing Options” on page 74.
   - **ACV probe** is the active content verification probe string. For more information, refer to “Using Active Content Verification (ACV)” on page 83.
   - **ACV response** is the active content verification response string. For more information, refer to “Using Active Content Verification (ACV)” on page 83.
   - **server agent port** is the port used to contact server agents. The default port is 1510. See Appendix A, ”Using Server Agents” for more information.

9. Set the flags:
   - **disable** causes the cluster to be unavailable. Use this flag when you are experimenting with a cluster’s settings and you do not want the cluster to listen for requests.
   - **server agent** has Equalizer use server agents gather performance statistics from the servers in the cluster. If you enable this option, you must run Server Agent daemons on each server in the cluster and must specify a value in **server agent port** (default is port 1510). See Appendix A, ”Using Server Agents” for more information about configuring server agents.
   - **ignore case** causes all of the cluster’s match rules to use case insensitive comparisons when this box is checked. You can override this setting by changing **ignore case** for a specific match rule.

10. For HTTP and HTTPS clusters, choose from the following options:
    - **spoof** causes Equalizer to spoof the client IP address when Equalizer routes a request to a server in a virtual cluster; that is, the IP address of the *client* is sent to the server, not the IP address of the Equalizer. This option is on by default. If you disable this option, the server receiving the request will see the Equalizer’s address as the client address because the TCP connection to the client is terminated when the request is routed. When spoof is enabled, the servers in the cluster must use the Equalizer as the default gateway for routing.
    - **persist** instructs Equalizer to use cookies to maintain a persistent session between a client and a particular server. This option is on by default. Equalizer “stuffs” a cookie into the server’s response header on its way back to the client. This cookie uniquely identifies the server to which the client was just connected. With **persist** enabled, Equalizer routes only the first request from a client using load balancing criteria; subsequent client requests are

routed to the same selected server for the entire session (while the cookie is valid -- see **cookie age**, below).

- **once only** limits Equalizer to parsing headers (and executing match rules) for only the first request of any client making multiple requests across a single TCP connection. This option is on by default. If this option is turned off, then Equalizer will parse the headers of every client request. See “Enabling the Once Only Flag for Persistent Connections” on page 82.

> **Note –** Although it is permitted by the software, it is *not* recommended to define a Layer 7 cluster with **persist** and **once only** both turned off, and with no match rules. By defining a Layer 7 cluster in such a way, you are essentially disabling Layer 7 processing, while still incurring extra overhead for the Layer 7 cluster. If your application requires a cluster with no persistence, header processing, or match rules, then we recommend that you define a Layer 4 UDP or TCP cluster for the best performance.

11. For HTTP and HTTPS clusters, if you enable the **persist** flag, the following options appear in the cluster parameters. Use these parameters to specify cookie behavior for the cluster:

- **cookie age** sets the time, in seconds, over which the client browser maintains the cookie (0 means the cookie never expires). After the specified number of seconds have elapsed, the browser can delete the cookie and any subsequent client requests will be handled by Equalizer’s load-balancing algorithms.
- **cookie generation** is a value added to cookies when the cookie scheme is 2 or greater. In order for cookies to be valid, cookie generation must match the equivalent number embedded in the cookie. Conversely if you need to invalidate old cookies, increment this number.
- **cookie scheme** (only displayed when the **advanced** flag is enabled) specifies the format of the cookie to be used for the cluster as an integer between **0** and **2** (default is **2**):

Cookie Scheme	Cookie Format
0	Constructs a cookie which will be named in such a way that so as long as the cluster maintains the same IP address, servers can be added to and removed from the cluster without invalidating all of the existing cookies. **This cookie stores the cluster IP and port, and the server IP and port.**
1	Constructs a cookie which will be valid across all clusters with the same IP address (not port specific). A requirement for this to be useful is that all clusters on that IP address share the same set of servers. **This cookie stores the Cluster IP, and Server IP and port.**
2	Constructs a cookie which will be valid across all clusters with the same IP address (using any port), and the same server within those clusters (with the server using any port). A requirement for this to be useful is that all clusters on that IP address share the same set of servers. **This cookie encodes the Cluster IP and Server IP.**

- **cookie domain** limits the presented cookie only to servers whose host name is within the specified domain. For example, if the cookie domain is `coyotepoint.com`, the browser will only present the cookie to servers in the `coyotepoint.com` domain (for example, `www.coyotepoint.com` or `my.coyotepoint.com`).

- **cookie path** presents the cookie only when the path component of the request URI has the same prefix as that of the specified path. For example, if the cookie path is `/store/`, the browser presents the cookie only if the request URI includes a path such as `/store/mypage.html`.
- **always** includes a cookie in the response whether or not the server actually set a cookie. If this flag is not enabled, Equalizer only sends a persistence cookie when the server sends a cookie of its own.

12. For HTTPS clusters, choose from the following options:
    - **x509 verify** has Equalizer check that the certificate meets the X.509 standard when you upload a certificate. Certain self-signed or chained certificates will not pass this verification and in that instance, you will want to disable the test. To see this flag, check the **advanced** flag.
    - **dont munge** forces Equalizer to pass responses from the cluster's servers without rewriting them. In the typical Equalizer setup, you configure the servers in an HTTPS cluster to listen and respond using HTTP; Equalizer communicates with the clients using SSL. If a server sends an HTTP redirect using the Location: header, this URL most likely will not include the `https:` protocol. Equalizer rewrites (munges) responses from the server so that they are HTTPS. You can direct Equalizer pass responses from the server without rewriting them by enabling the **dont munge** flag.
13. For L4 TCP and L4 UDP clusters, choose from the following options:
    - **sticky time** is the number of seconds that Equalizer should "remember" connections from clients. If you don't need sticky connections, set this option to 0. For more information, refer to "Enabling Sticky Connections" on page 80.
    - **intercluster sticky** is an option that when enabled ensures that Equalizer directs requests from a particular user to the same server, even if the connection is to a different virtual cluster. For more information, refer to "Enabling Sticky Connections" on page 80.
    - **probe ssl** (L4 TCP only) causes Equalizer to use SSL when it sends the **ACV probe** string. For more information, refer to "Using Active Content Verification (ACV)" on page 83.
14. Click the **commit** button to add the virtual cluster.

Equalizer can refuse an Add Cluster command for several reasons, including:

- Attempting to add a cluster address that is already configured or is configured as a server address
- Specifying an invalid cluster names
- Specifying an invalid IP address or port number
- Attempting to add more clusters than are supported by Equalizer

Advanced Cluster Fields and Flags

When you check the **advanced** flag check box at the bottom of the **add cluster** or **modify cluster** screens, additional fields are displayed. For most operations the default values are acceptable. The modifiable fields are described below:

- **probe_port** is used to select one port on the Equalizer to be used to for all content probes of the system (such as ACV) as well as protocol-specific health checks. It works for both Layer 4 and Layer 7 clusters.

  In previous implementations, probing was always done on the server port. However with a port range (see Step 6 on page 68), it cannot be assumed that the first port in the range will have a service running on it.

  By default, the **probe_port** field is set to zero and the Equalizer uses the **start_port** (for L4) or **port** (for L7) field value for the probe port. To change the default behavior, set **probe_port** to a specific port number.

  A **probe_port** value can be set on the servers in the cluster as well; see Step 7 on page 89 under “Adding a Server to a Cluster”.

  (Note that the server agent port remains a separate port that is used only for server agent communication; see Step 8 on page 69.)

- **max_conn** sets the maximum number of connections for all servers in the cluster. This value can be overridden by the **max_conn** setting for an individual server. See “Setting Maximum Connections per Server” on page 94.

- **netmask** is the netmask that applies to this cluster and is used to define an IP subnet that is different than the IP subnet defined for the external interface. It is assumed that the customer has the proper routing in place for clients to access multiple IP subnets defined on the Equalizer.

- **send buffer** applies to L7 clusters and is the amount of memory in kilobytes reserved by each L7 proxy process to store outgoing data before it is placed on the network interface.

- **receive buffer** applies to L7 clusters and is the amount of memory in kilobytes reserved by each L7 proxy process to store data that has been received on an interface before it is processed by an L7 proxy process.

- **request max** applies to L7 clusters and is the maximum number of kilobytes allotted for HTTP request headers.

- **response max** applies to L7 clusters and is the maximum number of kilobytes allotted for HTTP response headers.

- **probe delay** is the number of seconds between probes of the cluster’s servers. This applies to both TCP probes and ACV probes (if enabled). Also see the global parameters **probe_interval**, **probe_timeout**, **probe_delay**, and **strikeout threshhold** under “Modifying System Parameters” on page 43.

- **agent probe** is an optional string that is sent to an agent when an agent probe occurs. See Appendix A, ”Using Server Agents” for more information.

- **connect timeout** applies to L7 clusters and is the time in seconds that Equalizer waits for a server to respond to a connection request.

- **client timeout** applies to L7 clusters and is the time in seconds that Equalizer waits before closing an idle client connection. Valid values are between 1 and 150.

- **server timeout** applies to L7 clusters and is the time in seconds that Equalizer waits before closing an idle server connection.

- **custom header** applies to HTTPS clusters and allows you to specify a custom HTTP header that Equalizer will insert into incoming requests; this header indicates to the servers in the cluster that the request was received in HTTPS and unencrypted on Equalizer before being forwarded to the cluster; see “Specifying a Custom Header for HTTPS Clusters” on page 85 for more information.

- **cipher suite** applies to HTTPS clusters and is used to restrict cipher suites for incoming HTTPS requests. If a client request comes into Equalizer that does not use a cipher in this list, the connection is refused.

  For an Equalizer with no Xcel SSL Accelerator Card installed, and for Xcel II (newer generation) Cards, the following setting for **cipher suite** is used.

  ```
  AES128-SHA:DES-CBC3-SHA:RC4-SHA:RC4-MD5:AES256-SHA
  ```

  For an Xcel I (older generation) SSL Accelerator Card, this field will contain:

  ```
  DES-CBC3-SHA:RC4-SHA:RC4-MD5:AES256-SHA
  ```

  In previous releases, the EXP-RC4-MD5 ciphers were included in **cipher suite** for older browsers that only support 40-bit encryption. If some clients for your web services support only 40-bit encryption, you can add EXP-RC4-MD5 to the **cipher suite** list.

  > **Note –** EDH/DHA cipher suites which use ephemeral Diffie-Hellman keys are not recommended. They will work, but if they are added to the cipher suite string, they will have a major impact on performance. Cipher suites using eliptical curve (EC) cryptography are not supported. Please see “Supported Cipher Suites” on page 185 in **Appendix D, ”HTTPS Cluster Certificates”**, for a list of cipher suites supported by Equalizer.

  Besides its use with Xcel, this field can also be used to specify a custom cipher suite required by the servers in a cluster. For example, if your servers are required to support medium and high encryption using SSLv3 *only*, you could specify the following string for **cipher suite**, which will cause all non-SSLv3 client requests to be refused:

  ```
  ALL:!ADH:!EXPORT56:RC4+RSA:+HIGH:+MEDIUM:!LOW:!SSLv2:+SSLv3:+EXP:+eNULL
  ```

  This field requires a string in the format of the Apache **mod_ssl** directive **SSLCipherSuite**; see **http://httpd.apache.org/docs/2.0/mod/mod_ssl.html#sslciphersuite** for examples.

  You can also clear out the contents of this field, and all client requests received that use a cipher not supported by Xcel will still be accepted, but will be decrypted without using the Xcel card. Requests using ciphers supported by Xcel will be processed by the Xcel card.

- **sub-daemon max** applies to HTTPS clusters and is the maximum number of sub-daemons servicing the cluster.

- **session cache timeout** applies to HTTPS clusters and is number of seconds that Equalizer waits before disposing of an SSL session cache entry.

- **session cache kbytes** applies to HTTPS clusters and maximum number of kilobytes allotted to an SSL session cache.

- **client certificate verification depth** applies to HTTPS clusters and indicates the depth to which certificate checking is done on the client certificate chain. The default of 2 indicates that the client certificate (level 0) and two levels above it (levels 1 and 2) are checked; any certificates above level 2 in the chain are ignored. You should only need to increase this value if the Certificate Authority that issued your certificate provided you with more than 2 chained certificates in addition to your client certificate. See **Appendix D, ”HTTPS Cluster Certificates”**.

- **certify_client** applies to HTTPS clusters and indicates whether the server asks the client for a client certificate when a client request is received. The connection will succeed even if the client does not provide a certificate; but, if one is provided by the client it will be validated. See **Appendix D, ”HTTPS Cluster Certificates”**.
- **require certificate** applies to HTTPS clusters and indicates whether the server requires a client certificate when a client request is received. If the client does not provide a certificate, the connection is refused. This flag takes precedence over the **certify_client** flag; i.e., if both of these flags are specified, the client certificate is required. See **Appendix D, ”HTTPS Cluster Certificates”**.
- **verify once** applies to HTTPS clusters and indicates that the server will verify certificates only on the first client request, even if SSL is renegotiated. See **Appendix D, ”HTTPS Cluster Certificates”**.
- **ssl_unclean_shutdown** applies to HTTPS clusters and should be checked if you see errors (cannot see pages) while trying to maintain HTTPS persistent connections over HTTP/1.1. This problem especially applies to connections between Internet Explorer and Apache Servers and usually occurs intermittently.

Deleting a Virtual Cluster

You cannot delete a cluster with servers assigned to it. So, before attempting to delete the cluster, delete all servers from the cluster. For information about removing servers from a cluster, refer to “Deleting a Server” on page 90.

To delete a cluster, follow these steps:

1. Log into the Equalizer Administration Interface in edit mode.
2. In the left frame, click the name of the cluster to be deleted. The cluster’s parameters appear in the right frame.
3. Select **Delete Cluster** from the local menu.
4. When prompted, click **OK** to confirm that you want to remove the cluster permanently.

Configuring a Cluster’s Load-Balancing Options

Configure load balancing policy and response settings for each cluster independently. Multiple clusters do not need to use the same load balancing configuration even if the same physical server machines host them. For example, if one cluster on port 80 handles HTML traffic and one on port 8000 serves images, you can configure different load balancing policies for each cluster.

When you use adaptive load balancing (that is, you have ***not*** set the cluster’s load balancing policy to round robin or static weight), you can adjust Equalizer to optimize cluster performance. For more information, see “Adjusting a Server’s Static Weight” on page 90.

Equalizer’s Load Balancing Policies

Equalizer supports the following load balancing policies, each of which is associated with a particular algorithm that Equalizer uses to determine how to distribute requests among the servers in the cluster:

- **round robin** load balancing distributes requests equally among all the servers in the cluster. Equalizer dispatches the first incoming request to the first server, the second to the second server, and so on. When Equalizer reaches the last server, it repeats the cycle. If a server in the cluster is down, Equalizer does not send requests to that server. This is the default method.

The round robin method does not support Equalizer's adaptive load balancing feature; so, Equalizer ignores the servers' static weights and does not attempt to dynamically adjust server weights based on server performance.

- **static weight** load balancing distributes requests among the servers depending on their static weights. A server with a higher static weight gets a higher percentage of the incoming requests. Think of this method as a *weighted round robin* implementation. Static weight load balancing does not support Equalizer's adaptive load balancing feature; Equalizer does not dynamically adjust server weights based on server performance.
- **adaptive** load balancing distributes the load according to the following performance indicators for each server.

  Server response time is the length of time for the server to begin sending reply packets after Equalizer sends a request.

  Active connection count shows the number of connections currently active on the server.

  Server agent value is the value returned by the server agent daemon running on the server.
- **fastest response** load balancing dispatches the highest percentage of requests to the server with the shortest response time. Equalizer does this carefully: if Equalizer sends too many requests to a server, the result can be an overloaded server with slower response time. The Fastest Response policy optimizes the cluster-wide response time.

  Under Fastest Response, Equalizer checks the number of active connections and server agent values (if configured); but both of these have less of an influence than they do under adaptive load balancing. Even if a server's response time is the fastest in the cluster but its active connection count and server agent values are high, Equalizer might not dispatch new requests to that server.
- **least connections** load balancing dispatches the highest percentage of requests to the server with the least number of active connections. In the same way as Fastest Response, Equalizer tries to avoid overloading the server so it checks the server's response time and server agent value. Least Connections optimizes the balance of connections to servers in the cluster.
- **server agent** load balancing dispatches the highest percentage of requests to the server with the lowest server agent value. In a similar way to Fastest Response, Equalizer tries to avoid overloading the server by checking the number of connections and response time. This method only works if server agents are enabled. For more information about server agents, see "Configuring a Cluster to Use Server Agents" on page 78.

Equalizer's Load Balancing Response Settings

The **responsiveness** setting controls how aggressively Equalizer adjusts the servers' dynamic weights. Equalizer provides five response settings: Slowest, Slow, Medium, Fast, and Fastest. The response setting affects the dynamic weight spread, weight spread coefficient, and optimization threshold that Equalizer uses when it performs adaptive load balancing:

- **Dynamic Weight Spread** indicates how far a server's dynamic weight can vary (or *spread*) from its static weight.

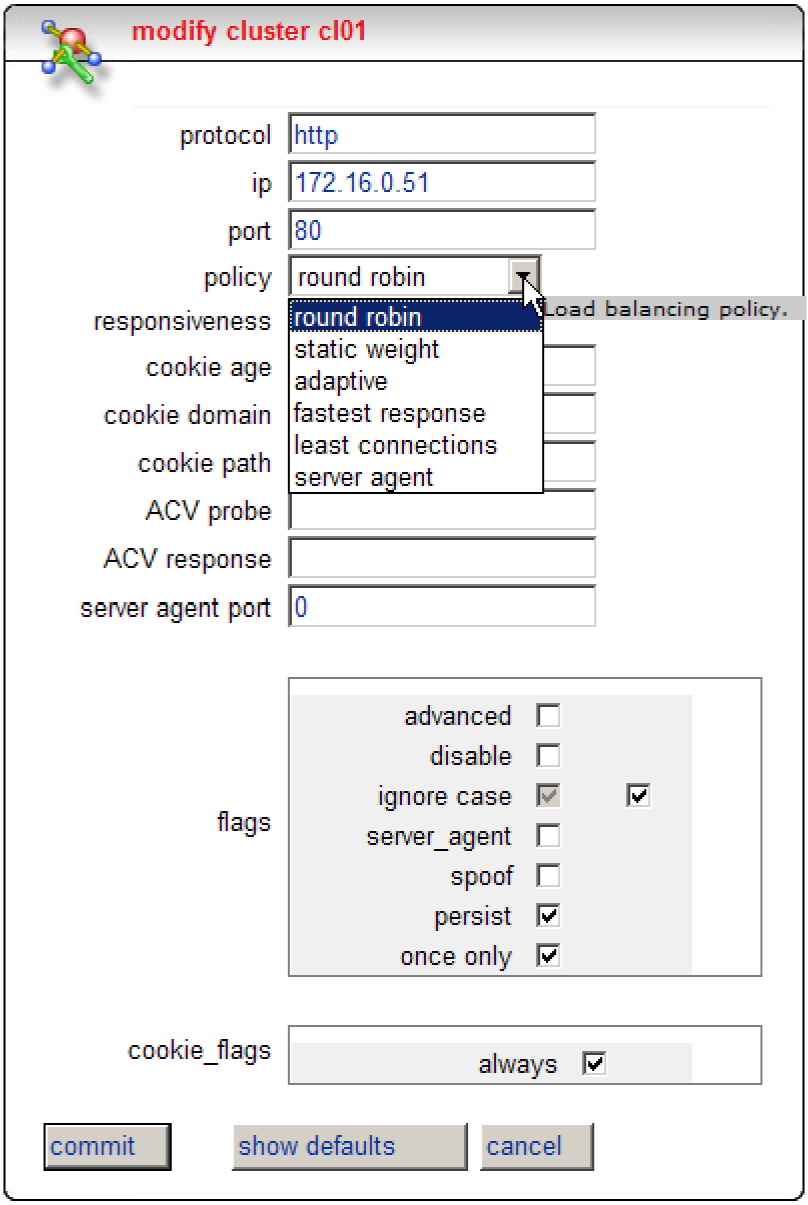
- **Weight Spread Coefficient** regulates the speed of change to a server’s dynamic weight. The weight spread coefficient causes dynamic weight changes to happen more slowly as the difference between the dynamic weight and the static weight increases.
- **Optimization Threshold** controls how frequently Equalizer adjusts dynamic weights. If Equalizer adjusts server weights too aggressively, oscillations in server weights can occur and cluster-wide performance can suffer. On the other hand, if Equalizer does not adjust weights often enough, server overloads might not be compensated for quickly enough and cluster-wide performance can suffer.

Modifying Equalizer’s Load Balancing Options

To change a cluster’s load-balancing options (see Figure 39), follow these steps:

1. Log into the Equalizer Administration Interface in Edit mode.
2. In the left frame, click the name of the cluster whose parameters to be changed. Equalizer displays the cluster’s parameters in the right frame.

3. Select **menu** > **Change Cluster Parameters**. Equalizer opens the modify cluster screen in the right frame.



modify cluster cl01

protocol	http
ip	172.16.0.51
port	80
policy	round robin
responsiveness	
cookie age	
cookie domain	
cookie path	
ACV probe	
ACV response	
server agent port	0

flags

advanced	□	
disable	□	
ignore case	☑	☑
server_agent	□	
spoof	□	
persist	☑	
once only	☑	

cookie_flags

always	☑

Figure 39 Changing load balancing options

4. Select a **policy**.

5. Choose a **responsiveness**.

6. Click the **commit** button.

Aggressive Load Balancing

After you fine-tune the static weights of each server in the cluster, you might discover that Equalizer is not adjusting the dynamic weights of the servers at all: the dynamic weights are very stable, even under a heavy load. In this case, you might want to set the cluster's load balancing response parameter to *fast*. Then Equalizer tries to optimize the performance of your servers more aggressively; this should improve the overall cluster performance. For more information about setting server weights, see "Adjusting a Server's Static Weight" on page 90.

Dynamic Weight Oscillations

If you notice a particular server's dynamic weight oscillates (for example, the dynamic weight varies from far below 100 to far above 100 and back again), you might benefit by choosing *slow* response for the cluster. You should also investigate the reason for this behavior; it is possible that the server application is behaving erratically.

Providing FTP Services on a Virtual Cluster

Virtual clusters that provide service on the FTP control port (port 21) must be layer 4 and have special requirements:

- FTP clusters occupy two virtual cluster slots, even though only one appears. This permits Equalizers NAT subsystem to rewrite server-originated FTP data connections as they "gateway" to the external network.
- FTP data connections always have a sticky time of one second. This is necessary to support the passive mode FTP data connection that most web browsers use.
- FTP virtual clusters do not support port redirection.

For more information about supporting passive mode FTP data connections, refer to "Enabling Passive FTP Connections" on page 53.

Configuring a Cluster to Use Server Agents

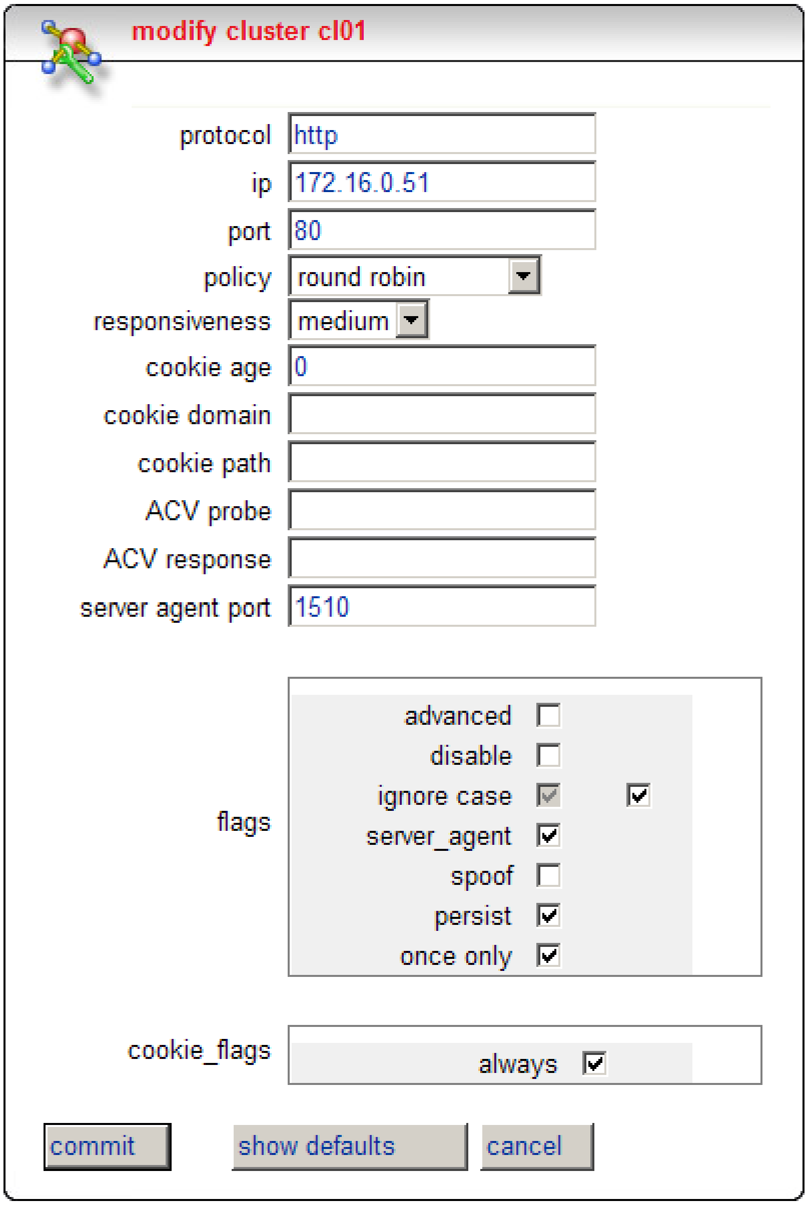
A *server agent* collects performance statistics from a server. If you configure a cluster to use server agents, Equalizer periodically contacts the server agent daemon running on each server and downloads the server performance statistics. You can also customize server agents to report on server resource availability; then Equalizer can stop sending requests to a server if a database or other vital resource is unavailable.

> **Note –** When you configure a cluster to use server agents, each server in the cluster ***must*** run a server agent daemon, so that the agent can provide status information to the Equalizer. If no agent is running on a server in a cluster configured to use the server agent load balancing policy, then the Equalizer will load balance without using the agent return value for that server (unless **pedantic agent** is set for the cluster, in which case Equalizer regards that server as down).

To configure a cluster to use server agents (see Figure 40), follow these steps:

1. Log into the Equalizer Administration Interface in edit mode.
2. In the left frame, click the name of the cluster to be configured. The cluster's parameters appear in the right frame.
3. Select **Change Cluster Parameters** from the local menu. The modify cluster screen opens in the right frame.
4. Check the **server agent** checkbox.

5. In the **server agent port** field, specify the port used to contact the server agent; the default port is 1510.



modify cluster cl01

protocol	http
ip	172.16.0.51
port	80
policy	round robin
responsiveness	medium
cookie age	0
cookie domain	
cookie path	
ACV probe	
ACV response	
server agent port	1510

flags

advanced	☐	
disable	☐	
ignore case	☑	☑
server_agent	☑	
spoof	☐	
persist	☑	
once only	☑	

cookie_flags

always ☑

Figure 40 Configuring a cluster to use server agents

6. If your agent needs to have a string sent to it before it will respond, click the **advanced** checkbox and then provide the string to be sent to the agent in the **agent probe** field.

7. Click the **commit** button.

For information about writing your own server agents and using agents to monitor server resource availability, see “Using Server Agents” on page 161.

Enabling Persistent Sessions

Equalizer provides several methods by which sessions between clients and servers can be made *persistent*; that is, it is possible to route a series of requests from a particular client to the same server, rather than have the Equalizer load balance each request in the series -- potentially sending each request to a different server.

For Layer 4 clusters, persistent sessions are enabled using the **sticky time** cluster parameter and (optionally) the **inter-cluster sticky** cluster flag. See “Enabling Sticky Connections” on page 80.

For Layer 7 clusters, persistent sessions are enabled using the **persist** or the **once only** cluster flags (which can be enabled together or separately). See “Enabling Cookies for Persistent Sessions” on page 81 and “Enabling the Once Only Flag for Persistent Connections” on page 82.

Enabling Sticky Connections

For L4 TCP and L4 UDP clusters, which only support L4 load balancing, you can use IP-address based sticky connections to maintain persistent sessions.

The **sticky time** period is the length of time over which Equalizer ensures that it directs new connections from a particular client to the same server. The timer for the sticky time period begins to expire as soon as there are no active connections between the client and the cluster. If Equalizer establishes a new connection to the cluster, Equalizer resets the timer for the sticky time period.

When you enable sticky connections, the memory and CPU overhead for a connection increase. This overhead increases as the sticky period increases. You should use the shortest reasonable period for your application and avoid enabling sticky connections for applications unless they need it. For most clusters, a reasonable value for the sticky time period is 600 seconds (that is,10 minutes). If your site is extremely busy, consider using a shorter sticky time period.

When you enable **inter-cluster stickiness**, you can ensure that Equalizer directs requests from a particular client to the same server even if the connection is to a different virtual cluster. Inter-cluster stickiness only works for L4 clusters. Although L7 clusters automatically provide inter-cluster stickiness, inter-cluster stickiness will not work between L4 and L7 clusters.

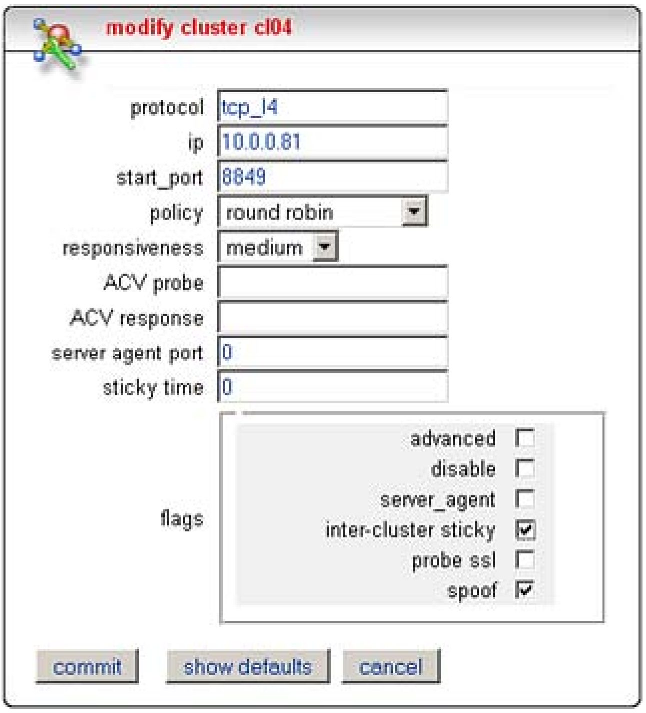
You must enable inter-cluster stickiness for all the clusters to be bound together. The clusters with enabled inter-cluster stickiness should contain identical sets of server IP addresses. For example:

```
Cluster www.coyotepoint.com:http
   Server srv1@192.168.0.5
   Server srv2

Cluster www.coyotepoint.com:https
   Server srv1@192.168.0.5
   Server srv2
```

To enable sticky connections (see Figure 41), follow these steps:

1. Log into the Equalizer Administration Interface in edit mode.
2. In the left frame, click the name of the cluster to be configured. The cluster’s parameters appear in the right frame.

3. Select **menu** > **Change Cluster Parameters**. The modify cluster screen opens in the right frame.



modify cluster cl04

protocol	tcp_l4
ip	10.0.0.81
start_port	8849
policy	round robin
responsiveness	medium
ACV probe	
ACV response	
server agent port	0
sticky time	0

flags

- advanced ☐
- disable ☐
- server_agent ☐
- inter-cluster sticky ☑
- probe ssl ☐
- spoof ☑

Figure 41 Setting the sticky time period

4. In the **sticky time** field, specify the sticky time period in seconds greater than zero.
5. To direct all requests from a particular client to the same server even if the connection is to a different virtual cluster, check the **inter-cluster sticky** checkbox.

> **Note –** You can turn on inter-cluster stickiness only if you have enabled sticky connections by specifying a sticky time greater than zero.

6. Click the **commit** button.

Enabling Cookies for Persistent Sessions

For HTTP and HTTPS clusters that support Layer 7 (L7) load balancing, you can enable the **persist** check box to use cookies to maintain a persistent session between a client and a particular server for the duration of the session.

When you use cookie-based persistence, Equalizer “stuffs” a cookie into the server’s response header on its way back to the client. This cookie uniquely identifies the server to which the client was connected and is included automatically in subsequent requests from the client to the same cluster. Equalizer can use the information in the cookie to route the requests to the same server. If the server is unavailable, Equalizer automatically selects a different server.

This option is enabled by default. Also see the descriptions of the **always**, **cookie age**, **cookie domain**, and **cookie path** cluster parameters under “Adding a Virtual Cluster” on page 67.

Enabling the Once Only Flag for Persistent Connections

Since HTTP 1.1, web browsers and servers have been able to negotiate persistent connections over which multiple HTTP transactions could take place, by specifying a *keep-alive* option in the request header. This is useful when several TCP connections are required in order to satisfy a single client request.

For example, before HTTP 1.1, if a browser wished to retrieve the file *index.html* from the server `www.coyotepoint.com`, the browser would take the following actions:

1. Browser opens TCP connection to `www.coyotepoint.com`.
2. Browser sends request to server "**GET /index.html**".
3. Server responds with the content of the page (a bunch of HTML).
4. Server closes connection.
5. Browser determines that there are objects (images) in the HTML document that need to be retrieved, so the browser repeats Steps 1 to 4 for each of the objects.

As you can imagine, there is a lot of overhead associated with opening and closing the TCP connections for each image. The way HTTP 1.1 optimizes this is by allowing multiple objects (pages, images, etc) to be fetched and returned across one TCP socket connection. The client requests that the server keep the connection open by adding the request header **Connection: keep-alive** to the request.

If the server agrees, the server will also include **Connection: keep-alive** in its response headers, and the client is able to send the next request over the persistent HTTP connection without the bother of opening additional connections. This is how Equalizer behaves.

For a Layer 7 cluster, Equalizer evaluates (and possibly changes) both the request and response headers that flow between the client and server (the request and response bodies are not examined). Match rules are applied to each client header, cookies may be inserted, and headers may be rewritten. When a client includes **keep-alive** in its headers, there is a fair amount of work required by the Equalizer to determine when the next set of request headers is ready to be parsed (evaluated), since there may be quite a lot of data going across the connection between sets of headers. To reduce this workload, the **once only** flag instructs the Equalizer to evaluate (and potentially modify) only the *first* set of headers in a connection.

So, in our example above, only the headers in the request for the *index.html* file are evaluated; the subsequent requests to obtain the images are not evaluated.

To summarize, enabling **once only** has the following effects:

- Match rules are only evaluated for the first request in a **keep-alive** connection.
- Header modifications such as insertion of persistence cookies and header rewriting will only be performed on the first response in a **keep-alive** connection.

An example of the latter behavior concerns the insertion of headers related to HTTPS processing. For an HTTPS cluster, Equalizer offloads SSL processing from the servers in the cluster; that is, it processes the HTTPS request and forwards it in HTTP to the server. When it does this, it inserts special headers into the request to indicate that the request was received by Equalizer in HTTPS and processed into HTTP (see "HTTPS Header Insertion" on page 85). If **once only** is set, these special headers are only inserted into the *first* request in a connection; the remainder of the requests in the connection are still processed, but no headers are inserted. Therefore, if you want to be able to parse for these headers in every request on the server end, you need to disable the **once only** flag for the cluster.

The **once only** flag is enabled by default when adding an L7 cluster. In general, it is more efficient to enable **once only**, but in situations where match rule evaluation is very important, or where any of the above effects are undesirable, **once only** should be disabled.

Using Active Content Verification (ACV)

Active Content Verification (ACV) is a mechanism for checking the validity of a server. When you enable ACV for a cluster, Equalizer requests data from each server in the cluster and verifies that the returned data contains a character string that indicates that the data is valid. You can use ACV with most network services that support a text-based request/response protocol, such as HTTP. However, you cannot use ACV with UDP-based services.

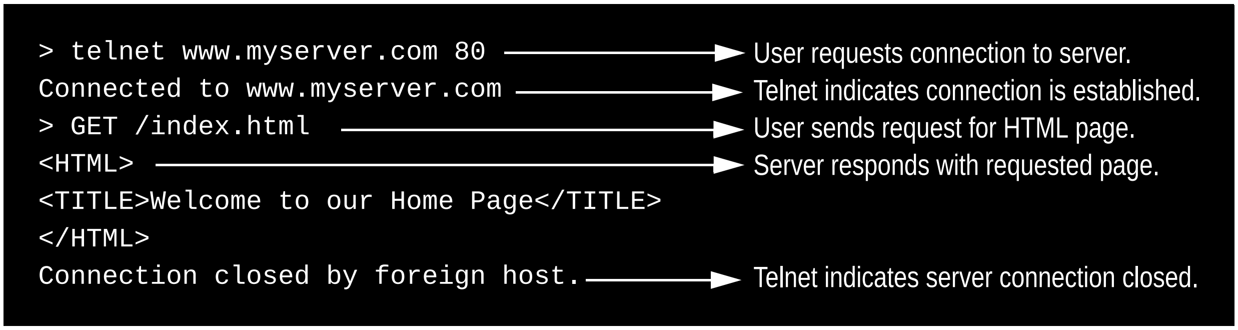
Controlling Server Verification Information

Specifying an *ACV probe string* and an *ACV response string* is one way to control the information that Equalizer uses to verify the servers. Equalizer uses the probe string to request data from each server. To verify the server's content, Equalizer searches the returned data for the response string. Equalizer expects to receive a response within the number of seconds specified by the **probe timeout** global parameter (default 10) when performing active content verification. Equalizer will make a number of successive attempts to reach the server equal to the **strikeout threshold** cluster parameter, separated by the number of seconds specified by the **probe delay** cluster parameter.

If there is no response or the response string does not appear in the first 1024 characters of the response, the verification fails; once the number of failures equals the **strikeout threshold**, Equalizer marks the server down and stops routing requests to that server.

If requests come in that contain cookies for a persistent connection to the down server, Equalizer will attempt to route the packets to the server in the cookie, and when this fails Equalizer sends the request to the next available server in the cluster (depending on the load balancing algorithm for the cluster). For the client, this means that any connection-related data stored on the downed server (such as a shopping cart) will be lost, and the client will need to restart any operations begun that depend on that data.

How ACV works is best explained using an example. The HTTP protocol enables you to establish a connection to a server, request a file, and read the result. Figure 42 illustrates the connection process when a user requests a telnet connection to an HTTP server and requests an HTML page.



```
> telnet www.myserver.com 80
Connected to www.myserver.com
> GET /index.html
<HTML>
<TITLE>Welcome to our Home Page</TITLE>
</HTML>
Connection closed by foreign host.
```

User requests connection to server.

Telnet indicates connection is established.

User sends request for HTML page.

Server responds with requested page.

Telnet indicates server connection closed.

Figure 42 Retrieving content from a server via telnet.

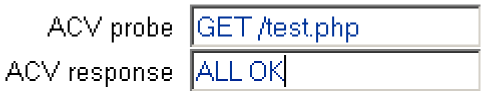
Equalizer can perform the same exchange automatically and verify the server's response by checking the returned data against an expected result.

Specify an *ACV probe string* and an *ACV response string* to control the information that Equalizer uses to perform the verification. Equalizer uses the probe string to request data from each server. To verify the server's content, Equalizer searches the returned data for the response string.

For example, you can use "`GET /index.html`" as the *ACV probe string* and you can set the response string to some text, such as "`Welcome`" in the example in Figure 42, which appears on the home page.

Similarly, if you have a Web server with a PHP application that accesses a database, you can use ACV to ensure that all the components of the application are working. You could set up a PHP page called **test.php** that accesses the database and returns a page containing "ALL OK" if there are no problems.

Then you would enter the following values on the **add cluster** or **modify cluster** screens:



ACV probe: GET /test.php
ACV response: ALL OK

If the page that is returned contains the correct response string (in the first 1000 characters, including headers) the server is marked "up"; if "ALL OK" were not present, the server is marked down.

The response string should be text that appears only in a valid response. This string is case-sensitive. An example of a poorly chosen string would be "HTML", since most web servers automatically generate error pages that contain valid HTML.

Enabling ACV

To enable ACV, follow these steps:

1. Log into the Equalizer Administration Interface in Edit mode.
2. In the left frame, click the name of the cluster to be configured. The cluster's parameters appear in the right frame.
3. Select **Change Cluster Parameters** from the local menu. The modify cluster screen opens in the right frame.
4. In the **ACV probe** field, specify a non-empty string. Equalizer sends this string to each server in the cluster to request verifiable data.

> **Note –** When you set up a L7 cluster and add a probe string, `\r\n` (that is, a "carriage return" followed by a "line feed") is automatically added to the end of the string. On the other hand, when you set up a L4 cluster and add a probe string, `\r\n` is *not* automatically added to the end of the string. The reason for this different behavior is that L7 "knows" the protocol is HTTP/HTTPS but L4 does not know the protocol to be used for the probe. If required for an L4 cluster, these characters need to be added manually.

5. In the **ACV response** field, specify a case-sensitive string that is not empty. Equalizer uses this string to verify the data with which the server responds to the ACV probe. For content

verification to succeed, the specified string must appear in the first 1024 characters of the server’s response (including any headers).

6. Click the **commit** button.

HTTPS Header Insertion

When a connection is established by a client for an HTTPS cluster, Equalizer performs the SSL processing on the request (this is called SSL offloading), and adds some additional headers to the client's request before forwarding the request on to a server:

```
X-LoadBalancer: CoyotePoint Equalizer
X-Forwarded-For: (cluster's IP address)
```

If the client provides an SSL certificate, the following are also added:

```
X-SSL-Subject: (certificate's X509 subject)
X-SSL-Issuer: (certificate's X509 issuer)
X-SSL-notBefore: (certificate not valid before info)
X-SSL-notAfter: (certificate not valid after info)
X-SSL-serial: (certs serial number)
X-SSL-cipher: (cipher spec)
```

If these headers are present in a request received by a server, then the server knows that the request was originally an HTTPS request and was processed by Equalizer before being forwarded to the server.

These headers are inserted into every request if the **once only** flag is disabled; if **once only** is enabled, then only the first request in a connection will have these headers inserted.

Some application may require a special header in the request, and the following section describes how Equalizer can be configured to provide a custom HTTPS header for such applications.

Specifying a Custom Header for HTTPS Clusters

Some applications, such as Microsoft Outlook Web Access (OWA), may require that all incoming client requests use the Secure Sockets Layer (SSL) protocol, meaning that all client requests must have the `https://` protocol in the URI.

If OWA is running on a server in an Equalizer Layer 7 cluster, then OWA will receive all requests with `http://` in the URI, since Equalizer performs SSL processing before passing the requests on to the server.

OWA allows for SSL offloading through the use of a special header, as explained in the following Microsoft technical article:

http://technet.microsoft.com/en-us/library/578a8973-dc2f-4fff-83c6-39b1d771514c.aspx.)

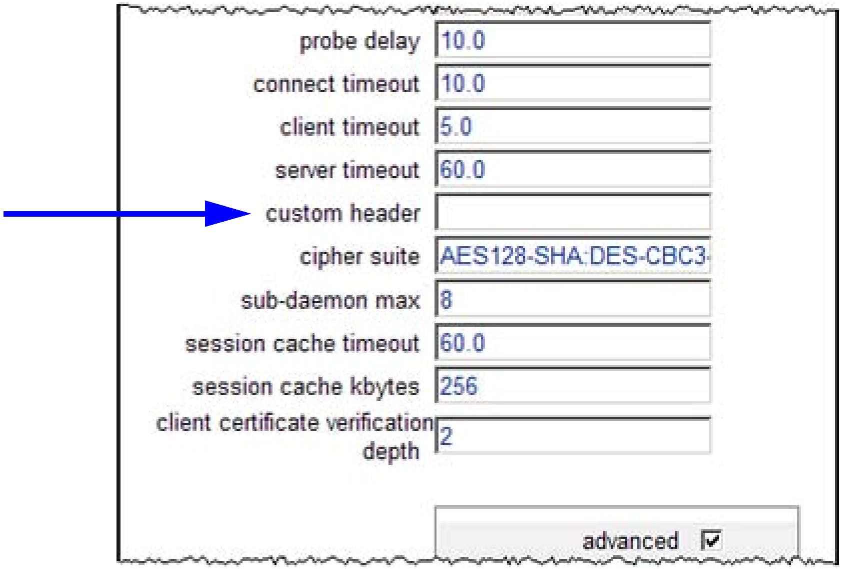
Two things are necessary when running OWA behind Equalizer:

- configure OWA to watch HTTP traffic for requests containing a custom header that indicates that the request was originally an SSL request that was processed by SSL offloading hardware (i.e., Equalizer) before reaching OWA (see the above article for instructions)

- configure the Equalizer cluster to add the custom header to all requests before sending them on to the OWA server (this is explained below)

Equalizer provides the ability to specify a custom header for HTTPS clusters. The following procedure shows you how to add a custom header to a new or existing HTTPS cluster definition, using the header required for an OWA server as an example.

1. Log into the Equalizer Administration Interface in Edit mode.
2. Do ***one*** of the following:
   a. Create a new virtual cluster: select **Add > Virtual Cluster** from the main menu bar. The **add cluster** screen appears in the right frame.
   b. Modify an existing virtual cluster: select the cluster name in the left frame, and then select **menu > Change Cluster Parameters** in the right frame. The **modify cluster** screen appears in the right frame.
3. If it is not already checked, select the **advanced** check box in the **flags** section of the screen in the right frame. The **custom header** field appears as shown in the following figure:



probe delay	10.0
connect timeout	10.0
client timeout	5.0
server timeout	60.0
custom header	
cipher suite	AES128-SHA:DES-CBC3-
sub-daemon max	8
session cache timeout	60.0
session cache kbytes	256
client certificate verification depth	2

advanced ☑

Figure 43 The custom header field

4. Type the following in the **custom header** field:

```
Front-End-Https: on
```

5. Set other parameters and flags for the cluster as desired; see “Adding a Virtual Cluster” on page 67 for more details.
6. Select **commit** to create or modify the cluster.

Performance Considerations for HTTPS Clusters

Layer 7 HTTPS clusters have several features which are not present in HTTP clusters. The two most important of these features are:

- the injection of custom headers to relay to the server the fact that Equalizer terminated the HTTPS connection and performed SSL processing on the incoming request (see the previous section, above)
- the "munging", or translation, of HTTP redirects to HTTPS redirects (see the description of the **dont munge** flag under “Adding a Virtual Cluster”, in Step 12 on page 71)

One flag which frequently affects the behavior of these options is the **once only** flag. This flag is present to speed up processing of HTTP requests by only looking at the first request, but since HTTPS has a lot of overhead associated with it anyway, turning this flag off does not reduce HTTPS performance. Furthermore, having this flag on for HTTPS clusters causes some applications to not function as needed.

In general, it is recommended to turn the **once only** flag off for HTTPS clusters. This is particularly true if you're using Microsoft Internet information Service (IIS) on the servers in your cluster.

For most applications, Xcel will sustain several hundred HTTPS transactions per second with no noticeable degradation in performance either of the cluster or Equalizer.

In terms of bulk data throughput, the theoretical maximum throughput for Xcel/HTTPS is roughly 50% of that for the Equalizer in HTTP mode: Equalizer models with gigabit Ethernet can move HTTP traffic at wire speed (1Gbit/s) for large transfers, while Xcel can encrypt only approximately 400Mbit/s with 3DES/SHA1 or 600Mbit/s with RC4/MD5. This reflects the fact that Xcel is primarily a transaction accellerator, not a bulk data encryptor. It is noteworthy, however, that even when moving bulk data at 600Mbit/s, Xcel removes the entire load of HTTPS/SSL processing from the servers in the cluster.

One final issue to be aware of is that Xcel supports only 3DES and RC4 encryption; it does not support AES. It also does not support SSL or TLS cipher suites that use ephemeral or anonymous Diffie-Hellman exchange (cipher suites whose names contain "EDH", "DHE", or "ADH").

The default configuration for HTTPS clusters created with an Xcel card present in the system will not use the modes described above. If, however, one either modifies the **cipher suite** string in the advanced cluster properties to use them (or, creates a cluster before installing the Xcel card and then adds an Xcel card to the system), it is possible that they may be negotiated with clients. This will not lead to incorrect operation of the system, but will cause encryption to occur in software (which does not perform as well as the Xcel card).

Managing Servers

In this section, you will learn how to work with servers: adding them, adjusting their static weight, shutting them down, and deleting them.

Server Software Configuration

Please observe the following guidelines and restrictions when configuring the software that is running on your servers:

- If the **spoof** flag is turned on for a cluster (the default), you should configure your network topology so that Equalizer is the gateway for ***all*** traffic for its virtual clusters. Each server in a cluster should be configured to use Equalizer as its default gateway. This way, all packets that come through Equalizer from clients will pass back through Equalizer and then to the clients.

You do *not* need to configure Equalizer as the gateway for the servers if you have *disabled* the IP **spoof** flag for the cluster.

- Server responses (and client requests) must contain 64 or fewer headers; any packet that contains more than 64 headers is dropped by Equalizer (along with the connection), and a message like the following is printed to Equalizer's event log:

  ```
  Warning: Dropping connection from ip-address -- too many headers
  ```

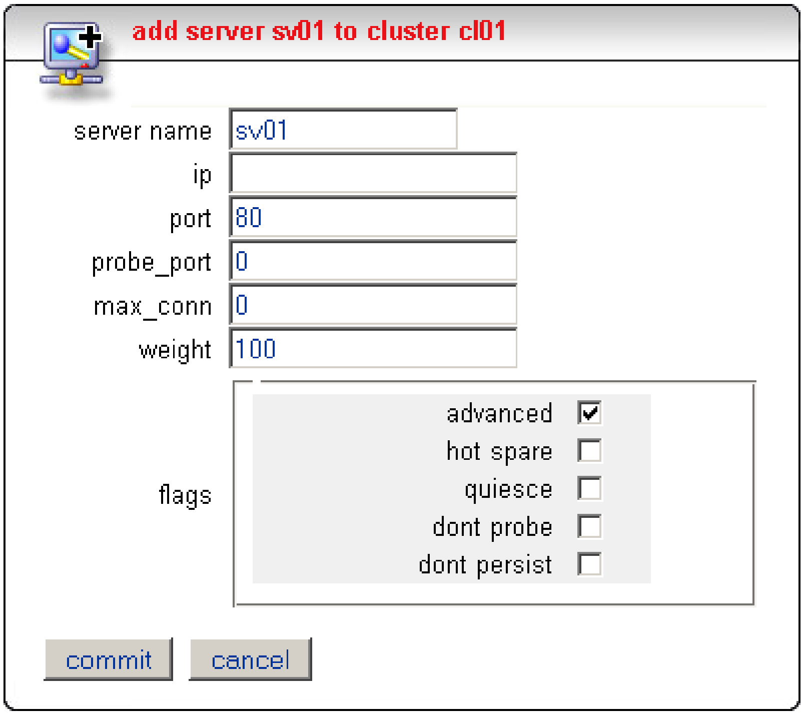
  Make sure that your server software is configured to return 64 headers or less in any response it sends back through Equalizer.

  If your application must use 64 headers or more in server responses, then you can turn the **spoof** flag off so that server responses go back to the client *without* going through Equalizer. Be aware, however, that this has no effect on the client side; any packets from the client with more than 64 headers will still be dropped by Equalizer (and a warning appended to the event log). In most cases, client requests do not include that many headers.

Adding a Server to a Cluster

To add a server (see Figure 44) to a virtual cluster, follow these steps:

1. Log into the Equalizer Administration Interface in Edit mode.
2. In the left frame, click the name of the cluster to which you want to add the server. The cluster's parameters appear in the right frame.
3. Select **menu** > **Add Server**. The add server screen opens in the right frame. (The figure below shows the screen with the **advanced** check box enabled.)



add server sv01 to cluster cl01

server name	sv01
ip	
port	80
probe_port	0
max_conn	0
weight	100
flags	advanced ☑ hot spare □ quiesce □ dont probe □ dont persist □

Figure 44 Adding a server

4. Enter **server name**, which is the server's logical name, or accept Equalizer's default. Each server in a cluster must have a unique name that begins with an alphabetical character, not a numeral (for example, *Phoenix*).
5. Enter **ip** which is the IP address of the server endpoint you are adding to the cluster.
6. Enter **port**, which is the port number of the service on the server machine. Unless you want to set up port redirection, you can usually accept the default value, which is the same as the port of the virtual cluster.

> **Note –** Equalizer performs all the encryption and decryption for HTTPS clusters, so traffic between the Equalizer and the servers in an HTTPS cluster uses the HTTP protocol. When you add servers to an HTTPS cluster, you should configure them on port 80.

If a *port range* has been defined for the server's cluster (see Step 6 on page 68, under "Adding a Virtual Cluster"), then the **port** field in the **add server** or **modify server** screen refers to the first port on which to start servicing the cluster **start_port**. For example:

Cluster Port Range	Server Port	Port Mapping (external to internal)

start_port = 80
end_port = 90
	80	80 to 80
		81 to 81
		...
		90 to 90

start_port = 80
end_port = 90
	100	80 to 100
		81 to 101
		...
		90 to 110

If there is no service running on one or more ports in the port range, Equalizer will still attempt to forward traffic to that port and return an error code to the client, just like what would happen if the client was connecting to the server directly.

7. Enable the **advanced** flag at the bottom of the screen to set a value for **probe_port**. The default is 0, which means that the server will use the value in the **port** field (see above) as the **probe_port**. Change this value if you want to use another port for health check robes form the Equalizer. (Note that the probe_port on the Equalizer is set in the cluster menus; see "Advanced Cluster Fields and Flags" on page 71.)
8. Enable the **advanced** flag at the bottom of the screen to set a value for **max_conn**. This sets the maximum number of connections for the server, and overrides the **max_conn** setting for the cluster (if any). See "Setting Maximum Connections per Server" on page 94 for more information.
9. Enter **weight**, which determines a starting point (static weight) for the percentage of requests to route to each server. For information about selecting an appropriate static weight, refer to "Adjusting a Server's Static Weight" on page 90.
10. Enable the **hot spare** check box if you plan to use this server as a backup server, in case the other servers in the cluster fail. Checking **hot spare** forces Equalizer to direct incoming connections to this server only if *all* the other servers in the cluster are down. You should only configure *one* server in a cluster as a hot spare.

For example, you might configure a server as a hot spare if you are using licensed software on your servers and the license allows you to run the software only on one node at a time. In this situation, you could configure the software on two servers in the cluster and then configure one of those servers as a hot spare. Equalizer will use the second server only if the first goes down, enabling you to make your application available without violating the licensing terms or having to buy two software licenses.

11. Enable the **quiesce** check box to avoid sending new requests to the server. This is usually used in preparation for shutting down an HTTP or HTTPS server. Please see “Shutting Down a Server Gracefully” on page 92.
12. Enable the **advanced** flag if you want to set the **dont probe** check box; when set, **dont probe** disables TCP health check probes for the server. This is usually used to disable probe checks for a particular server without changing the probe settings for the server’s cluster.
13. Enable the **advanced** flag if you want to set the **dont persist** check box; when set, **dont persist** disables persistence for the server when the **persist** flag (Layer 7 cluster) or a non-zero **sticky time** (Layer 4 cluster) is set on the cluster. For a Layer 7 cluster, this means that no cookie will be inserted into the response header on the way back to the client. For a Layer 4 cluster, no sticky record is set. This flag is usually used to disable persistence for a hot spare. For an example, see “Using a Hot Spare in a Cluster with a Maximum Connections Limit” on page 96.
14. Click the **commit** button.

Equalizer can refuse an Add Server command for several reasons, including:

- Attempting to add a server address that is already configured or is configured as a cluster address
- Specifying an invalid IP address or port number
- Attempting to add more servers than are supported by Equalizer

Deleting a Server

To delete a server from a virtual cluster, follow these steps:

1. Log into the Equalizer Administration Interface in edit mode.
2. In the left frame, click the name of the server to be removed.
3. Select **Delete Server** from the local menu.
4. When prompted, click **OK** to confirm that you want to remove the server from the cluster.

If you attempt to delete a server with active connections, a confirmation dialog box appears. Click **Force** to remove the server anyway. This action removes the server and deletes the active connections and the user sessions they represent. To cancel the deletion, click **Cancel**.

Adjusting a Server’s Static Weight

Equalizer uses a server’s static weight as the starting point for determining the percentage of requests to route to that server. Equalizer assigns servers with a higher static weight a higher percentage of the load.

Values for server weights can be in the range 20-200 (and 0, which essentially disables the server). When you install servers, set each server's static weight value in proportion to its “horsepower.” All the static weights in a cluster do not need to add up to any particular number; *it’s the ratio of the* *assigned server weight for a server to the total of all the server weights that determines the amount of traffic sent to a server.*

For example, you might assign a server with 4 dual-core 64-bit processors operating at 3.40GHz a value of 100 and a server with 2 dual-core 64-bit processors operating at 1.86GHz a value of 50. The first server will initially receive approximately 66% (100 divided by 150) of the traffic. The second server will initially get about 33% (50 divided by 150) of the traffic. It's important to note that setting the static weights of these servers to 100 and 50 is equivalent to setting the static weights to 180 and 90.

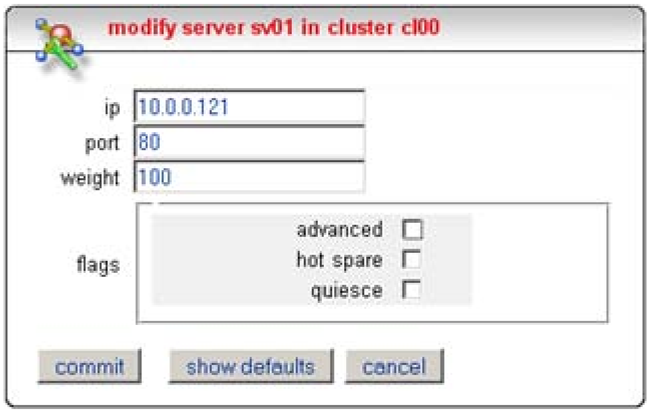
If Equalizer is performing adaptive load balancing (ALB), you should generally use higher static weights. When you have enabled Equalizer's ALB feature (that is, the load balancing policy is ***not*** set to round robin or static weight), using higher static weights will produce finer-grained load balancing. Higher weights enable Equalizer to adjust server weights more gradually; increasing the weight by 1 produces a smaller change if the starting weight is 100 than it does if the starting weight is 50.

However you set the static weights, Equalizer will adjust the weight of servers dynamically as traffic goes through the cluster. Dynamic server weights might vary from 50-150% of the statically assigned values. To optimize cluster performance, you might need to adjust the static weights of the servers in the cluster based on their performance.

Note – Equalizer stops dynamically adjusting server weights if the load on the cluster drops below a certain threshold. For example, if web traffic slows significantly at 4:00 AM PST, Equalizer will not modify server weights until traffic increases again. Because a server's performance characteristics can be very different under low and high loads, Equalizer optimizes only for the high-load case. Keep this in mind when you configure new Equalizer installations; to test Equalizer's ALB performance, you'll need to simulate expected loads.

To change a server's static weight (see Figure 45), follow these steps:

1. Log into the Equalizer Administration Interface in edit mode.
2. In the left frame, click the name of the server to be modified. The server's parameters appear in the right frame.
3. Select **Change Server Parameters** from the local menu. The modify server screen opens in the right frame.



modify server sv01 in cluster cl00

ip 10.0.0.121

port 80

weight 100

flags

advanced □

hot spare □

quiesce □

Figure 45 Changing a server's static weight

4. Enter the new weight in the **weight** field.
5. Click the **commit** button.

Setting Static Weights for Homogenous Clusters

If all the servers in a cluster have the same hardware and software configurations, you should set their static weights to the same value initially. We recommend that you use a static weight of 100 and set the load-balancing response parameter to *medium*.

As with any new configuration, you will need to monitor the performance of the servers under load for two to three hours. If you observe that the servers differ in the load they can handle, adjust their static weights accordingly and again monitor their performance. You should adjust server weights by small increments; for example, you might set the static weight of one server to 110 and the other to 90. Fine-tuning server weights to match each server's actual capability can easily improve your cluster's response time by 5 to 10%.

> **Note –** Equalizer's ALB algorithm can take 10-15 minutes to fine-tune cluster performance when you change static weights. After you change static weights, wait 30 minutes before you judge the cluster's ALB performance.

Setting Static Weights for Mixed Clusters

Equalizer enables you to build heterogeneous clusters using servers of widely varying capabilities. Adjust for the differences by assigning static weights that correspond to the relative capabilities of the available servers. This enables you to get the most out of your existing hardware, so you can use an older server side-by-side with a new one.

After you assign relative static weights, monitor cluster performance for two to three hours under load. You will probably fine-tune the weights and optimize performance of your cluster two or three times.

Continue monitoring the performance of your cluster and servers and watch for any trends. For example, if you notice that Equalizer *always* adjusts the dynamic weights so that the weight of one server is far below 100 and the weight of another is far above 100, the server whose dynamic weight is consistently being reduced might have a problem.

Shutting Down a Server Gracefully

To avoid interrupting user sessions, make sure that a server to be shut down or deleted from a cluster no longer has any active connections. When a server's static weight is zero, Equalizer will not send new requests to that server. Connections that are already established continue to exist until the client and server application end them or they time out because they are idle.

To shut down servers in a generic TCP or UDP (L4) cluster, you can set the server's weight to zero and wait for the existing connections to terminate. However, you need to quiesce servers in HTTP and HTTPS (L7) clusters to enable servers to finish processing requests for clients that have a persistent session with the server.

When you quiesce a server, Equalizer does not route new connections from new clients to the server, but will still send requests from clients with a persistent session to the quiescing server. Once all the persistent sessions on the server have expired, you can set the server's static weight to zero; then Equalizer will not send additional requests to the server.

Note that while a server is quiescing, it will still receive new requests *if all of the other servers in the cluster are unavailable*. This behavior prevents any new requests from being refused, but may lengthen the time needed to terminate all active persistent connections.

Removing a Layer 7 Server from Service

To remove a Layer 7 server from service, follow these steps:

1. In the left frame, click the name of the server to be quiesced. The server's parameters appear in the right frame.
2. Select **menu > Change Server Parameters** from the local menu. The modify server screen opens in the right frame.
3. Check the **quiesce** checkbox; then click **commit** to save your changes.
4. Click on **View > Cluster Summary** in the main menu and select a refresh interval in the drop-down box. Watch the quiescing server's number of **active** connections. Once there are no active connections shown, select **menu > Change Server Parameters** to set the server's weight to zero; click **commit** to save the change.
5. Click on the server name in the left frame and check the number of **total** connections (click the server name to refresh). If this number does not go to zero after a reasonable period of time, then there are clients that still have open persistent connections to the server. To make sure that these connections are not dropped, but are renegotiated after you take the server down, click on the cluster name in the left frame and increment the **cookie generation** parameter by 1; then click **commit**.

   This change invalidates all currently held cookies on all clients, and forces the client to renegotiate the connection, rather then the connection being dropped.

To ensure that no cookie ever persists beyond a given time period, you can change the **cookie age** cluster parameter from the default of 0 to some number of seconds that is reasonable for your application. Then, you only need to wait that number of seconds after quiescing the server and changing its weight to 0 before it's safe to take the server down. Note that this only applies to cookies created after the change is committed.

Removing a Layer 4 Server from Service

To remove a Layer 4 server from service, follow these steps:

1. In the left frame, click the name of the server to be removed. The server's parameters appear in the right frame.
2. Select **Change Server Parameters** from the local menu The **modify server parameters** dialog box opens in the right frame.
3. Set the server's weight to **0**; click **commit** to save the change. This action prevents Equalizer from routing new connections to the server.
4. Click on **View > Cluster Summary** in the main menu and select a refresh interval in the drop-down box. Watch the server's number of **active** and **sticky** connections. Once both of these numbers are **0**, click on the server name in the left frame and check the number of total connections (click the server name to refresh). Once that number is **0** and the server's **idle time** is greater than your application's session lifetime, you can take the server offline.

Setting Maximum Connections per Server

A new feature has been added for the HTTP, HTTPS, and L4 TCP cluster types that allows you to set a hard upper limit on the number of active connections per server. When a server in the cluster reaches the maximum connections limit, requests will not be routed to that server until the number of active connections falls below the limit.

Typical reasons to set a maximum number of connections include:

- implementing a connection limit that is required due to software limitations, such as an application that can service a limited number of concurrent requests
- implementing license restrictions that are not enforced by software; such as limiting the number of active connections to an application that is licensed for a limited number of concurrent connections
- setting a threshold that will limit resource utilization on the cluster

The **max_conn** limit can be set on a cluster or on individual servers in a cluster, and behaves as described below:

- Setting **max_conn** when you create a cluster sets the maximum number of connections for all subsequently created servers in the cluster.
- Setting (or changing) **max_conn** when modifying an existing cluster *does not* set (or change) **max_conn** on any of the existing servers in the cluster. If you want the new **max_conn** limit to apply to existing servers, you will need to set (or change) **max_conn** on each existing server.
- Setting **max_conn** when you create or modify a server overrides the **max_conn** setting for the cluster.
- The **max_conn** limit may be ignored on Layer 7 clusters with **persist** enabled. The **persist** option tells Equalizer to insert a session cookie into all responses back to the client. When Equalizer gets another request containing the cookie, and the **max_conn** limit has already been reached, it accepts the request anyway. However, if a hot spare is defined for the cluster, it sends the request to the hot spare instead. If **persist** is not enabled, **max_conn** is always enforced.
- The **max_conn** limit may be ignored on L4 TCP clusters with a non-zero **sticky time**. The **sticky time** option tells Equalizer to keep a “sticky record” so an L4 connection can be persistent. When Equalizer gets another request on a connection that already has a sticky record, and the **max_conn** limit has already been reached, it accepts the request anyway. However, if a hot spare is defined for the cluster, it sends the request to the hot spare instead. If no **sticky time** is set, **max_conn** is always enforced.
- A new flag, **dont persist**, has been introduced. It is intended to be used to override persistent connections for a hot spare in an L4 or L7 cluster that has a maximum connection limit. See the section “Using a Hot Spare in a Cluster with a Maximum Connections Limit” on page 96.

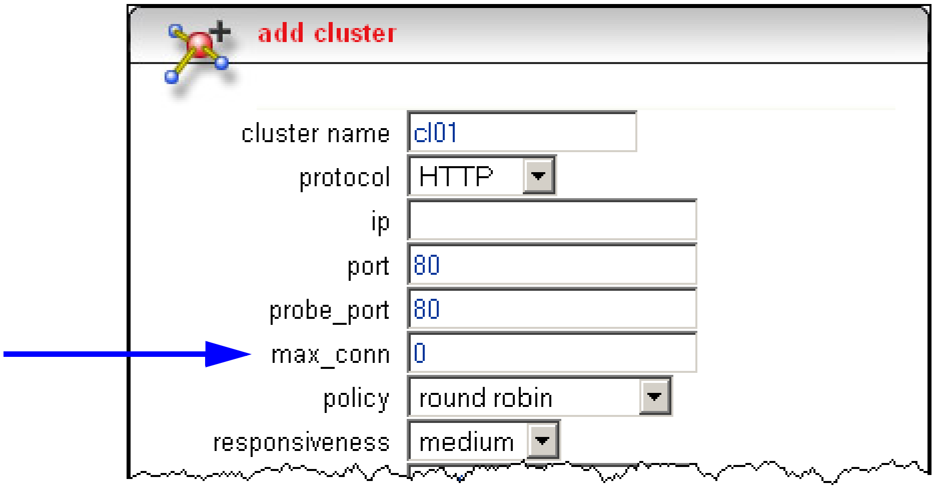
Setting Maximum Connections on a Cluster:

1. Log into the Equalizer Administration Interface in Edit mode.
2. Do ***one*** of the following:

a. Create a new virtual cluster: select **Add** > **Virtual Cluster** from the main menu bar. The **add cluster** screen appears in the right frame.

b. Modify an existing virtual cluster: select the cluster name in the left frame, and then select **menu** > **Change Cluster Parameters** in the right frame. The **modify cluster** screen appears in the right frame.

3. If it is not already checked, select the **advanced** check box in the **flags** section of the screen in the right frame. The **max_conn** field is near the top of the right-hand screen, as shown in the following figure:



cluster name	cl01
protocol	HTTP
ip	
port	80
probe_port	80
max_conn	0
policy	round robin
responsiveness	medium

4. Set **max_conn** to a positive integer between 0 and 65535. A zero (the default) indicates that there is no maximum connection limit.

5. Set other parameters and flags for the cluster as desired; see Chapter 6, "*Administering Virtual Clusters*", in the *Installation and Administration Guide*, for more details.

6. Select **commit** to create or modify the cluster.

7. If you modified an existing cluster with servers already defined, you will need to modify each server to use the new maximum connections limit.

Setting Maximum Connections on a Server:

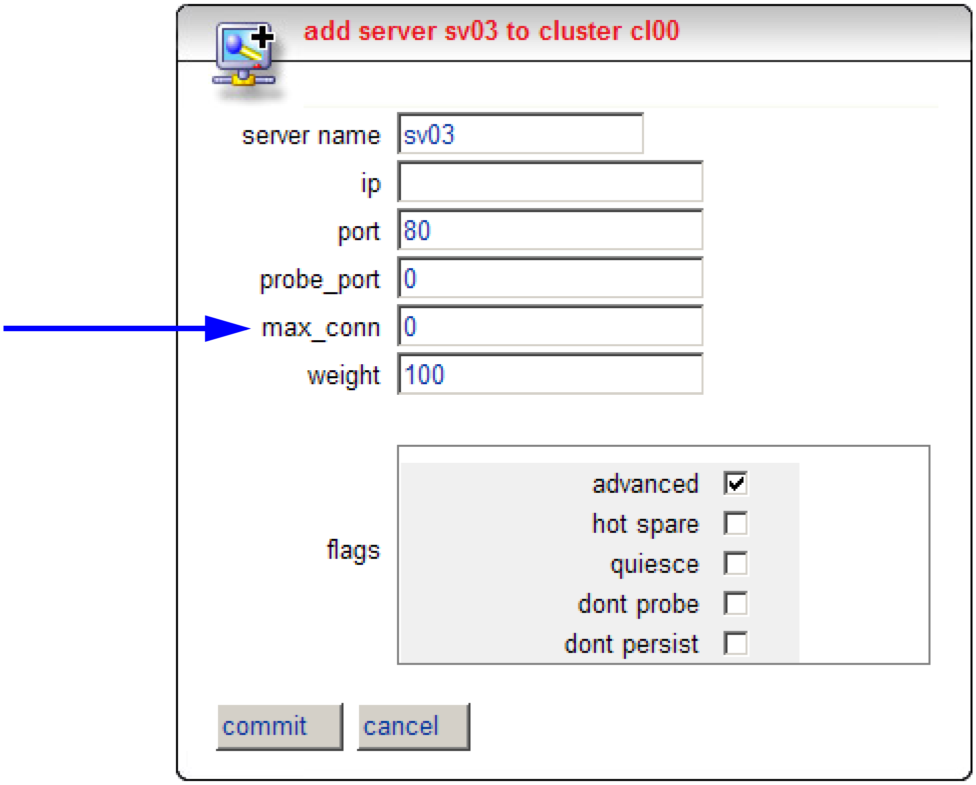
1. Log into the Equalizer Administration Interface in Edit mode.

2. Do ***one*** of the following:

a. Create a new server: select a cluster name in the left frame, and then select **menu** > **Add Server**. The **add server** screen appears in the right frame.

b. Modify an existing server: select the server name in the left frame, and then select **menu** > **Change Server Parameters** in the right frame. The **modify server** screen appears in the right frame.

3. If it is not already checked, select the **advanced** check box in the **flags** section of the screen in the right frame. The **max_conn** field is near the middle of the right-hand screen, as shown in the following figure:



add server sv03 to cluster cl00

server name	sv03
ip	
port	80
probe_port	0
max_conn	0
weight	100
flags	advanced ☑ hot spare □ quiesce □ dont probe □ dont persist □

4. Set **max_conn** to a positive integer between 0 and 65535. A zero (the default) indicates that the maximum connections limit set on the cluster applies to this server.

5. Set other parameters and flags for the server as desired; see Chapter 6, “*Administering Virtual Clusters*”, in the *Installation and Administration Guide*, for more details.

6. Select **commit** to create or modify the server.

Using a Hot Spare in a Cluster with a Maximum Connections Limit

When a maximum connections limit is set on all the servers in a cluster (either by setting **max_conn** on the cluster or on each individual server), it is often desirable to define a **hot spare** server for the cluster, so that any attempted connections to the cluster that occur after the **max_conn** limit has been reached are directed to the hot spare instead of being refused or sent to the server anyway because of a persistent connection.

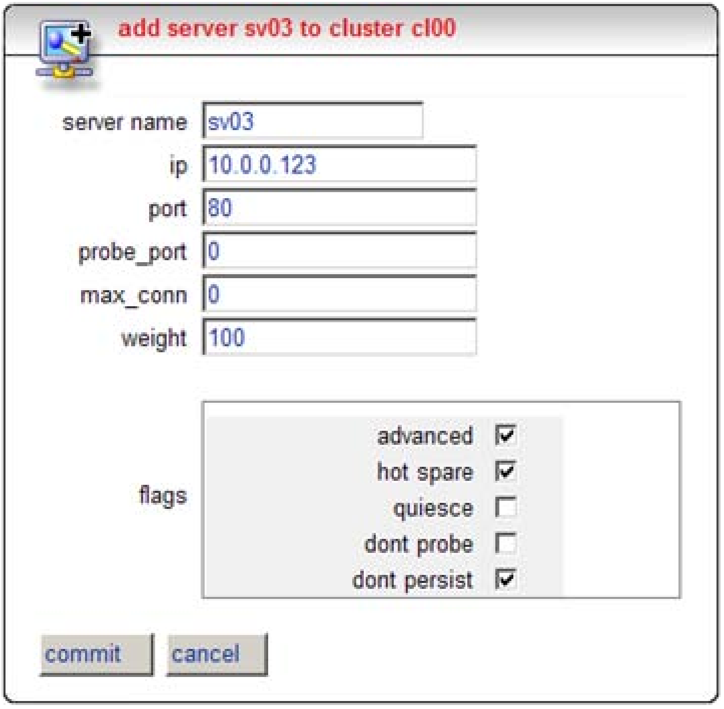
In this case, the hot spare could be configured to return a page to the client that contains text explaining the reason the connection has been refused. For example, the hot spare could return a page that says “All servers are currently busy -- please try again later.”.

The hot spare server should be configured as follows:

- Set **max_conn** to zero (0), so that all connection requests sent to the hot spare are accepted.

- Enable the **hot spare** flag. This specifies that any requests refused by all the other servers in the cluster because they reached their **max_conn** limit (or are down) will be forwarded to the hot spare server.
- Enable the **dont persist** flag. We do not want connections made to the hot spare to persist. Each connection to the cluster must first be load balanced amongst the other servers in the cluster and only go to the hot spare if all the other servers have reached their **max_conn** limit.

The following screen shows an example of adding a server with the above settings:



add server sv03 to cluster cl00

server name: sv03
ip: 10.0.0.123
port: 80
probe_port: 0
max_conn: 0
weight: 100

flags:
advanced ☑
hot spare ☑
quiesce □
dont probe □
dont persist ☑

Testing Virtual Cluster Configuration

1. If you use a two-network configuration: after you have configured a virtual cluster and added servers, telnet to each of the virtual clusters configured on the Equalizer from a system on the external network.

   When you telnet to a virtual cluster from the external test machine, Equalizer should connect you to one of the servers configured in the cluster. Repeatedly connect to the same virtual cluster using several sessions to make sure that Equalizer routes the connections to different servers in the cluster. (Equalizer does not necessarily select the servers in a round-robin fashion; it uses the algorithm defined for the cluster to determine the server that gets the next connection.)

   You also can use a client tool such as a Web browser to perform this test.

2. From a client machine on the Internet, connect to each virtual cluster using a Web browser.

For help in resolving configuration problems, see Appendix E, ”Troubleshooting”. Also visit the **Coyote Point Support Portal** (`http://www.coyotepoint.com/support.php`) for more help.

7 Monitoring Equalizer Operation

System status information and performance statistics can be gathered and displayed from within the Equalizer Administrative Interface; higher-end Equalizers can also be monitored using standard Simple Network Management Protocol (SNMP) utilities:

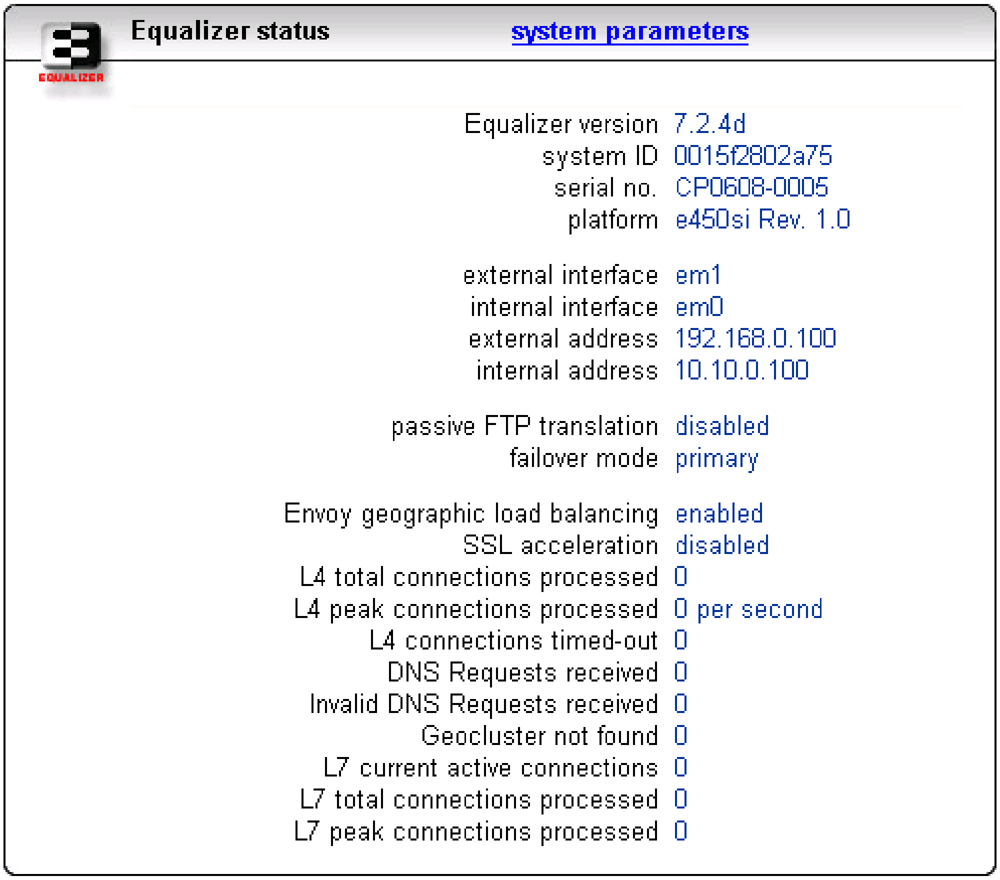
- **To display the current Equalizer status** (a summary of global parameters and usage statistics), see “Displaying Equalizer Status” on page 100.
- **To display messages** recorded in the operating system, cluster, and server logs, see “Displaying the System Event Log” on page 101.
- **To display a status report** on all virtual clusters, see “Displaying the Virtual Cluster Summary” on page 103
- **To examine the current configuration of a cluster** and plot a graph showing its usage over time, see “Displaying Cluster Information” on page 104 and “Plotting Cluster Performance History” on page 105.
- **To examine the current configuration of a server** and plot a graph showing its usage over time, see“Displaying Server Information” on page 107 and “Plotting Server Performance History” on page 108.
- **To examine the current configuration of a geographic cluster** and plot a graph showing its usage over time, see “Displaying Geographic Cluster Parameters” on page 110 and “Plotting Geographic Cluster Performance History” on page 111.
- **To examine the current configuration of a site within a geographic cluster** and plot a graph showing its usage over time, see “Displaying Site Information” on page 112 and “Plotting Site Performance History” on page 113.
- **To configure logs used to collect plotting data**, see “Configuring Plot Logs” on page 113.
- To export usage statistics to a comma separated value (csv) file, see “Exporting Usage Statistics” on page 114.
- **To configure log forwarding, email notification of events, or execution of an external command on events**, see “Configuring Custom Event Handling” on page 114.
- **To configure the SNMP agent** on an Equalizer Model E450si (or above), see “Browsing Equalizer Configurations using SNMP” on page 117.

Displaying Equalizer Status

To display a summary of global parameters and usage statistics for your Equalizer:

1. Log into the Equalizer Administration Interface.
2. At the top of the column in the left frame, click the **Equalizer** entry (or select **View > Equalizer status** from the main menu in the right frame). The **Equalizer status** screen appears in the right frame.

The Equalizer status screen (see Figure 46 on page 100) displays information about Equalizer's operation modes and overall connection statistics.



Equalizer status   system parameters

Equalizer version	7.2.4d
system ID	0015f2802a75
serial no.	CP0608-0005
platform	e450si Rev. 1.0
external interface	em1
internal interface	em0
external address	192.168.0.100
internal address	10.10.0.100
passive FTP translation	disabled
failover mode	primary
Envoy geographic load balancing	enabled
SSL acceleration	disabled
L4 total connections processed	0
L4 peak connections processed	0 per second
L4 connections timed-out	0
DNS Requests received	0
Invalid DNS Requests received	0
Geocluster not found	0
L7 current active connections	0
L7 total connections processed	0
L7 peak connections processed	0

Figure 46   Equalizer status information

- **Equalizer version** shows the current, running version of the Equalizer software.
- **system ID** shows the unique identifier for the Equalizer unit. [Note: in previous releases, this was shown with a colon (:) separating each pair of numbers.]
- **serial no.** is the hardware serial number; this is the same as the serial number on the tag on the back of Equalizer's metal housing.
- **platform** shows the type of Equalizer
- **external interface** is the name of this interface.

- **internal interface** is the name of this interface.
- **external address** is Equalizer’s external IP address.
- **internal address** is Equalizer’s internal IP address.
- **passive FTP Translation** indicates whether PASV FTP mode is enabled or disabled.
- **failover mode** signifies whether this Equalizer is a primary or backup unit.
- **Envoy geographic load balancing** denotes whether geographic load balancing is currently enabled. This information appears only on the E350 and E450 platforms.
- **SSL acceleration** shows whether the optional Xcel™ card is installed, which enables SSL acceleration.
- **L4 total connections processed** is the number of Layer 4 (L4) connections processed.
- **L4 peak connections processed** shows the peak number of L4 connections processed per second since the Equalizer was last booted.
- **L4 connections timed-out** displays the number of L4 connections that have timed out.
- **L7 current active connections** is the number of active Layer 7 (L7) connections.
- **L7 total connections processed** shows the number of L7 connections processed.
- **L7 peak connections processed** is the peak number of L7 connections processed per second since Equalizer was started.

If Envoy is enabled, the following DNS status information appears at the bottom of the Current Status section:

- **DNS requests received** displays the total number of DNS requests received.
- **invalid DNS requests received** shows the number of invalid DNS requests received.
- **Geocluster not found** shows the number of requests received for a geocluster that was not found.

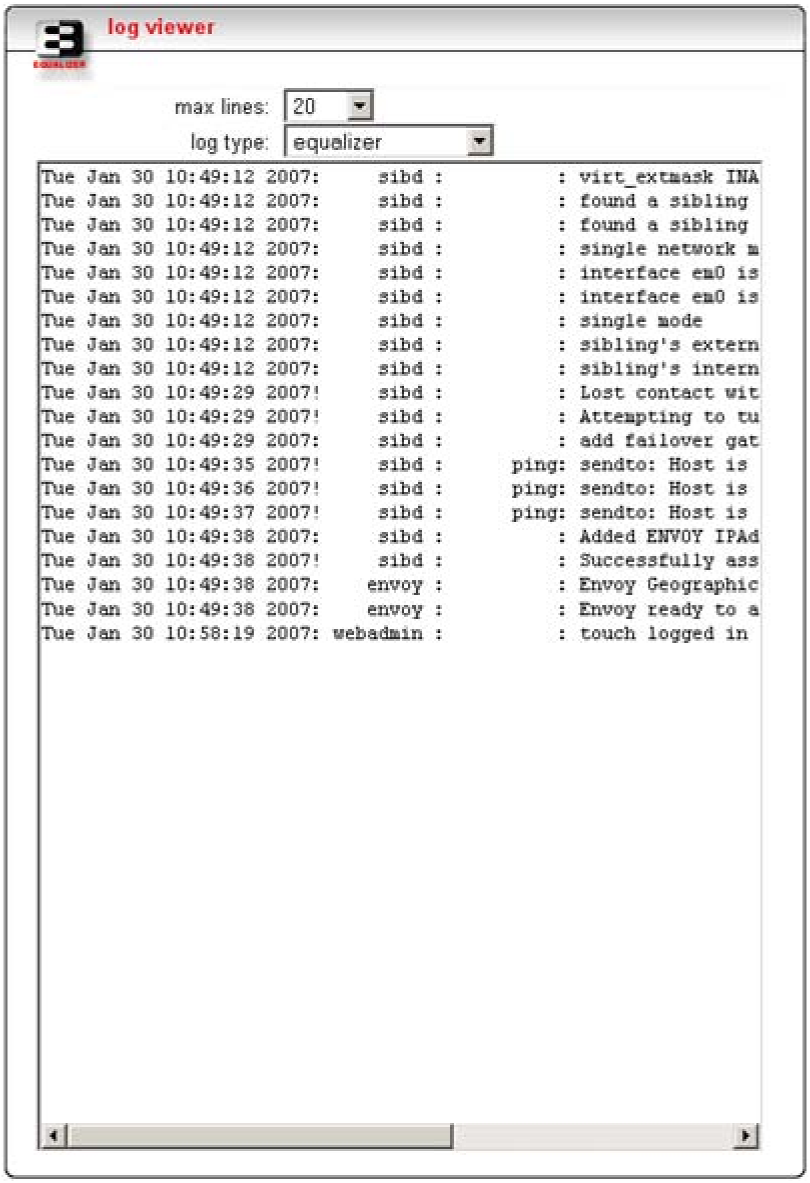
Click on **system parameters** at the top right of the window to display the full list of global parameters. These are described in Chapter 5, under “Modifying System Parameters” on page 43.

Displaying the System Event Log

The System Event Log (see Figure 47) displays start-up and server status messages. You can view the last 20, 50, 100, 200, 500, or 1000 entries.

To view the system event log and optionally change the number of entries on display, follow these steps:

1. Log into the Equalizer Administration Interface in either view or edit mode.
2. Select **Event Log** from the View menu in the main menu bar. The **log viewer** screen appears in the right frame (see Figure 47 on page 102).



max lines: 20

log type: equalizer

```
Tue Jan 30 10:49:12 2007:     sibd :           : virt_extmask INA
Tue Jan 30 10:49:12 2007:     sibd :           : found a sibling
Tue Jan 30 10:49:12 2007:     sibd :           : found a sibling
Tue Jan 30 10:49:12 2007:     sibd :           : single network m
Tue Jan 30 10:49:12 2007:     sibd :           : interface em0 is
Tue Jan 30 10:49:12 2007:     sibd :           : interface em0 is
Tue Jan 30 10:49:12 2007:     sibd :           : single mode
Tue Jan 30 10:49:12 2007:     sibd :           : sibling's extern
Tue Jan 30 10:49:12 2007:     sibd :           : sibling's intern
Tue Jan 30 10:49:29 2007!     sibd :           : Lost contact wit
Tue Jan 30 10:49:29 2007!     sibd :           : Attempting to tu
Tue Jan 30 10:49:29 2007:     sibd :           : add failover gat
Tue Jan 30 10:49:35 2007!     sibd :       ping: sendto: Host is
Tue Jan 30 10:49:36 2007!     sibd :       ping: sendto: Host is
Tue Jan 30 10:49:37 2007!     sibd :       ping: sendto: Host is
Tue Jan 30 10:49:38 2007:     sibd :           : Added ENVOY IPAd
Tue Jan 30 10:49:38 2007!     sibd :           : Successfully ass
Tue Jan 30 10:49:38 2007:    envoy :           : Envoy Geographic
Tue Jan 30 10:49:38 2007:    envoy :           : Envoy ready to a
Tue Jan 30 10:58:19 2007: webadmin :           : touch logged in
```

Figure 47 Viewing the system event log

3. To change the number of lines displayed, select a value from the drop-down list.
4. To look at the logs for the Equalizer, a virtual cluster, or the operating system, select a log from the **log type** drop-down list.

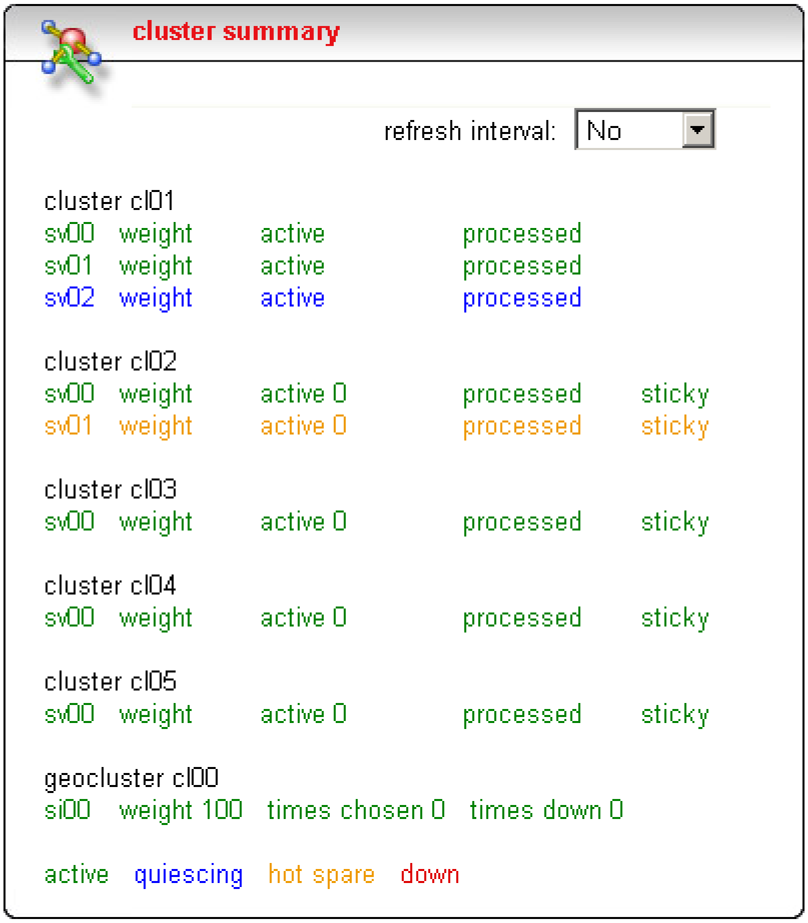
To export the contents of a log, you can copy text from the **log viewer** screen and paste it into another application (such as Windows Notepad); then, save the text to a file.

Displaying the Virtual Cluster Summary

The Virtual Cluster Summary (see Figure 48) lists the currently configured virtual clusters and their associated servers as well as the weight and status of each server.

To view the Virtual Cluster Summary, follow these steps:

1. Log into the Equalizer Administration Interface in either view or edit mode.
2. Select **View > Cluster Summary** from the main menu bar. The **cluster summary** screen appears in the right frame.



cluster summary

refresh interval: No

```
cluster cl01
sv00   weight      active          processed
sv01   weight      active          processed
sv02   weight      active          processed

cluster cl02
sv00   weight      active 0        processed     sticky
sv01   weight      active 0        processed     sticky

cluster cl03
sv00   weight      active 0        processed     sticky

cluster cl04
sv00   weight      active 0        processed     sticky

cluster cl05
sv00   weight      active 0        processed     sticky

geocluster cl00
si00   weight 100  times chosen 0  times down 0

active   quiescing   hot spare   down
```

Figure 48 Viewing cluster summary information

This summary displays the status at the time the page was loaded. To set this information to automatically refresh, select a refresh interval.

Equalizer periodically queries servers that have gone down to determine if they have become available again. When a server comes back online, Equalizer begins to route requests to the server, slowly increasing the server’s weight to its full capability.

For each server, the summary displays the following information:

- **weight**: The server weights determine the relative proportion of connection requests that Equalizer routes to each server. If you have enabled automatic load balancing, these weights are the current, dynamically-adjusted values, not the static weights initially assigned by the administrator.

- **active**: The number of current connections to the server.
- **processed**: The total number of connections that have been processed by the server since the system was rebooted.
- **sticky** (Layer 4 clusters only): The number of “sticky records” currently held by Equalizer. Each one of these represents a Layer 4 connection to an L4 TCP or UDP cluster with a non-zero **sticky time**. See “Enabling Sticky Connections” on page 80.

For each site in a geocluster, the summary displays the following information:

- **weight**: The server weights determine the relative proportion of connection requests that Equalizer routes to each server. If you have enabled automatic load balancing, these weights are the current, dynamically-adjusted values, not the static weights initially assigned by the administrator.
- **times chosen**: the number of times this site was selected by geographic load balancing to respond to a client request.
- **times down**: the number of times this site was down when geographic load balancing was attempting to select a site to respond to a client request. **Note**: this does not indicate current site status. Current site status is indicated by the color of the site in the table: green indicates up, red indicates it is down, and yellow indicates the site is a hot spare.

The cluster summary indicates the following server states:

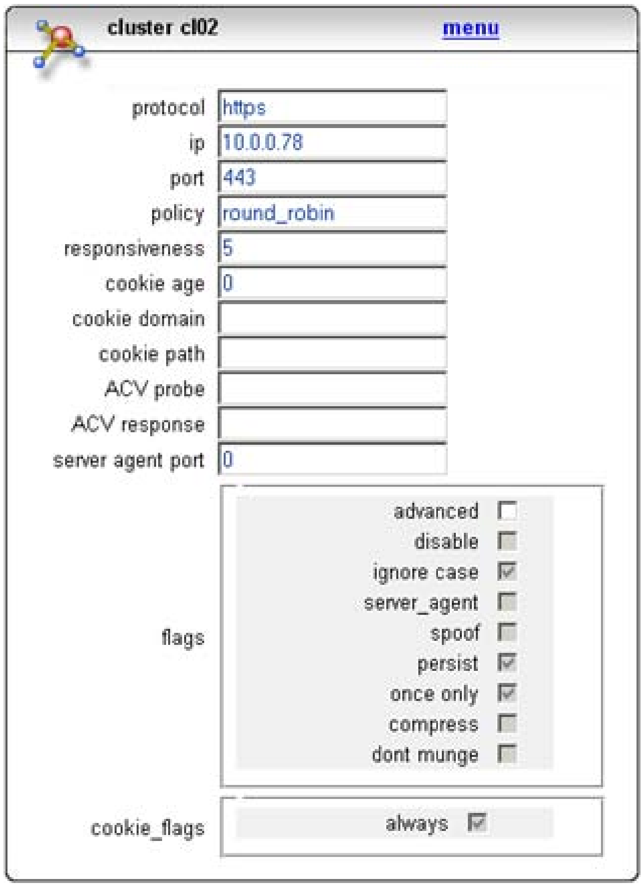
- Servers shown in green are currently active.
- Servers shown in blue are quiescing, that is, handling current connections but not accepting new ones.
- Servers shown in yellow are configured as hot spares.
- Servers shown in red are down. Equalizer monitors the status of active servers by periodically probing the IP address and Port specified by the server endpoint. If these probes fail the number of times specified by the **strikeout threshold** system parameter (see page 44), it marks the server *down*, gives the server a weight of zero, and stops routing new requests to that server. A server probe might fail even if the server machine is up and running. For instance, if the HTTP server daemon fails on a server machine, Equalizer will refuse connections to that endpoint.

Displaying Cluster Information

The **cluster** screen (see Figure 49) displays information about a cluster’s configuration. To display the parameters for a cluster, follow these steps:

1. Log into the Equalizer Administration Interface in either view or edit mode.

2. In the left frame, click the name of the cluster whose parameters you want to view. The **cluster** screen appears in the right frame.



protocol	https
ip	10.0.0.78
port	443
policy	round_robin
responsiveness	5
cookie age	0
cookie domain	
cookie path	
ACV probe	
ACV response	
server agent port	0
flags	advanced □ disable □ ignore case ☑ server_agent □ spoof □ persist ☑ once only ☑ compress □ dont munge □
cookie_flags	always ☑

Figure 49 Viewing cluster information

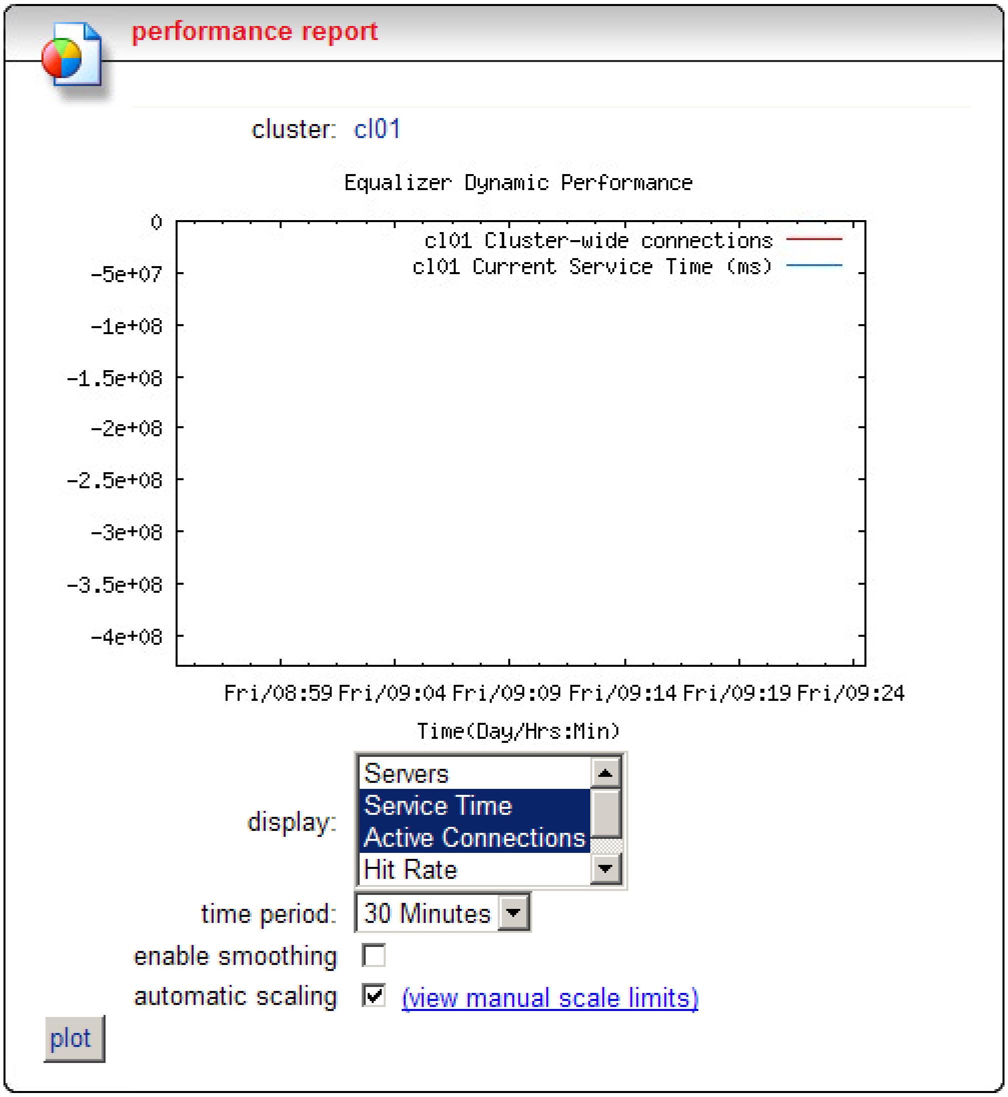
The **cluster** screen shows the selected load balancing policy, the load-balancing responsiveness setting, the persistence parameters, and the server agent parameters. For more information about how Equalizer uses these parameters, see “Adding a Virtual Cluster” on page 67.

Plotting Cluster Performance History

The Plot Cluster History feature (see Figure 50) enables you to view a graphical representation of the performance history for any cluster. To plot the performance history for a cluster:

1. Log into the Equalizer Administration Interface in either view or edit mode.
2. In the left frame, click the name of the cluster whose history you want to view.

3. Select **menu > Plot Cluster History** in the **cluster** screen. The graphical history for the selected cluster appears.



cluster: cl01

display: Servers / **Service Time** / **Active Connections** / Hit Rate

time period: 30 Minutes

enable smoothing □

automatic scaling ☑ (view manual scale limits)

Figure 50 Viewing a cluster's graphical history

By default, the service time and active connections are plotted for the previous five minutes. To change the information plotted, select the categories and duration you want to plot and click the **Plot** button.

The **enable smoothing** checkbox smooths out peaks and valleys in the graph.

The **automatic scaling** checkbox ssales the graph automatically to show all values. To set upper and lower limits on the data presented, click on **view manual scale limits**.

To zoom in on a portion of the graph, click the target area.

You can plot five values for a cluster:

- **Servers** is the average computed load of all the servers in the cluster. Because server computed loads are normalized by the cluster-wide average, the cluster-wide average should be 100. Certain events (for example, rapid fluctuations in the load, rebooting servers, and restarting application daemons such as httpd) can cause spikes in the computed load for the cluster.

- **Service Time** is the average service time of all of the servers in the cluster. The service time is the time it takes a server to start sending reply packets once it receives a client request. The average service time is a reasonable indication of the overall performance of the cluster.
- **Active Connections** is the total number of active connections on the servers in the cluster.
- **Hit Rate** is the number of connections served by the cluster each second. This is a good indication of how many "hits" the site is getting.
- **Server Agent** is the average of the dynamic server agent values for all servers in the cluster. If you have not configured server agents, this value defaults to 50 (that is, the value 50 is used by the load balancing algorithm).

For more information about these values, see the descriptions in "Plotting Server Performance History" on page 108.

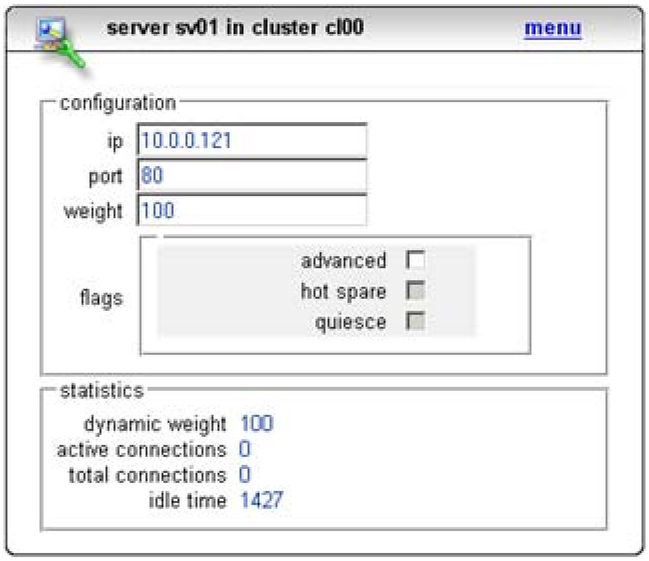
Displaying Server Information

The server screen (see Figure 51) provides information about a particular server, including the following:

- The server's name and the name of the cluster to which the server belongs.
- The server's IP address and port.
- The static weight the administrator assigned to the server.
- Other configuration information such as being a hot spare or being quiesced.

To display a server's parameters, follow these steps:

1. Log into the Equalizer Administration Interface in either view or edit mode.
2. In the left frame, click the name of the server whose parameters you want to view. The server's parameters appear in the right frame.



server sv01 in cluster cl00 — menu

configuration

ip 10.0.0.121

port 80

weight 100

flags: advanced □ hot spare □ quiesce □

statistics

dynamic weight 100

active connections 0

total connections 0

idle time 1427

Figure 51 Viewing server information

The **dynamic weight** is the current weight assigned by Equalizer to the server. The **active connections** and **total connections** show current connections statistics. The number of seconds spent idle is shown as **idle time**. For a Layer 4 server, **idle time** is replaced by **sticky connections**; a Layer 4 connection to an L4 TCP or UDP cluster with a non-zero **sticky time**. See "Enabling Sticky Connections" on page 80.

Plotting Server Performance History

The Plot Server feature (see Figure 52) enables you to view a graphical representation of the performance history for any server.

To plot the a server's performance history, follow these steps:

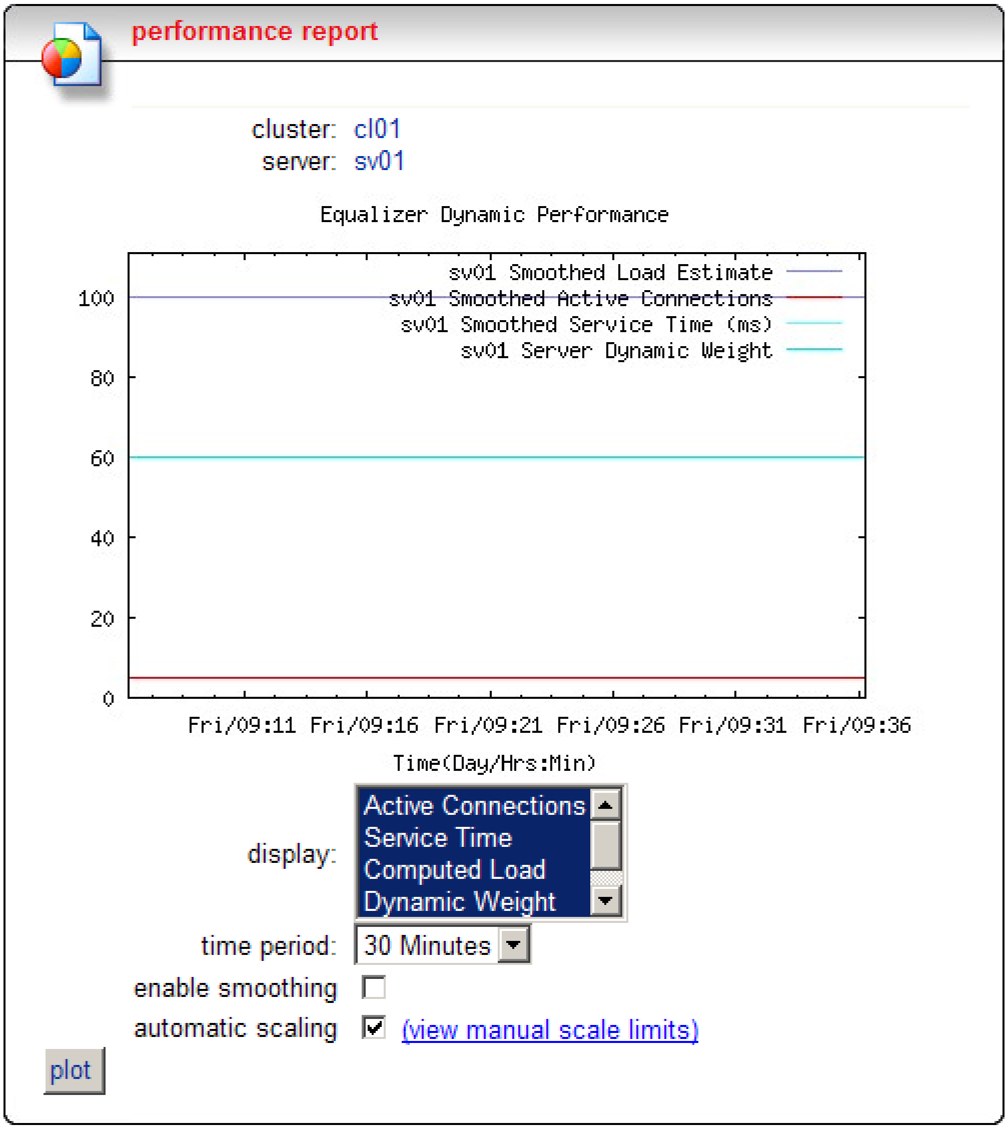
1. Log into the Equalizer Administration Interface in either view or edit mode.
2. In the left frame, click the name of the server whose history you want to view.
3. Select **Plot Server History** from the local menu in the server screen. The graphical history for the selected server appears. See Figure 52 on page 109.

   By default, the active connections, service time, computed load, and dynamic weight are plotted for the previous 30 minutes. To change the information plotted, select the categories and duration you want to plot and click the **Plot** button.

The **enable smoothing** checkbox smooths out peaks and valleys in the graph.

The **automatic scaling** checkbox ssales the graph automatically to show all values. To set upper and lower limits on the data presented, click on **view manual scale limits**.

To zoom in on a portion of the graph, click the area in which you are interested.



performance report

cluster: cl01
server: sv01

Equalizer Dynamic Performance
sv01 Smoothed Load Estimate
sv01 Smoothed Active Connections
sv01 Smoothed Service Time (ms)
sv01 Server Dynamic Weight
100
80
60
40
20
0
Fri/09:11
Fri/09:16
Fri/09:21
Fri/09:26
Fri/09:31
Fri/09:36
Time(Day/Hrs:Min)

display: Active Connections, Service Time, Computed Load, Dynamic Weight

time period: 30 Minutes

enable smoothing ☐

automatic scaling ☑ (view manual scale limits)

Figure 52 Viewing a server's graphical history

You can plot five values for a server:

- **Active Connections** shows the number of active connections on the server. Equalizer “smooths” the connection count using a sliding-window smoothing algorithm before being plotted. If you have enabled the sticky timer, note that the number of active connections on a server will be higher.
- **Service Time** indicates the time it takes a server to start sending reply packets once it has received a client request. This value is very small for servers that are primarily serving static HTML pages—typically 100-200 milliseconds. If the server is serving many active pages and cgi-bins, this value will be much higher. The service time increases when the server is under heavy load because client requests are queued until the server can handle them.

- **Computed Load** is a measure of the performance of the server relative to the overall performance of the cluster. Equalizer tries to normalize the cluster-wide computed load value to 100. If the server's computed load value is above 100, it is performing below the overall cluster performance.

  Equalizer derives a server's computed load value from its service time, number of active connections, and server agent value (if configured). It is also takes into account the load balancing policy used by the cluster.

  Ideally, a server's computed load should be around 100, though values in the range 85 to 115 are reasonable. If the server's computed load is higher than 115, the server is not performing well and you may need to add servers or upgrade to better servers. If you are using adaptive load balancing, Equalizer lowers the server's dynamic weight to reduce the number of connections sent to that server. If the server's computed load value is less than 85, the server is performing very well and Equalizer will attempt to improve cluster-wide performance by increasing the server's dynamic weight to direct more traffic to it. Such adjustments to the server's weight will in turn affect its computed load value.

- **Dynamic Weight** is the percentage of incoming traffic that Equalizer dispatches to this server. For example, if the cluster has three servers with dynamic weights of 100, 80, and 120, the first server will get 100/(100+80+120) or 33.3% of the incoming traffic.

  If a server is down, its dynamic weight is zero. If a server crashes and reboots, the period that the server was down shows up as a gap in the dynamic weight plot.

  If you are not using adaptive load balancing (for example, the load balancing policy is set to *round robin* or *static weight*), Equalizer does not use dynamic weights. For more information about setting the load balancing policy and adaptive load balancing, refer to "Configuring a Cluster's Load-Balancing Options" on page 74.

- **Server Agent** is the value that the server agent daemon returns. When queried, the server agent returns a value in the range -1 to 100. If you have not configured the cluster to use the server agent or the server agent daemon is not running on this server, the server agent value defaults to 50 (that is, a value of 50 is used by the load balancing algorithm).

  Server agent values above 60 to 70 indicate that the server is overloaded. If this persists and you have enabled adaptive load balancing, Equalizer responds by reducing the server's dynamic weight so that fewer requests are routed to the server.

> **Note –** If all your servers have server agent values above 70, you probably have more traffic than your servers can handle efficiently. In this case, Equalizer can help by intelligently managing the overload, but the long-term solution is to upgrade the servers or add new ones.

Displaying Geographic Cluster Parameters

If you have installed Envoy for your Equalizer, you can view information about each of the geographic clusters that you have configured. For more information about Envoy, refer to Chapter 8, "Administering Geographic Clusters" on page 149.

To view the cluster-wide parameters, follow these steps:

1. Log into the Equalizer Administration Interface in either view or edit mode.
2. In the left frame, click the name of the geographic cluster whose parameters you want to view. The Geographic Cluster Parameters screen appears in the right frame.

This page contains the following information:

- **Geographic Cluster**, which is the name of the cluster.
- **DNS TTL**, which is the amount of time, in seconds, that a name server is allowed to cache the domain information.
- **MX Exchanger**, which is the fully-qualified domain name that Equalizer will return if Equalizer receives a “mail exchanger” request for this geographic cluster. The mail exchanger is the host responsible for handling email sent to users in the domain.
- **Load Balancing Method** indicates the load-balancing method: round trip, adaptive, site load, or site weight. (For descriptions of these methods, refer to “Configuring a Geographic Cluster’s Load-Balancing Options” on page 156.)
- **Load Balancing Response** shows the type of response: slowest, slow, medium, fast, or fastest. This value controls how aggressively Equalizer adjusts the site’s dynamic weights. (For more information about the response settings, refer to “Adding a Geographic Cluster” on page 155.)
- **ICMP Triangulation** shows whether you have enabled ICMP triangulation, which routes client requests to the closest site geographically.

Plotting Geographic Cluster Performance History

If you have installed Envoy for your Equalizer, you can use the Plot Geographic Cluster feature to view a graphical representation of the performance history for the selected geographic cluster.

You can plot four values for a geographic cluster:

- **Request Rate** shows the number of requests received for the cluster per minute.
- **Active Requests** displays the number of requests that Equalizer is in the process of routing.
- **Network Latency** displays the average triangulation time when at least one site was able to respond. (This value does not include clients for which the default site was selected.)
- **Site Summary** shows the number of requests directed to all sites in the cluster for the specified duration. This plot appears by default when the plot site page is opened.

> **Note –** You can only display the site summary separately; you cannot plot the site summary on the same graph as the other values.

To plot the performance history for a geographic cluster, follow these steps:

1. Log into the Equalizer Administration Interface in either view or edit mode.
2. In the left frame, click the name of the geographic cluster whose history you want to view. The Geographic Cluster Parameters appear in the right frame.

3. Select **Plot GeoCluster History** from the local menu in the Geographic Cluster Parameters frame. The graphical history for the selected cluster appears in the right frame. By default, the site summary for the previous 30 minutes appears.
4. To change the information being plotted, select the categories and duration to be plotted; then click the **Plot** button. (To zoom in on a portion of the graph, click the area in which you are interested.)

Displaying Site Information

If you have installed Envoy, you can view configuration and status information for particular sites in a geographic cluster.

To view the information for a particular site, follow these steps:

1. Log into the Equalizer Administration Interface in either view or edit mode.
2. In the left frame, click the name of the site whose information you want to view. The Site Parameters page appears in the right frame.

The Site Parameters page displays the following site parameters:

- **Geographic Cluster** is the name of the geographic cluster to which this site belongs.
- **Site** is the name of the target site.
- **Site IP Address** is the site’s IP address.
- **Static Weight** shows the static weight assigned to the site.
- **Default Site** indicates whether the target site is the default site.
- **Resource** shows the IP address and port of the resource being monitored for this site.
- **Agent’s Address** is the IP address of the Equalizer agent running on the site.
- **Resource Keepalive** shows the number of seconds between resource availability checks. If a resource fails its availability check, its site will not be returned to clients. Even after a resource is declared dead, Equalizer performs availability checks to determine when the resource is restored.

In addition to the site parameters, the site’s current status appears as follows:

- **Resource Load** shows the load on the above resource that the Equalizer agent calculates. The load incorporates data on resource response time, number of active requests, and load-balancing variables.
- **Agent Retries** shows the number of probes Equalizer re-sent to its agent.
- **Agent Misses** shows the number of Equalizer-to-agent probes that received no response. Interruptions in network connectivity between the Equalizer server and site agents and site failures can result in missed probes.
- **Triangulation Time-outs** indicates the number of agent-to-client triangulation probes that timed out before Equalizer received a response.
- **Resource Errors** indicates the number of Equalizer-to-agent probes that returned a resource-unavailable error. If the Envoy on the remote site determines that the requested resource is unavailable, it returns a resource unavailable error.

- **Site Returned** shows the number of clients directed to this site. You can compare this number with the values for other sites to determine the relative number of users sent to each site. If a value for one site is zero and the others are non-zero, consider why the zero site has no traffic.
- **Returned as Default** indicates the number of clients directed to the default site.
- **Average Ping Time** shows the average triangulation time for all clients successfully contacted from this site. This represents all of the triangulation probes—whether or not this site was selected to process the request. This value gives you an idea of the network latency from this site to the user population. You can compare this value with the same value for other sites.

Plotting Site Performance History

If you have installed Envoy, the Plot Site feature enables you to view a graphical representation of the performance history for the selected site. To plot the performance history for a site, follow these steps:

1. Log into the Equalizer Administration Interface in either view or edit mode.
2. In the left frame, click the name of the site whose history you want to view. The Site Parameters appear in the right frame.
3. Select **Plot Site History** from the local menu in the Site Parameters frame. The graphical history for the selected site appears in the right frame.

By default, Equalizer plots the Request Rate and Resource Down values for the previous 30 minutes. To change the information plotted, select the categories and duration to be plotted; then click the **Plot** button. (To zoom in on a portion of the graph, click the area in which you are interested.)

You can plot the following six values for a site:

- **Probes Missed** is the number of requests in which an agent failed to reply to Equalizer’s probes.
- **Triangulation Errors** shows the number of ICMP ECHO requests that the agent at this site sent to clients and for which the agent received no response.
- **Resource Down** indicates that the target resource failed to respond during the period plotted.
- **Site Chosen** shows the number of times that Equalizer returned this site in response to a client query.
- **Network Latency** shows the average network distance, in milliseconds, between the agent at this site and the clients that made DNS requests.
- **Resource Load** is the relative workload of this site during the plotted period.

Configuring Plot Logs

A global parameter, **log hours**, controls the size and number of plot logs that Equalizer uses to collect the data for plotting performance graphs. This parameter is set in the **modifying system parameters** screen, displayed by selecting **Equalizer > Global Configuration** from the main menu of the Equalizer Administrative Interface.

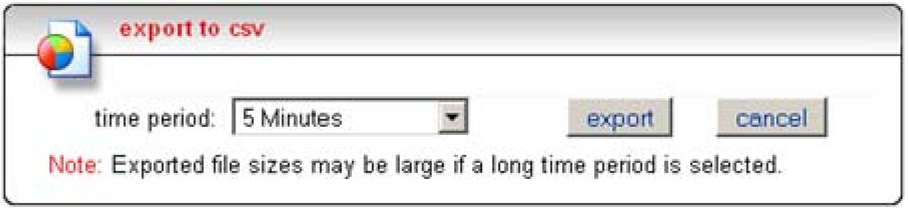
log hours is the target number of hours of plot log data to retain. A zero in this field allots the numbers of hours based on the available memory. Note that the number of hours of log data retained is limited by the amount of memory and disk space. If you define a large number of clusters and servers, this will limit the amount of time over which log data can be retained. For example, a system with 10 clusters each with 10 servers might only be able to retain about 4 hours of log data.

Exporting Usage Statistics

You can export usage statistics, including the data collected for plotting cluster and server histories, to a comma separated value (**.csv**) file that can be opened in any program (such as Excel) that accepts comma separated data as input. The data is exported to the browser in a file with the default name **export.csv**. All available statistical data is exported for the time period selected.

To export usage statistics, do the following:

1. Log into the Equalizer Administration Interface in Edit mode.
2. Select **View** > **Export to CSV** from the main menu bar; **export to csv** screen appears in the right frame.



time period: 5 Minutes

Note: Exported file sizes may be large if a long time period is selected.

Figure 53 The **export to csv** screen

3. Select the time period for which you want to export the data from the drop-down box.
4. Select **export** to download the file for saving via your browser.

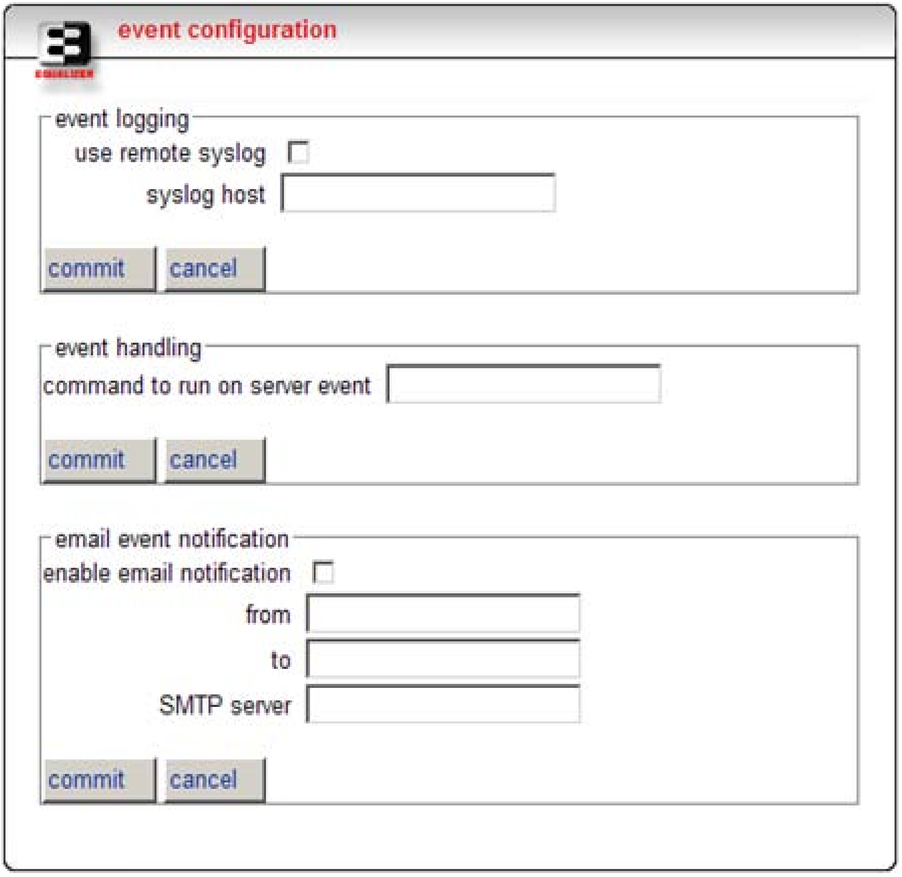
Configuring Custom Event Handling

You can configure Equalizer to perform certain actions when a server fails or other critical events occur. You can forward Equalizer log information to another machine, and specify a command to run or email to be sent when a server event occurs.

Forwarding Equalizer Log Information

You can forward Equalizer's System Event Log (see "Displaying the System Event Log" on page 101), to another machine that is running a **syslog** daemon. To specify a syslog host to which you will forward the log, follow these steps:

1. Log into the Equalizer Administration Interface in Edit mode.
2. Select **Equalizer > Global Configuration** from the main menu bar; then select **menu > Events** from the **modify system parameters** screen. The **event configuration** screen appears in the right frame.



event configuration

event logging

use remote syslog □

syslog host

event handling

command to run on server event

email event notification

enable email notification □

from

to

SMTP server

Figure 54 The event configuration screen

3. Check the **use remote syslog** checkbox.
4. In the s**yslog host** field, enter the hostname (not the IP address) of the machine to which you want to forward syslog messages. (Remember that the system you specify must be configured to be a syslog host; see the documentation for the operating system running on that system for more information.)
5. Click the **commit** button.

Specifying a Command to Run When a Particular Event Occurs

You can configure Equalizer to run a command that you specify (such as sending an e-mail or running a custom shell script) whenever server events occur. The following events trigger the specified command:

- Failure of a server
- Restoration of a failed server

- Failure of a server agent
- Restoration of a server agent
- Failover in a high-availability Equalizer pair

For example, to append a dated message to a log file whenever Equalizer detects a server failure, you could enter the following command:

```
echo ‘date‘ “System Failure.” >> /tmp/mylog
```

To specify a command to run, follow these steps:

1. Log into the Equalizer Administration Interface in Edit mode.
2. Select **Equalizer** > **Global Configuration** from the main menu bar; then select **menu** > **Events** from the **modify system parameters** screen. The **event configuration** screen appears in the right frame (see Figure 54 on page 115).
3. In the **command to run on server event** field, enter the command that you want Equalizer to run when it detects a server event.
4. Click the **commit** button.

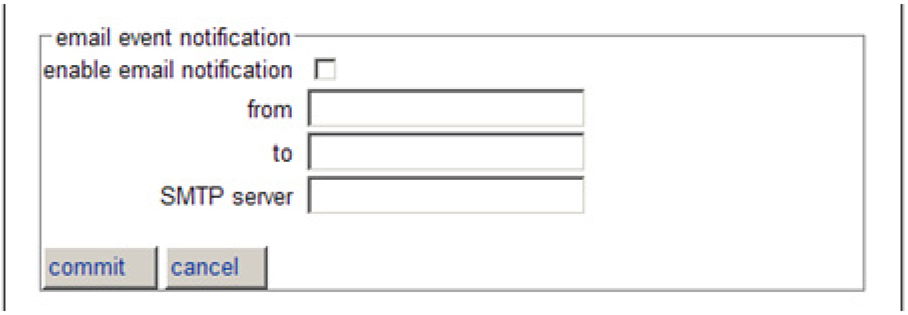
> **Note –** Any program that is specified in the command and that is to run for a server event must complete its work and terminate within one or two seconds to avoid interrupting Equalizer’s server failure detection facility.

Configuring Email Notification When a Particular Event Occurs

You can configure Equalizer to send an email notification whenever a server event occurs, for the same list of events shown above. You need to specify the sender and recipient email addresses, as well as the Simple Mail Transfer Protocol (SMTP) server for this feature to work. Any SMTP server will work with Equalizer, and usually will reside on another system on your network.

To configure, enable, or disable email notification, you can use either the **global parameters** screen or the **event configuration** screen. All email parameters are listed on both screens, and a change on one screen will be reflected on the other. The procedure below shows you how to use the **event notification** screen to configure, enable, and disable email notification. To use the **global parameters** screen, see the section “Modifying System Parameters” on page 43, in Chapter 5.

1. Log into the Equalizer Administration Interface in Edit mode.
2. Select **Equalizer** > **Global Configuration** from the main menu bar; then select **menu** > **Events** from the **modify system parameters** screen. The **event configuration** screen appears in the right frame.The **email notification** section of this screen is shown below.



3. In the **email event notification** section, enter the sender of the email in the **from** field using the format required by your SMTP server.

   The address format to use depends on how your SMTP server is configured. For many servers, the ***user@domain*** (e.g.: **admin@example.com**) format will be acceptable. Some servers can be configured to require sender and recipient addresses that conform strictly to the RFC821 standard. For example, a **postfix** SMTP server has an option called **strict_rfc821_envelopes** that, when enabled, requires that all addresses must be enclosed in angle brackets, as in **<*user@domain*>** (e.g.: **<admin@example.com>**). If such a server receives an email whose sender or recipient addresses are not enclosed in angle brackets, the server will return an address syntax error.

   Check the settings on your SMTP server to determine the address format you need to use, or ask your network administrator.

   If you leave the **from** field blank, the default address **events@*hostname.domain*** (e.g.: **events@sv01.example.com**) will be used. (The hostname and domain used are part of the global parameters specified when you set up the Equalizer hardware.)

4. Enter the recipient of the email in the **to** field using the format required by your SMTP server, as described in the previous step. A recipient address must be specified.
5. Enter the SMTP address used for forwarding email using either dot notation (`10.0.0.10`) or the hostname in the **SMTP server** field. The SMTP server must be listening on port 25.
6. Check the **enable email notification** checkbox (this box allows you to turn off email notification later without removing your email configuration, as shown in the next section).
7. Click the **commit** button. (If the **to** or **SMTP server** fields are blank, or if you do not check the **enable email notification** check box, you will not be able to commit the changes.)

Disabling Email Notification When a Particular Event Occurs

To disable email notification:

1. Follow Steps 1 and 2 in the previous section.
2. On the **event configuration screen**, clear the **enable email notification** checkbox.
3. Select **commit**.

Browsing Equalizer Configurations using SNMP

The Simple Network Management Protocol (SNMP) is an internet standard that allows a management station to monitor the status of a device over the network. SNMP organizes information about the Equalizer and provides a standard way to help gather that information. Using SNMP requires:

- An SNMP agent running on the system to be monitored.
- A Management Information Base (MIB) database on the system to be monitored.
- An SNMP management station running on the same or another system.

An SNMP agent and MIB databases are provided on Equalizer Models E450si and above, implemented for SNMPv1 and SNMPv2c.

A management station is not provided with Equalizer and must be obtained from a third party supplier. The management station is often used primarily to browse through the MIB tree, and so is sometimes called a MIB browser. One such management station that is available in a free personal edition is the iReasoning MIB Browser, available from `http://www.ireasoning.com`.

A MIB database is a hierarchical tree of variables whose values describe the state of the monitored device. A management station that want to browse the MIB database on a device sends a request to the SNMP agent running on the device. The agent queries the MIB database for the variables requested by the management station, and then sends a reply to the management station.

With SNMP, you can monitor the following information from the Equalizer MIBs:

Static configuration information, such as:

- Device name and Model
- Software version
- Internal and external IP addresses and netmasks
- Default gateway
- Failover alias

Equalizer's failover details

- Sibling Name
- Sibling Status (Primary or Secondary)

Dynamic configuration information, such as:

- Failover status
- NAT enabled
- L4 configuration state
- L7 configuration state
- Server Health check status
- Email status notification
- Cluster parameters (timeouts, buffers)
- Server parameters

Equalizer status

- L4 Statistics
- L7 Statistics

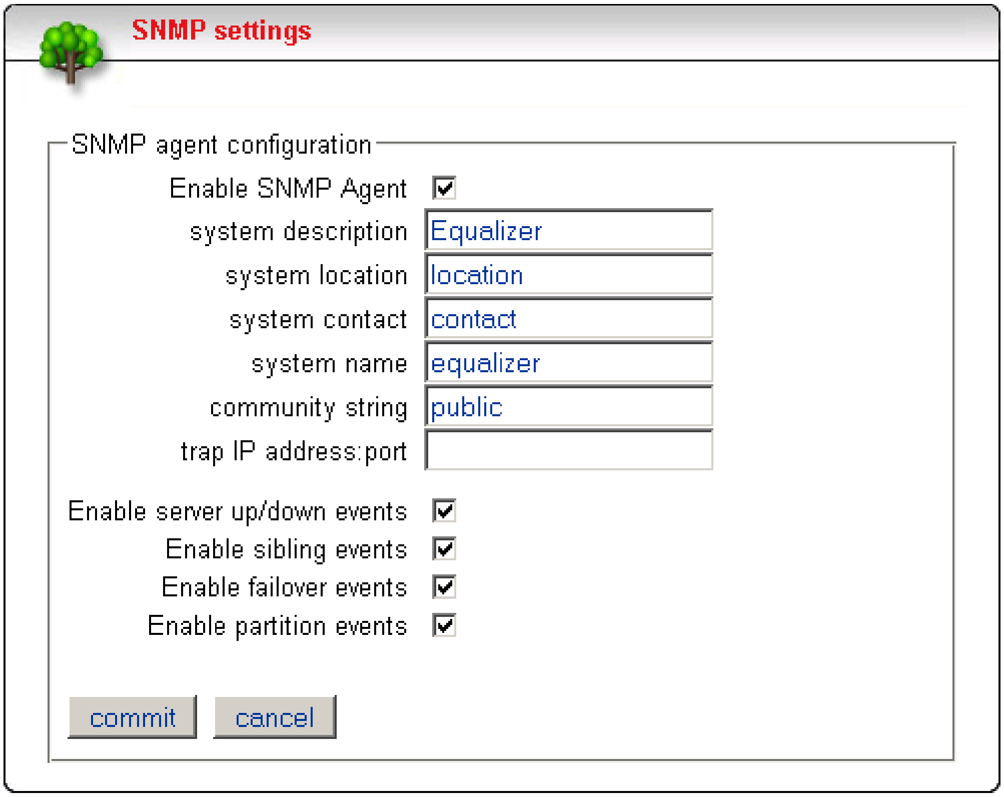
Equalizer cluster configuration

- L4 or L7 protocol of cluster
- Load balancing policy for cluster.
- IP address and port (or range)
- Sticky time and cross cluster sticky
- Cookie on or off

Enabling the SNMP Agent

The SNMP agent responds to outside SNMP requests, usually from an SNMP management station. To configure the SNMP agent, follow these steps from the Equalizer Administration Interface in Edit mode.

1. Choose **Equalizer > Global Configuration** from the main menu.
2. In the **modify system parameters** screen, select **menu > SNMP**. The **SNMP settings** screen appears in the right frame (see Figure 55).



SNMP settings

SNMP agent configuration

Enable SNMP Agent ☑

system description: Equalizer

system location: location

system contact: contact

system name: equalizer

community string: public

trap IP address:port:

Enable server up/down events ☑

Enable sibling events ☑

Enable failover events ☑

Enable partition events ☑

Figure 55 The SNMP settings screen.

3. Enter values for the **system description**, **system location**, **system contact**, and **system name**. Description is the user-assigned description of the Equalizer. Location describes its physical location. Contact is the name of the person responsible for this unit. Name is the administrative name for the Equalizer.
4. Enter a value for the **community string**. Any SNMP management console needs to send the correct community string along with all SNMP requests. If the sent community string is not correct, Equalizer discards the request and will not respond.
5. Enter an address and port in **trap IP address:port**. This specifies the IP address and port to which trap messages should be sent. Usually this is the IP address of the machine running the SNMP management station application. The port number used by default is 162, which is the default port used by SNMP management stations; it must match the port on which the SNMP management station is listening for traps.

6. Use the check boxes to enable the corresponding traps. The following table shows the traps that are enabled or disabled using the check boxes.

Enable server up/down events	This checkbox controls two traps, `cpsSysEqServerDownEv` and `cpsSysEqServerUpEv`. Equalizer triggers these traps when it detects either a server failure or a response from a failed server.
Enable sibling events	This checkbox controls two traps, `cpsSysEqSiblingContactLostEv` and `cpsSysEqSiblingContactOkayEv`. Equalizer triggers these traps whenever it is configured as part of a failover pair and it either loses contact or regains contact (respectively) with its sibling.
Enable failover events	This checkbox controls one trap, `cpsSysEqAssumedPrimaryRoleEv`. Equalizer sends this trap whenever it assumes primary status.
Enable partition events	This checkbox controls one trap, `cpsSysEqPartitionDetectedEv`. Equalizer sends this trap whenever it is in failover mode and detects that both Equalizers have assumed primary status.

7. Make sure the **Enable SNMP Agent** checkbox is turned on to start SNMP. To disable SNMP without removing your configuration, turn off the **Enable SNMP Agent** checkbox.

8. Click **commit** to save your changes.

Setting Up an SNMP Management Station

An SNMP management station is not provided with Equalizer. In order to use SNMP to manage an Equalizer, a third-party management console must be installed and configured on a machine that can access the Equalizer system. Configuration procedures are specific to the management console used.

At a minimum, the SNMP management console needs to be configured to:

- Use the Equalizer’s IP address and port 161 for SNMP requests.
- Use the **community string** specified in the above procedure.
- Use the address and port specified in the above procedure for SNMP traps (usually port 162 is used for this purpose, but this can be configured as shown in the above procedure).
- Use the Equalizer MIB definitions; these need to be loaded into the management console, following the instructions for the console. The Equalizer MIB source files are located at:

  ```
  http://<Equalizer-ip>/eqmanual/cpsreg.my
  http://<Equalizer-ip>/eqmanual/cpsequal.my
  ```

  In the above, `<Equalizer-ip>` is the IP address of the Equalizer. On the Equalizer, these are located in the directory */usr/local/www/eqmanual*.

MIB Description

Equalizer’s Management Information Base (MIB) contains five major sections. These sections describe Equalizer’s siblings (failover), configuration and status, clusters, servers, and events. Each object in the MIB contains a description field that describes the object’s purpose. All of the MIB objects are read-only; that is, SNMP **Set** operations are not supported.

The following is a summary description of the Equalizer MIB. The MIB source files contain detailed comments for each variable; these comments may also be displayed by the MIB browser when a variable is accessed.

Siblings

The main object that describes siblings is *cpsSysEqSiblings*. This describes any siblings for failover configurations.

Configuration and Status

The main object, *cpsSysEqualizer*, is the largest object in the MIB and contains many sub-objects. These sub-objects include:

eqStaticCfg - This group contains the static configuration information such as the name of the Equalizer, the software version, internal and external IP addresses and netmasks, default gateway, failover alias, etc.

eqDynamicCfg - This group consists of several sub-groups and contains no variables of its own. The sub-groups are:

> *eqGlobalDynamicCfg* - This group contains a number of global configuration items including failover status, whether or not outbound NAT is enabled, etc.
>
> *eqL4DynamicCfg* - This group contains configuration variables specific to Layer 4 load balancing, the state of passive FTP, idle timeout, stale timeout, etc.
>
> *eqL7DynamicCfg* - This group contains configuration variables specific to Layer 7 load balancing, including send and receive buffer sizes, the state of SSL encryption, etc.

eqStatus - This group consists of two sub-groups and contains no variables of it's own. The sub-groups are.

> *eqL4Status* - This group contains Layer 4 statistics such as number of connections processed, peak connections, and idle timeout count.
>
> *eqL7Status* - This group contains L7 statistics such as active connections, peak connections and total number of connections.

Clusters

The main object that describes clusters is *cpsSysEqClusters*. This consists of a set of tables describing the configuration of, and operational statistics for, all of the virtual clusters configured within the system.

Servers

The main object that describes servers is *cpsSysEqServers*. This consists of a set of tables describing the configuration of, and operational statistics for, all of the servers configured within each virtual cluster within the system.

Events

The main object that describes Equalizer events is *cpsSysEqEvents*. This contains variables that control whether or not traps are globally enabled and enable flags for each of the individual trap events.

8 Working with Match Rules

Why Match Rules?

The ability to make load balancing decisions based on the content of a client request is what separates Layer 7 processing from the processing options available at Layer 4. For Layer 7 clusters, Match Rules provide fine-grained control over load balancing decisions based on the content of the client request. If you need to be able to route requests to the servers in a cluster based on the content of the request, Match Rules are the answer.

Note – Match rules are supported on Equalizer Models E350 and higher models; they are *not* supported on E250 models.

Match Rules Overview

Layer 7 clusters can use logical constructs called “match rules” to control the processing of the incoming data stream from clients. Match rules extend the Layer 7 load balancing capabilities of HTTP and HTTPS clusters by allowing you to define a set of logical conditions which, when met by the contents of the request, trigger the load balancing behavior specified in the match rule.

Typically, a match rule selects the subset of servers that the load balancing algortihms will use for a particular request. By default, a request is load balanced over all the available non-spare servers in a cluster. Match rules allow you to select the group of servers that will be used to load balance the request.

For each virtual cluster, you can specify any number of match rules. For each match rule, you specify the subset of servers that can handle requests that meet the rule criteria.

A match rule provides for custom processing of requests within connections. Equalizer provides common and protocol-specific match functions that enable dynamic matching based on the request’s contents. Protocol-specific match functions typically test for the presence of particular attributes in the current request.

For example, a Layer 7 HTTP virtual cluster can specify matching on specific pathname attributes to direct requests to subsets of servers so that all requests for images are sent to the image servers. The difference between load balancing with and without match rules in such a situation is illustrated in the following figure.

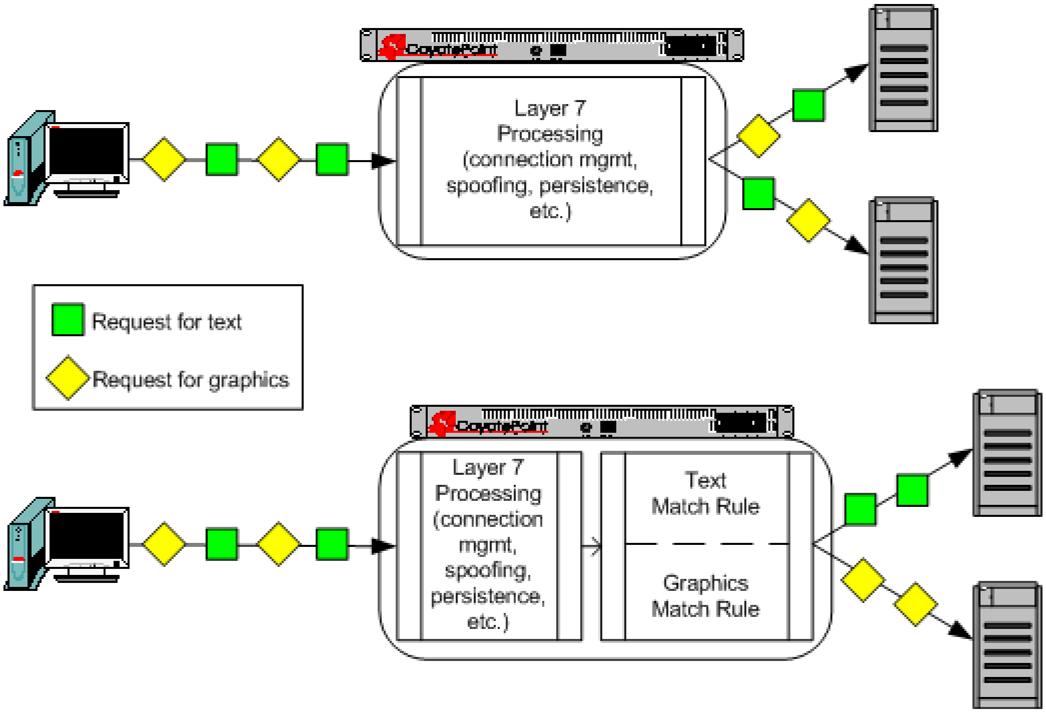


Figure 56 Conceptual Example of Match Rule Processing

Most client requests are a mix of requests for text and graphics. Layer 7 processing without Match Rules (top diagram in Figure 56) balances requests across all the available servers in the cluster, so that each server will see a mix of text and graphics requests. This means that all text and graphics must be available on each server.

Some sites may want to have one system serve only requests for graphics, and one system serve only text requests. By adding appropriate Match Rules (bottom diagram in Figure 56), Equalizer can examine each request to determine if the content requested is Text or Graphics, and send the request to the appropriate server. In this example, the servers need only hold the content they are serving, text or graphics.

Match Rule Processing

A match rule is like an if-then statement: an expression is evaluated and if it evaluates to true the body of the match rule applies to the request.

A match expression is a combination of match functions with logical operators, and can be arbitrarily complex. This allows for matching requests that have, for example:

```
(attribute A) AND NOT (attribute B)
```

If the match expression evaluates to *true*, then the data in the request has selected the match rule, and the match body applies.. The *match body* contains statements that affect the subsequent handling of the request.

Multiple match rules are checked in order. Once the data in the request selects a match rule, no further match rules are checked against the request. Equalizer makes a load balancing decision using the match body contents of the matching rule.

If the match expression evaluates to *false*, then each subsequent match rule in the list of match rules for the virtual cluster is processed until a match occurs. All virtual clusters have a **Default Match** rule, which always evaluates to *true* and which will use the entire set of servers for load balancing. The Default Match rule is always processed last.

Each virtual cluster can have any number of match rules, and each match rule can have arbitrarily complex match expressions. Keep in mind that Equalizer interprets match rules for every Layer 7 cluster connection, so it is a good idea to keep match rules as simple as possible.

> **Note –** Multiple requests may be received on the same TCP/IP connection. The default behavior of Equalizer is to load balance based only on the contents of the *first* request on a connection. This is indicated by the **once only** cluster flag, which is enabled by default (if this flag is not visible in the cluster flags, turn on the **advanced** flag).
>
> When using Match Rules, it is usually desriable to turn *off* the **once only** flag so that Equalizer matches against each individual request on the stream, not just the initial one.

General Match Expressions and Match Bodies

A match rule consists of a *match expression* and a *match body,* which identifies the operations to perform if the expression is satisfied by the request. Match syntax is as follows:

match *name* { *expression* } then { *body* }

Each match has a name, which is simply a label. The name must follow the same restrictions as those for cluster names and server names. All match names within a cluster must be unique.

Match Expressions

Match expressions affect the subsequent processing of the request stream using URI, host, or other information. Match expressions are made up of match functions, most of which are protocol-specific, joined by logical operators, optionally preceded by the negation operator, with sets of beginning and end parentheses for grouping where required. This may sound complex, and it can be, but typical match expressions are simple; it is usually best from a performance perspective to keep them simple.

The most simple match expression is one made up solely of a single match function. The truth value (*true* or *false*) of this expression is then returned by the match function. For example, a match function common to all Layer 7 protocols is the `any()` function, which always returns *true*, independent of the contents of the request data. So, the most simple match expression is:

any()

which will always result in the match rule being selected.

Use the logical NOT operator, (sometimes), to invert the sense of the truth value of the expression. So, you can use the NOT operator to logically invert a match expression, as follows:

NOT *expression*

giving rise to the next simplest example:

NOT any()

which always evaluates to *false* and always results in the match rule not being selected.

With the addition of the logical OR (`||`) and logical AND (&&) operators, you can specify complex expressions, selecting precise attributes from the request, as in this:

NOT happy() || (round() && happy())

Match expressions are read from left to right. Expressions contained within parentheses get evaluated before other parts of the expression. The previous expression would match anything that was not red or that was round and happy.

> **Note –** The the logical negation operator is displayed as “NOT”, rather than “!”.

Unlike the previous example, match functions correspond to certain attributes in a request header.

For example, a request URI for a web page might look like this:

```
Get /somedir/somepage.html   http/1.1
Accept: text/html, text/*, *.*
Accept-Encoding: gzip
Host: www.coyotepoint.com
User-Agent: Mozilla/4.7 [en] (Win98; U)
```

Various functions return true when their arguments match certain components of the request URI. Using the above request URI, for example, you could use several match functions:

- **pathname**() returns true if its argument matches `/somedir/somepage.html`
- **dirname**() returns true if its argument matches `/somedir/`
- **filename**() returns true if its argument matches `somepage.html`

Other functions can evaluate the contents of the `Host` header in the request URI above:

```
host (www.coyotepoint.com)
host_prefix (www)
host_suffix (coyotepoint.com).
```

Some function arguments can take the form of a regular expression[1]. Note that you cannot put regular expressions into match expressions except as an argument to a function whose definition supports regular expressions.

> **Note –** Matching regular *ex*pressions (using ***_regex**() functions) is many times more processing-intensive than using other match functions. It is usually possible to avoid using regular expressions by carefully crafting match expressions using other functions. For example, the following regular expression match:
>
> ```
> dirname_regex("(two|four|six|eight)")
> ```
>
> Can be replaced by the more efficient:
>
> ```
> dirname_substr("two") ||
> dirname_substr("four") ||
> dirname_substr("six") ||
> dirname_substr("eight")
> ```

[1] Regular expressions are specified according to IEEE Std 1003.2 (“POSIX.2”).

Match Bodies

Match bodies specify the actions to take if the match expression selects the request. This is specified in the form of statements that provide values to variables used by the load balancer to process the request. The most common (and most useful) match body selects the set of servers over which to apply the load balancing:

```
servers = all;
```

The `servers` assignment statement takes a comma-separated list of server names, which specifies the set of servers to be used for load balancing all requests that match the expression in the match rule. The reserved server names `all` and `none` specify respectively the set of *all* servers in the virtual cluster and *none* of the servers in the virtual cluster. If you do not assign servers, none will be available for load balancing; as a result, the connection to the client will be dropped.

In general, you can override most cluster-specific variables in a match body. (You can override protocol-specific variables as well, but that does not always make sense.) One useful example of overriding variables is as follows:

```
servers = s0, s1, s2;
flags = ! once_only;
```

which would load-balance across the specified servers (which first must be defined in the virtual cluster) and also turn off the `once_only` flag for the duration of processing of that connection.

Match Rule Definitions

Match rules are defined in the file */var/eq/eq.conf* with the definition of the cluster to which the match rule applies. A match rule as it appears in *eq.conf* looks like the following example:

```
match ma01 {
    client_ip("10.0.0.19")
 } then {
    flags              = !spoof, once_only;
    servers            = sv_102, sv_65;
 }
```

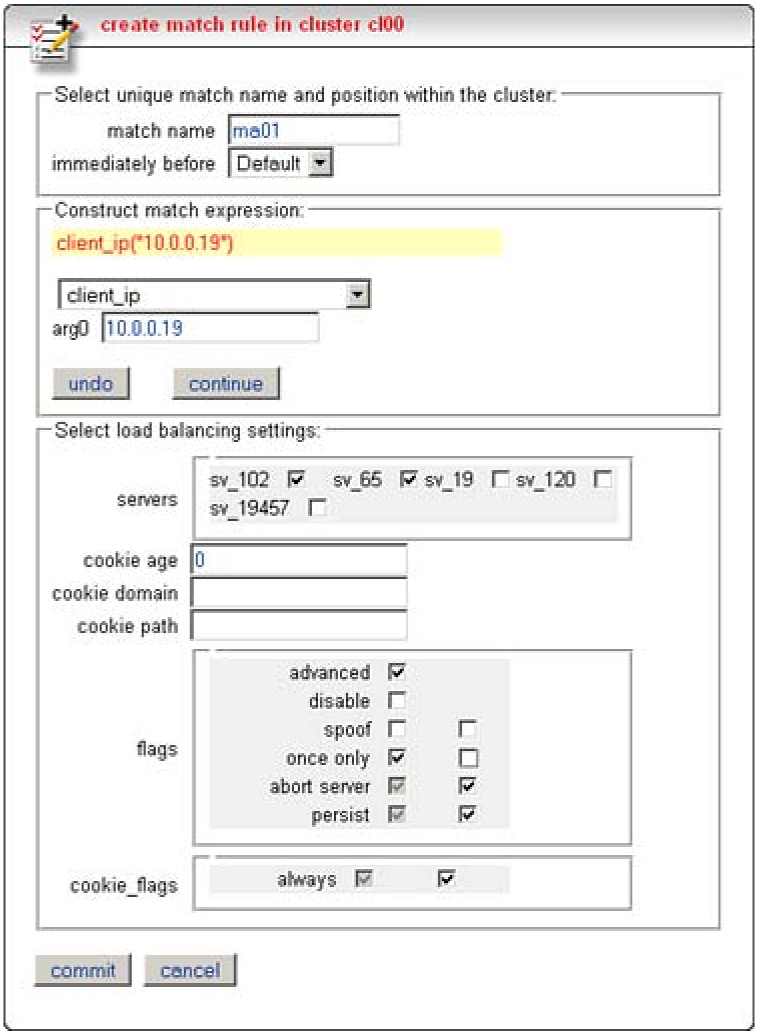
In this example (the match rule is named “ma01”), the match function, `client_ip`, has an argument that matches all requests from IP address `10.0.0.19`. Servers `sv_102` and `sv_65` are the only ones used for load balancing of matching requests. Additionally, this rule sets the `once_only` flag (that is, we perform processing only on the initial request on this connection) and clears the `spoof` flag (that is, when the connection is made to the server, the server sees a connection to the Equalizer, not to the client).

The Administration Interface allows you to create and modify match rules, without requiring a detailed knowledge of the configuration language syntax used in the *eq.conf* file. The interface validates match rules before saving them so that all saved rules are syntactically correct. For this reason, we recommend you use the interface to create and edit match rules, rather than editing the configuration file.

The interface does *not*, however, test the behavior of match rules. Match rules must be tested against a flow of incoming requests in order to determine if the behavior of the rule is what you expect.

Figure 57 on page 128 shows the match rule defined above as it would be displayed in the Administrative Interface.

The **Construct match expression** section of the screen shows the expression that is evaluated against the incoming request. If the expression evaluates to *true*, the **Select load balancing settings** section specifies the servers that will be used to satisfy the incoming request, as well as the flags that will be set for the request. The next section of this document explains these settings in detail.



create match rule in cluster cl00

Select unique match name and position within the cluster:

match name: ma01

immediately before: Default

Construct match expression:

client_ip("10.0.0.19")

client_ip

arg0: 10.0.0.19

Select load balancing settings:

servers: sv_102 ☑ sv_65 ☑ sv_19 □ sv_120 □ sv_19457 □

cookie age: 0

cookie domain:

cookie path:

flags:

advanced	☑	
disable	□	
spoof	□	□
once only	☑	□
abort server	☑	☑
persist	☑	☑

cookie_flags: always ☑ ☑

Figure 57 Example match rule

Managing Match Rules

Before constructing a match rule, you should first understand the general concepts of match rules covered in “General Match Expressions and Match Bodies” on page 125.

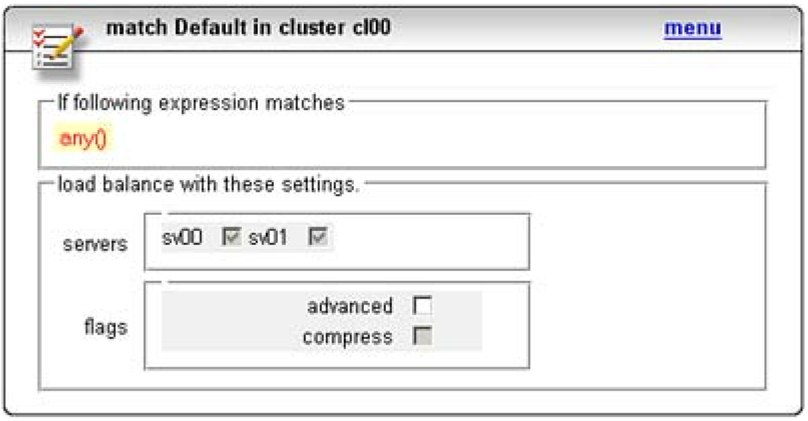
The Default Match Rule

All Layer 7 clusters created via the Equalizer Administration Interface start with a single match rule (named Default) that matches all requests and selects all servers.

```
match Default {
any()
} then {
servers = all;
}
```

The default rule specifies that all servers defined in the cluster should be used for load balancing the request, and that all flag settings for the request will be inherited from the cluster flag settings. This rule is always the last match rule in the ordered list of match rules for a cluster. You cannot modify, delete, or move this match rule.

The Default rule can be viewed by clicking in the left frame on **match Default** for any Layer 7 cluster. (If you have not created a Layer 7 cluster, see "Working with Virtual Clusters" on page 66). Figure 58 shows the default match rule for a cluster with two servers.



match Default in cluster cl00

If following expression matches

any()

load balance with these settings.

servers: sv00 ☑ sv01 ☑

flags: advanced □ compress □

Figure 58 A Default match rule shown in the Match Rule dialog box

Note that although the Default match rule cannot be modified or deleted, it can be overridden by placing a rule prior to it that selects `any()` request, and selects other servers and flags than the Default rule.

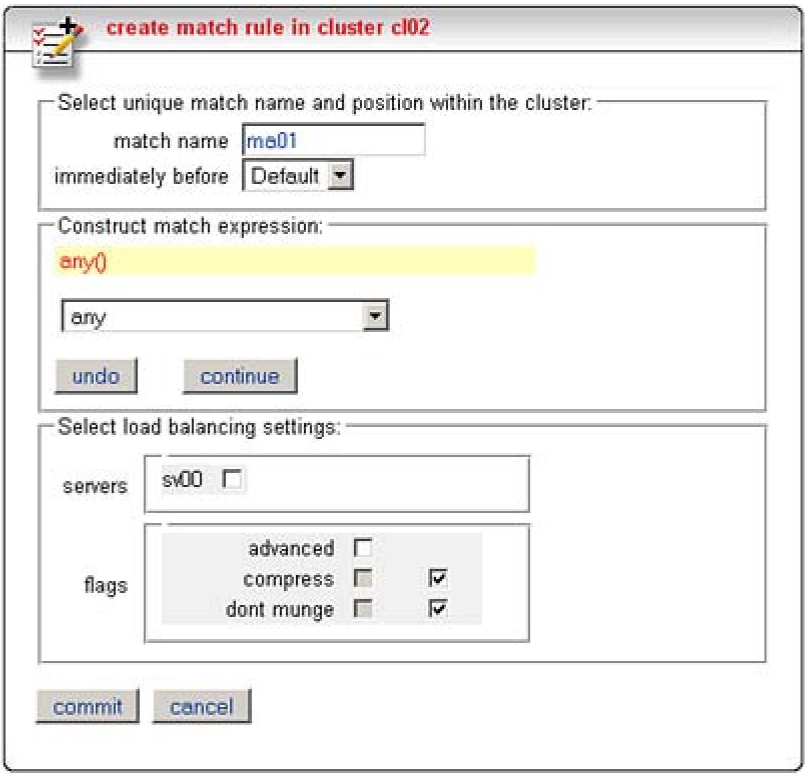
The following section shows you how to create a new Match Rule.

Creating a New Match Rule

To add a match rule to a virtual cluster, follow this general procedure:

1. Log into the Equalizer Administration Interface in edit mode.
2. In the left frame, click the name of the Layer 7 cluster to which you want to add the rule. The **cluster** screen appears in the right frame.

3. In the **cluster** screen, select **menu > Add Match Rule**. The **create match rule** screen appears in the right frame.



create match rule in cluster cl02

Select unique match name and position within the cluster:

match name: ma01

immediately before: Default

Construct match expression:

any()

any

Select load balancing settings:

servers: sv00 □

flags:

advanced □

compress □ ☑

dont munge □ ☑

Figure 59 Example Match Rule dialog box for a cluster with five servers

4. Enter a name for the new rule in the **match name** field.

   All match names within a cluster must be unique.

5. Select the placement of the rule by choosing a rule from the **immediately before** list box.

   The ordering of match rules is important, as they are processed from first to last until one of them evaluates to *true*, at which time the match body is processed. The initial match expression of a new rule, NOT any() is one that will always evaluate to *false* meaning that this match rule will never be selected. It is good practice to be cautious when adding new match rules to ensure that all the traffic to a cluster does not get mishandled. A new match rule will not be committed until you click **commit.** You can cancel the entire process by clicking **cancel**.

6. To place or modify a match function, click the appropriate part of the expression.

   The part of the expression that editor will directly affect is red and the affiliated parts to the selection are green. Pay attention to the colors of various parts of the match expression, these colors show what will be affected.

7. From the drop-down list below the match expression, select the match function with which you want to build or edit the rule. (To learn more about match functions, refer to “Match Functions” on page 132.)

   The drop-down list of edit actions are different depending on what you select in the expression and whether the cluster is HTTP or HTTPS. All lists have some common match functions and structural editing operators. In any list of edit actions, *selection* refers to the green and red parts of the match expression and *self* refers to the red portion.

Some of the structural editing operators include the function you are replacing (for example, replace with host AND any). When modifying the structure of an `any` function, it may be helpful to temporarily change the function to something more distinct (so that you will not have to interpret the expression, “replace with any AND any”).

8. Click the **continue** button, Equalizer shows the new version of the match expression.

   Depending on the new function, you may have to fill in information in the **arg0** and **arg1** text boxes. These fields supply arguments, as required, to the selected match function.

   If there are any syntax errors, an error screen appears when you click the **continue** button. This most likely occurs if there are missing arguments or syntax errors in the argument strings.

   If you click a different part of the match expression without clicking the **continue** button first, you will lose any changes since you last clicked **continue**.

9. You construct complicated Boolean expressions using the structural editing operators.

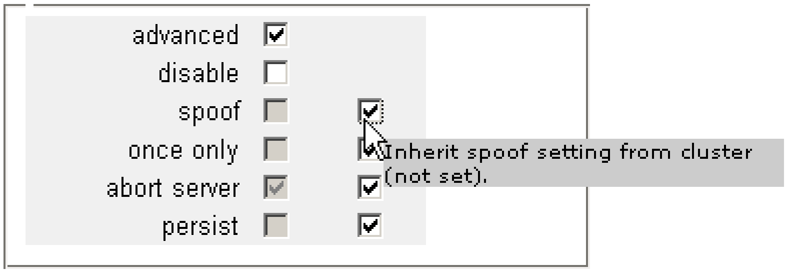
10. To undo the latest changes, click the **undo** button.

11. To add to or change the match expression, repeat steps 6 through 10. Equalizer continues to show your additions and modifications.

12. In the **servers** section, enable the check boxes to the right of the servers that you want to use to load balance requests that match the criteria set in the previous steps.

> **Caution – If you do not enable a check box for at least one server, *Equalizer will drop the connection for any request that matches the rule*.**

13. Check **advanced** if you want to override the inheritance of the **spoof**, **once only**, **abort_server**, or **persist** flags. The two columns of check boxes to the right of each flag allow you to specify that the flag setting for a request that is selected by the match rule is either the same as the cluster setting, or overridden for this match.



advanced ☑
disable □
spoof □ ☑
once only □ ☑
abort server ☑ ☑
persist □ ☑

Inherit spoof setting from cluster (not set).

Figure 60 Flag settings field for match rules

As the tooltip shown in the figure above indicates, the right-hand check box for each flag, if set, indicates that the flag setting will be inherited from the cluster setting -- in the example above, the **spoof** setting will be inherited from the cluster, on which spoof is not set. To override this setting for this rule, clear the right-hand “inherit” check box, and then enable the left-hand check box for **spoof**.

14. Check **disable** to indicate that this rule should not be processed. (This check box is often used to debug match rules, so that a match rule can be temporarily disabled during testing without deleting its definition.)
15. Click the **commit** button to save your new Match Rule definition.

Modifying a Match Rule

To edit a match rule, follow these steps:

1. Log into the Equalizer Administration Interface in edit mode.
2. In the left frame, click the name of the match rule to be changed.
3. In the right frame, click **menu > Edit Match Rule** from the local menu. The **modify match rule** screen opens in the right frame.
4. Make the desired changes to the match expression, using steps similar to those in the previous section, “Creating a New Match Rule” on page 129.
5. Make the desired changed to the list of servers and flags.
6. To save your changes, click the **commit** button.

Removing a Match Rule

To delete a match rule, follow these steps:

1. Log into the Equalizer Administration Interface in edit mode.
2. In the left frame, click the name of the match rule to be deleted.
3. In the right frame, click **Menu** and select **Delete This Match** from the local menu.
4. Click **OK** to confirm that you want to permanently remove the match rule.

Match Functions

To build or edit a match expression, click part of the expression to edit its arguments or to select a match function or logical expression from a dynamic drop-down list. The part of the expression that you click on is highlighted and determines the contents of the dorp-down list. For instance, if the current selection is a match function, the arguments to the function are displayed so you can edit them, along with a list of items that can replace the function.

In the Administration Interface, logical operators and constructs are introduced using special entries in the drop-down list for expressions. These allow you to build complex boolean expressions in match rules. See the section “Logical Operators and Constructs in the GUI” on page 138.

The combination of match functions and logical operators provides a great deal of control over request processing based on the contents of the request’s HTTP headers and the destination URI of the request.

The following table lists the non-URI functions supported by Equalizer match rules:

Table 61: non-URI Match Functions

non-URI Match Function	Description
any()	This function always evaluates to *true*.
client_ip(*string*)	This function evaluates to *true* only if the IP address of the client machine making the connection matches the *string* argument. The *string* can be a simple IP address (e.g., “192.168.1.110”), or an IP address in Classless Inter-Domain Routing (CIDR) notation (e.g., “192.168.1.0/24”). This function can be useful in restricting match expressions to a particular client or group of clients, which can aid in debugging a new match rule when a cluster is in production. Only the specified clients match the rule, leaving other clients to be handled by other match rules.
debug_message(*string*)	This function always evaluates to *true*. It writes the *string* argument to the Event Log for the cluster (**View > Event Log**). This function can be logically ANDed and ORed with other functions to write debug messages. *Use this function for testing and debugging only. Do not use it in production environments, since it has a negative impact on performance.*
ignore_case()	This function always evaluates to *true*, and is intended to be used to apply the **ignore_case** flag for comparisons when it is *not set* on the cluster. When this function is ANDed with other functions, it has the effect of forcing case to be ignored for any comparisons done by the match rule.
observe_case()	This function always evaluates to *true*, and is intended to be used to override the **ignore_case** flag for comparisons when it is *set* on a cluster. When this function is ANDed with other functions, it has the effect of forcing case to be honored for any comparisons done by the match rule.
http_09()	This function takes no arguments and evaluates to *true* if the HTTP protocol used by the request appears to be HTTP 0.9. This is done by inference: if an explicit protocol level is absent after the request URI, then the request is considered HTTP 0.9.
method(*string*)	This function evaluates to *true* if the *string* argument exactly matches the Request Method (e.g., GET, POST, etc.) specified in the request. Note that by default Equalizer forwards packets to servers without determining whether or not the method specified in the request is valid (i.e., is a method specified in Section 9 of RFC2616). One use of the **method**() function is to be able to override this default behavior and prevent invalid requests from being forwarded to a server.
[header match functions]	[No exact match **header()** function is supplied. See “Match Function Notes” on page 136, for the supported values for *header*.]
header_prefix(*header*, *string*)	This function evaluates to *true* if the selected *header* is present and if the string-valued argument *string* is a prefix of the associated header text.
header_suffix(*header*, *string*)	This function evaluates to *true* if the selected *header* is present and if the argument *string* is a suffix of the associated header text.

Table 61: non-URI Match Functions

non-URI Match Function	Description
header_substr(*header*, *string*)	This function evaluates to *true* if the selected *header* is present and if the string-valued argument *string* is a sub-string of the associated header text.
header_regex(*header*, *string*)	This function evaluates to *true* if the selected *header* is present and if the string-valued argument *string*, interpreted as a regular expression, matches the associated header text.
ssl2()	HTTPS only. This function evaluates to *true* if the client negotiated the encrypted connection using SSL version 2.0.
ssl3()	HTTPS only. This function evaluates to *true* if the client negotiated the encrypted connection using SSL version 3.0.
tls1()	HTTPS only. This function evaluates to *true* if the client negotiated the encrypted connection using TLS version 1.0.

In addition to the functions in the preceding table, a set of functions is provided that allows you to process requests based on the various components of a request’s destination URI.

A URI has the following parts (as defined in RFC1808):

```
<scheme>://<hostname>/<path>;<params>?<query>#<fragment>
```

In addition, Equalizer further breaks up the `<path>` component of the URI into the following components:

```
<directory><filename>
```

The following figure illustrates how Equalizer breaks up a URI into the supported components:

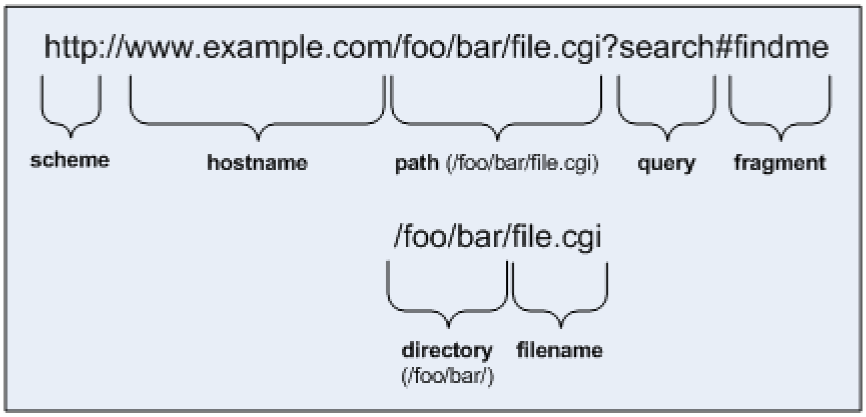


Figure 62 URI components

Note that the following components of the URI do not have corresponding match functions:

- Match functions for the `<scheme>` component are not necessary, since a cluster must be configured to accept only one protocol: HTTP *or* HTTPS.
- Match functions for the optional `<params>` component are not provided. Use the **pathname*()** and **filename*()** functions to match characters at the end of the **path** and **filename** components.

- Match functions for the optional `<fragment>` component are not provided. The fragment portion of a URI is not transmitted by the browser to the server, but is instead retained by the client and applied after the reply from the server is received.

The following table lists the URI matching functions that match text in the URI components shown in Figure 62.

Table 63: URI-based Match Functions

URI Match Function	Description
host(*string*)	This function evaluates to *true* if the *string* argument exactly matches the hostname portion of the request. *In the case of HTTP 0.9, the host is a portion of the request URI. All other HTTP protocol versions require a Host header to specify the host, which would be compared to the string.* [Also see Item 5 on page 138.]
host_prefix(*string*)	This function evaluates to *true* if the *string* argument is a prefix of the hostname portion of the URI path. The prefix of the hostname includes all text up to the first period ("www" in "www.example.com"). [Also see Item 5 on page 138.]
host_suffix(*string*)	This function evaluates to *true* if the *string* argument is a suffix of the hostname portion of the URI path. The suffix of the hostname includes all text after the first period in the hostname ("example.com" in "www.example.com"). [Also see Item 5 on page 138.]
pathname(*string*)	This function evaluates to *true* if the *string* argument exactly matches the path component of the request URI.
pathname_prefix(*string*)	This function evaluates to *true* if the *string* argument is a prefix of the path component of the request URI.
pathname_suffix(*string*)	This function evaluates to *true* if the *string* argument is a suffix of the path component of the request URI.
pathname_substr(*string*)	This function evaluates to *true* if the *string* argument is a substring of the path component of the request URI.
pathname_regex(*string*)	This function evaluates to *true* if the *string* argument, interpreted as a regular expression, matches the path component of the request URI.
dirname(*string*)	This function evaluates to *true* if the *string* argument exactly matches the directory portion of the path component of the request URI. The path component is the entire directory path, including the trailing slash (for example, "/foo/bar/" is the directory portion of "/foo/bar/file.html").
dirname_prefix(*string*)	This function evaluates to *true* if the *string* argument is a prefix of the directory portion of the path component of the request URI. The leading slash must be included in the *string* (for example, "/fo" is a prefix of "/foo/bar/file.html").
dirname_suffix(*string*)	This function evaluates to *true* if the *string* argument is a suffix of the directory portion of the path component of the request URI. The trailing slash must be included in the *string* (for example, "ar/" is a suffix of the directory portion of "/foo/bar/file.html").
dirname_substr(*string*)	This function evaluates to *true* if the *string* argument is a substring of the directory portion of the path component of the request URI.
dirname_regex(*string*)	This function evaluates to *true* if the *string* argument, interpreted as a regular expression, matches the directory portion of the path component of the request URI.

Table 63: URI-based Match Functions

URI Match Function	Description
filename(*string*)	This function evaluates to *true* if the *string* argument exactly matches the filename portion of the URI path. *This portion includes only the text after the last trailing path component separator (/), as that is considered part of the directory* (for example, “file.html” is the filename portion of “/foo/bar/file.html”).
filename_prefix(*string*)	This function evaluates to *true* if the *string* argument is a prefix of the filename portion of the URI path.
filename_suffix(*string*)	This function evaluates to *true* if the *string* argument is a suffix of the filename portion of the URI path.
filename_substr(*string*)	This function evaluates to *true* if the *string* argument is a substring of the filename portion of the URI path.
filename_regex(*string*)	This function evaluates to *true* if the *string* argument, interpreted as a regular expression, matches the filename portion of the URI path.
query(*string*)	This function evaluates to *true* if the *string* argument exactly matches the (optional) query component of the request URI. The query, if present, appears in a URI following a question mark (?). The syntax of a query is application specific, but generally is a sequence of key/value pairs separated by an ampersand (&).
query_prefix(*string*)	This function evaluates to *true* if the *string* argument is a prefix of the query portion of the URI path.
query_suffix(*string*)	This function evaluates to *true* if the *string* argument is a suffix of the query portion of the URI path.
query_substr(*string*)	This function evaluates to *true* if the *string* argument is a substring of the query portion of the URI path.
query_regex(*string*)	This function evaluates to *true* if the *string* argument, interpreted as a regular expression, matches the query portion of the URI path.

Match Function Notes

1. **Considering Case in String Comparisons:** String comparisons performed by match functions honor the setting of the **ignore case** cluster parameter: if it is set on the cluster (the default), then all match rule functions used for that cluster are case insensitive; that is, the case of strings is ignored. For example, the string “ab” will match occurrences of “ab”, “Ab”, “aB”, and “AB”. If **ignore case** is *not* set on the cluster, then all string comparisons are by default case sensitive (the string “ab” will match only “ab”). To override the **ignore case** flag setting on the cluster for a match function or block of functions, you must logically AND the **observe_case()** or **ignore_case()** functions with the match function or block. For example, if **ignore case** is set on the cluster, you would use the following construct to force the **header_substr()** function to make case sensitive string comparisons:

```
(observe_case() AND header_substr("host", "MySystem"))
```

2. **Regular Expressions:** Some match functions have *prefix*, *suffix*, *substr*, or *regex* variants. The *regex* variants interpret an argument as a regular expression to match against requests. Regular expressions can be very costly to compute, so use the *prefix*, *suffix*, or *substr* variants of functions (or Boolean combinations of prefix and suffix testing), rather than the *regex* function variants, for best performance. For example, the following regular expression match:

   ```
   dirname_regex("(two|four|six|eight)")
   ```

   Can be replaced by the more efficient:

   ```
   dirname_substr("two") ||
   dirname_substr("four") ||
   dirname_substr("six") ||
   dirname_substr("eight")
   ```

   Equalizer supports POSIX regular expression syntax only. See Appendix C, “Regular Expression Format” for a description.

3. **Supported Headers:** All of the **header_*(*header, string*)** match functions take a *header* argument, which selects the header of interest. If this header is not present in the request, the match function evaluates to *false*. Otherwise, the text associated with the header is examined depending on the particular function.

   Although HTTP permits a header to span multiple request lines, none of the functions matches text on more than one line. In addition, Equalizer will only parse the first instanmce of a header. If, for example, a request has multiple **cookie** headers, Equalizer will only match against the first **cookie** header in the request.

   The list of supported headers for the *header* argument are as follows:

Table 64: Supported HTTP Headers for Matching

accept	**expect**	**proxy-authorization**
accept-charset	**from**	**range**
accept-encoding	**host**	**referer**
accept-language	**if-match**	**te**
authorization	**if-modified-since**	**trailer**
cache-control	**if-none-match**	**transfer-encoding**
connection	**if-range**	**upgrade**
content-length	**if-unmodified-since**	**user-agent**
cookie	**max-forwards**	**via**
date	**pragma**	**warning**

4. **HTTPS Protocol Matching:** Equalizer permits the construction of virtual clusters running the HTTPS protocol. HTTPS is HTTP running over an encrypted transport, typically SSL version 2.0 or 3.0 or TLS version 1.0. All of the functions available for load balancing HTTP clusters

are available for HTTPS clusters. In addition, there are some additional match functions [`ssl2`(), `ssl3`(), and `tls1`()], that match against the protocol specified in an HTTPS request.

Note – Given that HTTPS runs encrypted using SSL and TLS as the transport, in order to perform any Layer 7 processing, the Equalizer must terminate the SSL/TLS encrypted connection. This can have deleterious effects on performance, as the encryption and decryption process is resource-intensive. A hardware accelerator, Xcel, is available which can be added to the Equalizer platform to ameliorate this problem.

5. **Supported Characters in URIs:** The characters permitted in a URI are defined in RFC2396. Equalizer supports all characters defined in the standard for all Match Functions that have a URI as an argument, with one exception: currently the "-" (dash) character in URIs is not matched by the **host*()** functions. The workaround is to use the **header*()** functions instead to match on the **Host** header. Also note that the ASCII space character is not permitted in URIs -- it is required to be encoded by all conforming browsers as "%20" (see Section 2.4 of RFC2396).

Logical Operators and Constructs in the GUI

In addition to the Match Functions listed in the previous section, the Equalizer Administrative Interface provides the following logical operators and constructs that allow you to combine the match functions into logical expressions, and manipulate the functions in the match expression. All of these operators and constructs affect the part of the match expression that is currently selected (highlighted in red) in the graphical interface.

Table 65: Match Rule Logical Operators and Constructs

negate function	This function negates (or reverses) the value of the expression that comes immediately after it in the match definition. When using the GUI to construct a match rule, choosing this function negates the currently selected function in the match rule expression and appears on screen as the string "`NOT`". In the *eq.conf* file, it negates the function immediately following it and appears as an exclamation point (!).
delete selection	Removes the currently selected portion of the match expression.
replace with AND	Replaces the currently selected logical operator with "AND".
replace with OR	Replaces the currently selected logical operator with "OR".
replace with any AND any	Replaces the currently selected logical construct with "any() AND any()".
replace with any OR any	Replaces the currently selected logical construct with "any() OR any()".
replace with self AND any	Replaces the currently selected logical construct with the current selection logically AND'ed with the "any()" function.
replace with self OR any	Replaces the currently selected logical construct with the current selection logically OR'ed with the "any()" function.
replace with any AND self	Replaces the currently selected function or logical construct with the "any()" function logically AND'ed with the current selection.
replace with any OR self	Replaces the currently selected function or logical construct with the "any()" function logically OR'ed with the current selection.

Table 65: Match Rule Logical Operators and Constructs

replace with any AND *function*	Replaces the currently selected function or logical construct with the “any()” function logically AND’ed with the current selection.
replace with any OR *function*	Replaces the currently selected *function* with the “any()” function logically OR’ed with the current selection.
replace with *function* AND any	Replaces the currently selected *function* with the current selection logically AND’ed with the “any()” function.
replace with *function* OR any	Replaces the currently selected *function* with the current selection logically OR’ed with the “any()” function.
swap left and right	When a logical operator is selected (i.e., AND or OR), switches the order of the left and right sides of the logical expression (e.g., “A AND B” becomes “B AND A”).
replace with left	When a logical operator is selected (i.e., AND or OR), replaces the entire logical expression with the left side of the logical expression (e.g., “A AND B” becomes “A”).
replace with right	When a logical operator is selected (i.e., AND or OR), replaces the entire logical expression with the left side of the logical expression (e.g., “A AND B” becomes “B”).

Example Match Rules

This section shows you how to create a few of the most commonly used types of match rules:

- “Parsing the URI” on page 139
- “Disabling Persistent Connections for One or More Servers” on page 140
- “Dedicated Image and Content Servers” on page 142

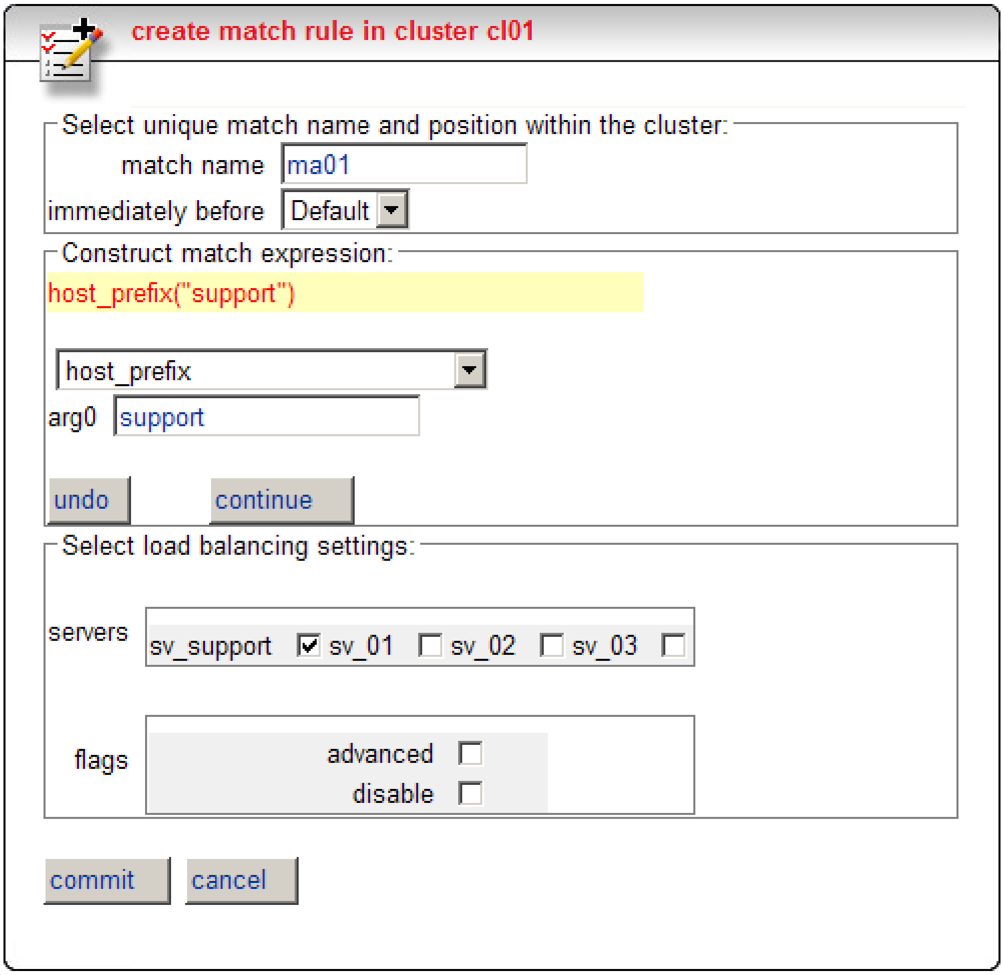
Parsing the URI

In this example, we want to direct requests to a particular server based on the hostname used in the URI contained in the request. We want all requests for URIs that start with “support” to go to one server, and all other requests that do *not* match this rule to be load balanced across all servers in the cluster.

To do this, we will construct one match rule that parses the URI; if the URI contains the string “**support**”, it forwards the request to the server **sv_support**. For this example, we assume that a cluster with four servers (**sv_support**, **sv_01**, **sv_02**, **sv_03**) has already been defined.

1. Log into the Equalizer Administration Interface in Edit mode.
2. In the left frame, click the name of the Layer 7 cluster to which you want to add the rule. The **cluster** screen appears in the right frame.
3. In the **cluster** screen, select **menu** > **Add Match Rule**. The **create match rule** screen appears in the right frame.
4. Type **support** into the **match name** text box.
5. In the **Construct match expression** field, select **replace with host_prefix** from the drop-down list box, and then select **continue**.

6. Replace the text in the **arg0** text box with **support**. Select **continue**.
7. In the **server** field, select **sv_support**. The screen should now look like this:



create match rule in cluster cl01

Select unique match name and position within the cluster:

match name: ma01

immediately before: Default

Construct match expression:

host_prefix("support")

host_prefix

arg0: support

Select load balancing settings:

servers: sv_support ☑ sv_01 □ sv_02 □ sv_03 □

flags: advanced □ disable □

8. Select the **commit** button to create the **support** rule. The home screen for the Administrative Interface is displayed. In the left-frame cluster list, **Match support** should now appear above **Match Default** for the cluster.

Disabling Persistent Connections for One or More Servers

Persistent connections to servers are enabled by the **persist** cluster flag, which is enabled by default when you create a cluster. If a cluster has a mix of servers that require persistent connections as well as some that do not, overall performance would generally be improved by disabling persistent connections for those servers that do not require it.

This procedure shows you how to disable the **persist** flag for one or more of the servers in a cluster, using a match rule. The match rule needs to select all the incoming requests destined for servers that don’t require persistent connections.

The match expression that you use in the match rule depends on how the match rule can determine if an incoming request will be routed to a server that does not require persistent connections.

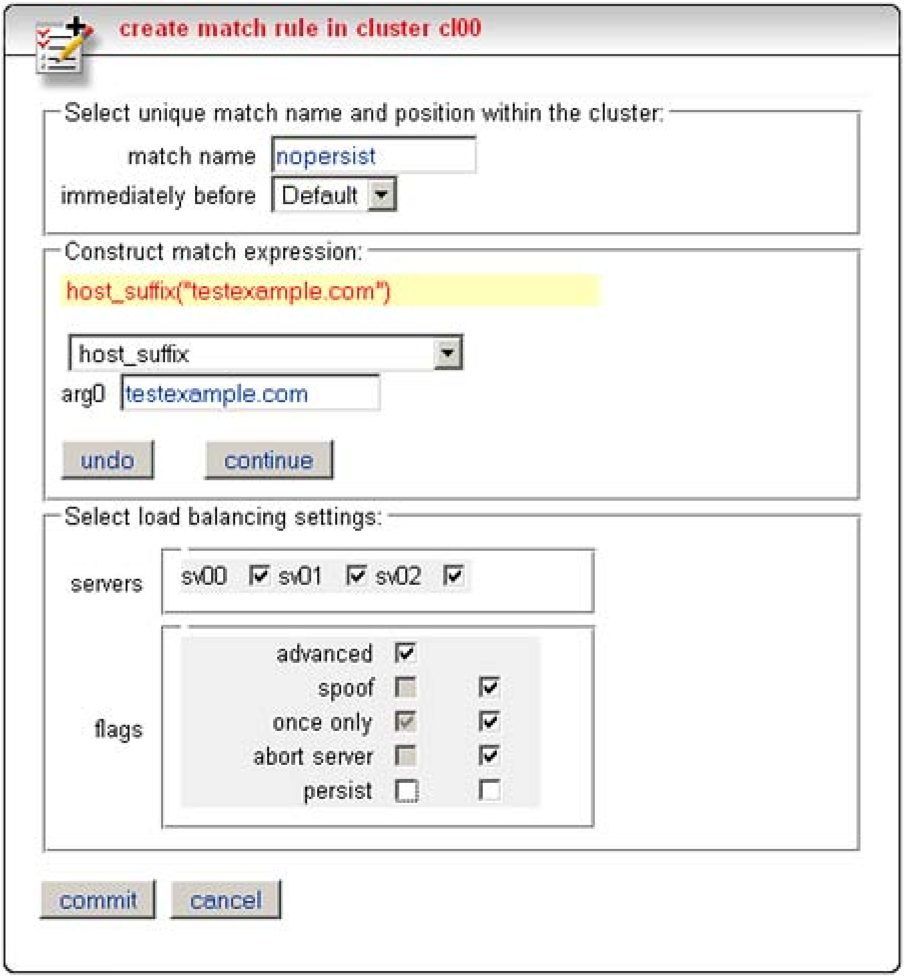
In this example, we assume that we can determine this by examining the hostname used in incoming requests. Any request containing a hostname in the following format will not require a persistent connection:

```
name.testexample.com
```

We'll assume that any request with a hostname having the format ***name*.testexample.com** will not require persistent connections. We'll use the `host_suffix()` match rule function to match the hostname. For this example, we assume that a cluster with three servers (**sv00**, **sv01**, **sv02**) has already been defined. We will construct a match rule that turn off **persist** for any request that contains the host suffix "**testexample.com**"; this request will be balanced across all three servers in the cluster.

1. Log into the Equalizer Administration Interface in Edit mode.
2. In the left frame, click the name of the Layer 7 cluster to which you want to add the rule. The **cluster** screen appears in the right frame.
3. In the **cluster** screen, select **menu > Add Match Rule**. The **create match rule** screen appears in the right frame.
4. Type **nopersist** into the **match name** text box.
5. In the **Construct match expression** field, select **replace with host_suffix** from the drop-down list box, and then select **continue**.
6. Replace the text in the **arg0** text box with **testexample.com**. Select **continue**.
7. In the **servers** field, enable the check boxes for the servers **sv00**, **sv01**, and **sv02**.

8. In the **flags** field, enable the **advanced** check box. In the list of flags that appears, disable the right-hand check box for the **persist** flag; then, disable the left-hand check box next to **persist**. The **create match rule** screen should now look like this:



create match rule in cluster cl00

Select unique match name and position within the cluster:

match name: nopersist

immediately before: Default

Construct match expression:

host_suffix("testexample.com")

host_suffix

arg0: testexample.com

Select load balancing settings:

servers: sv00 ☑ sv01 ☑ sv02 ☑

flags:

advanced	☑	
spoof	□	☑
once only	☑	☑
abort server	□	☑
persist	□	□

9. Select the **commit** button to create the **nopersist** rule. The home screen for the Administrative Interface is displayed. In the left-frame cluster list, **Match nopersist** should now appear above **Match Default** for the cluster.

Dedicated Image and Content Servers

In this example, we want to direct all requests for images to a particular set of server, and balance the remainder of requests across the other servers in the cluster. The image servers are all connected to a common storage device that contains the images. The remaining servers are all dedicated to serving particular content for different web sites. For this example, we assume that a cluster with five servers as shown below has already been defined

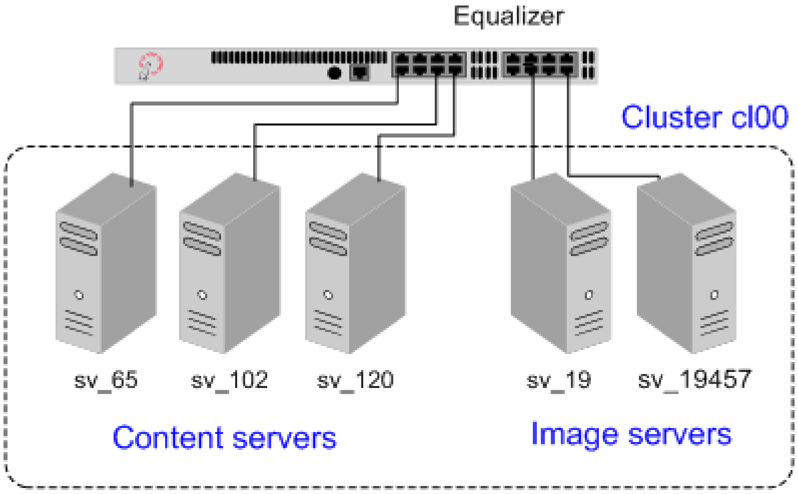


Figure 66 Match Rule Example: Dedicated Image and Content Servers

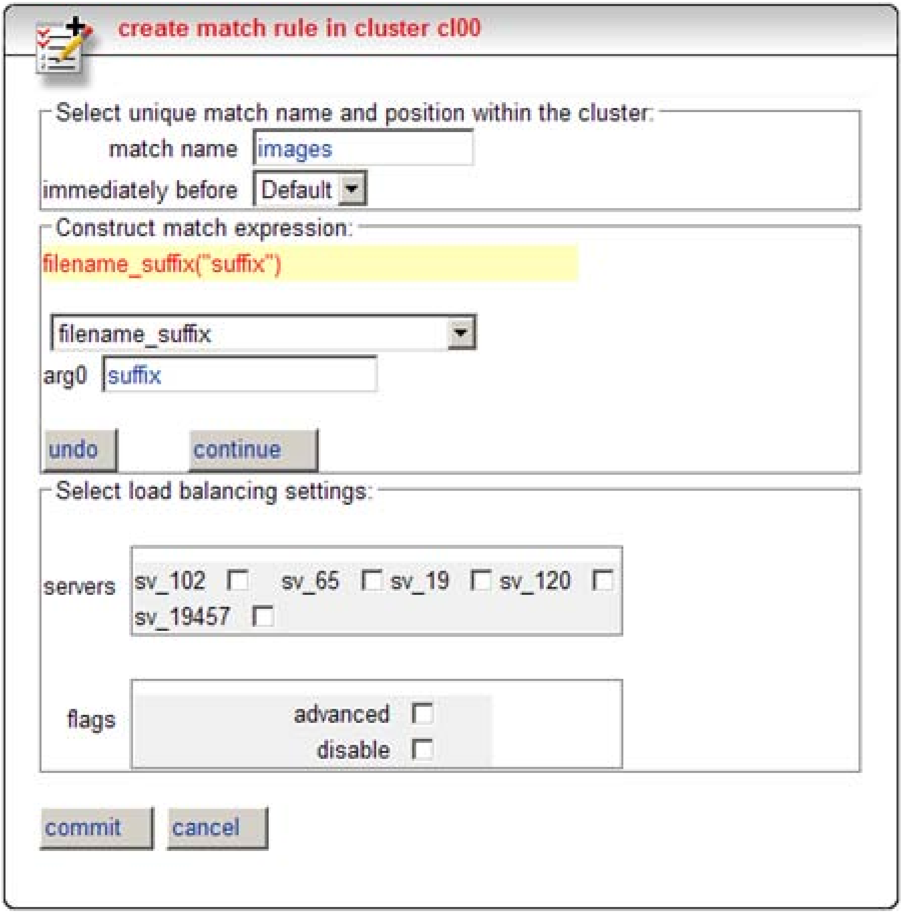
We want to maintain persistent connections for the web site servers, assuming that some of the websites may need to maintain sessions for applications such as shopping carts, email, etc. Persistent connections are not necessary for the image servers, since they access the images from common storage and have no need to maintain client sessions, so there is no need to incur the performance impact of maintaining session information.

To do this, we'll create two match rules, as follows:

1. Log into the Equalizer Administration Interface in Edit mode.
2. In the left frame, click the name of the Layer 7 cluster to which you want to add the rule. The **cluster** screen appears in the right frame.
3. In the **flags** field, make sure that the **once only** and **persist** flags are not checked. This is necessary because these flags, if enabled, cause only the first request in a connection to be evaluated. Since we want content to come from one set of servers and images from another, we want the servers that will have persistent connections to be chosen by the match rules.

   If one or both of these flags is enabled, select **menu** > **Change Cluster Parameters** and make sure both flags are not checked. Select **commit** to return to the **cluster** screen.
4. In the **cluster** screen, select **menu** > **Add Match Rule**. The **create match rule** screen appears in the right frame.
5. Type **images** into the **match name** text box. In this match rule, we'll construct an expression that will match all the filename extensions of the images to be served. These requests will go to the image servers.

6. In the **Construct match expression** field, select **replace with filename_suffix** from the drop-down list box, and then select **continue**. The screen should now appear as shown below:



create match rule in cluster cl00

Select unique match name and position within the cluster:

match name images

immediately before Default

Construct match expression:

filename_suffix("suffix")

filename_suffix

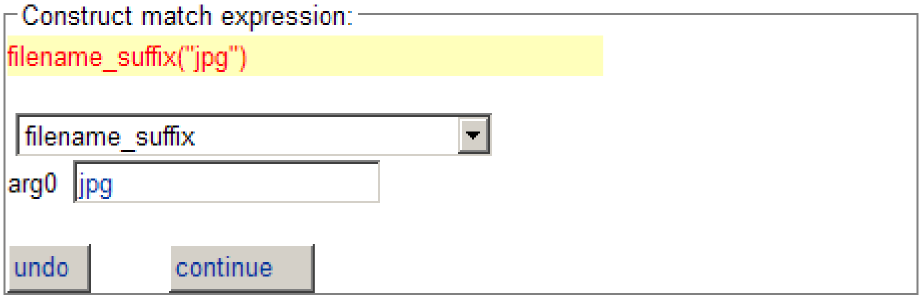
arg0 suffix

Select load balancing settings:

servers sv_102 □ sv_65 □ sv_19 □ sv_120 □ sv_19457 □

flags advanced □ disable □

7. Select the text in the **arg0** text box and type **jpg**. Select **continue**. The **Construct match expression** field should now appear as shown below:



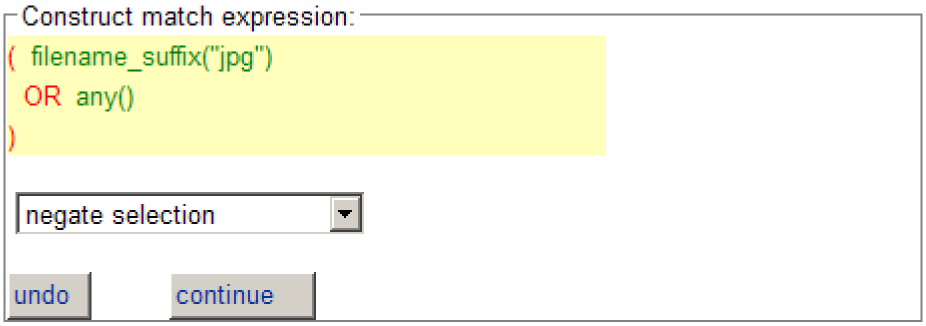
Construct match expression:

filename_suffix("jpg")

filename_suffix

arg0 jpg

8. In the drop-down box shown in the figure above, select **replace with filename_suffix OR any**, and then select **continue**. The **Construct match expression** field should now appear as shown below:

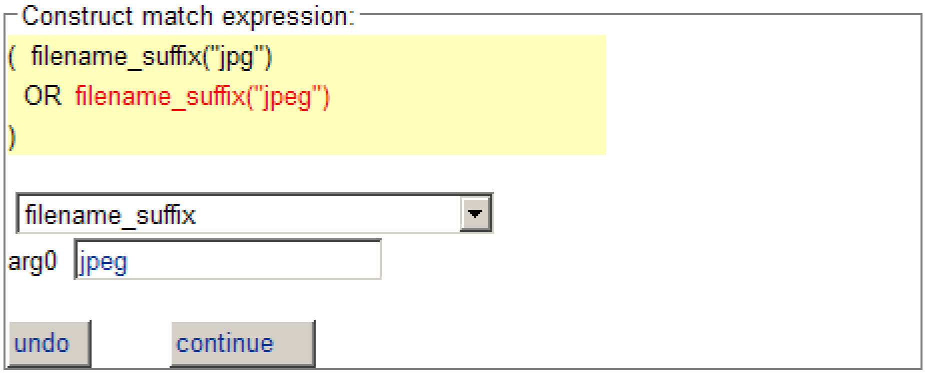


Construct match expression:

```
( filename_suffix("jpg")
  OR any()
)
```

9. Click on the **any()** function so that it is highlighted in red. Then select **replace with filename_suffix** from the drop-down box and click **continue**.

10. Type **jpeg** into the **arg0** text box and click **continue**. The **Construct match expression** field should now look like this:



Construct match expression:

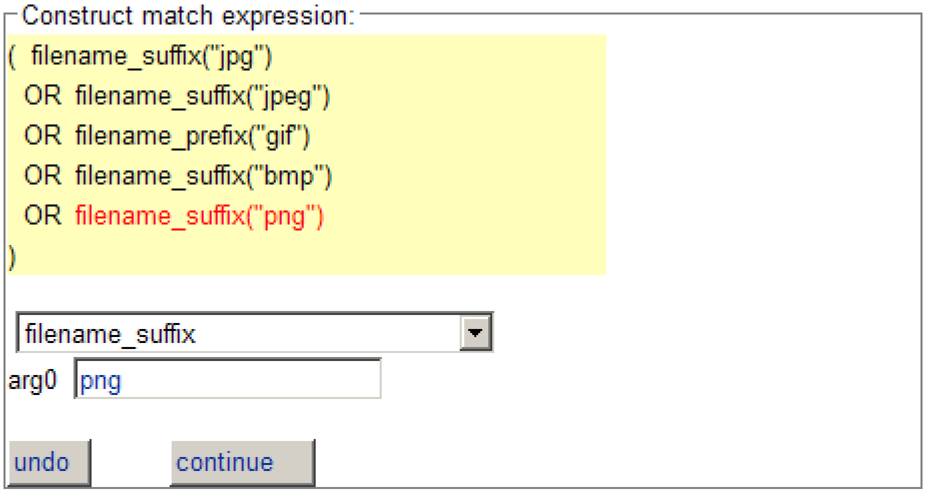
```
( filename_suffix("jpg")
  OR filename_suffix("jpeg")
)
```

filename_suffix

arg0 jpeg

11. Repeat Steps 8, 9, and 10 for each of the other filename suffixes on our example servers -- **gif, bmp**, and **png**

When you are done, the **Construct match expression** field should now look like this:



Construct match expression:

```
( filename_suffix("jpg")
  OR filename_suffix("jpeg")
  OR filename_prefix("gif")
  OR filename_suffix("bmp")
  OR filename_suffix("png")
)
```

filename_suffix

arg0 png

12. In our example, we want all the images to be served from either **sv_19** or **sv_19457**. Enable the check boxes for these two servers in the **servers** field. (Note that the check box for a server appears *after* the server name.) We don’t need to set any flags on this rule, so select the **commit** button to create the **images** rule. The home screen for the Administrative Interface is displayed. In the left-frame cluster list, **Match images** should now appear above **Match Default** for the cluster.

13. The **images** rule we created selects all the requests for image files; now we need a rule to determine which servers will receive all the other requests. The Default rule is not sufficient, and in fact we don’t want it to be reached, since it could send a request for content to one of the image servers. So, we’ll create another rule with the same match expression as the Default [**any()**], but a restricted list of servers. This effectively *replaces* the Default match rule with one of our own.

    In the left frame, click the name of the Layer 7 cluster to which you want to add the rule. The **cluster** screen appears in the right frame.

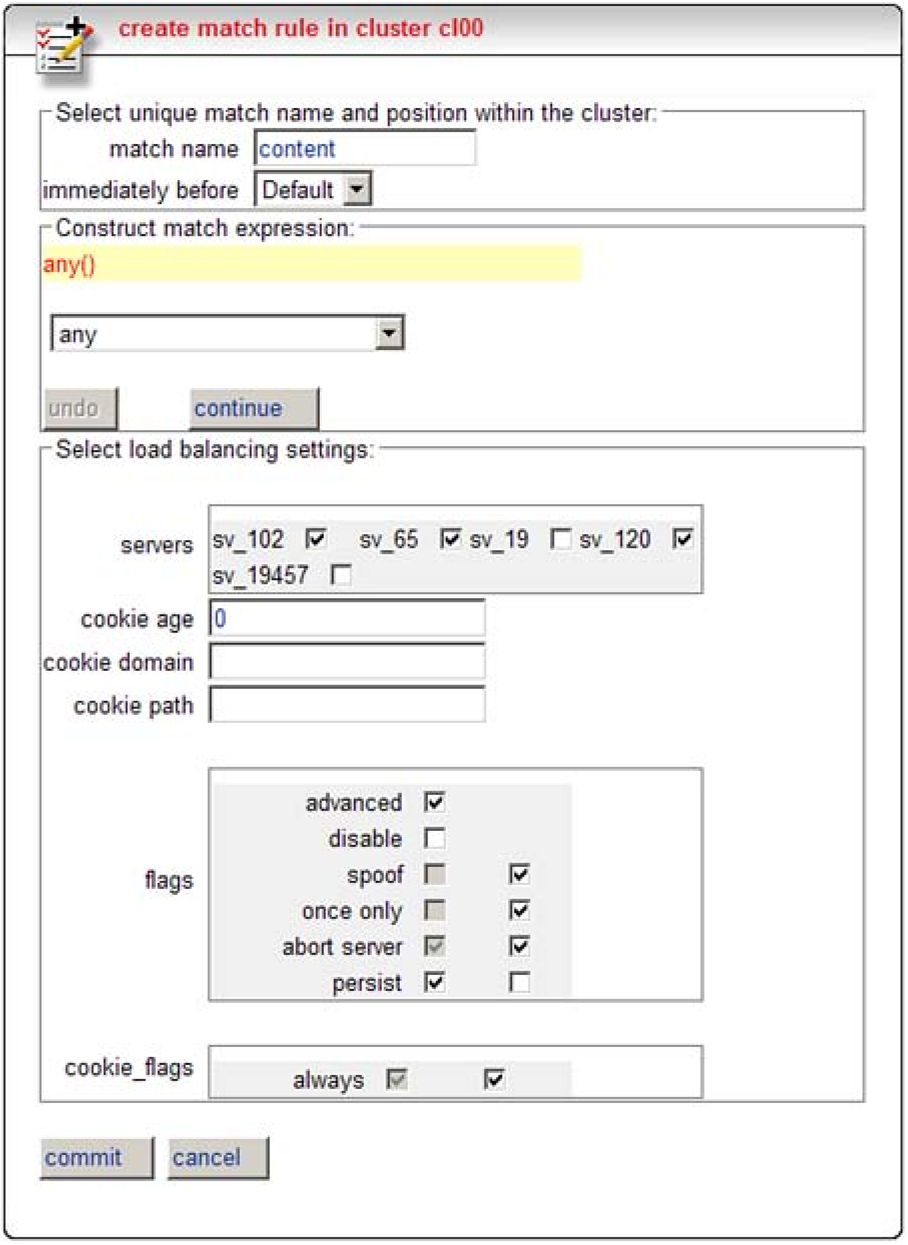
14. In the **cluster** screen, select **menu > Add Match Rule**. The **create match rule** screen appears in the right frame.

15. Type **content** into the **match name** text box, and select **Default** in the **immediately before** drop-down box.

16. In the **servers** field, enable the check boxes for the servers **sv_102**, **sv_65**, and **sv_120**.

17. In the **flags** field, enable the **advanced** check box. In the list of flags that appears, disable the right-hand check box for the **persist** flag; then, enable the left-hand check box next to **persist**. (Remember that in our example we’re enabling **persist** for the content servers, so that persistent

sessions can be maintained by the applications that run on these servers.) The **create match rule** screen should now look like this:



create match rule in cluster cl00

Select unique match name and position within the cluster:

match name: content

immediately before: Default

Construct match expression:

any()

any

Select load balancing settings:

servers: sv_102 ☑ sv_65 ☑ sv_19 □ sv_120 ☑ sv_19457 □

cookie age: 0

cookie domain:

cookie path:

flags:

advanced	☑	
disable	□	
spoof	□	☑
once only	□	☑
abort server	☑	☑
persist	☑	□

cookie_flags: always ☑ ☑

18. Select the **commit** button to create the **content** rule. The home screen for the Administrative Interface is displayed. In the left-frame cluster list, **Match content** should now appear above **Match Default** for the cluster.

9 Administering Geographic Clusters

Geographic Load Balancing with Envoy

The Envoy geographic load balancer, an optional software add-on for the Equalizer product line, supports geographic clustering and load balancing. Geographic clustering and load balancing enables requests to be automatically distributed across servers in different physical locations or on different networks.

Envoy Overview

Equalizer and its set of servers in a particular location forms a *site* (or Envoy site). A geographic cluster contains multiple sites, and Equalizer's geographic load balancing technology balances incoming requests across those sites.

When a client uses DNS to resolve the address of a domain name, it performs a recursive search with a number of name servers to resolve that address. Envoy is the last name server in this search. The name server in the recursive chain immediately before Envoy returns a list of Envoy sites. The client sends requests, one at a time, to each of the Envoy sites until it reaches an active site. If the Envoy site is active, Envoy performs the following steps to determine the site in the geographic cluster that should handle the request:

1. If, for example, Site A in Figure 67 is the first active Envoy site accessed by the client, Site A then identifies the geographic cluster that has been configured with the requested domain name—in this example, `www.coyotepoint.com`.

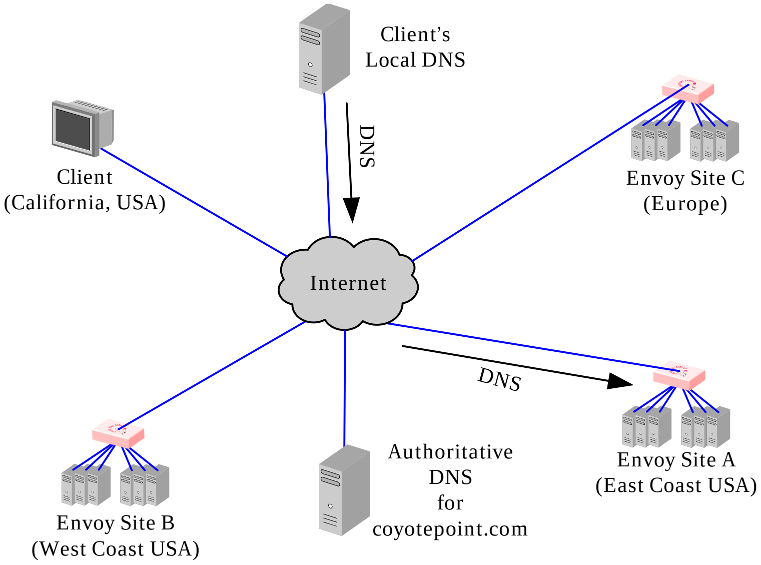

Client's Local DNS
DNS
Client (California, USA)
Envoy Site C (Europe)
Internet
DNS
Envoy Site A (East Coast USA)
Envoy Site B (West Coast USA)
Authoritative DNS for coyotepoint.com

Figure 67 Sending name resolution requests to an Equalizer in a geographic cluster

It does this by sending a *geographic query protocol probe* (GQP) to each site; the probe is received by a special Envoy *agent* running at each site in the cluster (the agent for a site is configured when you configure Envoy for the site). These probes contain information about the requesting client and the resource that is being resolved. Site A also queries its local Envoy agent (see Figure 68).

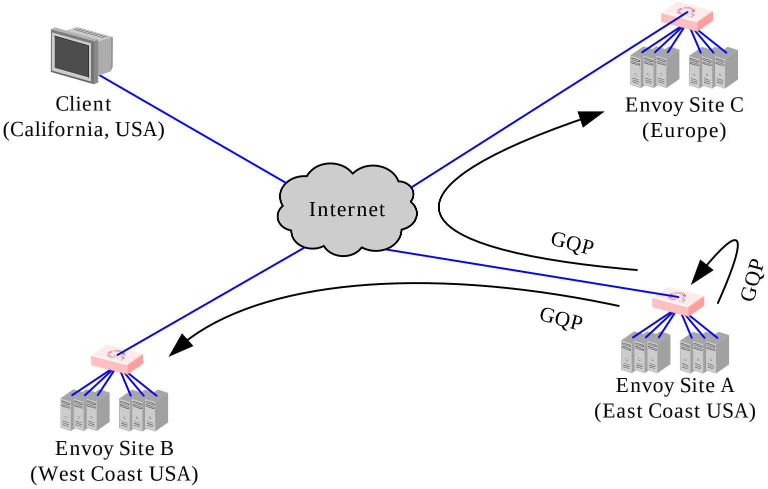


Figure 68 The selected Equalizer queries other Equalizers and its own servers in the geographic cluster

2. The Envoy agent at each site checks the availability of the requested resource (see Figure 69) and sends a reply to Site A via GQP:
    - If the resource is not available at the agent's site, the agent sends an error message to Equalizer.
    - If the resource is available at the agent’s site, the agent sends a message indicating the availability of the resource back to site A via GQP.

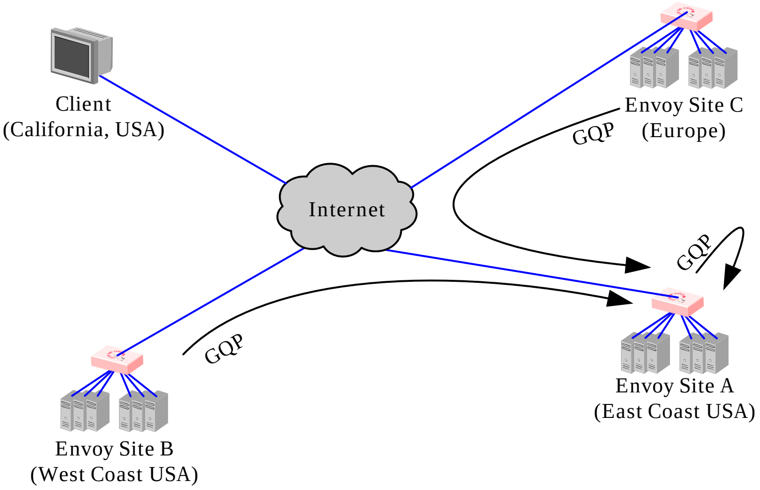


Figure 69 The selected Equalizer receives availability and triangulation (latency) information

Note that if ICMP triangulation is enabled for the geocluster, the agent at a site where the resource is available first sends an ICMP echo request (*ping*) to the requesting *client*. When the echo reply from the client is received, the agent includes latency information in its reply to the

Envoy site that sent the geographic probe (Site A). This provides more accurate client location information to Envoy in the case where a resource is available at more than one site. Envoy will choose the site that will best serve the client according to the latency information received.

3. The site that sent the geographic probe, Site A, returns the address of the best Envoy site to the requesting client's local DNS (see Figure 70).

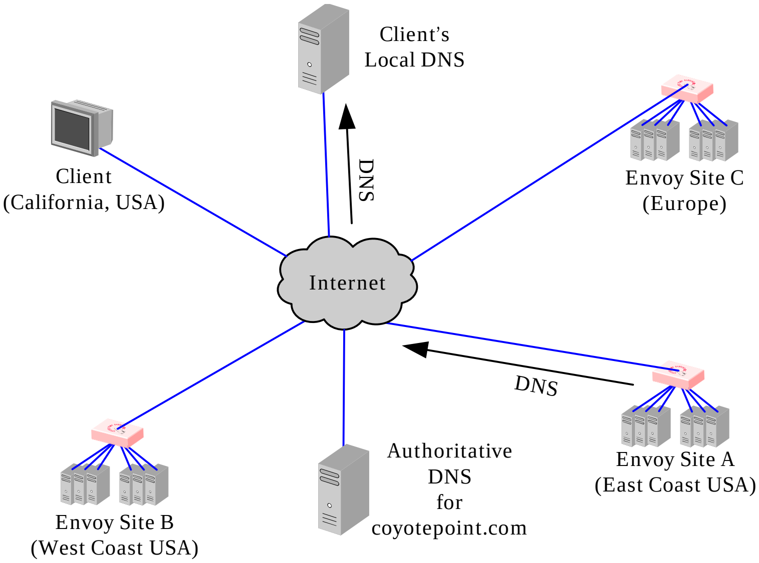


Figure 70 The client's local DNS receives the best Equalizer site

4. The selected Equalizer uses the information gathered from probing each site to determine the site that is best able to process the request for the client and then forwards the request to that site (see Figure 71). This site then responds to the client and the connection is thereafter managed by the chosen site (in our example, Site B).

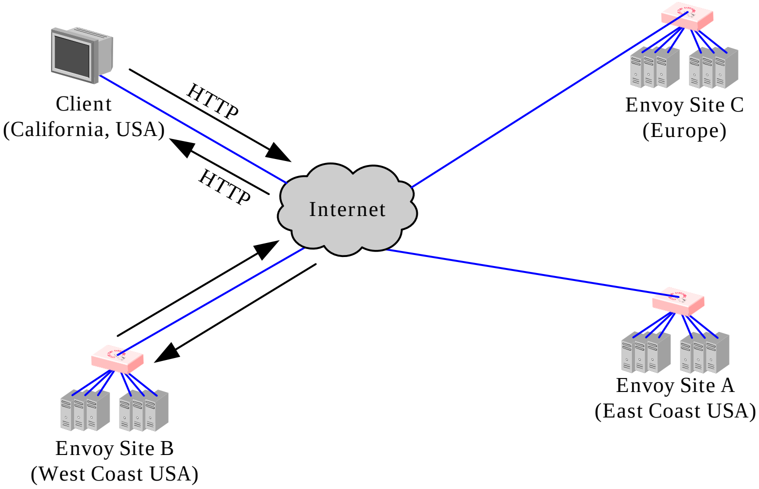


Figure 71 Site B handles the client's connection

Licensing and Configuring Envoy

Each site in an Envoy geocluster must have an Equalizer that is running Envoy, which must be licensed in order to run. Envoy software is pre-installed on each Equalizer and is enabled through the registration and licensing process.

After you have licensed Envoy and completed Envoy and DNS configuration described in this section, you can set up geographic clusters and define the available sites for each cluster.

Enabling Envoy

To license and enable Envoy, follow these steps:

1. Follow the registration procedure and make sure that you enter the serial number for your Envoy software on the registration website; see “Licensing Equalizer” on page 39 in Chapter 5.
2. Shut down the Equalizer and reboot the machine; see “Rebooting Equalizer” on page 63 in Chapter 5.
3. After the system reboots, confirm that Envoy is enabled. Log into the Equalizer Administration Interface and select **View > Equalizer Status**. The line **Envoy geographic load balancing** should indicate that Envoy is **enabled**.

Configuring the Authoritative Name Server to Query Envoy

You must configure the authoritative name server(s) for the domains that are to be geographically load balanced to delegate authority to the Envoy sites. You need to delegate each of the fully-qualified subdomains to be balanced. If your DNS server is run by an Internet Service Provider (ISP), then you need to ask the ISP to reconfigure the DNS server for Envoy. If you are running your own local DNS server, then you need to update the DNS server’s *zone file* for your Envoy configuration.

For example (see Figure 72), assume you must balance `www.coyotepoint.com` across a geographic cluster containing two Envoy sites, `east.coyotepoint.com` (at `192.168.2.44`) and `west.coyotepoint.com` (at `10.0.0.5`). In this case, you must configure the name servers that will handle the `coyotepoint.com` domain to delegate authority for `www.coyotepoint.com` to both `east.coyotepoint.com` and `west.coyotepoint.com`. When queried to resolve `www.coyotepoint.com`, `coyotepoint.com`’s name servers should return name server (NS) and alias (A) records for both Envoy sites.

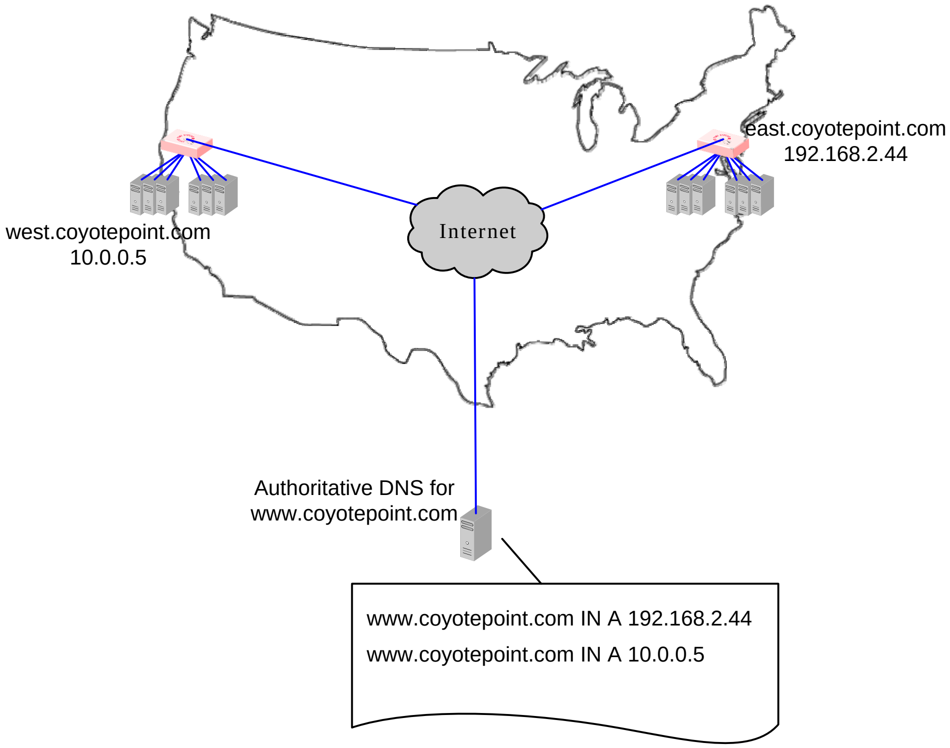


Figure 72 Two-site DNS example

An example of a DNS zone file for this configuration is shown below. In this example, the systems `ns1` and `ns2` are assumed to be the authoritative name servers (master and slave) for the `coyotepoint.com` domain.

```
$TTL 86400
coyotepoint.com. IN SOA ns1.coyotepoint.com. hostmaster.coyotepoint.com. (
                          0000000000
                          00000
                          0000
                          000000
                          00000 )

coyotepoint.com.     IN NS ns1.coyotepoint.com.
coyotepoint.com.     IN NS ns2.coyotepoint.com.
www.coyotepoint.com. IN NS east.coyotepoint.com.
www.coyotepoint.com. IN NS west.coyotepoint.com.

ns1   IN A ns1-IP-address
ns2   IN A ns2-IP-address
east  IN A 192.168.2.44
west  IN A 10.0.0.5
```

Figure 73 Example DNS Zone File

In the example above, we left the domain parameters as zeros, since these vary widely between DNS installations. Please see the documentation for the version of DNS that you are using for more information on the zone file content and format.

Using Envoy with Firewalled Networks

Envoy sites communicate with each other using Coyote Point’s UDP-based Geographic Query Protocol (GQP). Similarly, Envoy sites communicate with clients using the DNS protocol. If you protect one or more of your Envoy sites with a network firewall, you must configure the firewall to permit the Envoy packets to pass through.

To use Envoy with firewalled networks, you need to configure the firewalls so that the following actions occur:

- Envoy sites communicate with each other on UDP ports 5300 and 5301. The firewall must allow traffic on these ports to pass between Equalizer/Envoy sites.
- Envoy sites and clients can exchange packets on UDP port 53. The firewall must allow traffic on this port to flow freely between an Envoy server and any Internet clients so that clients trying to resolve hostnames via the Envoy DNS server can exchange packets with the Envoy sites.
- Envoy sites can send ICMP echo request packets out through the firewall and receive ICMP echo response packets from clients outside the firewall. When a client attempts a DNS resolution, Envoy sites send an ICMP echo request (ping) packet to the client and the client might respond with an ICMP echo response packet.

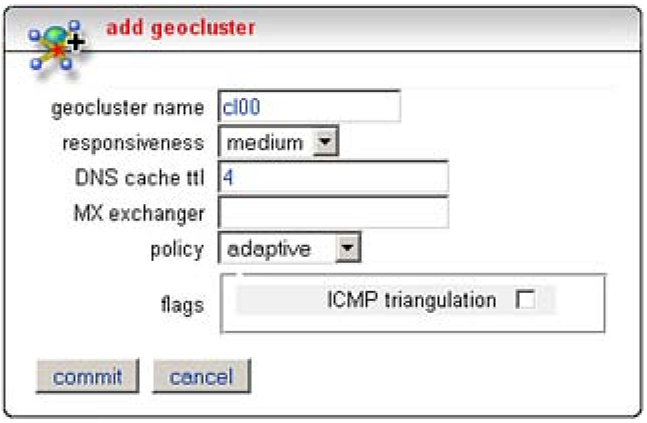
Working with Geographic Clusters

This section shows you how to add or delete a geographic cluster and how to configure a geographic cluster's load-balancing options. Configuring a geographic cluster and its sites is analogous to configuring a virtual cluster and its servers.

Adding a Geographic Cluster

To add a new geographic cluster, follow these steps:

1. Log into the Equalizer Administration Interface in edit mode.
2. Select **Geographic Cluster** from the Add menu in the main menu bar. The add geocluster screen appears in the right frame (see Figure 74).



add geocluster

geocluster name: cl00
responsiveness: medium
DNS cache ttl: 4
MX exchanger:
policy: adaptive
flags: ICMP triangulation □

Figure 74 Add geocluster screen

3. Enter the **geocluster name**, which is the fully-qualified domain name (FQDN) of the geographic cluster (for example, `www.coyotepoint.com`). The FQDN must include all name components up to the top level (com, net, org, etc). Do not include the trailing period.
4. Specify the **responsiveness**. This value controls how aggressively Equalizer adjusts the site's dynamic weights. Equalizer provides five response settings: Slowest, Slow, Medium, Fast, and Fastest. Faster settings enable Equalizer to adjust its load balancing criteria more frequently and permit a greater variance in the relative weights assigned to sites. Slower settings cause site measurements to be averaged over a longer period of time before Equalizer applies them to the cluster-wide load balancing; slower settings also tend to ignore spikes in cluster measurements caused by intermittent network glitches. We recommend that you select the *Medium* setting as a starting point.
5. Enter the **DNS cache ttl** (cache time-to-live), which is the length of time, in seconds, that the client's DNS server should cache the resolved IP address. Longer times will result in increased failover times in the event of a site failure, but are more efficient in terms of network resources. A reasonable value would be 120 (that is, 2 minutes).
6. Enter the **MX exchanger**, which is the fully qualified domain name to be returned if Equalizer receives a "mail exchanger" request for this geographic cluster. The mail exchanger is the host responsible for handling email sent to users in the domain.
7. Specify the **policy**:

- **round trip**: This method weights the client's network proximity more heavily than other criteria. This option only works if you enable Ping Triangulation.
- **adaptive**: This method takes all available information into account when selecting a site. This setting is a reasonable default.
- **site load**: This method weights the current load at each site more heavily than other criteria.
- **site weight**: This method weights the user-defined static weight for each site more heavily than other criteria.

8. Check or clear the **ICMP triangulation** checkbox. When you check ICMP triangulation, each Envoy site pings the client and collects latency information, which provides more accurate client location information. If you do not want to allow Equalizer to ping clients when they make a request, clear the ICMP triangulation checkbox.
9. Click the **commit** button to add the geographic cluster. An entry for the new geographic cluster appears in the left frame.

Equalizer can refuse an Add GeoCluster command for several reasons, including:

- Attempting to add a cluster for a FQDN that is already configured
- Attempting to add more clusters than are supported by Equalizer

Configuring a Geographic Cluster's Load-Balancing Options

You can change the load balancing policy and response settings for a geographic cluster from the geocluster screen. Configure these parameters independently for each geographic cluster. (For more information about the load balancing policy and response settings, see "Adding a Geographic Cluster" on page 155.)

You might want to fine-tune the static weights of the geographic cluster's sites to optimize cluster performance. For more information, see "Adjusting a Site's Static Weight" on page 159.

To change a geographic cluster's load-balancing options, follow these steps:

1. Log into the Equalizer Administration Interface in edit mode.
2. In the left frame, click the link formed by the domain name associated with the geographic cluster. The geocluster screen opens in the right frame.
3. Select **Change GeoCluster Parameters** from the local menu. The modify geocluster screen appears in the right frame.
4. Specify the **responsiveness**, which controls how aggressively Equalizer adjusts the site's dynamic weights: Slowest, Slow, Medium, Fast, and Fastest. The faster settings enable Equalizer to adjust its load balancing criteria more frequently and permit a greater variance in the relative weights assigned to sites. A slow setting causes site measurements to be averaged over a longer period of time before Equalizer applies them to the cluster-wide load balancing and tend to ignore spikes in cluster measurements caused by intermittent network glitches.
5. Specify the **DNS cache ttl** (cache time-to-live), which is the length of time, in seconds, that the client's DNS server should cache the resolved IP address. Longer times will result in increased failover times in the event of a site failure, but are more efficient in terms of network resources. A reasonable value would be 120 (that is, 2 minutes).

6. Specify the **MX exchanger**, which is the fully qualified domain name to be returned if Equalizer receives a “mail exchanger” request for this geographic cluster. The mail exchanger is the host responsible for handling email sent to users in the domain.
7. Select a **policy**. The policy determines the algorithm that Equalizer will use to distribute requests among the sites in the cluster:
   - **round trip**, which weights the client’s network proximity more heavily than other criteria.
   - **adaptive**, which takes all available information into account when selecting a site. This setting is a reasonable default.
   - **site load**, which weights the current load at each site more heavily than other criteria.
   - **site weight**, which weights the user-defined static weight for each site more heavily than other criteria.
8. Check or clear the **ICMP triangulation** checkbox. When you check ICMP triangulation, each Envoy site pings the client and collects latency information, which provides more accurate client location information. If you do not want to allow Equalizer to ping clients when they make a request, clear the ICMP triangulation checkbox.
9. Click the **commit** button.

Deleting a Geographic Cluster

To delete a geographic cluster, follow these steps:

1. Log into the Equalizer Administration Interface in edit mode.
2. In the left frame, click the name of the geographic cluster to be deleted. The geocluster screen appears in the right frame.
3. Select **Delete this GeoCluster** from the local menu in the geocluster screen.
4. When prompted, click **OK** to verify that you really want to remove the cluster. Equalizer deletes the GeoCluster and all its sites.

Working with Sites

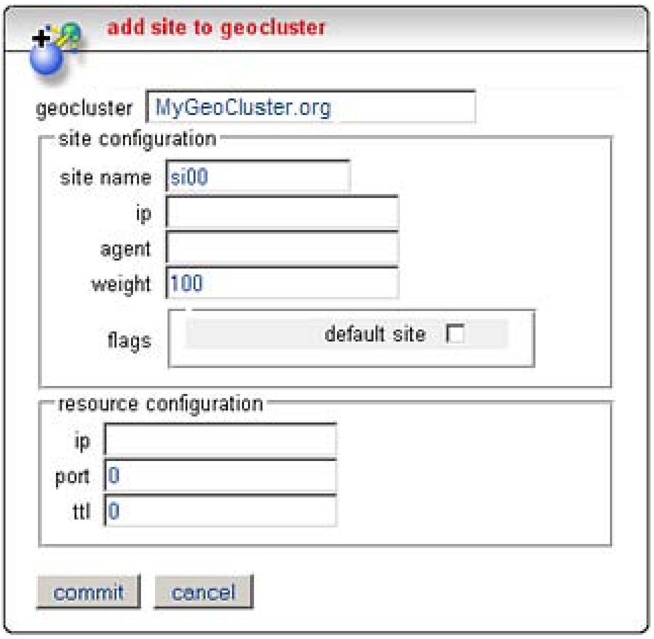
This section describes how to use Equalizer to add or delete a site from a geographic cluster and how to adjust a site’s static weight.

Adding a Site to a Geographic Cluster

To add a site to an existing geographic cluster, follow these steps:

1. Log into the Equalizer Administration Interface in edit mode.
2. In the left frame, click the name of the geographic cluster to which you want to add the site.

3. Select **Add Site** from the local menu. The add site screen opens in the right frame (see Figure 75).



geocluster MyGeoCluster.org

site name si00

ip

agent

weight 100

flags default site □

ip

port 0

ttl 0

Figure 75 Add site to geocluster screen

4. Enter the **site name**, which is a symbolic name that represents this site. For example, the east-coast site for `www.coyotepoint.com` might be `eastCOAST`.

5. Enter the **ip**, which is the IP address of the site. This is the address of an Equalizer cluster that is returned if the site is chosen.

6. Enter the **agent**, which is the IP address of the site monitoring agent. Usually, this is the external (or Envoy failover) address of the Equalizer at this site.

7. Enter the **Static Weight** value, which represents the site's capacity. (This value is similar to a server's static weight.) Valid values range between 10 and 200. Use the default of 100 if all sites are configured similarly; otherwise, adjust higher or lower for sites that have more or less capacity.

8. Check or clear the **Default** checkbox. You can designate only one site in a cluster as the default.
   - Equalizer returns a peer site's IP address based on the selected load balancing algorithms.
   - Choose the default site if the client's DNS server did not respond to ICMP echo requests from any site. This can happen if a firewall blocks ICMP packets between the client's DNS and the internet.

9. Enter the **ip** in the resource configuration section, which is the IP address of the resource that is monitored for this site. This must be the same address as a configured Equalizer cluster and is generally the same value as the site address. For example, `east.coyotepoint.com` might have resource `IP=192.168.0.5` and `Port=80` if this cluster were configured on Equalizer.

10. Enter the **port**, which is the TCP port number of the resource that is monitored for this site.

11. Enter the **ttl** value, which is how often the agent should probe the resource. A value of 100 results in the resource's availability being tested every 100 seconds.

12. Click the **commit** button.

Equalizer can refuse an Add Site command for several reasons, including attempting to add:

- A site with a name or IP address that is already configured
- More sites than are supported by Equalizer
- A default site when you have already configured a default site

Adjusting a Site’s Static Weight

Equalizer uses a site’s static weight as the starting point for determining what percentage of requests to route to that site. Equalizer assigns sites with a higher static weight a higher percentage of the load. The *relative* values of site static weights are more important than the actual values. For example, if two sites are in a geographic cluster and one has roughly twice the capacity of the other, setting the static weights to 50 and 100 is equivalent to setting the static weights to 100 and 200.

Dynamic site weights can vary from 50% to 150% of the assigned static weights. To optimize geographic cluster performance, you might need to adjust the static weights of the sites in the cluster based on their performance.

Site weights can range from 10 to 200. When you set up sites in a geographic cluster, you should set each site’s static weight value in proportion to its capacity for handling requests. It is not necessary for all of the static weights in a cluster to add up to any particular number.

To change a site’s static weight, follow these steps:

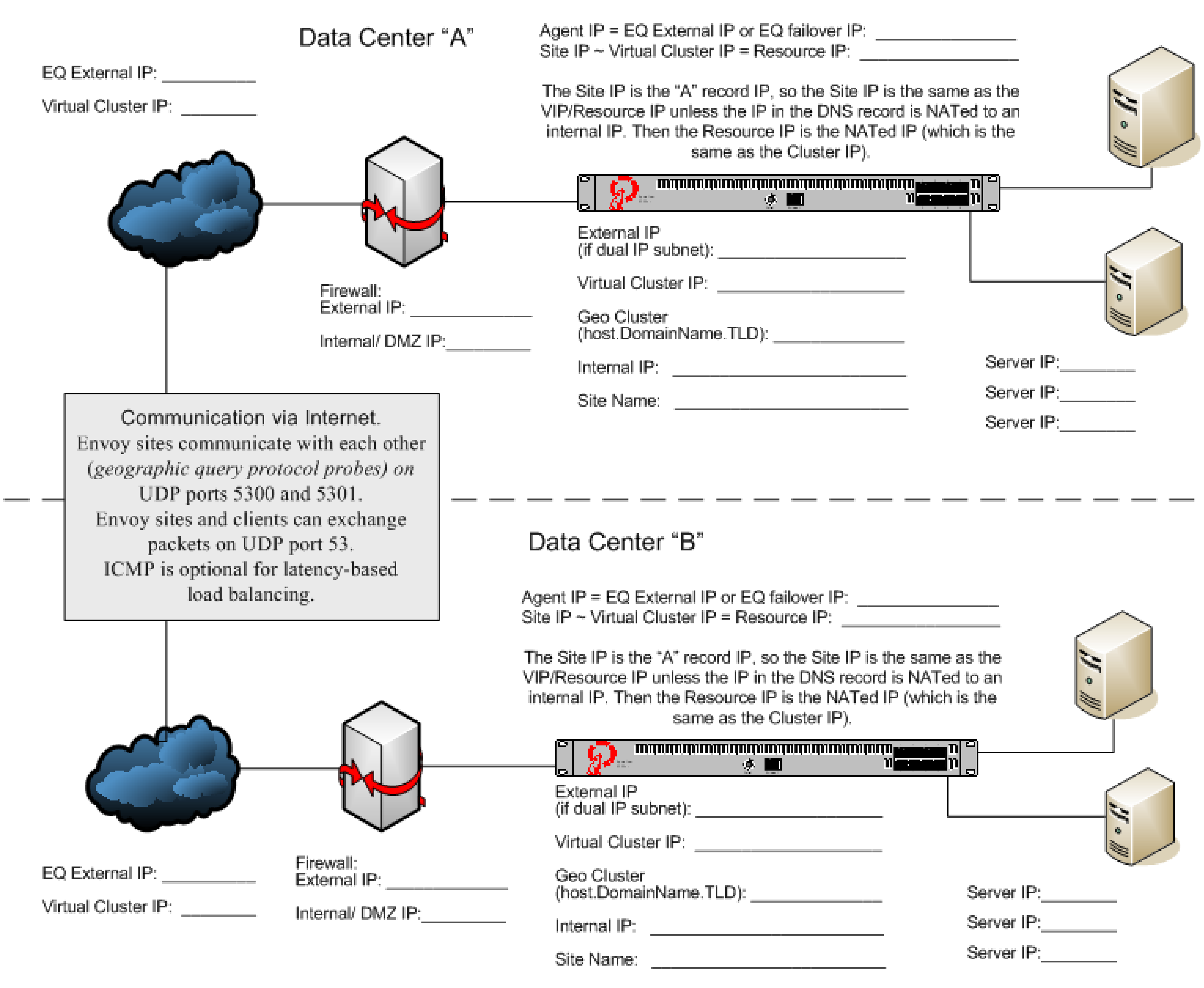
1. Log into the Equalizer Administration Interface in edit mode.
2. In the left frame, click the name of the site to be modified. The site screen opens in the right frame.
3. Select **Change GeoSite Parameters** from the local menu. The modify site screen appears in the right frame.
4. Enter the new weight in the **weight** field.
5. Click the **commit** button.

Deleting a Site from a Geographic Cluster

To delete a site from a geographic cluster, follow these steps:

1. Log into the Equalizer Administration Interface in edit mode.
2. In the left frame, click the name of the site to be deleted. The site screen appears in the right frame.
3. Select **Delete this GeoSite** from the local menu.
4. When prompted, click **OK** to confirm that you really want to remove the site.

Envoy Configuration Worksheet



Data Center "A"

EQ External IP: ____________

Virtual Cluster IP: __________

Firewall:
External IP: ________________

Internal/ DMZ IP:__________

Agent IP = EQ External IP or EQ failover IP: ________________
Site IP ~ Virtual Cluster IP = Resource IP: __________________

The Site IP is the "A" record IP, so the Site IP is the same as the VIP/Resource IP unless the IP in the DNS record is NATed to an internal IP. Then the Resource IP is the NATed IP (which is the same as the Cluster IP).

External IP
(if dual IP subnet): ______________________

Virtual Cluster IP: ______________________

Geo Cluster
(host.DomainName.TLD): ______________

Internal IP: ____________________________

Site Name: ____________________________

Server IP:________

Server IP:________

Server IP:________

Communication via Internet.
Envoy sites communicate with each other *(geographic query protocol probes)* on UDP ports 5300 and 5301.
Envoy sites and clients can exchange packets on UDP port 53.
ICMP is optional for latency-based load balancing.

Data Center "B"

Agent IP = EQ External IP or EQ failover IP: ________________
Site IP ~ Virtual Cluster IP = Resource IP: __________________

The Site IP is the "A" record IP, so the Site IP is the same as the VIP/Resource IP unless the IP in the DNS record is NATed to an internal IP. Then the Resource IP is the NATed IP (which is the same as the Cluster IP).

External IP
(if dual IP subnet): ______________________

Virtual Cluster IP: ______________________

Geo Cluster
(host.DomainName.TLD): ______________

Internal IP: ____________________________

Site Name: ____________________________

Server IP:________

Server IP:________

Server IP:________

EQ External IP: ____________

Virtual Cluster IP: __________

Firewall:
External IP: ________________

Internal/ DMZ IP:__________

A Using Server Agents

Introducing Server Agents

You can configure Equalizer's load balancing algorithms to accept direct feedback from servers that describe the current server load or availability of critical resources. This is done by writing a *server agent* and deploying it on your servers.

The agent must listen on and be able to respond to TCP connections on a well-known port. The response is in the form of an ASCII string (an integer between -1 and 100) that represents the current load on the server, or indicates that service is not available. If an agent indicates that service is unavailable, Equalizer will automatically stop sending requests to that server.

You configure server agents on a cluster-wide basis—all the servers in a virtual cluster must be running agents for server agents to be used for adaptive load balancing. When you have enabled server agents, Equalizer periodically probes the agent at each server's IP address through the configured agent port. Equalizer uses the collected server agent values when performing adaptive load balancing calculations. You configure Equalizer to use server agents through the Change Cluster dialog box. (For more information, see “Configuring a Cluster to Use Server Agents” on page 78.)

Agents and Load Balancing Policies

Agents work with all load balancing policies (see “Equalizer’s Load Balancing Policies” on page 74), except for **round robin** (which simply ignores any agent defined for the cluster). All the other policies use the integer returned by the agent as one factor in determining the server to which a new request is sent.

The **server agent policy** gives primary importance to the value returned by an agent over other load balancing factors (server weight, number of current connections, etc.).

Enabling Agents

Agents are enabled for a cluster by turning on the **server_agent** flag in the **add cluster** or **modify cluster** screens. Turing on this flag also changes the **agent port** setting from **0** (off) to **1510**, the default port used for agents. Make sure that any agent you write and deploy is listening on the same port number on the servers.

By default, Equalizer will open up a connection to the server agent’s IP/port, and wait for a response. If no response is received, then the Equalizer performs load balancing without the server agent value for that server.

Some agents, particularly those written in Java, may require that a string be sent to the agent before a response is sent back to Equalizer. The advanced field **agent probe** is provided for this purpose. If a string appears in this field, it is sent to the agent when an agent probe occurs.

There is also a **pedantic agent** flag that tells Equalizer to regard a server as down if there is no response from the server's agent. This flag is set in the **system parameters**, and when it is enabled it applies to *all* clusters that have agents. See "Modifying System Parameters" on page 43.

Writing Server Agents

You can write custom agents as shell scripts, or in Perl, C, or other languages.

Equalizer's agent protocol is extremely simple: when Equalizer connects to the agent's port, the agent must respond with an ASCII string (a number that represents the current condition of the server) and then close the port.

Conditions on the server are inidicated by the agent's return value as follows:

-1	Service is unavailable. Might also be used to indicate that a required resource, such as a database, is unavailable.
0 to 100	0 indicates that the server is very lightly loaded; 100 indicates that the server is very overloaded.

The code on the following page is a simple server agent example written in Perl. This code prompts for a constant value when the server agent program is started, and returns that value when a connection is made on port 1510 (configurable via the `$port` variable).

.

```
#!/usr/bin/perl -w
# serveragent.pl
#--------------------
#(c) Copyright 2007 Coyote Point Systems, Inc.

use strict;
use Socket;

# use port 1510 as default my $port = 1510;
my $proto = getprotobyname('tcp');

# take the response from the command line
my $response = shift;

# response has to be a valid server agent response
response==-1 or ($response > 0 and  $response<101)
    or die "Response must be between -1 and 100";

# create a socket and set the options, set up listen port

socket(SERVER, PF_INET, SOCK_STREAM, $proto) or die "socket: $!";
setsockopt(SERVER, SOL_SOCKET, SO_REUSEADDR, 1) or die "setsock: $!";
my $paddr = sockaddr_in($port, INADDR_ANY);

# bind to the port, then listen on it
bind(SERVER, $paddr) or die "bind: $!";
listen(SERVER, SOMAXCONN) or die "listen: $!";
print "Server agent started on port $port\n";

# accepting a connection
my $client_addr;
while ($client_addr = accept(CLIENT, SERVER)) {
        # find out who connected
        my ($client_port, $client_ip) = sockaddr_in($client_addr);
        my $client_ipnum = inet_ntoa($client_ip);

        # print who has connected -- this is for debugging only
        print "Connection from: [$client_ipnum]\n";
        # send them a message, close connection
        print CLIENT $response;
        close CLIENT;
}
```

Here is the output of the server program when it is started on the server:

```
$ ./serveragent.pl 50
Server agent started on port 1510
Connection from: [10.0.0.32]
```

Here is what I see when I **telnet** to the port:

```
$ telnet 10.0.0.120 1510
50
Connection to host lost.
```

This program is only an example because it doesn't make any useful calculations of what the server agent response should be. Such calculations need to be made by the customer depending on what the server agent program is monitoring.

B Using Reserved IP Addresses

Equalizer supports placing servers on *reserved*, non-routable networks such as the class A network 10.0.0.0 and the class C network 192.168.2.0. In environments in which the conservation of IP addresses is important, using reserved IP addresses can minimize the number of "real" IP addresses needed.

For example, an ISP hosting several hundred unique web sites replicated on three servers might not want to assign real IP addresses for all of them because each virtual cluster would consume four addresses: three on the back-end servers and one for the virtual cluster. In this case, the ISP might use 10.0.0.0 (the now-defunct Arpanet) as the internal network and assign virtual server addresses out this network for the servers. Figure 67 illustrates a typical reserved internal network.

> **Note –** Due to the additional overhead introduced by enabling outbound NAT, approach using reserved internal networks with caution.

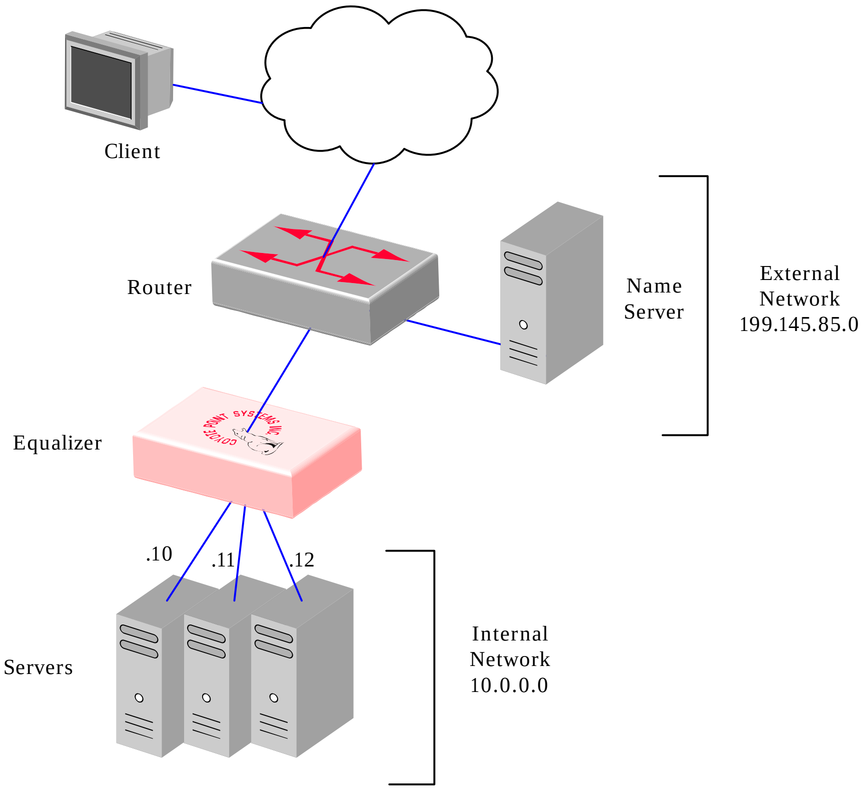

Client
Router
Name Server
External Network 199.145.85.0
Equalizer
COYOTE POINT SYSTEMS INC.
.10
.11
.12
Servers
Internal Network 10.0.0.0

Figure 76 Reserved Internal Network

If servers placed on a non-routable network need to communicate with hosts on the Internet for any reason (such as performing DNS resolution or sending email), you must configure Equalizer to perform *outbound NAT* (network address translation). When you have enabled outbound NAT, Equalizer translates connections originating from the servers on the reserved network so that external hosts will not see packets originating from non-routable addresses.

To enable Equalizer to perform outbound NAT, follow these steps:

1. Open the Equalizer Administration Interface and log in under edit mode.
2. In the left frame, click the **Equalizer** entry at the top of the column.
3. Select **Change Equalizer Parameters** from the local menu. The modify system parameters screen appears in the right frame.
4. Check the **enable outbound NAT**.
5. Click the **commit** button.

Note – If you're using a Failover configuration, use the same outbound NAT setting on both Equalizers.

You will find a worksheet for configuring and using reserved IP addresses in "Equalizer Configuration Worksheets" on page 16.

C Regular Expression Format

Equalizer supports only IEEE Std 1003.2 (POSIX.2) regular expressions in Match Rules. There are many other variants and extensions of regular expressions, including those found in Perl, Java, various shell languages, and the traditional Unix **grep** family of utilities; these variants are not supported in Match Rules.

Regular expressions can be difficult to create and debug, and can use significvant system resources to process. We recommend you use regular expressions only when no other Match Rule function will provide the functionality you require.

To aid in creating correct and efficient regular expressions, you can use a regular expression evaluator; many of these are available for download on the internet. Two free online regular expression evaluators are also available at the following websites:

> **http://www.rexv.org/** (choose POSIX tab)
> **http://www.projects.aphexcreations.net/rejax/** (choose PHP POSIX Language)

Terms

The terms in this section describe the components of regular expressions.

- A *regular expression* (RE) is one or more non-empty branches, separated by pipe symbols (|). An expression matches anything that matches one of the branches.
- A *branch* consists of one or more concatenated pieces. A branch matches a match for the first piece, followed by a match for the second, and so on.
- A *piece* is an atom optionally followed by a single *, +, or ?, or by a bound.
    - An atom followed by an asterisk matches a sequence of 0 or more matches of the atom.
    - An atom followed by a plus sign matches a sequence of 1 or more matches of the atom.
    - An atom followed by a question mark matches a sequence of 0 or 1 matches of the atom.
- A *bound* consists of an open brace ({) followed by an unsigned decimal integer, between 0 and 255 inclusive. You can follow the first unsigned decimal integer with a comma, or a comma and a second unsigned decimal integer. Close the bound with a close brace (}). If there are two integers, the value of the first may not exceed the value of the second.

Learning About Atoms

An *atom* followed by a bound that contains one integer *i* and no comma matches a sequence of exactly *i* matches of the atom. An atom followed by a bound that contains one integer *i* and a comma matches a sequence of *i* or more matches of the atom. An atom followed by a bound containing two integers *i* and *j* matches a sequence of *i* through *j* (inclusive) matches of the atom. An atom can consist of any of the following:

- A regular expression enclosed in parentheses, which matches a match for the regular expression.
- An empty set of parentheses, which matches the null string.
- A bracket expression.
- A period (.), which matches any single character.
- A carat (^), which matches the null string at the beginning of a line.
- A dollar sign ($), which matches the null string at the end of a line.
- A backslash (\) followed by one of the following characters: ^.[$()|*+?{\, which matches that character taken as an ordinary character.
- A backslash (\) followed by any other character, which matches that character taken as an ordinary character (as if the \ had not been present).
- A single character with no other significance, which simply matches that character. **Note that regular expressions are case-insensitive.**
- An open brace ({) followed by a character other than a digit is an ordinary character, not the beginning of a bound. It is illegal to end a real expression with a backslash (\).

Creating a Bracket Expression

A *bracket expression* is a list of characters enclosed in brackets ([...]). It normally matches any single character from the list. If the list begins with ^, it matches any single character not from the rest of the list. Two characters in a list that are separated by '-' indicates the full range of characters between those two (inclusive) in the collating sequence; for example, '[0-9]' in ASCII matches any decimal digit. It is illegal for two ranges to share an endpoint; for example, 'a-c-e'. Ranges are very collating-sequence-dependent, and portable programs should avoid relying on them.

- To include a literal ']' in the list, make it the first character (following an optional '^').
- To include a literal '-', make it the first or last character, or the second endpoint of a range.
- To use a literal '-' as the first endpoint of a range, enclose it in '[.' and '.]' to make it a collating element (see below).

With the exception of these and some combinations using '[' (see next paragraphs), all other special characters, including '\', lose their special significance within a bracket expression.

Within a bracket expression, a collating element (a character, a multi-character sequence that collates as if it were a single character, or a collating-sequence name for either) enclosed in '[.' and '.]' stands for the sequence of characters of that collating element. The sequence is a single element of the bracket expression's list. A bracket expression containing a multi-character collating element can thus match more than one character; e.g., if the collating sequence includes a 'ch' collating element, then the real expression '[[.ch.]]*c' matches the first five characters of 'chchcc'.

Within a bracket expression, a collating element enclosed in '[' and `]' is an equivalence class, representing the sequences of characters of all collating elements equivalent to that one, including itself. (If there are no other equivalent collating elements, the treatment is as if the enclosing delimiters were '[.' and '.]'.) For example, if 'x' and 'y' are the members of an equivalence class, then '[[x]]', '[[y]]', and '[xy]' are all synonymous. An equivalence class may not be an end-point of a range.

Within a bracket expression, the name of a character class enclosed in '[:' and ':]' stands for the list of all characters belonging to that class.

There are two special cases of bracket expressions: the bracket expressions '[[:<:]]' and '[[:>:]]' match the null string at the beginning and end of a word respectively. A word is defined as a sequence of word characters that is neither preceded nor followed by word characters. A word character is an alnum character (as defined by ctype(3)) or an underscore. This is an extension, compatible with but not specified by IEEE Std 1003.2 ("POSIX.2"), and should be used with caution in software intended to be portable to other systems.

Matching Expressions

If a real expression could match more than one substring of a given string, the real expression matches the one starting earliest in the string. If the real expression could match more than one substring starting at that point, it matches the longest. Subexpressions also match the longest possible substrings, subject to the constraint that the whole match be as long as possible, with subexpressions starting earlier in the real expression taking priority over ones starting later. Note that higher-level subexpressions thus take priority over their lower-level component subexpressions.

Match lengths are measured in characters, not collating elements. A null string is considered longer than no match at all. For example, 'bb*' matches the three middle characters of 'abbbc', '(wee|week)(knights|nights)' matches all ten characters of 'weeknights', when '(.*).*' is matched against 'abc' the parenthesized subexpression matches all three characters, and when '(a*)*' is matched against 'bc' both the whole real expression and the parenthesized subexpression match the null string.

D HTTPS Cluster Certificates

The sections below tell you how to get your Layer 7 HTTPS clusters running with certificates. Please read these sections completely before beginning to work with certificates on Equalizer.

While this document tells you all you need to know to use certificates with HTTPS clusters, it is *not* a primer on HTTPS, SSL, or certificates. There are many resources on the Internet, in trade publications, and in books on these topics; in addition, most SSL certificate vendors offer basic SSL overviews on their websites.

Using Certificates in HTTPS Clusters

The HTTPS protocol supports encrypted, secure communication between clients and servers. It requires that a Secure Sockets Layer (SSL) authentication handshake occur between a client and a server in order for a connection request to succeed.

When a client requests an HTTPS connection to a web server, the server (which has already been set up to support SSL connections) sends a *server certificate* to the client for verification. The client checks the content of the certificate against a local database of *Certificate Authorities*, and if it finds a match the connection is made. If no match is found (as is often the case with self-signed certificates), the browser will display a warning and ask if you want to continue with the connection.

A further level of trust can be enabled by setting the server up to request a *client certificate* in addition to the server certificate. Copies of the client certificate are pre-installed on both client and server. When the server sends the server certificate to the client, it also sends a request for a certificate from the client. Once the client accepts the server certificate as described above, it sends the client certificate to the server for verification. The server compares the client certificate it receives with its local copy of the client certificate, and if they match the connection is made.

A server certificate is required for an HTTPS connection; a client certificate is optional.

HTTPS and Equalizer Clusters

In the typical HTTPS scenario described above, the client and server are communicating directly, and the server is doing all the work of encrypting and decrypting packets, comparing certificates, and authenticating clients. If you have many systems servicing requests for the same website, you'll need to install certificates on each server.

With Equalizer, you do not need to install a certificate on every server in a Layer 7 HTTPS cluster. Since certificates are associated with host names and not IP addresses, you only need a server certificate for each HTTPS cluster and the certificates are installed only on Equalizer -- not on each server. This reduces maintenance by reducing the number of certificates required for a group of systems serving content for the same host name.

When a client requests a connection to an HTTPS cluster, Equalizer establishes the HTTPS connection with the client, off loading SSL processing from all the servers in the HTTPS cluster.

Equalizer communicates with the clients via HTTPS; the traffic between Equalizer and the servers in an HTTPS cluster is HTTP (i.e., unencrypted). Compared to the typical scenario where each server is establishing direct HTTPS connections with clients, encrypting and decrypting packets, and serving content as well, SSL offloading improves the overall performance of the cluster.

For even better performance, an optional Xcel SSL Acceleration Card can be installed in Equalizer. With Xcel, all SSL processing is done by the Xcel card, enhancing overall HTTPS throughput. For more information on Xcel, please visit the Coyote Point website (`www.coyotepoint.com`) and Support Portal (`support.coyotepoint.com`)

Note that HTTPS and certificates can be used on servers in Layer 4 TCP and UDP clusters, but you *will* need to install a server and client certificate on *each* server in the cluster (since Equalizer is not doing any HTTPS/SSL processing in Layer 4). In this scenario, no certificates are installed on Equalizer.

About Certificates and HTTPS Clusters

Each Layer 7 HTTPS cluster requires a *server* certificate; a *client* certificate is optional.

Web servers (such as Apache) and browsers (such as Internet Explorer and Firefox) are delivered with pre-installed Trusted Root Certificates. Trusted Root Certificates are used to validate the server and client certificates that are exchanged when an HTTPS connection is established.

Equalizer supports self-signed certificates, as well as signed certificates from Trusted Root Certificate Authorities and from Certificate Authorities (CAs) without their own Trusted Root CA certificates. If a CA without its own Trusted Root CA certificate issues your certificate, you will need to install at least two certificates: a server certificate and a chained root (or intermediate) certificate for the CA. The intermediate certificate associates the server certificate with a Trusted Root certificate.

Similarly, if you want to use client certificates with an HTTPS cluster, you’ll need to get a signed client certificate from a CA, or create a self-signed certificate. A client certificate needs to be installed on each client that will access the Equalizer cluster, as well as on Equalizer. The same client certificate can be used on all clients (i.e., you don’t need to buy or create a separate certificate for each client system).

Just as with server certificates, you may need to install a client certificate and a chained root certificate, if you obtain your certificates from a CA without its own Trusted Root CA certificate. Some sites prefer to use self-signed certificates for clients, or set up their own local CA to issue client certificates.

For several good tutorials on how to get your certificates signed, please see:

```
http://sial.org/howto/openssl/
```

Whichever method you choose, follow these general guidelines for certificates you want to use with Equalizer:

- Equalizer accepts both the **x509 PEM** or **PKCS12** certificate formats; PEM files usually have a *.pem* extension; PKCS12 files usually have a *.pfx* extension. Most CA vendors provide certificates in PEM format.
- If you are using an Xcel I accelerator card, use a private key **bit length** that is a multiple of **8** (e.g., 1024, 2048, etc.). This restriction does not apply to newer generation Xcel II cards.
- When uploading certificates to Equalizer, the certificates and private key must be contained in a single plain text file, in the following order:

- server certificate
- private key
- chained root (intermediate) certificates (if any)

Enabling HTTPS with a Server Certificate

The following are the steps to follow to obtain and install a server certificate, and verify that it works.

1. **Generate a Server Certificate Signing Request or a Self-Signed Server Certificate.**

   To get a server certificate, do *one* of the following:

   a. **Create a Certificate Signing Request (CSR) and send it to a Certificate Authority for signing.** This provides the highest level of trust to the client, as the client can be assured that the certificate it receives from the server (in this case, Equalizer) was approved (i.e., digitally signed) by a trusted third party. Thus, the client has the assurance of a third party that the server to which it is connecting is identifying itself legitimately (and is not impersonating the legitimate server's identity). See the section "Generating a CSR and Getting It Signed by a CA" on page 175.

   b. **Create a certificate and sign it yourself.** This provides a lower level of trust, since the client is essentially trusting the server to identify itself. Self-signed certificates are relatively easy to counterfeit, and are only recommended for use on internal, non-production, or test configurations. See the section "Generating a Self-Signed Certificate" on page 176.

2. **Create the HTTPS cluster.**

   When creating an HTTPS cluster, the default flags and parameters are acceptable for most server certificate configurations. However, if the server certificate you have does not sctrictly conform to the standard x509 format, disable the **x509 verify** flag (enable the **advanced** flag to see it in the flag section of the **add cluster** or **modify cluster** screens). Many self-signed and some chained certificates may not be in strict x509 format.

   For more information on creating HTTPS clusters, see Chapter 6, "*Administering Virtual Clusters*", in the *Equalizer Installation and Administration Guide*.

3. **Install the Server Certificate on Equalizer.**

   Use the Equalizer Administration Interface to install the server certificate. See the section "Installing a Server or Client Certificate for an HTTPS Cluster" on page 178.

4. **Try connecting to the Cluster via HTTPS.**

   From a client browser, open **https://*cluster***, where ***cluster*** is the network node name or IP address of the HTTPS cluster. The browser may notify you that it is accepting a certificate from the server and ask for confirmation. Once you accept the certificate, the requested page should be displayed.

Enabling HTTPS with Server and Client Certificates

The following are the steps to follow to obtain and install both server and client certificates, and verify that they work.

1. **Perform the procedure in the previous section** ("Enabling HTTPS with a Server Certificate" on page 173) **to enable HTTPS with a server side certificate.**

2. **Generate a Client Certificate Signing Request or a Self-Signed Client Certificate.**

   In Step 1, you created a server certificate. Now, follow the same procedure to generate a client certificate; do *one* of the following:

   a. **Create a Certificate Signing Request (CSR) and send it to a Certificate Authority for signing**. See the section "Generating a CSR and Getting It Signed by a CA" on page 175.

   b. **Create a certificate and sign it yourself**. See the section "Generating a Self-Signed Certificate" on page 176.

   Many organizations choose to use third-party signed certificates for their HTTPS clusters, and use self-signed certificates for their clients.

3. **Modify the HTTPS cluster to request a client certificate.**

   a. Select the HTTPS cluster in the left frame of the Equalizer Administrative Interface and then select **menu > Change Cluster Parameters** in the right frame.

   b. Select the **advanced** flag to display advanced options.

   c. Enable the **certify_client** flag; this tells Equalizer to request a client certificate when a client attempts to connect to this cluster.

   d. By default, the **client certificate verification depth** is set to 2. This number indicates the number of levels in a certificate chain that the Equalizer will process before stopping (and refusing the connection). This default will need to be raised if you received more than one chained root certificate in addition to a client certificate from your Certificate Authority. Note that this setting has an impact on performance, since SSL operations are resource intensive.

   e. By default, Equalizer requests a client certificate, but does not *require* the client to provide one. Enable the **require certificate** flag to require that a client return a valid certificate before connecting.

   f. By default, the client's certificate will be re-validated if the SSL connection needs to be renegotiated. (Renegotiation is a feature of SSL, can occur for any of a number of reasons, and may be initiated by Equalizer or the client browser.) Enable the **verify once** flag to tell Equalizer *not* to re-evaluate the client certificate even if SSL renegotiation occurs. This can have a positive performance impact if many SSL renegotiations are occurring during normal operations.

   g. Select **commit** to save your changes to the cluster definition.

   For more information on creating HTTPS clusters, see Chapter 6, "*Administering Virtual Clusters*", in the *Equalizer Installation and Administration Guide*.

4. **Install the Client Certificate on Equalizer.**

   Use the Equalizer Administration Interface to install the client certificate. See the section "Installing a Server or Client Certificate for an HTTPS Cluster" on page 178.

5. **Install the Client Certificate on all clients.**

   Import the client certificate into the client browser's list of certificates. On Firefox, open **Tools > Options > Advanced > View Certificates**. On Internet Explorer, open **Tools > Internet Options > Content > Certificates**. Refer to the documentation for your browser for instructions.

6. **Try connecting to the Cluster via HTTPS.**

   From a client browser, open **https://*cluster***, where ***cluster*** is the network node name or IP address of the HTTPS cluster. The browser may notify you that it is accepting a certificate from the server and ask for confirmation. Once you accept the certificate, the server should ask for a client certificate; your browser may ask you to choose one. After the client certificate is sent to the server and accepted, the requested page should be displayed.

Generating a CSR and Getting It Signed by a CA

Most CA vendors provide a means of generating a Certificate Signing Request (CSR) on their websites, and we recommend that you use the CA website to generate the CSR.

A CSR can also be generated using the OpenSSL tools on any system, including Windows. The examples below were executed on a Windows system with the OpenSSL tools installed.

Note that only the most basic **openssl** command options are shown. See the **openssl**(1) and **req**(1) manual pages at `http://www.freebsd.org/cgi/man.cgi` for more information. Many certificate vendors also provide tools on their websites for entering a CSR.

Generating a CSR using OpenSSL

1. Navigate to an appropriate directory on your system, and create a new directory to hold your CSR, certificate, and private key.

2. Generate the CSR by entering this command:

   ```
   openssl req -new -newkey rsa:1024 -out cert.csr
   ```

   This begins an interactive session to generate a CSR, and also generates a new private key to be output into a file named *privkey.pem*. The key length you use (1024 in this example) can be any multiple of 8. If you already have a private key, use **-key *filename*** (instead of **-newkey rsa:1024**) to specify the file containing the private key. The key length you use (i.e., 1024 in this example) can be any multiple of 8.

   After generating the private key, the following prompts are displayed (example responses shown):

   ```
   Enter PEM pass phrase: <password>
   Verifying - Enter PEM pass phrase: <password>
   Country Name (2 letter code) [AU]:US
   State or Province Name (full name) [Some-State]:New York
   Locality Name (eg, city) []:Millerton
   Organization Name (eg, company) [Internet Widgits Pty Ltd]:CPS Inc.
   Organizational Unit Name (eg, section) []:Engineering
   Common Name (eg, YOUR name) []:mycluster.example.com
   Email Address []:admin@example.com
   ```

   Make sure you remember the **password** you specify, as you will need it to install and use the certificate.

   For a *server certificate*, the **Common Name** provided must be the DNS-resolvable fully qualified domain name (FQDN) used by the Equalizer cluster. When a client receives the certificate from the server, the client browser will display a warning if the **Common Name** does not match the hostname of the request URI.

For a *client certificate*, the **Common Name** in the client's copy of the certificate is only compared to the **Common Name** in the copy of the client certificate on the server, so **Common Name** can be any value.

3. Visit the website of an SSL Certificate Authority (CA) to submit the *cert.csr* file to the CA.
4. Once the CA returns your signed certificate (usually in email), go to the section "Preparing a Signed CA Certificate for Installation" on page 177.

Generating a Self-Signed Certificate

To generate a self signed certificate in PEM format:

1. Generate a self-signed x509 format certificate by entering this command:

```
openssl req -new -x509 -newkey rsa:1024 -out selfcert.pem -days 1095
```

This creates a self-signed certificate (*selfcert.pem*) that will be valid for 1095 days (about three years) and also generates a new private key to be output into a file named *privkey.pem*. The key length you use (1024 in this example) can be any multiple of 8. If you already have a private key, use **-key *filename*** instead of **-newkey rsa:1024** to specify the file containing the private key. The key length you use (i.e., 1024 in this example) can be any multiple of 8.

After generating the private key, the following prompts are displayed (example responses shown):

```
Enter PEM pass phrase: <password>
Verifying - Enter PEM pass phrase: <password>
Country Name (2 letter code) [AU]:US
State or Province Name (full name) [Some-State]:New York
Locality Name (eg, city) []:Millerton
Organization Name (eg, company) [Internet Widgits Pty Ltd]:CPS Inc.
Organizational Unit Name (eg, section) []:Engineering
Common Name (eg, YOUR name) []:myclient.example.com
Email Address []:admin@example.com
```

Depending on the tool you use to create the certificate, you may also be asked for a challenge password and other optional information. Make sure you remember the **password** (and, if prompted, the challenge password) you specify, as you will need it to install the certificate.

The **Common Name** provided must be the DNS-resolvable fully qualified domain name (FQDN) used by the Equalizer cluster. For a *server certificate*, when the client receives the certificate from the server, the browser will display a warning if the **Common Name** does not match the hostname of the request URI. For a *client certificate*, the **Common Name** in the client's copy of the certificate is only compared to the **Common Name** in the copy on the server, so this can be any value.

2. Combine the private key and certificate into one file, using a command like the following:

```
cat selfcert.pem privkey.pem > clustercert.pem
```

3. You can now install your self signed certificate and private key file, *clustercert.pem*, on Equalizer and your clients, as appropriate.

Preparing a Signed CA Certificate for Installation

When you receive your signed certificate back from your CA, you'll get one or more *.pem* files in return, or you'll get one or more mail messages from the CA. The files or messages contain your signed certificate and any necessary intermediate certificates required by the CA's chain of trust.

If you get your certificates in the mail, save each one to an ASCII text file with a *.pem* extension. Make sure you use a text editor such as **Notepad** (Windows) or **vi** (Unix/Linux) to save the files as text files.

Note that if you are using IIS, see the section “Using IIS with Equalizer” on page 182.

If you get only *one* certificate (the signed server certificate) from your CA, then:

1. Save it to a text file (e.g., *servcert.pem* for a server certificate, or *clientcert.pem* for a client certificate).
2. Open a new text file and read both the signed certificate and your private key (in this order) into the file. (The private key was created previously when you generated your CSR.) Save the file as a plain text file. On a Unix system, like Equalizer, you can do this with a command like one the following:

   ```
   cat servcert.pem privkey.pem > clustercert.pem
   ```

   ```
   cat clientcert.pem privkey.pem > clientprivcert.pem
   ```

   Whatever method you use, the file should look like this when you are done:

   ```
   -----BEGIN CERTIFICATE-----
   ...
   -----END CERTIFICATE-----
   -----BEGIN RSA PRIVATE KEY-----
   ...
   -----END RSA PRIVATE KEY-----
   ```

   Make sure you save the file as a plain text file.
3. Install the file into Equalizer as instructed in the section “Installing a Server or Client Certificate for an HTTPS Cluster” on page 178.

If the CA uses chained root, or intermediate, certificates, then you'll receive (or need to download from the CA) more than one *.pem* file: the server certificate, plus any intermediate certificates needed to establish the chain of trust back to a Root CA certificate installed on your web server or client browser.

If you get *more than one* certificate (the signed server certificate plus one or more intermediate certificates) from your CA, then:

1. Save each certificate to a separate text file (e.g., *servcert.pem*, *intmcert.pem*).
2. Open a new text file and read the signed certificate, your private key, and any intermediate certificates (in this order) into the file. (Your private key was created previously, when you generated the CSR.) Save the file as a plain text file. On a Unix system, like Equalizer, you can do this with a command like one of the following:

   ```
   cat servcert.pem privkey.pem intmcert.pem > clustercert.pem
   ```

   ```
   cat clientcert.pem privkey.pem intmcert.pem > clientprivcert.pem
   ```

   Whatever method you use, the file should look like this when you are done:

```
-----BEGIN CERTIFICATE-----
...
-----END CERTIFICATE-----
-----BEGIN RSA PRIVATE KEY-----
...
-----END RSA PRIVATE KEY-----
-----BEGIN CERTIFICATE-----
...
-----END CERTIFICATE-----
Add more certificates here if needed in the chain...
```

Make sure you save the file as a plain text file.

3. Install the file into Equalizer as instructed in the section “Installing a Server or Client Certificate for an HTTPS Cluster” on page 178.

Installing a Server or Client Certificate for an HTTPS Cluster

Your certificate authority may issue you either a single signed client or server certificate, or a signed certificate plus one or more chained root certificates (also called “intermediate” certificates). The certificate or certificates you receive establish a chain of trust that ends at a trusted root certificate installed on your web server (and on every client that interacts with the web server).

You must install all the certificates you receive on Equalizer to complete the installation process for HTTPS clusters. To install them on Equalizer, certificates must be in a single file, in either x509 (*.pem*) or PKCS12 (*.pfx*) format; see the section “Preparing a Signed CA Certificate for Installation” on page 177.

To install certificates onto Equalizer, follow these steps:

1. Copy the file containing the certificate and private key information (*clustercert.pem* in the examples above; *clustercert.pfx* if you used IIS) to the machine from which you will log into the Equalizer Administrative Interface. Note the location.
2. Log into the Equalizer Administration Interface.
3. In the left frame, click the name of the HTTPS cluster for which you want to install a certificate. The cluster’s parameters appear in the right frame.
4. Select **menu** > **Manage SSL Certificates**. The **install SSL certificate** screen appears in the right frame.

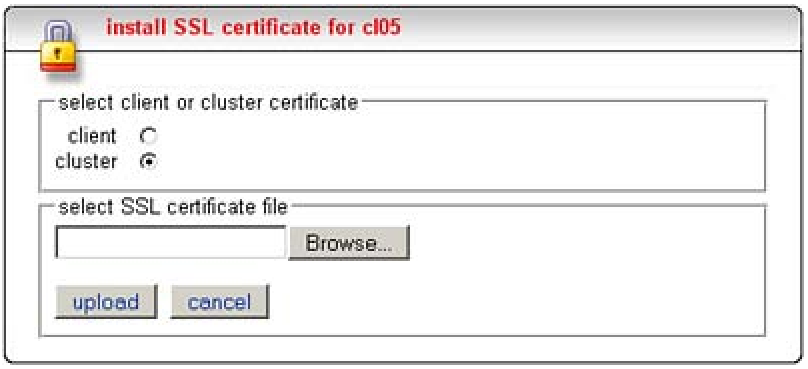

install SSL certificate for cl05
select client or cluster certificate
client ○
cluster ●
select SSL certificate file
Browse...
upload
cancel

Figure 77 The install certificate screen

5. If your Equalizer has an Xcel SSL Accelerator Card installed, a check box labelled **use secure key storage** will appear at the top of the **install SSL certificate** screen. If you do not have an Xcel Card, then this option will not appear on the screen.

   Checking this box tells Equalizer to store your private key in write-only memory on the Xcel card so that no one can access it. See the section “Using Certificates with the Xcel SSL Accelerator Card” on page 180, for more information.

   > **Caution –** If you do not check this box (or you do not have an Xcel card), your key is kept on Equalizer (in the directory */var/eq/ssl*) and will be accessible to anyone who can log into Equalizer. It is therefore essential that you restrict the ability of non-authorized personnel to access Equalizer, since any user can log in and copy or remove your private key. All Equalizer logins should be password protected with non-trivial passwords to restrict access to your private keys, and passwords should be given only to trusted personnel.

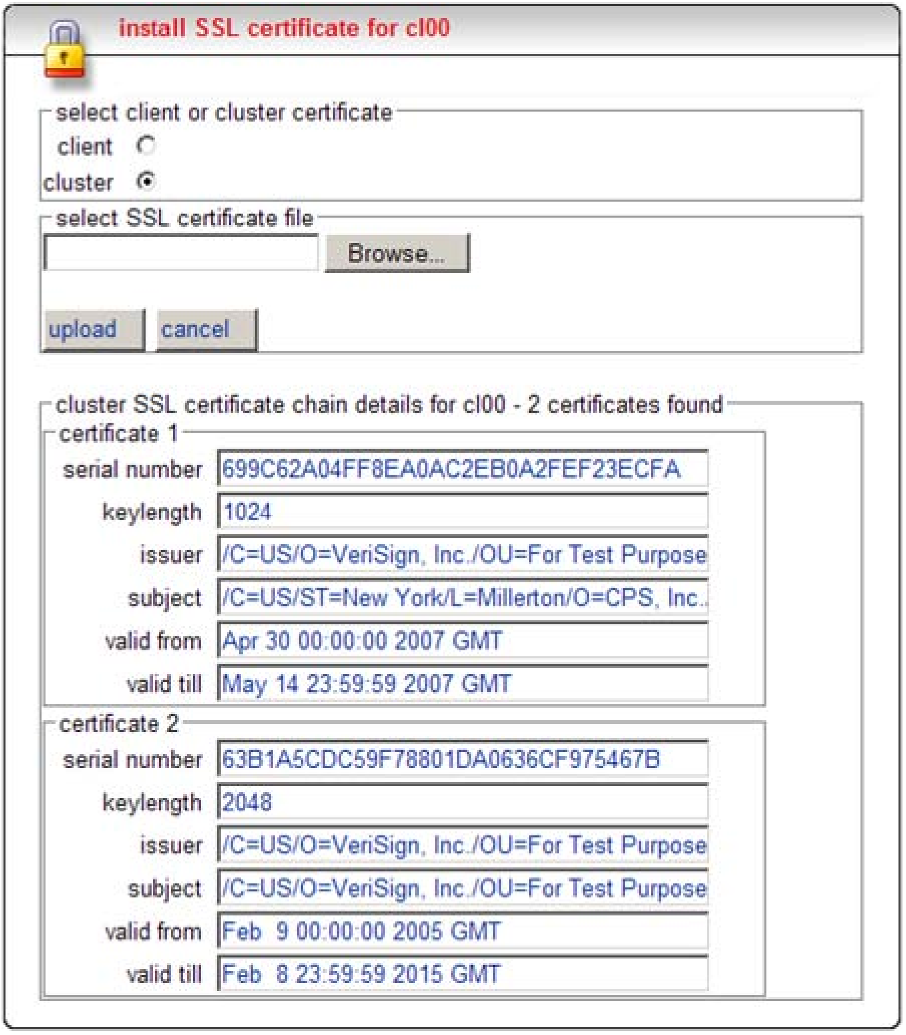
6. If you are installing a *server* certificate, leave the **cluster** radio button selected; if you are installing a *client* certificate, make sure that the **client** radio button is selected.

7. Enter the full path name of the certificate file (or click **Browse** to select the file). Click **upload** to install the certificate on Equalizer. You’ll be prompted for a password, which is the password (PEM pass phrase) you provided when you generated the CSR for the certificate (or created the self-signed certificate).

   Note: Uploading the certificate can fail for a number of reasons. If the **x509 verify** cluster flag is enabled, Equalizer will attempt to verify that the certificate is compliant with the X.509 standard. Certain self-signed or chained certificates will not pass this verification. If you have trouble uploading your certificate, you may need to start this procedure again and, in Step 3, disable **x509 verify** (and **commit** the change) before proceeding.

8. After the upload is complete, select **menu > Manage SSL Certificates** again to verify the certificate details displayed in the **Install SSL certificate** screen. The screen should now show the certificate details, as in the example below. In this example, a file containing a server

certificate, its private key, and an intermediate certificate were uploaded to Equalizer, and the display shows details for both certificates.



install SSL certificate for cl00

select client or cluster certificate

client ○

cluster ●

select SSL certificate file

cluster SSL certificate chain details for cl00 - 2 certificates found

certificate 1

serial number	699C62A04FF8EA0AC2EB0A2FEF23ECFA
keylength	1024
issuer	/C=US/O=VeriSign, Inc./OU=For Test Purpose
subject	/C=US/ST=New York/L=Millerton/O=CPS, Inc.
valid from	Apr 30 00:00:00 2007 GMT
valid till	May 14 23:59:59 2007 GMT

certificate 2

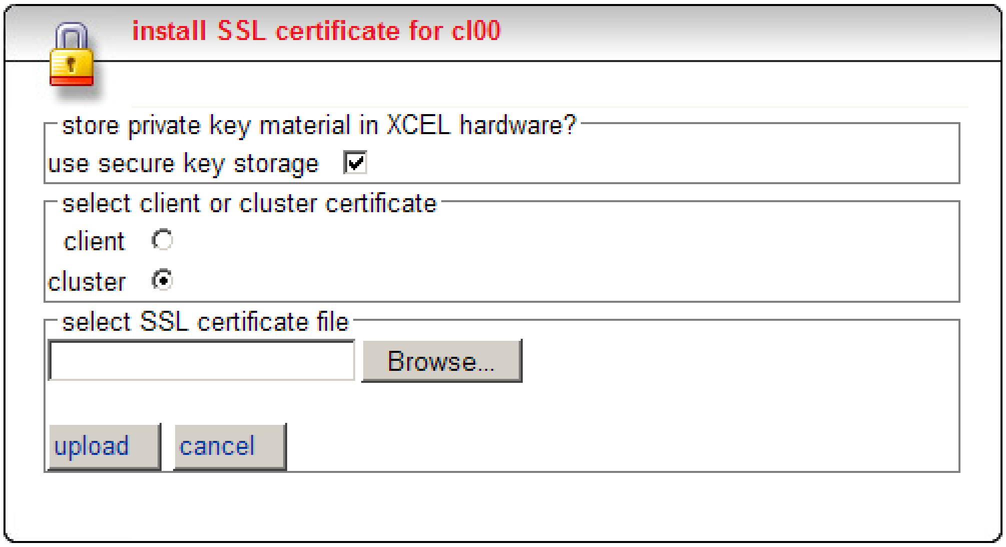
serial number	63B1A5CDC59F78801DA0636CF975467B
keylength	2048
issuer	/C=US/O=VeriSign, Inc./OU=For Test Purpose
subject	/C=US/O=VeriSign, Inc./OU=For Test Purpose
valid from	Feb 9 00:00:00 2005 GMT
valid till	Feb 8 23:59:59 2015 GMT

9. If the certificate you just installed on Equalizer is a client certificate, you’ll also need to install the certificate on each client. This usually involves converting the PEM format certificate into PKCS12 format; see the section “Converting a Certificate from PEM to PKCS12 Format” on page 184.

Using Certificates with the Xcel SSL Accelerator Card

The Equalizer Xcel SSL Accelerator Card is an add-on for Equalizer that provides **secure key storage** (SKS) as well as hardware-based SSL encryption and decryption. All private keys uploaded to an Equalizer with an installed Xcel card can be placed in write-only memory that can only be accessed by the accelerator hardware. This prevents unauthorized access to your private keys.

If your Equalizer has an Xcel SSL Accelerator Card installed, a check box labelled **use secure key storage** will appear on the **install SSL certificate** screen, as shown below.



install SSL certificate for cl00

store private key material in XCEL hardware?

use secure key storage ☑

select client or cluster certificate

client ○

cluster ●

select SSL certificate file

Checking this box tells Equalizer to store your private key in write-only memory on the Xcel card so that no one can access it.

> **Caution –** If you do not check this box (or you do not have an Xcel card), your key is kept on Equalizer (in the directory */var/eq/ssl*) and will be accessible to anyone who can log into Equalizer. It is therefore essential that you restrict the ability of non-authorized personnel to access Equalizer, since any user can log in and copy or remove your private key. All Equalizer logins should be password protected with non-trivial passwords to restrict access to your private keys, and passwords should be given only to trusted personnel.

The Xcel card provides 128 kilobits of memory for private keys. This will hold up to 32 four-kilobit (4096-bit) keys, 64 two-kilobit (2048-bit) keys, or 128 one-kilobit (1024-bit) keys. The key length used for private keys to be stored on an older Xcel I card *must* be a multiple of 8.

Note that if you install the Xcel card in an Equalizer that already has HTTPS clusters with certificates defined, you must delete the HTTPS clusters and add them again in order to store the private keys on the Xcel card in SKS.

Clearing Secure Key Storage

Over time, it is possible for the SKS memory on Xcel to become full. When SKS is full, the following error is returned when you try to add another key (or replace an existing key):

```
Call to 'cert2sks' failed.
Error initializing RSA material
Using stdin
Could not allocate RSA key (N8_NO_MORE_RESOURCE).
Died at /usr/local/sbin/cert2sks line 286.
```

When this happens, you can do one of two things:

- Uncheck the **use secure key storage** check box when adding the SSL certificate; the private key will be kept on the Equalizer instead of in SKS.
- Clear SKS memory (using the procedure below); this removes all keys from SKS and will free up any space taken by keys that are no longer used. This assumes you have not already used all 128kb of space on the Xcel card. If you do this, you'll need to re-add all your certificates for all your HTTPS clusters whose keys were kept in SKS.

To clear SKS memory on the Xcel card:

1. Log into Equalizer as *root* over the serial line.
2. Enter the following command:

```
SKSManager -R -u 0
```

3. After the operation completes (which should take about 1 minute), re-add all certificates for all HTTPS clusters.

Using Certificates in Failover Configurations

In failover configurations, client and server certificates are *not* part of the configuration settings that are transferred between the failover peers when configuration changes are made on one of the failover systems. For this reason, you must install the server certificates (and the client certificates, if used) on *both* of the failover peers.

Using IIS with Equalizer

Using Internet Information Services (IIS) is optional when creating and managing certificates for Equalizer Layer 7 HTTPS clusters and clients. In fact, one of the advantages of using Equalizer is that only one server certificate is required for an HTTPS cluster. The cluster certificate is installed on Equalizer, *not* on the servers in the HTTPS cluster. So, you do not need to use IIS on each server to create and install certificates. This reduces the amount of effort spent administering server certificates.

For Layer 4 TCP and UDP clusters, certificates are *not* installed on Equalizer, and you *will* need to install a server certificate on *each* server in the cluster (since Equalizer is not doing any HTTPS/SSL processing in Layer 4). Generating a CSR and installing a signed certificate on Windows using IIS is shown in the procedure below.

Note that IIS does not support the creation of self-signed certificates. You must create the self-signed certificate on Equalizer (see “Generating a Self-Signed Certificate” on page 176) or another system that supports the OpenSSL tools; then, use IIS to import the certificate into the proper certificate store (usually, the **Personal** store) on Windows.

For more information on using IIS, please refer to the IIS documentation from Microsoft.

Generating a CSR and Installing a Certificate on Windows Using IIS

1. If you have not already installed Internet Information Services (IIS), use the **Add and Remove Programs** wizard (under **Control Panel**) to install it. Click on **Add/Remove Windows**

Components and turn on the check box next to **Internet Information Services (IIS)**; click **Next** and follow the wizard's instructions.

2. Select **Control Panel > Administrative Tools > Internet Information Services**.
3. For a cluster (server) certificate, navigate to the website for which the CSR is intended. For a client certificate, navigate to any website or the default. Right click on the website and select **Properties**.
4. Select the **Directory Security** tab and click the **Server Certificate** button.
5. Select **Next**, and follow the Certificate Wizard prompts:
    a. Select **Create a new certificate**, and then **Next**.
    b. Select **Prepare the request now, but send it later**, and then **Next**.
    c. Type a **Name** for the certificate and select a **Bit Length** that is a multiple of 8. For most purposes, a bit length of 1024 is adequate. Longer bit lengths increase security at the expense of more SSL processing. Select **Next**.
    d. Type in an **Organization** (e.g., **MyCompany, Inc.**) and **Organizational Unit** (e.g., **Marketing**); then select **Next**.
    e. Type in the **Common name** for the certificate, and then select **Next**.

       For a *server certificate*, the **Common Name** provided must be the DNS-resolvable fully qualified domain name (FQDN) used by the Equalizer cluster. When a client receives the certificate from the server, the client browser will display a warning if the **Common Name** does not match the hostname of the request URI.

       For a *client certificate*, the **Common Name** in the client's copy of the certificate is only compared to the **Common Name** in the copy of the client certificate on the server, so **Common Name** can be any value.
    f. Type in a **Country/Region**, **State/province**, and **City/locality**; then select **Next**.
    g. The last step in the wizard is to name and locate the new CSR. The default name and location will be *c:\certreq.txt* unless you choose otherwise.
6. Visit the SSL vendor's website to submit your certificate request.
7. Once the SSL vendor has mailed the new signed certificate back to you, do one of the following:
    a. If you are using this certificate with a Layer 4 cluster, copy the new certificate onto the system on which you generated the request and double-click to install. If this is a server certificate for a server in a Layer 4 TCP or UDP cluster, make sure you attach it to the appropriate web site. If this is a client certificate, make sure you place the certificate in the **Personal** certificate store.
    b. If you are using the certificate with a Layer 7 cluster, export your new SSL certificate with your private key, so that it can be installed on Equalizer:
        A. In IIS, right click on the website for which the certificate was generated and navigate through **Properties > Directory Security > View Certificate > Details**.
        B. Select **Copy to File**, then **Next**.
        C. Select **Yes**, export the private key; then **Next**.
        D. Select **PKCS #12 (.PFX)**; check **Enable strong protection**; then **Next**.
        E. Type and confirm the password; then **Next**.

F. Enter a file name, e.g. ***C:\clustercert.pfx***; then click **Next**.

G. Click **Finish**.

H. Click **Ok** if the export was successful.

I. The certificate is now ready to be uploaded to the cluster via the Equalizer Administration Interface; see “Installing a Server or Client Certificate for an HTTPS Cluster” on page 178.

Converting a Certificate from PEM to PKCS12 Format

Many browsers, such as FireFox and Internet Explorer, require private keys and certificates in PKCS12 format for installation. In order to install client and intermediate certificates into these browsers, you will first have to convert them from PEM format to PKCS12 format. (Note: if you created your certificate using IIS as explained in the previous section, then your certificate is already in PKCS12 format; it can be installed directly into a browser without conversion.)

Like PEM format, PKCS12 format supports having all your certificates and your private key in one file, as discussed above in the section “Preparing a Signed CA Certificate for Installation” on page 177. If you followed the instructions in that section and created the file *clientprivcert.pem* (containing the client certificate, the private key, and any intermediate certificates), then converting the file to PKCS12 is simple:

```
openssl pkcs12 -export -in clientprivcert.pem -out clientprivcert.pfx
```

The resulting file, *clientprivcert.pfx*, can now be installed into all client browsers that will be accessing the cluster that requires a client certificate.

In **Internet Explorer**, certificates are installed by selecting **Tools > Internet Options** from the main menu, selecting the **Content** tab, and pressing the **Certificates** button. Select the **Personal** tab and then the **Import** button.

In **FireFox**, certificates are installed by selecting **Tools > Options** from the main menu, selecting **Advanced**, selecting the **Encryption** tab, and pressing the **View Certificates** button. When the **Certificate Manager** appears, select the **Your Certificates** tab and then the **Import** button.

Supported Cipher Suites

The following tables show the **cipher suites** supported by Equalizer. See the discussion of the **cipher suites** parameter in “Advanced Cluster Fields and Flags” on page 71.

No Xcel and Xcel II Card

The following cipher suites are supported by the base Equalizer software and by the Xcel II (newer generation) SSL Acccelerator Card:

OpenSSL Cipher Suite Name	TLS/SSL Cipher Suite Names
AES128-SHA	TLS_RSA_WITH_AES_128_CBC_SHA
DES-CBC3-SHA	TLS_RSA_WITH_3DES_EDE_CBC_SHA SSL_RSA_WITH_3DES_EDE_CBC_SHA
RC4-SHA	TLS_RSA_WITH_RC4_128_SHA SSL_RSA_WITH_RC4_128_SHA
RC4-MD5	TLS_RSA_WITH_RC4_128_MD5 SSL_RSA_WITH_RC4_128_MD5
AES256-SHA	TLS_RSA_WITH_AES_256_CBC_SHA
The cipher suites below are supported but are not recommended.	
EXP-RC4-MD5	TLS_RSA_EXPORT_WITH_RC4_40_MD5 SSL_RSA_EXPORT_WITH_RC4_40_MD5 SSL_CK_RC4_128_EXPORT40_WITH_MD5
EDH-RSA-DES-CBC3-SHA	TLS_DHE_RSA_WITH_3DES_EDE_CBC_SHA SSL_DHE_RSA_WITH_3DES_EDE_CBC_SHA
DHE-RSA-AES128-SHA	TLS_DHE_RSA_WITH_AES_128_CBC_SHA
DHE-RSA-AES256-SHA	TLS_DHE_RSA_WITH_AES_256_CBC_SHA

Xcel I Card

The following cipher suites are supported by the older generation Xcel I SSL Accelerator cards.

OpenSSL Cipher Suite Name	TLS/SSL Cipher Suite Names
DES-CBC3-SHA	TLS_RSA_WITH_3DES_EDE_CBC_SHA SSL_RSA_WITH_3DES_EDE_CBC_SHA
RC4-SHA	TLS_RSA_WITH_RC4_128_SHA SSL_RSA_WITH_RC4_128_SH
RC4-MD5	TLS_RSA_WITH_RC4_128_MD5 SSL_RSA_WITH_RC4_128_MD5

E Troubleshooting

You usually can diagnose Equalizer installation and configuration problems using standard network troubleshooting techniques. This section identifies some common problems, the most likely causes, and the best solutions.

For additional Troubleshooting information, as well as the most up to date documentation, supplements, and technical articles, please visit the Coyote Point Support website:

```
http://www.coyotepoint.com/support.php
```

Equalizer Doesn’t Boot

Serial cable not connected to Equalizer

Check the cable connection.

Clients Time Out While Trying to Contact a Virtual Cluster

Equalizer is not gatewaying reply packets from the server

Log on to the server(s) and check the routing tables. Perform a `traceroute` from the server to the client. Adjust the routing until Equalizer's address shows up in the `traceroute` output.

⚠ **All packets sent from the server back to clients must pass through Equalizer.**

Test client is on the same network as the servers

If the test client is on the same network as the servers, the servers will probably try to send data packets directly to the client, bypassing Equalizer. You can correct this by adding *host routes* on the servers so that the servers send their reply packets via Equalizer.

No active servers in the virtual cluster

Possible solutions:

- Check the Equalizer Summary page. Are there any servers in that virtual cluster? Are all the servers marked DOWN?
- Log onto the server an run the `netstat` command (Unix servers). If the `netstat` output shows connections in the SYN-RCVD state, the server is not forwarding its reply packets to Equalizer.

Equalizer is not active

Is Equalizer functioning? Try to `ping` the administration address. If you do not get a response, “Equalizer Doesn’t Respond to Pings to the Admin Address” provides additional troubleshooting information.

Primary and Backup Equalizer Are in a Conflict Over Primary

Certain switches (often those from Cisco and Dell) have Spanning Tree enabled by default. This can cause a delay in the times that the network is accessible and cause the backup Equalizer to enter into failover mode. If you cannot disable Spanning Tree, enable FastPort for all ports connected to the Equalizers.

Backup Equalizer Continues to Boot

Primary and Backup Equalizer Are in a Conflict over Primary

Certain Dell and Cisco switches have Spanning Tree enabled by default. This can cause a delay in the times that the network is accessible and cause the backup Equalizer to enter into failover mode. If you cannot disable Spanning Tree, enable PortFast for all ports connected to the Equalizers.

Can’t View Equalizer Administration Pages

Equalizer is not active

Is Equalizer functioning? Try to `ping` the administration address. If you do not get a response, see “Equalizer doesn’t respond to pings to the admin address” below, which provides additional troubleshooting information.

Equalizer Administration Interface Unresponsive

Clear your browser cache; or, close your browser and open it again to establish a new connection.

Equalizer Administration Page Takes a Long Time to Display

DNS server configured on Equalizer is not responding

Possible solutions:

- Check that Equalizer has IP connectivity to the name server configured using the serial configuration utility.
- If you want to disable DNS lookups on Equalizer, specify a name server IP address of 0.0.0.0 in Equalizer's serial configuration utility.

Equalizer Doesn’t Respond to Pings to the Admin Address

Equalizer is not powered on

Check that power switch is on and the front panel LED is lit. Connect the keyboard and monitor, cycle the power, and watch the startup diagnostic messages.

Equalizer isn't connected to your network

Check the network wiring.

Administration address not configured on the external interface

This applies to dual network configurations. Use the Equalizer Configuration Utility to set the IP address and netmask for external interface. Be sure to commit your changes.

Browser Hangs When Trying to Connect Via FTP to an FTP Cluster

FTP server returns its private IP address in response to a “PASV” command

Enable PASV mode FTP translation on the Advanced options page of the Equalizer Administration utility. (For more information, see “Enabling Passive FTP Connections” on page 53.) This behavior is likely to cause problems if you’re using reserved internal addresses for the server

Return Packets from the Server Aren’t Routing Correctly

IP spoofing is enabled

This problem normally occurs in a single network setup. When you enable IP spoofing, clustered servers see the client’s IP address. If the server tries to reply directly to the client, the client will reject the reply (it had sent its request to a different address).

Run a `traceroute` to ensure that routes from a server to a client go through Equalizer and not directly back to the client. If Equalizer does not appear, modify the route to include Equalizer. Alternatively, you can disable IP spoofing.

Web Server Cannot Tell Whether Incoming Requests Originate Externally or Internally

IP Spoofing is not enabled

Check the cluster’s configuration and enable IP spoofing. This causes Equalizer to pass the client’s IP address. Make sure that responses from the server go through the Equalizer.

Why aren't my clusters working if the server status is "up"?

There are several reasons this could be happening. Make sure that Equalizer is being used as the default gateway on all your servers, and that the server service or daemon is running. Sometimes additional host or network routes will need to be added to the clustered servers in single network. The **traceroute** (Unix) and **tracert** (Windows) commands area useful diagnostic tools. Trace from the clustered server back to any client that is not able to resolve the cluster address. If Equalizer is not showing up as the first hop, routing is the cause of the problem.

Context Help Does Not Appear

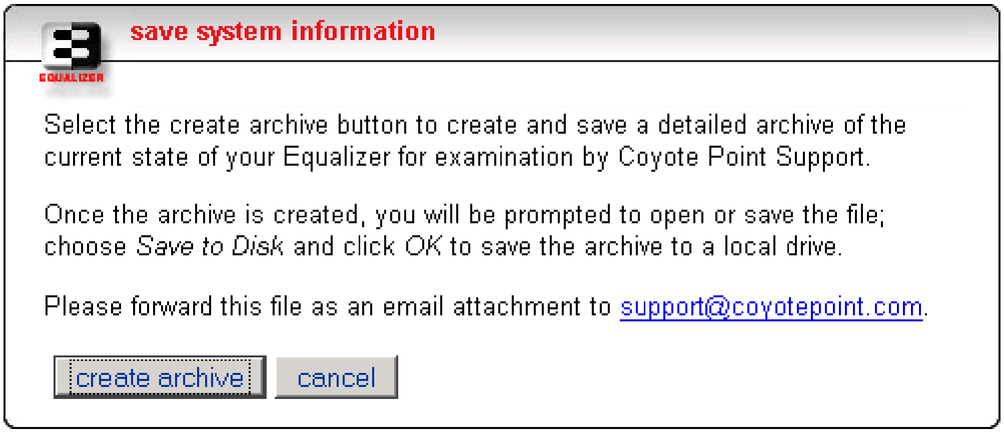
Turn off the Pop-up Blocker for your browser. In FireFox, select **Tools** > **Options** > **Content** and disable the **Block popup windows** check box. In Internet Explorer, select **Tools** > **Internet Options** > **Privacy** and disable the **Turn on Pop-up Blocker** check box.

Creating a System Information Archive

Use the **Help** > **Save System Info** command to create an archive that contains various configuration files, logs, and other information used by Coyote Point Support to help diagnose problems you are having with Equalizer. (This functionality was available in earlier releases only by logging into Equalizer as *root* via the serial line or SSH, and using the **eqcollect** command.)

To create the archive:

1. Log into the administrative interface using the **touch** login.
2. Select the **Help** > **System Info** command. The following is displayed:



Select the create archive button to create and save a detailed archive of the current state of your Equalizer for examination by Coyote Point Support.

Once the archive is created, you will be prompted to open or save the file; choose *Save to Disk* and click *OK* to save the archive to a local drive.

Please forward this file as an email attachment to support@coyotepoint.com.

Figure 78 **Help** > **Save System Info** screen

3. Select **create archive** to create the archive; once the archive is collected, a dialog box is displayed by your browser to save the archive to a local file.
4. Open your email client, and send the file you saved to **support@coyotepoint.com** as an attachment. Explain the nature of your problem in the email, or just include the support ticket number you were given previously by Coyote Point Support.

Restoring Access to the Administrative Interface

If all access to the Administrative Interface is disabled, do the following to enable access again:

1. Log into Equalizer using the serial line or SSH as *root*.
2. Enter the following command exactly as shown:

   ```
   parse_config -a -H 1 -i /var/eq/eq.conf -E -I -F -p -s
   ```

   - `-a`: tells the program to affect the server, rather than just checking the config.
   - `-H 1`: restart Apache after '1' seconds (0 does NOT work)
   - `-i /var/eq/eq.conf`: location of the `eq.conf` file to parse
   - `-E, -I, -F`: start on External IP, Internal IP, Failover IP; respectively
   - `-p, -s`: enable HTTP protocol, HTTPS protocol; respectively
3. Running the above command does *not* update the *eq.conf* file. To restore access in the current configuration, log into the Administrative Interface, change the global parameter `gui access flags`, and commit your changes (see “Modifying System Parameters” on page 43).

F License and Warranty

SOFTWARE LICENSE

PLEASE READ THESE TERMS AND CONDITIONS CAREFULLY BEFORE USING THE SOFTWARE. BY USING THIS SOFTWARE YOU AGREE TO BE BOUND BY THE TERMS AND CONDITIONS OF THIS LICENSE. IF YOU DO NOT AGREE WITH THE TERMS OF THIS LICENSE, PROMPTLY RETURN THE UNUSED SOFTWARE, MANUAL, AND RELATED EQUIPMENT AND HARDWARE (WITH PROOF OF PAYMENT) TO THE PLACE OF PURCHASE FOR A FULL REFUND.

Coyote Point Systems, Inc. (“Coyote Point Systems”) and its suppliers grant to Customer (“Customer”) a nonexclusive and nontransferable license to use the Coyote Point Systems software (“Software”) in object code form solely on a single central processing unit owned or leased by Customer or otherwise embedded in equipment provided by Coyote Point Systems. Customer may make one (1) archival copy of the software provided Customer affixes to such copy all copyright, confidentiality, and proprietary notices that appear on the original. EXCEPT AS EXPRESSLY AUTHORIZED ABOVE, CUSTOMER SHALL NOT COPY, IN WHOLE OR IN PART, SOFTWARE OR DOCUMENTATION; MODIFY THE SOFTWARE; REVERSE COMPILE OR REVERSE ASSEMBLE ALL OR ANY PORTION OF THE SOFTWARE; OR RENT, LEASE, DISTRIBUTE, SELL, OR CREATE DERIVATIVE WORKS OF THE SOFTWARE.

Customer agrees that aspects of the licensed materials, including the specific design and structure of individual programs, constitute trade secrets and/or copyrighted material of Coyote Point Systems. Customer agrees not to disclose, provide, or otherwise make available such trade secrets or copyrighted material in any form to any third party without the prior written consent of Coyote Point Systems. Customer agrees to implement reasonable security measures to protect such trade secrets and copyrighted material. Title to Software and documentation shall remain solely with Coyote Point Systems.

This License is effective until terminated. Customer may terminate this License at any time by destroying all copies of Software including any documentation. This License will terminate immediately without notice from Coyote Point Systems if Customer fails to comply with any provision of this License. Upon termination, Customer must destroy all copies of Software. Software, including technical data, is subject to U.S. export control laws, including the U.S. Export Administration Act and its associated regulations, and may be subject to export or import regulations in other countries. Customer agrees to comply strictly with all such regulations and acknowledges that it has the responsibility to obtain licenses to export, reexport, or import Software.

This License shall be governed by and construed in accordance with the laws of the State of New York, United States of America, as if performed wholly within the state and without giving effect to the principles of conflict of law. If any portion hereof is found to be void or unenforceable, the remaining provisions of this License shall remain in full force and effect. This License constitutes the entire License between the parties with respect to the use of the Software.

Restricted Rights - Coyote Point Systems' software is provided to non-DOD agencies with RESTRICTED RIGHTS and its supporting documentation is provided with LIMITED RIGHTS. Use, duplication, or disclosure by the U.S. Government is subject to the restrictions as set forth in subparagraph “C” of the Commercial Computer Software - Restricted Rights clause at FAR 52.227-

19. In the event the sale is to a DOD agency, the U.S. Government's rights in software, supporting documentation, and technical data are governed by the restrictions in the Technical Data Commercial Items clause at

DFARS

252.227-7015 and DFARS 227.7202.

LIMITED WARRANTY

This document includes Limited Warranty information for Coyote Point Systems products. For products purchased in the European Union, please refer to the European Union Amendment.

General Terms.

EXCEPT AS EXPRESSLY SET FORTH IN THIS LIMITED WARRANTY, COYOTE POINT SYSTEMS MAKES NO OTHER WARRANTIES OR CONDITIONS, EXPRESS OR IMPLIED, INCLUDING ANY IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE, NONINFRINGEMENT OR ARISING FROM A COURSE OF DEALING, USAGE, OR TRADE PRACTICE. COYOTE POINT SYSTEMS EXPRESSLY DISCLAIMS ALL WARRANTIES AND CONDITIONS NOT STATED IN THIS LIMITED WARRANTY. ANY IMPLIED WARRANTIES THAT MAY BE IMPOSED BY LAW ARE LIMITED IN DURATION TO THE LIMITED WARRANTY PERIOD. SOME STATES OR COUNTRIES DO NOT ALLOW A LIMITATION ON HOW LONG AN IMPLIED WARRANTY LASTS OR THE EXCLUSION OR LIMITATION OF INCIDENTAL OR CONSEQUENTIAL DAMAGES FOR CONSUMER PRODUCTS. IN SUCH STATES OR COUNTRIES, SOME EXCLUSIONS OR LIMITATIONS OF THIS LIMITED WARRANTY MAY NOT APPLY TO YOU.

This Limited Warranty applies to the Coyote Point Systems software and hardware products sold by Coyote Point Systems, Inc., its subsidiaries, affiliates, authorized resellers, or country distributors (collectively referred to in this Limited Warranty as “Coyote Point Systems”) with this Limited Warranty.

Software. Coyote Point Systems warrants that: (i) the media on which the Software is furnished will be free of defects in materials and workmanship under normal use; and (ii) the Software substantially conforms to its published specifications during the Limited Warranty Period. Except for the foregoing, the Software is provided AS IS.

Hardware. Coyote Point Systems warrants that the Hardware product will be free from defects in material and workmanship under normal use during the Limited Warranty Period.

The Limited Warranty Period is for one year from the date of shipment. Your dated delivery receipt, showing the date of shipment of the product, is your proof of the shipment date. You may be required to provide proof of purchase as a condition of receiving warranty service. You are entitled to warranty service according to the terms and conditions of this document if a repair to your Coyote Point Systems software or hardware is required within the Limited Warranty Period. This Limited Warranty extends only to the original purchaser of this Coyote Point Systems product and is not transferable to anyone who obtains ownership of the Coyote Point Systems product from the original purchaser.

Coyote Point Systems products are manufactured using new materials or new and used materials equivalent to new in performance and reliability. Replacement products are guaranteed to have functionality at least equal to our published specifications. Replacement parts are warranted to be free from defects in material or workmanship for the remainder of the Limited Warranty Period. Repair or replacement of a part will not extend the Limited Warranty.

During the Limited Warranty Period, Coyote Point Systems will repair or replace the defective component parts or the hardware product. All component parts or hardware products removed under this Limited Warranty become the property of Coyote Point Systems. Coyote Point Systems, at its discretion, may elect to provide you with a replacement unit of Coyote Point Systems' choosing that is at least equivalent to your Coyote Point Systems product in hardware performance. Coyote Point Systems reserves the right to elect, at its sole discretion, to give you a refund of your purchase price instead of a replacement. This is your exclusive remedy for defective products.

To request Limited Warranty service, you must contact Coyote Point Systems Technical Support, which can be reached at (888) 891-8150 or via E-mail at support@coyotepoint.com. Coyote Point Systems Technical Support will determine the nature of the problem, and if a return is necessary, issue a Return Materials Authorization (RMA). No returned product will be accepted without an RMA number obtained in advance and clearly marked on the outside of the shipping container. All products to be returned must be in the original manufacturer's undamaged packaging along with all accessories shipped with the original product including cables, handles and manuals. If you did not retain the original packaging materials, there may be a charge for replacement packaging.

If a defective product is returned, the cost of incoming freight and insurance is the responsibility of the customer. The cost of return freight is the responsibility of Coyote Point Systems, if shipped within the United States. Shipments to other locations will be freight collect. You are responsible for missing or physically damaged parts on the returned defective product, if they are not covered under the product Limited Warranty. You are responsible for all customs fees, taxes or VAT that may be due (excluding income taxes). A product returned for repair but found to be in good working order will be charged a $75 “No Trouble Fee”.

Coyote Point Systems does not warrant that the operation of this product will be uninterrupted or error-free. Coyote Point Systems is not responsible for damage that occurs as a result of your failure to follow the instructions that came with the Coyote Point Systems product.

Restrictions.

This Limited Warranty does not extend to software errors that can not be reproduced, or for any product from which the serial number has been removed, or that has been damaged or rendered defective (a) as a result of accident, misuse, abuse, or other external causes; (b) by operation outside the usage parameters stated in the user documentation that shipped with the product; (c) by the use of parts not manufactured or sold by Coyote Point Systems; or (d) by modification or service by anyone other than (i) Coyote Point Systems, (ii) a Coyote Point Systems authorized service provider, or (iii) your own installation of end-user replaceable Coyote Point Systems or Coyote Point Systems approved parts.

COYOTE POINT SYSTEMS WILL NOT HAVE ANY LIABILITY FOR ANY DAMAGES ARISING FROM THE USE OF THE PRODUCTS IN ANY HIGH-RISK ACTIVITY, INCLUDING, BUT NOT LIMITED TO, THE OPERATION OF NUCLEAR FACILITIES, AIRCRAFT NAVIGATION OR COMMUNICATION SYSTEMS, AIR TRAFFIC CONTROL, MEDICAL SYSTEMS, LIFE SUPPORT, OR WEAPONS SYSTEMS.

These terms and conditions constitute the complete and exclusive warranty agreement between you and Coyote Point Systems regarding the Coyote Point Systems product you have purchased. These terms and conditions supersede any prior agreements or representations including representations made in Coyote Point Systems sales literature or advice given to you by Coyote Point Systems or an agent or employee of Coyote Point Systems that may have been made in connection with your purchase of the Coyote Point Systems product. No change to the conditions of this Limited Warranty is valid unless it is made in writing and signed by an authorized representative of Coyote Point Systems.

Limitation of Liability

IF YOUR COYOTE POINT SYSTEMS SOFTWARE OR HARDWARE PRODUCT FAILS TO WORK AS WARRANTED ABOVE, YOUR SOLE AND EXCLUSIVE REMEDY SHALL BE REPAIR OR REPLACEMENT (INCLUDING REFUND). COYOTE POINT SYSTEMS' MAXIMUM LIABILITY UNDER THIS LIMITED WARRANTY IS EXPRESSLY LIMITED TO THE LESSER OF THE PRICE YOU HAVE PAID FOR THE PRODUCT OR THE COST OF REPAIR OR REPLACEMENT OF ANY SOFTWARE OR HARDWARE COMPONENTS THAT MALFUNCTION IN CONDITIONS OF NORMAL USE. COYOTE POINT SYSTEMS IS NOT LIABLE FOR ANY DAMAGES CAUSED BY THE PRODUCT OR THE FAILURE OF THE PRODUCT TO PERFORM, INCLUDING ANY LOST PROFITS OR SAVINGS OR DATA, OR SPECIAL, INCIDENTAL, OR CONSEQUENTIAL DAMAGES, OR PUNITIVE DAMAGES. COYOTE POINT SYSTEMS IS NOT LIABLE FOR ANY CLAIM MADE BY A THIRD PARTY OR MADE BY YOU FOR A THIRD PARTY.

THIS LIMITATION OF LIABILITY APPLIES WHETHER DAMAGES ARE SOUGHT, OR A CLAIM MADE, UNDER THIS LIMITED WARRANTY OR AS A TORT CLAIM (INCLUDING NEGLIGENCE AND STRICT PRODUCT LIABILITY), A CONTRACT CLAIM, OR ANY OTHER CLAIM. THIS LIMITATION OF LIABILITY CANNOT BE WAIVED OR AMENDED BY ANY PERSON. THIS LIMITATION OF LIABILITY WILL BE EFFECTIVE EVEN IF YOU HAVE ADVISED COYOTE POINT SYSTEMS OR AN AUTHORIZED REPRESENTATIVE OF COYOTE POINT SYSTEMS OF THE POSSIBILITY OF ANY SUCH DAMAGES. THIS LIMITATION OF LIABILITY, HOWEVER, WILL NOT APPLY TO CLAIMS FOR PERSONAL INJURY. THE FOREGOING LIMITATIONS SHALL APPLY EVEN IF THE ABOVE-STATED WARRANTY FAILS OF ITS ESSENTIAL PURPOSE.

THIS LIMITED WARRANTY GIVES YOU SPECIFIC LEGAL RIGHTS. YOU MAY ALSO HAVE OTHER RIGHTS THAT MAY VARY FROM STATE TO STATE OR FROM COUNTRY TO COUNTRY. YOU ARE ADVISED TO CONSULT APPLICABLE STATE OR COUNTRY LAWS FOR A FULL DETERMINATION OF YOUR RIGHTS.

IN THE EVENT OF INCONSISTENCY BETWEEN ANY TERMS OF THIS DISCLAIMER OF WARRANTIES AND LIMITED WARRANTY AND ANY TRANSLATION THEREOF INTO ANOTHER LANGUAGE, THE ENGLISH LANGUAGE VERSION SHALL PREVAIL.

THIS DISCLAIMER OF WARRANTIES AND LIMITED WARRANTY ARE GOVERNED BY THE LAWS OF THE STATE OF NEW YORK, UNITED STATES OF AMERICA, WITHOUT REGARD TO THE CONFLICT OF LAWS PROVISIONS THEREOF. THE UNITED NATIONS CONVENTION ON CONTRACTS FOR THE INTERNATIONAL SALE OF GOODS SHALL NOT APPLY TO THESE TERMS IN ANY RESPECT.

THIS DISCLAIMER OF WARRANTIES AND LIMITED WARRANTY ARE SUBJECT TO THE TERMS OF SALE OF THE COYOTE POINT SYSTEMS' PRODUCT.

G Additional Requirements

Short-Circuit Protection

Warning

This product relies on the building's installation for short-circuit (overcurrent) protection. Ensure that a fuse or circuit breaker no larger than 120 VAC, 15A U.S. (240 VAC, 10A international) is used on the phase conductors (all current-carrying conductors).

Attention Pour ce qui est de la protection contre les courts-circuits (surtension), ce produit dépend de l'installation électrique du local. Vérifier qu'un fusible ou qu'un disjoncteur de 120 V alt., 15 A U.S. maximum (240 V alt., 10 A international) est utilisé sur les conducteurs de phase (conducteurs de charge).

Warnung Dieses Produkt ist darauf angewiesen, daß im Gebäude ein Kurzschluß- bzw. Überstromschutz installiert ist. Stellen Sie sicher, daß eine Sicherung oder ein Unterbrecher von nicht mehr als 240 V Wechselstrom, 10 A (bzw. in den USA 120 V Wechselstrom, 15 A) an den Phasenleitern (allen stromführenden Leitern) verwendet wird.

Power Supply Cord

CAUTION: THE POWER SUPPLY CORD IS USED AS THE MAIN DISCONNECT DEVICE, ENSURE THAT THE SOCKET-OUTLET IS LOCATED/INSTALLED NEAR THE EQUIPMENT AND IS EASILY ACCESSIBLE.

ATTENTION: LE CORDON D'ALIMENTATION EST UTILISÉ COMME INTERRUPTEUR GÉNÉRAL. LA PRISE DE COURANT DOIT ÊTRE SITUÉE OU INSTALLÉE À PROXIMITÉ DU MATÉRIEL ET ÊTRE FACILE D'ACCÉS.

Warnung: Das Netzkabel dient als Netzschalter. Stellen Sie sicher, das die Steckdose einfach zugänglich ist.

Installation into an Equipment Rack

When operating the unit in an equipment Rack, take the following precaution:

- Make sure the ambient temperature around the unit (which may be higher than the room temperature) is within the limit specified for the unit.
- Make sure there is sufficient airflow around the unit.
- Make sure electrical circuits are not overloaded - consider the nameplate rating of all the connected equipment, and make sure you have over current protection.
- Make sure the equipment is properly grounded.
- Make sure no objects are placed on top of the unit.

Chassis Warning—Rack-Mounting and Servicing

Warning To prevent bodily injury when mounting or servicing this unit in a rack, you must take special precautions to ensure that the system remains stable. The following guidelines are provided to ensure your safety:

- This unit should be mounted at the bottom of the rack if it is the only unit in the rack.
- When mounting this unit in a partially filled rack, load the rack from the bottom to the top with the heaviest component at the bottom of the rack.
- If the rack is provided with stabilizing devices, install the stabilizers before mounting or servicing the unit in the rack.

Attention Pour éviter toute blessure corporelle pendant les opérations de montage ou de réparation de cette unité en casier, il convient de prendre des précautions spéciales afin de maintenir la stabilité du système. Les directives ci-dessous sont destinées à assurer la protection du personnel :

- Si cette unité constitue la seule unité montée en casier, elle doit être placée dans le bas.
- Si cette unité est montée dans un casier partiellement rempli, charger le casier de bas en haut en plaçant l'élément le plus lourd dans le bas.
- Si le casier est équipé de dispositifs stabilisateurs, installer les stabilisateurs avant de monter ou de réparer l'unité en casier.

Warnung Zur Vermeidung von Körperverletzung beim Anbringen oder Warten dieser Einheit in einem Gestell müssen Sie besondere Vorkehrungen treffen, um sicherzustellen, daß das System stabil bleibt. Die folgenden Richtlinien sollen zur Gewährleistung Ihrer Sicherheit dienen:

- Wenn diese Einheit die einzige im Gestell ist, sollte sie unten im Gestell angebracht werden.
- Bei Anbringung dieser Einheit in einem zum Teil gefüllten Gestell ist das Gestell von unten nach oben zu laden, wobei das schwerste Bauteil unten im Gestell anzubringen ist.
- Wird das Gestell mit Stabilisierungszubehör geliefert, sind zuerst die Stabilisatoren zu installieren, bevor Sie die Einheit im Gestell anbringen oder sie warten.

Battery

A lithium battery is included with this board. Do not puncture, mutilate, or dispose of the battery in a fire. There is a danger of explosion if the battery is incorrectly replaced. Replace only with the same or equivalent type recommended by the manufacturer. Dispose a used battery according to the manufacturer instructions and in accordance with your local regulations.

Specifications

Power requirements:

- 115/230 VAC auto selecting 60/50 Hz 3.0A

Power Consumption:

Knowing how much power is consumed by a product while using all the hardware resources allows site administrators to plan how many units can be connected to their power circuits without overload.

The power consumption information shown in the tables below was captured during the following operational stages of the product:

- **Rush-in** current -- when the product is powered ON
- **No Load** -- when the product is booted from OS but no resource-hungry process is running
- **100% CPU** -- when 100% processor load is emulated on the product

The following data is captured during the test, at both 110V and 220V:

- **Watts** -- total power consumed by product
- **PF/VA** -- Power Factor in Volt-Amps (a ratio of the real power and apparent power consumed by the product)
- **V/KHz** -- Voltage in kilohertz
- **Amp** -- total current consumed by product

Table 1: Power Consumption at 110V/60Hz

Model	110V/60Hz	Watts	PF/VA	V/KHz	Amp
E550si					
	Rush-in	120	1	120.6	1.703
	No Load	90.6	0.965	119.3	0.789
	100% CPU	115.9	0.97	119.3	1.003
E450si					
	Rush-in	150.3	0.994	119.8	1.623
	No Load	91.3	0.963	119.5	0.793
	100% CPU	143.8	0.985	119.3	1.24
E350si					
	Rush-in	157.6	0.984	121	1.455
	No Load	85.7	0.963	120.4	0.738
	100% CPU	143.8	0.982	120.3	1.21
E250si					
	Rush-in	52.3	0.933	120	1.1
	No Load	41.7	0.901	120	0.39
	100% CPU	46.8	0.924	120	0.43

Table 2: Power Consumption at 220V/50Hz

Model	220V/50Hz	Watts	PF/VA	V/KHz	Amp
E550si					
	Rush-in	114.76	0.447	220	1.2
	No Load	87.2	0.813	220	0.48
	100% CPU	110.7	0.862	220	0.577
E450si					
	Rush-in	146.6	0.943	220	0.778
	No Load	89.3	0.824	221	0.49
	100% CPU	140.3	0.917	220	0.702
E350si					
	Rush-in	137.2	0.935	221	0.898
	No Load	83.2	0.801	222	0.47
	100% CPU	124.7	0.901	221.5	0.685
E250si					
	Rush-in	45.9	0.823	220	0.258
	No Load	39.1	0.781	220	0.226
	100% CPU	43.9	0.795	220	0.249

Operating Environment

- **Temperature**: 40 - 105 °F, 5 - 40 °C.
- **Humidity**: 5 - 90%, non-condensing.

Physical Dimensions

Model	Weight	Height	Width	Depth
E250si	10.9 lbs.	1.75 in.	19 in.	11.75 in.
E350si / E450si / E550	15 lbs.	1.75 in.	19 in.	15.75 in.

Regulatory Certification

Please see the product data sheets on the Coyote Point Website (`www.coyotepoint.com`) for product certification details.

Glossary

This glossary defines some of the key terms used in this document. Some of the glossary definitions are based on RFC1208, “A Glossary of Networking Terms.”[1]

active content verification (ACV) Active Content Verification; an Equalizer mechanism for checking the validity of a server. ACV does not support UDP-based services.

administration address The IP address assigned to Equalizer on the internal network. *See* internal network and IP address.

administration interface The browser-based interface for setting up and managing the operation of Equalizer.

address translation The modification of external addresses to standardized network addresses and of standardized network addresses to external addresses.

agent An application that gathers or processes information for a larger application. *See* server agent.

aggregation A summary of all the data that is computed from detailed information. *See* sticky network aggregation.

alias A nickname that replaces a long name or one that is difficult to remember or spell. *See* IP alias.

aliased IP address A nickname for an IP address. *See* IP alias.

algorithm Instructions, procedures, or formulas used to solve a problem.

application layer Layer 7 (L7); the highest layer of standards in the Open Systems Interconnection (OSI) model (according to The MIcrosoft Press *Computer Dictionary*), which helps a user perform work such as transferring files, formatting e-mail messages, and accessing remote computers.

atom The smallest part of a regular expression in Equalizer. *See* branch, piece, and regular expression.

authoritative name server A name server that maintains the complete information for a particular part of the domain name space. *See* name server.

back-end server A physical server on the internal network that receives connection requests from Equalizer.

backup Equalizer The backup unit, which replaces the primary Equalizer if that Equalizer fails. *See* hot backup and primary Equalizer.

bound A character that represents the limit of part of a regular expression.

bracket expression In a regular expression, a list of characters enclosed in brackets ([...]).

branch In an Equalizer regular expression, a complete piece of a regular expression. You can concatenate and/or match branches. *See* atom, piece, and regular expression.

cache An area in which information is temporarily stored.

Class A An ISO/IEC 11801 standard for twisted pair cabling rated to 100 KHz; similar to Category 1 cabling. Use the Class A standard for voice and low frequency applications. According to the Microsoft Press *Computer Dictionary*, you can use Class A networks "for sites with few networks but numerous hosts." *See* ISO/IEC.

Class B An ISO/IEC 11801 standard for twisted pair cabling rated to 1 MHz; similar to Category 2 cabling. Use the Class B standard for medium bit rate applications. *See* ISO/IEC.

Class C An ISO/IEC 11801 standard for twisted pair cabling rated to 16 MHz; similar to TIA/EIA Category 3 cabling. Use the Class C standard for high bit rate applications, in which the network allocates 24 bits for the IP address network-address field. A Class C network allocates 24 bits for the IP address network-address field and 8 bits for the host field. *See* ISO/IEC.

cluster A set of networked computer systems that work together as one system. *See* server cluster and virtual cluster.

cluster address The IP address assigned to a particular cluster configured on Equalizer.

computed load A measure of the performance of a server relative to the overall performance of the cluster of which the server is a part.

cookie Data that a Web server stores on a client on behalf of a Web site. When a user returns to the Web site, the server reads the cookie data on the client, providing the Web site all the saved information about the user.

daemon An application that runs in the background and performs one or more actions when events trigger those actions.

DNS Domain Name System or Domain Name Service; used to map domain names to Internet servers in order to link to IP addresses or map IP addresses to domain names. *See* IP address.

DNS TTL The amount of time, in seconds, that a name server is allowed to cache the domain information. *See* DNS and TTL.

domain The highest level in an IP address and the last part of the address in the URL. The domain identifies the category under which the Web site operates. For example, in www.coyotepoint.com, com is the domain, where com represents a *commercial* site. *See* domain name, IP address, and subdomain. *See also* DNS.

domain name The owner of an IP address. The next highest level in an IP address and the next-to-last part of the address. For example, in www.coyotepoint.com, coyotepoint is the domain name. *See* domain, IP address, and subdomain. *See also* DNS.

dynamic weight The weight that Equalizer assigns to a particular server during operation. *See* server weight, static weight, and weight.

echo The transmittal of data that has been sent successfully back to the originating computer. *See* ping. *See also* CMP echo request.

edit mode One of two modes in which Equalizer can be administered. In edit mode, you can view and modify parameters. *See* view mode.

EIA Electronic Industries Association; a trade association that sets standards for electrical and electronic components.

endpoint An IP address-port pair that identifies the start or end of an address; a value that ends a process.

[1] O. Jacobsen and D. Lynch, Interop, Inc. March 1991.

Envoy Equalizer add-in; software that supports geographic clustering and load balancing. *See* geographic cluster, geographic load balancing, and load balancing. *See also* intelligent load balancing.

Equalizer Administration Interface An Equalizer window with which you can monitor Equalizer's operation; view statistics; add, modify, or clusters; add, modify, and delete servers; and shut down a server or Equalizer through a Javascript-enabled browser.

Equalizer Configuration Utility An Equalizer feature that enables you to configure Equalizer, set parameters, and shut down and upgrade Equalizer.

external address The IP address assigned to Equalizer on the external network.

external interface A network interface used to connect Equalizer to the external network. *See* interface, internal interface, and network interface.

external network The subnet to which the client machines and possibly the Internet or an intranet are connected.

failover The act of transferring operations from a failing component to a backup component without interrupting processing.

firewall A set of security programs, which is located at a network gateway server and which protect the network from any user on an external network. *See* gateway.

FQDN *See* Fully Qualified Domain Name (FQDN).

FTP File Transfer Protocol; rules for transferring files from one computer to another.

FTP cluster A virtual cluster providing service on the FTP control port (port 21). *See* cluster and virtual cluster.

Fully Qualified Domain Name (FQDN) The complete, registered domain name of an Internet host, which is written relative to the root domain and unambiguously specifies a host's location in the DNS hierarchy. For example, east is a hostname and east.coyotepoint.com is its fully qualified domain name. *See also* domain name.

gateway A network route that typically translates information between two different protocols.

geographic cluster A collection of servers (such as Web sites) that provide a common service over different physical locations. *See* cluster.

geographic load balancing Distributing requests as equally as possible across servers in different physical locations. *See* load balancing. *See also* intelligent load balancing.

geographic probe A query sent to a site in a geographic cluster to gather information so Equalizer can determine the site that is best able to process a pending request. *See* geographic cluster.

header One or more lines of data that identify the beginning of a block of information or a file.

hot backup Configuring a second Equalizer as a backup unit that will take over in case of failure. Also known as a hot spare. *See* backup Equalizer. *See also* primary Equalizer.

A server can also be used as a hot backup, or hot spare, within a cluster. If all the other servers in the cluster fail, the hot spare will begin processing requests for the cluster.

HTTP HyperText Transfer Protocol; the protocol with which a computer or user access information on the World Wide Web.

HTTPS HyperText Transfer Protocol (Secure); a server application programmed to run under the Windows NT operating system.

hub A device that joins all the components attached to a network.

ICMP
See Internet Control Message Protocol.

ICMP echo request
The act of repeating a stream of characters (for example, echoing on the computer screen characters as a user types those characters). *See* ping. *See also* echo.

ICMP triangulation
Routing client requests to the closest site geographically based on triangulation, a method of calculating the location of a site using the known locations of two or more other sites.

intelligent load balancing
A request for load balancing using Equalizer-based algorithms that assess the configuration options set for cluster and servers, real-time server status information, and information in the request itself. *See* algorithm and load balancing. *See also* geographic load balancing.

interface
The place at which two or more systems connect and communicate with each other. *See* external interface, internal interface, and network interface.

internal address
The IP address assigned to Equalizer on the internal network.

internal network
The subnet to which the back-end server machines are connected.

Internet Control Message Protocol (ICMP)
The ISO/OSI Layer 3, Network, protocol that controls transport routes, message handling, and message transfers during IP packet processing. *See* ICMP triangulation and ISO/OSI model.

IP
Internet protocol; the TCP/IP protocol that controls breaking up data messages into packets, sending the packets, and reforming the packets into their original data messages. *See* Internet protocol stack, IP address, packet, and TCP/IP.

IP address
A 32-bit address assigned to a host using TCP/IP. IP addresses are written in dotted decimal format, for example, 192.22.33.1.

ISO/IEC
International Organization for Standardization/International Electrotechnical Commission; international standards organizations.

ISO/OSI model
International Organization for Standardization/Open Systems Interconnection model, a standard that consists of seven layers that control how computers communicate with other computers over a network.

- Layer 1, Physical, which sets the rules for physical connections via hardware, is the lowest layer.
- Layer 2, Data-link, uses Layer 1 and its own rules to control coding, addressing, and transmitting information.
- Layer 3, Network, uses the prior two layers rules as well as its own rules to control transport routes, message handling, and message transfers.
- Layer 4, Transport, uses its rules and those of the previous layers to control accuracy of message delivery and service.
- Layer 5, Session, uses its rules and those of the previous layers to establish, maintain, and coordinate communication.
- Layer 6, Presentation, uses its rules and those of the previous layers to control text formatting and appearance as well as conversion of code.
- Layer 7, Application, uses its rules and those of the other layers to control transmission of information from one application to another. Layer 7 is the highest layer.

See Layer 4, Layer 7, and transport layer.

L4
See Layer 4.

L7
See Layer 7.

latency The time over which a signal travels over a network, from the starting point to the endpoint. *See* ping. *See also* CMP echo request and echo.

Layer 4 (L4) The transport layer; Layer 4 uses its rules and those of the previous three layers to control accuracy of message delivery and service.which controls accuracy of message delivery and service. *See* ISO/OSI model and Layer 7.

Layer 7 (L7) The application layer; Layer 7 uses its rules and those of the other layers to control transmission of information from one application to another. Layer 7 is the highest layer in the ISO/OSI model. *See* ISO/OSI model and Layer 4.

load A job that can be processed or transported once. *See* load balancing. *See also* geographic load balancing and intelligent load balancing.

load balancing Moving a load from a highly-used resource to a resource that is used less often so that operations are efficient. Equalizer balances loads over a wide physical area or by using algorithms that assess options and real-time information. *See* geographic load balancing and intelligent load balancing.

MX exchanger Mail exchanger; a fully qualified domain name to be returned if a server receives a mail exchanger request.

name server A server that stores information about the domain name space.

NAT Network Address Translation; an Internet standard that defines the process of converting IP addresses on a local-area network to Internet IP addresses. *See* NAT subsystem.

NAT subsystem The Equalizer subsystem responsible for transferring connections to and from the back-end servers.

netmask Address mask; a bit mask used to select bits from an Internet address for subnet addressing. The mask is 32 bits long and selects the network portion of the Internet address and one or more bits of the local portion.

Network Address Translation (NAT) *See* NAT.

network interface The place at which two or more networks connect and communicate with each other. *See* interface. *See* external interface, interface, and internal interface.

network route *See* gateway.

OSI network A network that uses the International Organization for Standardization/Open Systems Interconnection model. *See* ISO/OSI model, Layer 4, Layer 7, and transport layer.

packet A group of data that is transmitted as a single entity.

passive FTP connection An Equalizer option that rewrites outgoing FTP PASV control messages from the servers so that they contain the IP address of the virtual cluster rather than that of the server. *See* FTP and PASV.

PASV Passive mode FTP; a mode with which you can establish FTP connections for clients that are behind firewalls. *See* firewall, FTP, and passive FTP connections.

pattern match A pattern of ASCII or hexadecimal data that filters data.

payload The set of data to be transmitted. A payload contains user information, user overhead information, and other information that a user requests. A payload *does not* include system overhead information. Also known as the mission bit stream.

persistence The act of storing or retaining data for use at a later time, especially data that shows the state of the network before processing resumes. *See* cookie and IP-address-based persistence.

physical server A machine located on the internal network that provides services on specific IP addresses and ports. *See* server and virtual web server. *See also* authoritative name server, back-end server, name server, and proxy server.

piece An atom followed by a single *, +, or ?, or by a bound. *See* atom, branch, and regular expression.

ping A program used to test reachability of destinations by sending them an ICMP echo request and waiting for a reply. *See* echo and probe. *See also* CMP echo request

port The abstraction used by Internet transport protocols to distinguish among multiple simultaneous connections to a single destination host.

port number The number used to identify a service contact port, such as HTTP port 80.

primary Equalizer The primary unit that handles requests. If the primary Equalizer fails, the backup unit replaces it. *See also* backup Equalizer and hot backup.

probe An action that obtains status information about a computer, network, or device. *See* geographic probe and ping.

protocol A set of rules that govern adherence to a set of standards. *See* protocol stack.

protocol stack A layer of protocols that process network actions cooperatively and in tandem. *See* protocol.

proxy server A utility, which is part of a firewall, that helps the regular tasks of managing data transmittal from a network to the Internet and from the Internet to the network. *See also* firewall.

quiesce To become quiet or more quiet than previously.

RADIUS Remote Authentication Dial-In User Service; a protocol that authorizes and authenticates a user trying to link to a network or the Internet.

redirection The process of receiving input from or sending output to a different resource than usual.

regular expression (RE) One or more non-empty branches, separated by pipe symbols (|). An expression matches anything that matches one of the branches. *See* atom, branch and piece.

request packet A packet that contains information that requests a response. *See* packet and response packet.

reserved network A network consisting of “phony” IP addresses, which are not registered and cannot be made visible outside of the internal network.

resolution The process of interpreting all the messages between an IP address and a domain name address.

response packet A packet that contains information that responds to a request. *See* packet and request packet.

round robin The default load balancing policy which distributes requests equally among all servers in a virtual cluster, without regard to static weights or adaptive load balancing criteria. The first request received is routed to the first server in the list, the second request to the second server, and so on. When the last server is reached, the cycle starts again with the first server.

router A network device that facilitates the transmission (that is, *routing*) of messages.

routing table A database, which is static or dynamic, that contains a set of route addresses and routing information familiar to the router. A human being enters and updates the information in a static routing table; routers operate and constantly update a dynamic routing table.

RST The reset command, which instructs a device to end a connection.

Secure Sockets Layer (SSL) A protocol, which uses public-key encryption, that enables secure communications between a client and Web server, typically for guarding financial transactions.

server A computer or application that controls access to a network and its associated devices and applications. A server communicates with one or more clients as well as other servers. *See* authoritative name server, back-end server, name server, physical server, proxy server, and virtual web server.

server address The IP address of a server on the internal interface. Multiple IP addresses can be aliased to a single physical server. *See* server.

server agent An agent that provides Equalizer with real-time performance statistics for a specified server. *See* server.

server cluster A group of servers that are components in a network and joined through hardware or software. *See* cluster. *See also* FTP cluster, geographic cluster, and virtual cluster. *See* server.

server endpoint An IP address-port pair that identifies a physical or virtual server on the internal network to which Equalizer can route connection requests. *See* server.

server weight A value that indicates the relative proportion of connection requests that a particular server will receive. *See* dynamic weight, server, static weight, and weight.

session A logical connection between a server and a client that spans a series of individual client requests and server responses. The persistence of session data is maintained through the exchange of cookies in Layer 7, or through the sticky connections feature in Layer 4.

site A cluster of servers under Equalizer control that is part of a geographic cluster.

spoofing Fooling a system into thinking that a transmission comes from an authorized user when that may not be the case.

SSL *See* Secure Sockets Layer (SSL).

stack An area of reserved memory in which applications place status data and other data. *See* protocol stack.

stale connection A partially open or closed connection.

state Status; the current condition of a network, computer, or peripherals.

stateless A condition in which a server processes each request from a site independently and cannot store information about prior requests from that site. Each request stands on its own. *See also* DNS and RADIUS.

static weight The weight that an administrator assigns to a particular server. During operation, Equalizer dynamically adjusts the server weights (that is, dynamic weight), so a server's weight at a particular time might be different from the static weight originally set by the administrator. *See* dynamic weight, server weight, and weight.

sticky connection A connection in which a particular client remains connected to same server to handle subsequent requests within a set period of time.

sticky timer A countdown timer that tracks periods of inactivity between a particular client and server.

subdomain A section, which is formally named, that is under a domain name; analogous to the relationship between a subfolder and folder. For example, in www.coyotepoint.com, www is the subdomain. *See* domain, domain name, and IP address. *See also* DNS.

subnet Part of a network that has the same address as the network plus a unique subnet mask.

switch A device, which is attached to a network and which controls the route over which data is sent.

SYN/ACK Synchronize and acknowledge; a message that synchronizes a sequence of data information and acknowledges the reception of that information.

syslog A system log file, in which information, warning, and error messages are stored in a file, sent to a system, or printed.

TCP Transmission Control Protocol; the rules for the conversion of data messages into packets. *See* ISO/OSI model, Layer 4, packet, transport layer, and TCP/IP.

TCP/IP Transmission Control Protocol/Internet Protocol; the rules for transmitting data over networks and the Internet. *See* TCP.

Telnet Part of TCP/IP, a protocol that enables a user to log onto a remote computer connected to the Internet. *See* TCP/IP.

traceroute A utility that shows the route over which a packet travels to reach its destination.

Transmission Control Protocol (TCP) *See* TCP.

Transmission Control Protocol/Internet Protocol (TCP/IP) *See* TCP/IP.

transport layer *See* Layer 4. *See also* ISO/OSI model.

TTL Time-to-live, the length of time, in seconds, that a client’s DNS server should cache a resolved IP address.

User Datagram Protocol (UDP) Within TCP/IP, a protocol that is similar to Layer 4 (the transport layer). UDP converts data into packets to be sent from one server to another but does not verify the validity of the data. *See* ISO/OSI, TCP/IP, and transport layer.

view mode One of two modes in which Equalizer can be administered: edit and view. In view mode, you can view—but not edit—parameters. *See* edit mode.

virtual cluster An endpoint that acts as the network-visible port for a set of hidden back-end servers. *See* cluster, endpoint, FTP cluster, geographic cluster, and server cluster.

virtual server address An IP address that is aliased to a physical server that has its own, separate IP address. *See* virtual web server.

virtual web server Software that imitates HTTP server hardware. A virtual web server has its own domain name and IP address. *See* domain name, HTTP, IP address, server, and virtual server address. *See also* authoritative name server, back-end server, name server, physical server, and proxy server.

WAP *See* Wireless Application Protocol.

weight The relative proportion of a single item in a population of similar items. *See* dynamic weight, server weight, and static weight.

Wireless Application Protocol (WAP) A set of rules that govern access to the Internet through wireless devices such as cellular telephones, pagers, and two-way communication devices.

Index